

# Rebuilding indexes in an OEBS environment

By

**G.Piper**

**www.PIPER-Rx.com**

Jan 2011

Index maintenance is all too often overlooked and can, over time, have a serious impact on performance. Ever growing indexes account for large amounts of wasted space and failing to perform regular index rebuilds on dynamic application tables is just throwing away performance.

I have always advocated that not all performance issues are technical, with the majority of issues in a stable OEBS environment being related to user behaviour. Whilst index rebuilds are a technical process, it is the failure of the DBA (user) to perform regular index rebuilds on dynamic application tables that is the issue.

We will also show how normal DBA processes do not always readily translate to an OEBS environment.

## 1. The issue

When rows are deleted from a table, Oracle does not always reuse the index space until you:

- ❖ Delete all entries from the table
- ❖ Truncate the table
- ❖ Rebuild the index

In the following example I created a table [index\\_demo](#) and an index [index\\_demo\\_n1](#) and used the validate index command to demonstrate the number of deleted rows in our index:

[Index\\_stats](#) values after adding 50 rows to the [index\\_demo](#) table

NAME	LF_ROWS	DEL_LF_ROWS	PCT_DELETED_ROWS
INDEX_DEMO_N1	50	0	0

[Index\\_stats](#) values after deleting all rows to the [index\\_demo](#) table

NAME	LF_ROWS	DEL_LF_ROWS	PCT_DELETED_ROWS
INDEX_DEMO_N1	50	50	100

[Index\\_stats](#) values after truncating the [index\\_demo](#) table

NAME	LF_ROWS	DEL_LF_ROWS	PCT_DELETED_ROWS
INDEX_DEMO_N1	0	0	0

Truncating the table will release the index space, but truncating objects is not always available.

I understand that if you add the same values back to the index the index leaf may be reused, however this rarely happens in OEBS. A case in point is that the [request\\_id](#) used in [fnd\\_concurrent\\_requests\\_u1](#) is an incrementing sequence number.

## 2. Not at my site....

The following are two examples of what can happen to you:

### 1.1 Site 1 - A relatively small site

The Application is holding approximately 4 years of history sign-on audit data with no purge program in place. Once a purge is implemented the indexes will definitely need a rebuild...

The application also generates approximately 1,000 [fnd\\_concurrent\\_requests](#) records per day which are being purged nightly holding 30 days history on-line. The indexes had not been rebuilt for over 6 years 3 months thus in this instance the base table should be holding approximately 30,000 records whereas the primary key index [fnd\\_concurrent\\_requests\\_u1](#) would be holding approximately 2.25 million rows; 98.7% deleted rows

### 1.2 Site 2: - A larger site

The Application is holding approximately 1.5 years of history sign-on audit data, ([fnd\\_logins](#) is holding 13.5 million rows) with no purge program in place. Once a purge is implemented the indexes will definitely need a rebuild...

The application also generates approximately 35,900 [fnd\\_concurrent\\_requests](#) records per day which are being purged nightly holding 7 days history on-line. The indexes had not been rebuilt for over 4 years. Thus in this instance the base table should be holding approximately 251,000 records whereas the primary key index [fnd\\_concurrent\\_requests\\_u1](#) would be holding approximately 52 million rows; 99.9% deleted rows. This is an interesting one when you calculate the space the index is holding; I estimate over 1.4 Gb for [fnd\\_concurrent\\_requests\\_u1](#).

Also keep in mind there are nine (9) to eleven (11) indexes on the [fnd\\_concurrent\\_requests](#) table depending on your OEBS (Oracle E-Business Suite) version...

## 3. Monitoring products can help

One product I know of will regularly validate a list of indexes and alert when any have been found with over 30% deleted rows. This is resource intensive but very accurate.

Another product scans [dba\\_objects.last\\_ddl\\_time](#) for list of indexes and alerts if any of them have not been rebuilt in the past **X** months.

Both these products evaluate either the primary key or similar index for each of the target tables on the principle that if the primary key index has not been rebuilt then all other indexes on the object are likely not to have been rebuilt. The method of reviewing one index per table is a lot less resource intensive than reviewing all indexes on the target tables.

In both cases the products provide a standard list of indexes out of the box and both allow you to add your own indexes to the list for tables you know are dynamic.

I prefer the second method as regular maintenance rebuilds should form part of any DBA's normal maintenance program.

However, I was at one site that was running a proprietary monitor that sent a daily alert indicating the indexes on [fnd\\_concurrent\\_requests](#) require rebuilding. Rather than rebuilding the index the DBA disabled the alert as it was "annoying".

## 4. Assessing the damage at your site

Of course as part of your normal maintenance program you will have been regularly analysing your application's objects using one of the concurrent request programs like "Gather Schema Statistics" (not the [dbms\\_utility.analyze\\_schema](#) or [dbms\\_stats.gather\\_schema\\_stats](#) methods...) unfortunately these methods do not provide you with the number of deleted leaf rows in the index.

