

*** Create a simple table (as always) with 1M rows

```
SQL> CREATE TABLE bowie AS SELECT rownum id, sysdate-10000+mod(rownum,10000) date_field FROM dual CONNECT BY LEVEL <=1000000;
```

Table created.

*** Now create an index with just the default block size of 8K

```
SQL> CREATE INDEX bowie_i ON bowie(date_field);
```

Index created.

```
SQL> exec dbms_stats.gather_table_stats(ownname=>'BOWIE', tabname=>'BOWIE', estimate_percent=>null, method_opt=>'FOR ALL COLUMNS SIZE 1');
```

PL/SQL procedure successfully completed.

*** Now check the db_file_multiblock_read_count as this value is important to what happens next .

```
SQL> show parameter db_file_multiblock
```

NAME	TYPE	VALUE
db_file_multiblock_read_count	integer	16

*** Now flush the buffer cache and trace the session to *ACTUALLY* see what's happening.

```
SQL> set autotrace traceonly
```

```
SQL> alter system flush buffer_cache; -- note diff in 9i and may benefit if run several times
```

System altered.

```
SQL> alter session set events '10046 trace name context forever, level 12';
```

Session altered.

*** Now run a simple query to ensure Oracle performs an IFFS ...

```
SQL> SELECT /*+ index_ffs (bowie, bowie_i) */ count(*) FROM bowie where date_field > '01-JAN-1980';
```

Execution Plan

Plan hash value: 3049906432

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	8	727 (2)	00:00:09
1	SORT AGGREGATE		1	8		
* 2	INDEX FAST FULL SCAN	BOWIE_I	1000K	7812K	727 (2)	00:00:09

Predicate Information (identified by operation id):

2 - filter("DATE_FIELD">TO_DATE(' 1980-01-01 00:00:00', 'syyyy-mm-ddhh24:mi:ss'))

Statistics

0	recursive calls
0	db block gets
2662	consistent gets
2655	physical reads
0	redo size
411	bytes sent via SQL*Net to client
396	bytes received via SQL*Net from client

```
2 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
1 rows processed
```

*** Note: Execution plan uses an Index Fast Full Scan as expected

*** Note: The consistent gets is 2662.

*** If you look at the generated trace file, you'll notice the db file scattered read waits have a majority of blocks equal to the db_file_multiblock_read_count (16) as expected ...

```
PARSING IN CURSOR #1 len=93 dep=0 uid=88 oct=3 lid=88 tim=615409480715 hv=254146998
ad='1e22a880' sqlid='cbr6d7n7kbydq'
SELECT /*+ index_ffs (bowie, bowie_i) */ count(*) FROM bowie where date_field > '01-JAN-1980'
END OF STMT
PARSE #1:c=0,e=9072,p=0,cr=0,cu=0,mis=1,r=0,dep=0,og=1,tim=615409480707
BINDS #1:
EXEC #1:c=0,e=88,p=0,cr=0,cu=0,mis=0,r=0,dep=0,og=1,tim=615409495071
WAIT #1: nam='SQL*Net message to client' ela= 7 driver id=1413697536 #bytes=1 p3=0 obj#=12446
tim=615409495114
WAIT #1: nam='db file sequential read' ela= 10584 file#=8 block#=1033 blocks=1 obj#=78294
tim=615409524250
WAIT #1: nam='db file scattered read' ela= 13980 file#=8 block#=1034 blocks=16 obj#=78294
tim=615409551520
WAIT #1: nam='db file scattered read' ela= 1487 file#=8 block#=1050 blocks=16 obj#=78294
tim=615409554677
WAIT #1: nam='db file scattered read' ela= 1377 file#=8 block#=1066 blocks=16 obj#=78294
tim=615409557777
WAIT #1: nam='db file scattered read' ela= 1143 file#=8 block#=1082 blocks=16 obj#=78294
tim=615409561563
WAIT #1: nam='db file scattered read' ela= 1395 file#=8 block#=1098 blocks=16 obj#=78294
tim=615409578514
WAIT #1: nam='db file scattered read' ela= 16948 file#=8 block#=1114 blocks=16 obj#=78294
tim=615409597152
WAIT #1: nam='db file scattered read' ela= 1418 file#=8 block#=1130 blocks=16 obj#=78294
tim=615409600348
WAIT #1: nam='db file scattered read' ela= 86358 file#=8 block#=1146 blocks=15 obj#=78294
tim=615409688319
WAIT #1: nam='db file scattered read' ela= 14990 file#=8 block#=1161 blocks=16 obj#=78294
tim=615409704861
WAIT #1: nam='db file scattered read' ela= 1553 file#=8 block#=1177 blocks=16 obj#=78294
tim=615409708115
WAIT #1: nam='db file scattered read' ela= 1374 file#=8 block#=1193 blocks=16 obj#=78294
tim=615409711113
WAIT #1: nam='db file scattered read' ela= 1421 file#=8 block#=1209 blocks=16 obj#=78294
tim=615409722363
```

...

```
WAIT #1: nam='db file scattered read' ela= 13690 file#=8 block#=3769 blocks=16 obj#=78294
tim=615411896934
WAIT #1: nam='db file scattered read' ela= 1371 file#=8 block#=3785 blocks=16 obj#=78294
tim=615411899941
WAIT #1: nam='db file scattered read' ela= 10091 file#=8 block#=3801 blocks=15 obj#=78294
tim=615411911589
FETCH #1:c=328125,e=2417731,p=2655,cr=2662,cu=0,mis=0,r=1,dep=0,og=1,tim=615411912887
```

*** The number of blocks read per multiblock read is generally 16 blocks as expected as this is the value of the db_file_multiblock_read_count parameter.

