

- [Oracle](#)
- [Blogs Home](#)
- [Products & Services](#)
- [Downloads](#)
- [Support](#)
- [Partners](#)
- [Communities](#)
- [About](#)
- [Login](#)

Oracle Blog

[Oracle Database In-Memory](#)

Technical details, ideas and news on Oracle Database In-Memory

« [It's here! Oracle...](#) | [Main](#)

Getting started with Oracle Database In-Memory Part I

By Maria Colgan on [Jul 28, 2014](#)

Now that [Oracle Database 12.1.0.2](#) has been officially released, I can finally start sharing more technical details on how [Oracle Database In-Memory](#) (Database In-Memory) works.

I thought we should start this series of post right at the very beginning by answering the most fundamental question, how and when is Database In-Memory installed and enabled.

Let's start by doing a clean install of 12.1.0.2 and allowing the installer to create a typical single instance database.

```
SQL*Plus: Release 12.1.0.2.0 Production on Sun Jul 27 16:43:44 2014
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
```

So, has In-Memory been installed?

Yes, Oracle Database In-Memory is installed. How do I know? Oracle Database In-Memory is not a bolt on

technology to the Oracle Database. It has been seamlessly integrated into the core of the database as a new component of the Shared Global Area (SGA). When the Oracle Database is installed, Oracle Database In-Memory is installed. They are one in the same. You can't unlink it or choose not to install it.

The more important question is if In-Memory is automatically enabled or not?

The answer is **NO**. In order to prove this we are going to need to look at some of the new initialization parameters that control In-Memory.

```
SQL> show parameter INMEMORY
```

NAME	TYPE	VALUE
inmemory_clause_default	string	
inmemory_force	string	DEFAULT
inmemory_max_populate_servers	integer	0
inmemory_query	string	ENABLE
inmemory_size	big integer	0
inmemory_trickle_repopulate_servers_percent	integer	1
optimizer_inmemory_aware	boolean	TRUE

Six new initialization parameters with the `INMEMORY` prefix have been introduced to directly control the different aspects of the new in-memory functionality. There is also a new optimizer parameter that controls whether queries can use the `INMEMORY` or not.

Right now we are only interested in one of these parameters, `INMEMORY_SIZE` to determine if In-Memory is enabled.

Database In-Memory uses an In-Memory column store (IM column store), which is the new component of the SGA, called the In-Memory Area. Data in the IM column store does not reside in the traditional row format used by the Oracle Database; instead it uses a new column format. The size of the IM column store is controlled by the `INMEMORY_SIZE` parameter. As you can see the `INMEMORY_SIZE` parameter is set to 0 and therefore Database In-Memory is not enabled, as there is no IM column store allocated. We can also confirm the In-Memory Area is not allocated by querying `v$sga`.

```
SQL> SELECT name, value FROM v$sga;
```

NAME	VALUE
Fixed Size	7652568
Variable Size	1.1274E+10
Database Buffers	6.4693E+10
Redo Buffers	260780032

Still don't trust me? Let's confirm Database In-Memory is not enabled by examining the feature tracking information. I'm going to force the feature-tracking table to be updated before I query it.

```
SQL> exec dbms_feature_usage_internal.exec_db_usage_sampling(SYSDATE);

PL/SQL procedure successfully completed.

SQL>
SQL> select ul.name, ul.detected_usages
  2  from  dba_feature_usage_statistics ul
  3  where ul.version=(select max(u2.version)
  4  FROM  dba_feature_usage_statistics u2
  5  where u2.name = ul.name and ul.name like 'In-%')
  6  ;

no rows selected
```

If Database In-Memory isn't enabled out of the box how do you enable it?

Enabling Database In-Memory is actually a multi-step process.

Step1: First we must allocated memory for the IM column store by setting the `INMEMORY_SIZE` parameter to a non-zero value that is great than 100MB.

```
ALTER SYSTEM SET inmemory_size = 20G scope=spfile;
```

Since the IM column store is part of the SGA, we also need to ensure the `SGA_TARGET` parameter is set large enough to accommodate the new IM column store and all of the other existing components (buffer cache, shared pool, large pool etc.). By default, the installer set the `SGA_TARGET` to 71G, so I'm going to bump it by 20G.

```
ALTER SYSTEM SET sga_target = 91G scope=spfile;
```

Now let's bounce the database so these parameter changes can take effect.

