*** First create a tablespace with just the one datafile. It's a LMT, non-ASSM with Uniform size of 1M

```
SQL> CREATE TABLESPACE bowie_ts
  2      DATAFILE 'C:\ORACLE\ORADATA\FLOYD\bowie_ts01.DBF' SIZE 10M
  3      EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M
  4      SEGMENT SPACE MANAGEMENT MANUAL
  5  /

Tablespace created.
```

*** Now, create a table with an initial size that fills the tablespace (an initial of 9M should do the trick).

```
SQL> CREATE TABLE TABLE_A (stuff varchar2(200)) storage (initial 9M)
TABLESPACE bowie_ts;

Table created.
```

*** Can't now create another table in the tablespace, showing that the tablespace and hence data file is effectively full

```
SQL> CREATE TABLE TABLE_B (stuff varchar2(200)) TABLESPACE bowie_ts;
CREATE TABLE TABLE_B (stuff number) TABLESPACE bowie_ts
*
ERROR at line 1:
ORA-01658: unable to create INITIAL extent for segment in tablespace
BOWIE_TS
```

*** Lets just confirm the relative file number of the data file in this tablespace

```
SQL> SELECT file_id, relative_fno, file_name FROM dba_data_files WHERE
tablespace_name = 'BOWIE_TS';

   FILE_ID RELATIVE_FNO FILE_NAME
---------- ------------ ----------------------------------------
         7            7 C:\ORACLE\ORADATA\FLOYD\BOWIE_TS01.DBF
```

*** Let's now add another data file to the tablespace, this time it's only 2M so we can only effectively allocate the one extent

```
SQL> ALTER TABLESPACE bowie_ts ADD DATAFILE
'C:\ORACLE\ORADATA\FLOYD\bowie_ts02.DBF' SIZE 2M;

Tablespace altered.
```

*** Let's confirm the new datafile has been allocated with a greater relative file number than the first

```
SQL> SELECT file_id, relative_fno, file_name FROM dba_data_files WHERE
tablespace_name = 'BOWIE_TS';

   FILE_ID RELATIVE_FNO FILE_NAME
---------- ------------ ----------------------------------------
         7            7 C:\ORACLE\ORADATA\FLOYD\BOWIE_TS01.DBF
         8            8 C:\ORACLE\ORADATA\FLOYD\BOWIE_TS02.DBF


*** Good, 8 is > 7.
*** OK, let's create the new table table_B. This table must allocate an
extent in the new datafile ...


SQL> CREATE TABLE TABLE_B (stuff varchar2(200)) TABLESPACE bowie_ts;

Table created.


*** We shouldn't be able to allocate any more extents, the tablespace
should be full again


SQL> ALTER TABLE TABLE_B ALLOCATE EXTENT;
ALTER TABLE TABLE_B ALLOCATE EXTENT
*
ERROR at line 1:
ORA-01653: unable to extend table BOWIE.TABLE_B by 128 in tablespace
BOWIE_TS


*** Good. Let's confirm that table_b is indeed in the second data file


SQL> SELECT segment_name, file_id, relative_fno FROM dba_extents WHERE
segment_name='TABLE_B';

SEGMENT_NAME            FILE_ID RELATIVE_FNO
-------------------- ---------- ------------
TABLE_B                       8            8



*** OK, let's now create an index for table_b. This index can go in any
other tablespace, it doesn't matter where


SQL> CREATE INDEX table_b_idx ON table_b(stuff) TABLESPACE users;

Index created.


*** Now let's populate and fill the table with a whole bunch of duplicate
row entries for our index

*** We commit as we go until the table is full, hence keeping all the
inserted rows
```

```
SQL> BEGIN
  2  FOR i in 1..10000000 LOOP
  3    INSERT INTO TABLE_B VALUES ('David Bowie is one of the most
brilliant and influential rock acts on the planet');
  4    COMMIT;
  5  END LOOP;
  6  END;
  7  /
BEGIN
*
ERROR at line 1:
ORA-01653: unable to extend table BOWIE.TABLE_B by 128 in tablespace
BOWIE_TS
ORA-06512: at line 3


*** Note all rowids in the table have the same leading object id but is
followed by the 8 to signify the relative file number

*** As each rowid is greater than the previous (as the block id and row
number increase), the index has been performing nice efficient 90-10 splits

*** Therefore the pct_used value should be nice and high


SQL> ANALYZE INDEX table_b_idx VALIDATE STRUCTURE;

Index analyzed.

SQL> SELECT lf_rows, pct_used FROM index_stats;

   LF_ROWS   PCT_USED
---------- ----------
     10668         97



*** Good, 97% means the index is nice and compact and has indeed been
perform 90-10 splits

*** Now let's drop our initial table, the one that filled the first
datafile, bye bye table_A



SQL> DROP TABLE table_a PURGE;

Table dropped.


*** Now let's continue adding rows to our second table_B

*** As the first data file is now empty, we will now allocate all our new
extents from this datafile ...


SQL> BEGIN
  2  FOR i in 1..10000000 LOOP
  3  INSERT INTO TABLE_B VALUES ('David Bowie is one of the most brilliant
and influential rock acts on the planet');
```

```
  4  COMMIT;
  5  END LOOP;
  6  END;
  7  /
BEGIN
*
ERROR at line 1:
ORA-01653: unable to extend table BOWIE.TABLE_B by 128 in tablespace
BOWIE_TS
ORA-06512: at line 3


*** OK, table is all full again.

*** Let's check to ensure we have now added a whole bunch of extents in the
first data file this time


SQL> SELECT segment_name, file_id, relative_fno FROM dba_extents WHERE
segment_name='TABLE_B';

SEGMENT_NAME            FILE_ID RELATIVE_FNO
-------------------- ---------- ------------
TABLE_B                       8            8
TABLE_B                       7            7
TABLE_B                       7            7
TABLE_B                       7            7
TABLE_B                       7            7
TABLE_B                       7            7
TABLE_B                       7            7
TABLE_B                       7            7
TABLE_B                       7            7
TABLE_B                       7            7

10 rows selected.


*** Yes we have, all the new extents have a Relative File Number of 7

*** This means that the rowids for all these new rows have an 7 following
the object id in the rowid where previously it was an 8

*** Therefore, none of these rowids can possibly be the maximum rowid

*** Therefore none of the row entries can have the maximum value

*** Therefore, since moving to the first datafile, all leaf block splits on
the index have been 50-50 splits ...


SQL> ANALYZE INDEX table_b_idx VALIDATE STRUCTURE;

Index analyzed.

SQL> SELECT lf_rows, pct_used FROM index_stats;

   LF_ROWS   PCT_USED
---------- ----------
    107436         58
```

*** As a result, the PCT_USED of the index has plummeted from 97 to 58% ...