

You can use following steps to extract flash cache contents into an external table. You can also automate this task by creating user equivalency between compute and all storage nodes.

You will need the followings:

1. Access to all storage cells as CellAdmin user
2. A database directory for external table
3. An external table
4. Your Database Unique Name
5. 2 shell scripts (monitor_cache.sh & db_cache.sh)

Let's get started by login into any compute node as Oracle user and select a working directory. For simplicity reason, I will be using '/home/oracle' for external table and as my working directory.

1. Create database directory object for external table (Pick a better location)

```
CREATE OR REPLACE DIRECTORY db_cache as '/home/oracle';
```

2. Create External table using following script

```
CREATE TABLE exadata_cell_cache_usage (  
  dbUniqueName varchar2(35),  
  object_id number,  
  cachedKeepSize number,  
  cachedSize number,  
  hitCount number,  
  missCount number  
)  
ORGANIZATION EXTERNAL  
(  
  TYPE ORACLE_LOADER  
  DEFAULT DIRECTORY db_cache  
  ACCESS PARAMETERS (  
    RECORDS DELIMITED BY NEWLINE  
    FIELDS TERMINATED BY ','  
  )  
  LOCATION ('db_cache.out')  
)  
REJECT LIMIT UNLIMITED;
```

3. Create monitor_cache.sh shell script , you will need to plugin db_unique_name

```
cellcli << EOF
SPOOL db_cache_data.lst
list flashcachecontent attributes dbUniqueName,objectNumber, cachedKeepSize, cachedSize,
hitcount, misscount where dbUniqueName = '<db_unique_name>'
spool off
EXIT;
EOF
```

```
[oracle@ ~]$ cat monitor_cache.sh
cellcli << EOF
SPOOL db_cache_data.lst
list flashcachecontent attributes dbUniqueName,objectNumber, cachedKeepSize, cachedSize, hitcount, misscount where dbUniqueName =
spool off
EXIT;
EOF
```

4. Create db_cache.sh shell script

```
ssh celladmin@cellnode1_IP < monitor_cache.sh
ssh celladmin@cellnode2_IP < monitor_cache.sh
ssh celladmin@cellnode3_IP < monitor_cache.sh
scp celladmin@cellnode1_IP:/home/celladmin/db_cache_data.lst cell01data.lst
scp celladmin@cellnode2_IP:/home/celladmin/db_cache_data.lst cell02data.lst
scp celladmin@cellnode3_IP:/home/celladmin/db_cache_data.lst cell03data.lst
cat cell01data.lst | awk '{print $1,"$2","$3","$4","$5","$6}' >> cell01data.out
cat cell02data.lst | awk '{print $1,"$2","$3","$4","$5","$6}' >> cell02data.out
cat cell03data.lst | awk '{print $1,"$2","$3","$4","$5","$6}' >> cell03data.out
cat cell01data.out cell02data.out cell03data.out > db_cache.out
```

```
[oracle@ ~]$ cat db_cache.sh
ssh celladmin@ < monitor_cache.sh
ssh celladmin@ < monitor_cache.sh
ssh celladmin@ < monitor_cache.sh
scp celladmin@ :/home/celladmin/db_cache_data.lst cell01data.lst
scp celladmin@ :/home/celladmin/db_cache_data.lst cell02data.lst
scp celladmin@ :/home/celladmin/db_cache_data.lst cell03data.lst
cat cell01data.lst | awk '{print $1,"$2","$3","$4","$5","$6}' >> cell01data.out
cat cell02data.lst | awk '{print $1,"$2","$3","$4","$5","$6}' >> cell02data.out
cat cell03data.lst | awk '{print $1,"$2","$3","$4","$5","$6}' >> cell03data.out
cat cell01data.out cell02data.out cell03data.out > db cache.out
```

5. Execute db_cache.sh as Oracle user

```
ksh db_cache.sh
```

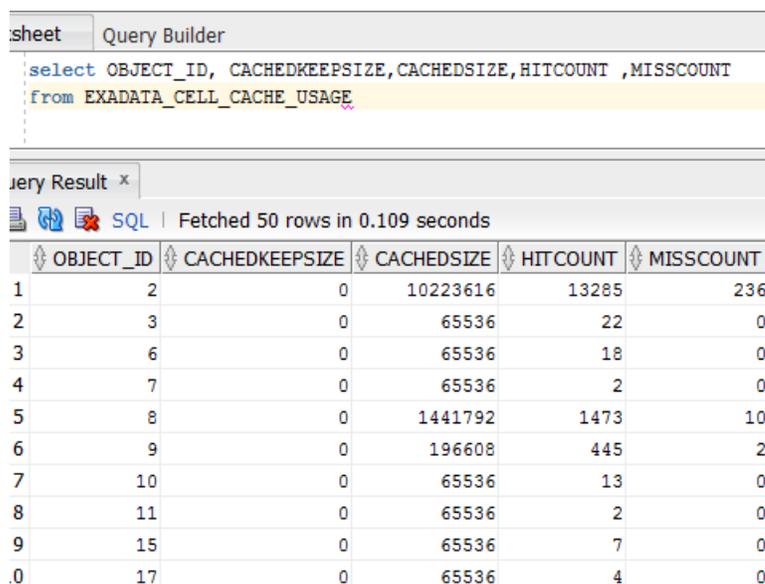
```
[oracle@ ~]$ ksh db_cache.sh  
Pseudo-terminal will not be allocated because stdin is not a terminal.  
celladmin@ 's password: █
```

