

*** Create a simple little table, in a non ASSM tablespace

```
SQL> CREATE TABLE reverse_details (id NUMBER, name VARCHAR2(20));
```

Table created.

*** Create a normal, non-reverse index based on the monotonically increasing ID column

```
SQL> CREATE INDEX normal_index ON reverse_details(id);
```

Index created.

*** Insert a whole bunch of rows with the ID monotonically increasing

```
SQL> INSERT INTO reverse_details SELECT rownum, 'David Bowie' FROM dual
CONNECT BY LEVEL <= 1000000;
```

1000000 rows created.

```
SQL> COMMIT;
```

Commit complete.

*** Gather stats on the table and index ...

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

PL/SQL procedure successfully completed.

*** Let's look at the Clustering Factor

```
SQL> SELECT i.index_name, i.clustering_factor, t.blocks, t.num_rows
FROM user_indexes i, user_tables t
WHERE i.table_name = t.table_name and t.table_name = 'REVERSE_DETAILS'
and i.index_name = 'NORMAL_INDEX';
```

INDEX_NAME	CLUSTERING_FACTOR	BLOCKS	NUM_ROWS
NORMAL_INDEX	3029	3033	1000000

1 row selected.

*** Well, at 3,029 on a table that has 3,033 blocks, it doesn't get much better than that ...

*** Let's rebuild the index as a Reverse Key Index

```
SQL> ALTER INDEX normal_index REBUILD REVERSE COMPUTE STATISTICS;
```

Index altered.

*** Let's look at the Clustering Factor now

```
SQL> SELECT i.index_name, i.clustering_factor, t.blocks, t.num_rows
       FROM user_indexes i, user_tables t
       WHERE i.table_name = t.table_name and t.table_name = 'REVERSE_DETAILS'
and i.index_name = 'NORMAL_INDEX';
```

INDEX_NAME	CLUSTERING_FACTOR	BLOCKS	NUM_ROWS
NORMAL_INDEX	999994	3033	1000000

1 row selected.

*** Well, at 999,994 on a table with 1,000,000 rows, it doesn't get much worse than that ...