What you need to do is validate the index using the following command:

```
VALIDATE index APPLSYS.FND_CONCURRENT_REQUESTS_U1;
```

Or

```
ANALYZE index APPLSYS.FND_CONCURRENT_REQUESTS_U1 validate structure;
```

**Note:** The validate index command does not affect the cost based optimiser stats or the optimiser. The command only populates the [index\\_stats](#) object.

You can then use the following SQL to show the number and percentage of deleted leaf rows:

```
SELECT name,  
       lf_rows,  
       del_lf_rows,  
       del_lf_rows * 100 /  
         decode(lf_rows, 0, 0.0001, lf_rows) pct_deleted_rows  
FROM sys.index_stats;
```

**Note:** The sys object [index\\_stats](#) only holds one (1) record at a time

## 5. When should an index be rebuilt?

There are many schools of thought here, most literature I have read over the years says that indexes are candidates for rebuild when there are more than 20 – 30% deleted leaf rows, In an OEBS environment I prefer to stick to the higher end of 30%.

In fact the best solution I have found over the years is to identify the indexes that are candidates for rebuild and rebuild those indexes once every 3 months for a high transaction site and once every 6 months for a low transaction site.

## 6. What Indexes should I be rebuilding?

The primary candidates are indexes related to the tables that are regularly purged as part of the normal purge program:

- ❖ Sign-on Audit
- ❖ Self-service sessions
- ❖ Concurrent requests
- ❖ Workflow
- ❖ ... etc
- ❖ and the 100+ application interface tables

The following SQL provides a list of indexes associated with the above processes listing indexes that have more than 5,000 rows, the age since the last rebuild and a rebuild identifier for any index that has not been rebuilt in the last 6 months:

```
SELECT dob.owner,
       dind.table_name,
       dob.object_name index_name,
       dind.num_rows row_count,
       initcap(dind.uniqueness) index_type,
       initcap(dind.partitioned) partitioned,
       dind.temporary,
       dind.generated,
       to_char(dob.created, 'DD-Mon-YY HH24:MI') created_date,
       to_char(dob.last_ddl_time, 'DD-Mon-YY HH24:MI') last_ddl_date,
       lpad(trunc((months_between(sysdate , dob.last_ddl_time)) /12 ),2,'0') ||':'||
       lpad(trunc((months_between(sysdate , dob.last_ddl_time)) -
                (trunc((months_between(sysdate , dob.last_ddl_time)) /12 ) *12 )),2,'0') ||':'||
       lpad(trunc((months_between(sysdate , dob.last_ddl_time)) -
                (trunc((months_between(sysdate , dob.last_ddl_time)) /12 ) *12 )
                -
                (trunc((months_between(sysdate , dob.last_ddl_time)) -
                (trunc((months_between(sysdate , dob.last_ddl_time)) /12 ) *12 )))
                * to_char(last_day(sysdate), 'DD')),2,'0') "Last Rebuilt YY-MM-DD",
       Decode( sign(months_between (sysdate, dob.last_ddl_time) - 6 ),
              -1, null, 'Rebuild') rebuild_status
FROM   dba_objects dob,
       dba_indexes dind
WHERE  dob.owner = dind.owner
       and dob.object_name = dind.index_name
       and dob.object_name in
           (SELECT index_name
            FROM dba_indexes
            WHERE (table_name in
                  ('FND_CONCURRENT_REQUESTS',
```

```

--Sign-on Audit Purge
'FND_LOGINS',
'FND_LOGIN_RESPONSIBILITIES',
'FND_LOGIN_FORMS',
-- LOBS
'FND LOBS',
-- Self Service Purge
'ICX_SESSIONS',
'ICX_SESSION_ATTRIBUTES',
'ICX_TRANSACTIONS',
'ICX_TEXT',
'ICX_CONTEXT_RESULTS_TEMP',
'ICX_FAILURES',
'ICX_REQUISITIONER_INFO',
'FND_SESSION_VALUES',
-- Workflow Purge
'WF_ACTIVITIES',
'WF_ACTIVITIES_TL',
'WF_ACTIVITY_ATTR_VALUES',
'WF_ACTIVITY_ATTRIBUTES',
'WF_ACTIVITY_ATTRIBUTES_TL',
'WF_ACTIVITY_TRANSITIONS',
'WF_COMMENTS',
'WF_ITEM_ACTIVITY_STATUSES',
'WF_ITEM_ACTIVITY_STATUSES_H',
'WF_ITEMS',
'WF_NOTIFICATION_ATTRIBUTES',
'WF_NOTIFICATIONS',
'WF_PROCESS_ACTIVITIES' )
or table_name like '%INTERFACE%'
and (num_rows > 5000
    or num_rows is null )
and dob.object type = 'INDEX'
ORDER by dind.table name,
      dob.object_name;

