



Craig S. Mullins

Database Performance Management

[Return to Home Page](#)

December 1998



DB2 update

Storage Impact of Type 2 Indexes

By Craig S. Mullins

Type 2 indexes provide numerous benefits to a DB2 subsystem. The primary benefit is the elimination of index locking. However, many newer DB2 features such as row level locking and uncommitted reads require Type 2 indexes. And with DB2 Version 6, to be available in 1999, Type 1 indexes are eliminated altogether, so you will be forced to move to Type 2 indexes.

But before rushing off to implement Type 2 indexes you would be wise to investigate the storage impact of migrating from Type 1 to Type 2 indexes.

So, just what will the impact of Type 2 index be with regard to storage requirements? The answer, not

surprisingly, is "it depends!" There are quite a few differences between Type 1 and Type 2 indexes that impact storage. The first difference is in the amount of usable space on an index page. A Type 2 leaf page has 4038 bytes of usable space; a Type 2 non-leaf page has 4046 bytes. Type 1 leaf and non-leaf pages have 4050 useable bytes per page. So, Type 2 indexes have less usable space per page.

Additionally, Type 2 indexes require an additional one-byte RID prefix in addition to the four-byte RID found in both Type 1 and Type 2 indexes. The new one-byte RID prefix found in a Type 2 index contains three flags: pseudo-deleted, possibly uncommitted, and RID hole follows.

Because Type 2 indexes have a different internal structure, two pieces of header information needed on Type 1 indexes are no longer required: the subpage header and the non-unique key header. Since Type 2 indexes do not use subpages, the 17-byte logical subpage header required of a Type 1 index is not in Type 2 indexes.

Non-unique Type 1 indexes have a six-byte header and will repeat an entry (header and key) if a key has more than 255 RIDs. Type 2 indexes have a two-byte header and can have more than 255 RIDs in each entry. The entry is only repeated if there is not enough room in a leaf page to hold all of the RIDs; the same is true for a Type 1 index. Type 2 indexes also have a two-byte MAPID for each key at the end of the page,

so total savings per key is two bytes (six bytes for the Type 1 header, minus two bytes for the Type 2 header and two bytes for the MAPID).

Type 2 indexes store truncated keys instead of the complete key. Only the portion of the key required to make it uniquely identifiable is stored on non-leaf pages. However if there are many duplicate keys so that the same key is on more than one leaf page, a Type 2 index will have RIDs stored in the non-leaf pages, causing more space to be used instead of less. This is due to Type 2 indexes keeping the RIDs in sequence.

Finally, Type 2 indexes are required for large table spaces. In this case the RID is five bytes (plus the one-byte RID prefix, which is still required).

As you can see, there is no clear-cut answer as to whether a Type 1 or Type 2 index will utilize more storage. In general though, you should favor creating Type 2 indexes instead of Type 1 because of the advantages they offer:

- There is no index locking with Type 2 indexes. Index locking is one of the predominant causes of contention in pre-V4 DB2 applications.
- Type 2 indexes are the only type supported for ASCII encoded tables.
- As touched upon earlier, many newer DB2 features can not be used unless Type 2 indexes are used; these features include large object

support, row level locking, data sharing, full partition independence, uncommitted reads, UNIQUE WHERE NOT NULL, and CPU and Sysplex parallelism.

- Furthermore, IBM is promoting Type 2 indexes as the standard and will remove support for Type 1 indexes in DB2 V6. As of DB2 V5, both Type 1 and Type 2 indexes are still supported, though.

Rules of Thumb for Index Storage

Taking all of the points above into consideration, here are some general rules of thumb on index storage requirements that you can apply when developing DB2 databases:

- A Type 1 index with a subpage of 16 usually wastes a lot of space. A Type 2 index will almost always use less space than a Type 1 with 16 subpages (but so will a Type 1 index with a subpage of 1).
- A Type 1 with a subpage of 1 usually will use slightly less space than a Type 2 index for both unique and non-unique keys. For the average user the space difference is relatively small and should not be a factor.
- Beware of Type 2 space usage if numerous row deletes occur. Type 1 indexes clean up after a delete, while DB2 pseudo-deletes index RID entries. A pseudo-delete is when DB2 marks the index entry for deletion, but does not physically delete it. When high levels of activity occur, you could encounter numerous pages of nothing but

pseudo-deleted RIDs. DB2 should periodically clean-up the pseudo-deleted entries, but in some cases users report seeing them staying around for weeks at a time wasting space. A reorganization or rebuild will clean up the pseudo-deleted RIDs and free the wasted space.

- Beware of space usage when numerous inserts occur. Type 1 index entries move around in the page and finally when a split occurs, one half of the index entries are moved to another page, usually causing the one half page to be wasted. This is known as the "half full" problem. Type 2 index pages will also split, but provision has been made at the end of a data set to avoid the "half full" problem. Also Type 2 indexes with non-unique keys will chain RIDs within a page. Each chain entry requires a chain pointer and the normal RID. The additional overhead is two bytes plus the Type 2 RID. All these problems can be solved by reorganizing the index.
- The user should monitor the disk space usage of both Type 1 and Type 2 indexes and reorganize the indexes when they grow too large or when performance problems arise.

Be sure to factor all of these issues into your index storage requirement exercises. The actual index sizing formulas are contained in the IBM DB2 manuals and you should use these calculations (or an automated space calculator) to arrive at actual index space requirements. Good luck planning the space requirements for your Type 2 index conversion.

From DB2 Update (Xephon), December 1998.

© 1999 Mullins Consulting, Inc. All rights reserved.

[Home.](#)