NAME	TYPE	VALUE
inmemory_clause_default	string	
inmemory_force	string	DEFAULT
inmemory_max_populate_servers	integer	16
inmemory_query	string	ENABLE
inmemory_size	big integer	20G
inmemory_trickle_repopulate_servers_percent	integer	1
optimizer_inmemory_aware	boolean	TRUE

NAME	VALUE
Fixed Size	7656416
Variable Size	1.1274E+10
Database Buffers	6.4693E+10
Redo Buffers	260775936
In-Memory Area	2.1475E+10

As you can see, we now have an IM column store. But Database In-Memory is still not in use because no objects have been populated into the IM column store. To confirm this we can look at two new v\$ views,

`v$IM_SEGMENTS` and `v$IM_USER_SEGMENTS` that indicate what objects are in the In-Memory Column Store.

```
SQL> SELECT v.owner, v.segment_name name, v.populate_status status
  2 FROM   v$im_segments v;

no rows selected
```

We can also confirm it by checking the feature tracking information again.

```
SQL> exec dbms_feature_usage_internal.exec_db_usage_sampling(SYSDATE);

PL/SQL procedure successfully completed.

SQL>
SQL> select ul.name, ul.detected_usages
  2 from   dba_feature_usage_statistics ul
  3 where  ul.version=(select max(u2.version)
  4 FROM   dba_feature_usage_statistics u2
  5 where  u2.name = ul.name and ul.name like 'In-%')
  6 ;

no rows selected
```

Step 2: Unlike a pure In-Memory database, not all of the objects in an Oracle database need to be populated in the IM column store. The IM column store should be populated with the most performance-critical data in the database. Less performance-critical data can reside on lower cost flash or disk. Of course, if your database is small enough, you can populate all of your tables into the IM column store. Only objects with the `INMEMORY` attribute are populated into the IM column store. The `INMEMORY` attribute can be specified on a tablespace, table, (sub)partition, or materialized view. In this case let's enable the `INMEMORY` attribute on one of the user tables `CUSTOMERS`.

```
ALTER TABLE SSB.customers INMEMORY;
```

By default Oracle automatically decides when to populate the table into the In-Memory Column Store. This is also referred to as “on demand”, as Oracle typically populates the table after it has been accessed for the first time. So, let's run a query on the `CUSTOMERS` table.

```
SELECT cust_valid, Count(*)
FROM customers
GROUP BY cust_valid;
```

Now if we check `v$IM_SEGMENTS` we see the `CUSTOMERS` table has been populated in the IM column

store.

```
SQL> SELECT v.owner, v.segment_name name, v.populate_status status
2 FROM v$im_segments v;
```

OWNER	NAME	STATUS
SSB	CUSTOMERS	COMPLETED

If we check the feature tracking information now, we will see Database In-Memory is enabled and being used.

```
SQL> exec dbms_feature_usage_internal.exec_db_usage_sampling(SYSDATE);

PL/SQL procedure successfully completed.

SQL>
SQL> select ul.name, ul.detected_usages
2 from dba_feature_usage_statistics ul
3 where ul.version=(select max(u2.version)
4 FROM dba_feature_usage_statistics u2
5 where u2.name = ul.name and ul.name like 'In-%')
6 ;
```

NAME	DETECTED_USAGES
In-Memory Column Store	3
In-Memory Aggregation	0

In next weeks post, I will explain in a lot more detail all of the different sub-clause of the `INMEMORY` attribute and what all of the columns in the new `v$` views represent.

[+Maria Colgan](#)

Category: Oracle Database In-Memory

Tags: [12c database getting started in-memory performance](#)

[Permanent link to this entry](#)

« [It's here! Oracle...](#) | [Main](#)

Comments:

Post a Comment:

- Name:
- E-Mail:
- URL:
- Notify me by email of new comments
- Remember Information?

- Your Comment:
- HTML Syntax: NOT allowed
- Please answer this simple math question

4 + 16 =

-

About

The Oracle Database In-Memory blog is written by the Oracle product management team and sheds lights on all thing In-Memory.

Search

Enter search term:

Search only this blog

Recent Posts

- [Getting started with Oracle Database In-Memory Part I](#)
- [It's here! Oracle Database In-Memory is officially released](#)
- [In-Memory Column Store versus the Buffer Cache](#)
- [Oracle Database In-Memory & the Optimizer](#)
- [Oracle Database In-Memory Population](#)
- [Oracle Database In-Memory Launch!](#)
- [Preview the Oracle Database In-Memory option](#)

Top Tags

- [12c](#)
- [database](#)
- [getting_started](#)
- [in-memory](#)
- [oow](#)
- [optimizer](#)
- [overview](#)
- [performance](#)
- [population](#)

Categories

- [Oracle](#)
- [Oracle Database In-Memory](#)

Archives

« July 2014

Sun Mon Tue Wed Thu Fri Sat

		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

[Today](#)

Bookmarks

- [Events](#)
- [Oracle Technology Network](#)
- [Oracle.com](#)

Technical Papers

- [Oracle Database In-Memory Data Sheet](#)
- [Oracle Database In-Memory Whitepaper](#)

Menu

- [Blogs Home](#)
- [Weblog](#)
- [Login](#)

Feeds**RSS**

- [All](#)
- [/Oracle](#)
- [/Oracle Database In-Memory](#)
- [Comments](#)

Atom

- [All](#)
- [/Oracle](#)
- [/Oracle Database In-Memory](#)
- [Comments](#)

The views expressed on this blog are those of the author and do not necessarily reflect the views of Oracle.

[Terms of Use](#) | [Your Privacy Rights](#) | [Cookie Preferences](#)