```

**Note:** The row counts require that you have gathered stats on the application objects.

### Example output

OWNER	TABLE_NAME	INDEX_NAME	ROW_COUNT	INDEX_TYPE	PARTITIONED	TEMPORARY	GENERATED	CREATED_DATE	LAST_DDL_DATE	Last Rebuilt...	REBUILD_STATUS
APPL...	FND_CONCURRENT_REQUESTS	FND_CONCURRENT_REQUESTS_N1	36962	Nonunique	No	N	N	25-Nov-02 01:57	26-Jan-06 05:10	04:11:18	Rebuild
APPL...	FND_CONCURRENT_REQUESTS	FND_CONCURRENT_REQUESTS_N10	49108	Nonunique	No	N	N	24-Jul-06 22:40	24-Jul-06 22:40	04:05:19	Rebuild
APPL...	FND_CONCURRENT_REQUESTS	FND_CONCURRENT_REQUESTS_N2	44967	Nonunique	No	N	N	25-Nov-02 01:57	26-Jan-06 06:18	04:11:18	Rebuild
APPL...	FND_CONCURRENT_REQUESTS	FND_CONCURRENT_REQUESTS_N3	30501	Nonunique	No	N	N	25-Nov-02 01:57	26-Jan-06 05:52	04:11:18	Rebuild
APPL...	FND_CONCURRENT_REQUESTS	FND_CONCURRENT_REQUESTS_N4	56509	Nonunique	No	N	N	25-Nov-02 01:57	26-Jan-06 06:07	04:11:18	Rebuild
APPL...	FND_CONCURRENT_REQUESTS	FND_CONCURRENT_REQUESTS_N5	5465	Nonunique	No	N	N	25-Nov-02 01:57	26-Jan-06 06:38	04:11:18	Rebuild
APPL...	FND_CONCURRENT_REQUESTS	FND_CONCURRENT_REQUESTS_N6	43511	Nonunique	No	N	N	25-Nov-02 01:57	26-Jan-06 04:37	04:11:18	Rebuild
APPL...	FND_CONCURRENT_REQUESTS	FND_CONCURRENT_REQUESTS_N7	46879	Nonunique	No	N	N	25-Nov-02 01:57	26-Jan-06 06:39	04:11:18	Rebuild
APPL...	FND_CONCURRENT_REQUESTS	FND_CONCURRENT_REQUESTS_N8	49129	Nonunique	No	N	N	25-Nov-02 01:57	26-Jan-06 06:26	04:11:18	Rebuild
APPL...	FND_CONCURRENT_REQUESTS	FND_CONCURRENT_REQUESTS_N9	45978	Nonunique	No	N	N	25-Nov-02 01:57	26-Jan-06 06:34	04:11:18	Rebuild
APPL...	FND_CONCURRENT_REQUESTS	FND_CONCURRENT_REQUESTS_U1	47206	Unique	No	N	N	25-Nov-02 01:57	26-Jan-06 05:34	04:11:18	Rebuild

This is just a starter list; you should add any indexes you know should be rebuilt on a regular basis to the SQL script.

## 7. Should I use a rebuild script I found on the web?

There are a number of scripts written by DBAs to validate and rebuild indexes, one such script can be found at:

<http://toadworld.com/KNOWLEDGE/KnowledgeXpertforOracle/tabid/648/TopicID/RBIN2/Default.aspx>

**Beware:** Before you run this in an OEBS environment you should be aware that a single validate index and [index\\_stats](#) and lookup can take approximately 5 plus seconds; there are approximately 1,300 indexes for APPLSYS alone, which should take 1 hour 48 minutes to run, excluding index rebuild time. The total OEBS application will have about 66,000 indexes (version dependant), to do all application modules would take approximately 3.8 days excluding index rebuild time.

There are also a number of **partitioned indexes** which require a different rebuild command to normal indexes.

This is a very good example of how some normal DBA processes do not always readily translate to an OEBS environment.

Have fun rebuilding your indexes... ☺

## 8. Want to know more?

There is loads more **FREE** information on this topic and all aspects of OEBS Application Administration at the **PIPER-Rx** website. After over 20+ years working with Oracle (the product, not the Company) and Oracle E-Business Suite (since Release 5) I have visited countless sites and pretty much seen it all when it comes to Applications Administration. Since the late 1990's I have spent more time sharing these learnings and the most popular papers and case studies I have presented are available at the **PIPER-Rx.com** website as well as a whole host of Tips and Reports I have used throughout my career.

All information at the **PIPER-Rx.com** website is **FREE** so why not check it out....I hope you find it useful! – **30,000+ downloaders to date can't be wrong!**

## 9. Disclaimer

*The material contained in this document is provided by the author "as is" and any express or implied warranties, including, but not limited to, any implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the author be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of any content or information, even if advised of the possibility of such damage. It is always recommended that you seek independent, professional advice before implementing any ideas or changes to ensure that they are appropriate.*

*Oracle®, Oracle Applications® & Oracle E-Business Suite® are registered trademarks of Oracle Corporation*

*TOAD® is a registered trademark of Quest Software*