Notes: - You will have to provide celladmin password for at least 6 times here. Once it's done you should have following files generated for you.

```
-rw-r--r-- 1 oracle oinstall 200 Jun 27 00:57 monitor_cache.sh  
-rw-r--r-- 1 oracle oinstall 673 Jun 27 01:19 db_cache.sh  
-rw-r----- 1 oracle oinstall 224531 Jun 27 01:28 cell01data.lst  
-rw-r----- 1 oracle oinstall 215589 Jun 27 01:28 cell02data.lst  
-rw-r----- 1 oracle oinstall 231492 Jun 27 01:28 cell03data.lst  
-rw-r--r-- 1 oracle oinstall 244459 Jun 27 01:28 cell01data.out  
-rw-r--r-- 1 oracle oinstall 234834 Jun 27 01:28 cell02data.out  
-rw-r--r-- 1 oracle oinstall 261678 Jun 27 01:28 cell03data.out  
-rw-r--r-- 1 oracle oinstall 740971 Jun 27 01:28 db_cache.out
```

6. Now query external table to see cache data.

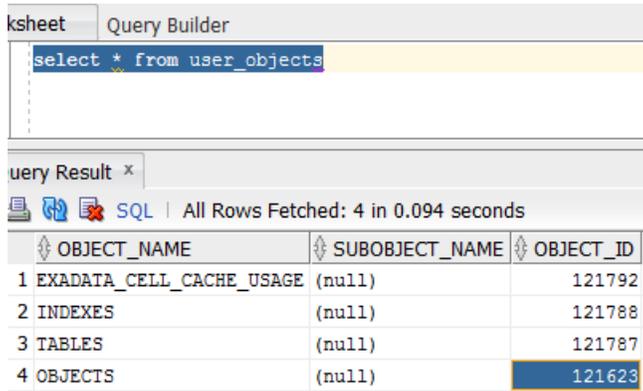
```
Select OBJECT_ID, CACHEDKEEPSIZE, CACHEDSIZE, HITCOUNT ,MISSCOUNT  
from EXADATA_CELL_CACHE_USAGE;
```



OBJECT_ID	CACHEDKEEPSIZE	CACHEDSIZE	HITCOUNT	MISSCOUNT	
1	2	0	10223616	13285	236
2	3	0	65536	22	0
3	6	0	65536	18	0
4	7	0	65536	2	0
5	8	0	1441792	1473	10
6	9	0	196608	445	2
7	10	0	65536	13	0
8	11	0	65536	2	0
9	15	0	65536	7	0
0	17	0	65536	4	0

7. Extract object_id of your table from dba_objects

select * from user_tables;

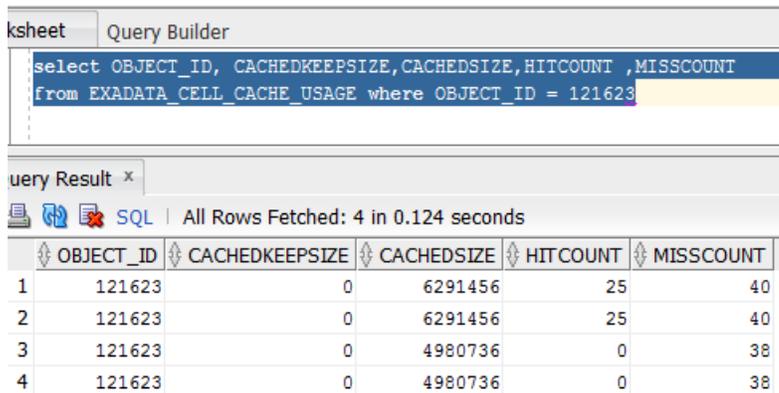


The screenshot shows a SQL Query Builder window with the query `select * from user_objects`. Below the query, the results are displayed in a table with 4 rows. The columns are OBJECT_NAME, SUBOBJECT_NAME, and OBJECT_ID. The fourth row, corresponding to the OBJECTS table, has its OBJECT_ID value, 121623, highlighted in blue.

	OBJECT_NAME	SUBOBJECT_NAME	OBJECT_ID
1	EXADATA_CELL_CACHE_USAGE	(null)	121792
2	INDEXES	(null)	121788
3	TABLES	(null)	121787
4	OBJECTS	(null)	121623

8. Finally you can query exadata_cell_cache_usage to query cache information about your object using object_id.

Select OBJECT_ID, CACHEDKEEPSIZE,CACHEDSIZE,HITCOUNT ,MISSCOUNT
from EXADATA_CELL_CACHE_USAGE where OBJECT_ID = 121623



The screenshot shows a SQL Query Builder window with the query `select OBJECT_ID, CACHEDKEEPSIZE,CACHEDSIZE,HITCOUNT ,MISSCOUNT from EXADATA_CELL_CACHE_USAGE where OBJECT_ID = 121623`. Below the query, the results are displayed in a table with 4 rows. The columns are OBJECT_ID, CACHEDKEEPSIZE, CACHEDSIZE, HITCOUNT, and MISSCOUNT. All rows show the same OBJECT_ID (121623) with varying cache statistics.

	OBJECT_ID	CACHEDKEEPSIZE	CACHEDSIZE	HITCOUNT	MISSCOUNT
1	121623	0	6291456	25	40
2	121623	0	6291456	25	40
3	121623	0	4980736	0	38
4	121623	0	4980736	0	38