*** Note also the elapsed times of the fetch at 2417731 or 2.4 of a second with 2662 CRS (consistent reads)

*** Now repeat the same exercise, but this time create the index in a bigger block size tablespace (in this example we've gone from 8K to 16K)

```
SQL> drop index bowie_i;
```

Index dropped.

```
SQL> CREATE INDEX bowie_i ON bowie(date_field) tablespace ts_16k;
```

Index created.

```
SQL> alter system flush buffer_cache;
```

System altered.

```
SQL> SELECT /*+ index_ffs (bowie, bowie_i) */ count(*) FROM bowie where date_field > '01-JAN-1980';
```

Execution Plan

Plan hash value: 3049906432

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	8	500 (2)	00:00:07
1	SORT AGGREGATE		1	8		
* 2	INDEX FAST FULL SCAN	BOWIE_I	1000K	7812K	500 (2)	00:00:07

Predicate Information (identified by operation id):

2 - filter("DATE_FIELD">TO_DATE(' 1980-01-01 00:00:00', 'syyyy-mm-ddhh24:mi:ss'))

Statistics

0 recursive calls
0 db block gets
1317 consistent gets
1310 physical reads
0 redo size
411 bytes sent via SQL*Net to client
396 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
1 rows processed

*** WOW !! Consistent reads have gone down dramatically from 2662 down to just 1317.

*** However, if you look at the actual trace file, we note some interesting details ...

```
SELECT /*+ index_ffs (bowie, bowie_i) */ count(*) FROM bowie where date_field > '01-JAN-1980'  
END OF STMT
```

```
PARSE #1:c=0,e=61,p=0,cr=0,cu=0,mis=0,r=0,dep=0,og=1,tim=626802114084
```

```
BINDS #1:
```

```
EXEC #1:c=0,e=58,p=0,cr=0,cu=0,mis=0,r=0,dep=0,og=1,tim=626802114261
```

```
WAIT #1: nam='SQL*Net message to client' ela= 3 driver id=1413697536 #bytes=1 p3=0 obj#=515  
tim=626802114293
```

```
WAIT #1: nam='db file sequential read' ela= 7681 file#=6 block#=5 blocks=1 obj#=78296  
tim=626802122045
```

```
WAIT #1: nam='db file scattered read' ela= 3711 file#=6 block#=6 blocks=8 obj#=78296  
tim=626802125884
```

```
WAIT #1: nam='db file scattered read' ela= 1413 file#=6 block#=14 blocks=8 obj#=78296  
tim=626802128684
```

```
WAIT #1: nam='db file scattered read' ela= 1447 file#=6 block#=22 blocks=8 obj#=78296  
tim=626802131649
```

```
WAIT #1: nam='db file scattered read' ela= 2014 file#=6 block#=30 blocks=8 obj#=78296  
tim=626802135222
```

```
WAIT #1: nam='db file scattered read' ela= 16734 file#=6 block#=38 blocks=8 obj#=78296  
tim=626802153598
```

```
WAIT #1: nam='db file scattered read' ela= 28220 file#=6 block#=46 blocks=8 obj#=78296  
tim=626802183366
```

```
WAIT #1: nam='db file scattered read' ela= 29689 file#=6 block#=54 blocks=8 obj#=78296  
tim=626802214601
```

....

```
WAIT #1: nam='db file scattered read' ela= 1451 file#=6 block#=1277 blocks=8 obj#=78296  
tim=626804860970
```

```
WAIT #1: nam='db file scattered read' ela= 1735 file#=6 block#=1285 blocks=8 obj#=78296
tim=626804864204
WAIT #1: nam='db file scattered read' ela= 1888 file#=6 block#=1293 blocks=8 obj#=78296
tim=626804867541
WAIT #1: nam='db file scattered read' ela= 1497 file#=6 block#=1301 blocks=8 obj#=78296
tim=626804870568
WAIT #1: nam='db file scattered read' ela= 1164 file#=6 block#=1309 blocks=6 obj#=78296
tim=626804873195
FETCH #1:c=140625,e=2759712,p=1310,cr=1317,cu=0,mis=0,r=1,dep=0,og=1,tim=626804874040
```

*** First thing to note is that the size of the multiblock reads has halved. Where previously Oracle was reading 16 blocks per multiblock read, now it's only reading 8 blocks at a time

*** The effective size of a multiblock read is actually identical.

*** The total elapsed time for the fetch is 2759712, where previously it was 2417731. In this specific example, it's actually worse than previously even though the CR value has gone down to just 1317 ...

*** Although consistent reads has dropped, behind the covers, Oracle is actually performing the same sized I/Os as it adjusts the size of the blocks read per multiblock read, based on the block size of the segment being read.