

Manual Object Reorganization

Roughly, for us, the execution time was 100 minutes per 1GB of really used space (data). Regarding resources, reorganizing can require up to 300MB of memory and up to 30 percent CPU. It requires a lot less on smaller systems.

- *reorg.sql* - script to reorg all tables in the database
- *reindex.sql* - script to rebuild invalidated indexes - called by *reorg.sql*
- *ts_coalesce.sql* - script to coalesce tablespaces - called by *reorg.sql*

The following method will keep the rest of the database online and available to users. For each table, there are two steps:

Step 1

The ALTER TABLE MOVE command will lock the table for changes, but will allow queries. While the table is moved, the new table will actually be a TEMPORARY segment in the destination tablespace, named something like, “52.42” for the duration of the reorg. The old table will continue to be there and is dropped (and the new table renamed to the old name) only when the new table build is finished successfully. The TEMP tablespace is normally not used. However, RBS and redo logs can take a serious hit.

If there is not enough space, the procedure will fail and the old table will remain in place. This procedure can be run by the schema owner or by the SYSTEM user. Relocating tables to other tablespaces can be done manually, by editing the generated *reorg.lst* script. If there is enough spare space, one can create one or two flip-flop tablespaces, dedicated to moving around reorganized objects, so that the objects are always rebuilt in only a few larger extents when moved to the other tablespace.