The aim of the OLAP cache is to improve the query performance by reading the data out of the cache instead of requesting it from the database. After a query is run for the first time the data set transferred from the database is saved in the storage systems of the OLAP cache. This cache entry can then be taken the next time. As it is a ‘global’ cache, different work processes can use entries of the cache and its data content is stored independently from user sessions.

Local Cache versus Global Cache

The OLAP Processor accesses two types of cache, the local cache and the global cache. The local cache can only keep the data of the current work process (roll memory) and only one user can access this data during that particular work process. Unlike the local cache, the global OLAP cache can be accessed by various query instances, the data content is stored independently from user sessions. When the global cache is switched off, the local cache is still active. At runtime, all drilldown states of the query are cached locally to start with and then stored globally if required. With ‘OLAP cache’ we refer to the global OLAP cache.

Cache Modes

Up to BW7.3, various cache modes are supported for the global cache. These differ in terms of their persistence medium. SAP recommends to use the mode ‘Query Aggregate’ (also known as BLOB/Cluster Enhanced) which is also the default setting. This cache mode has been introduced with BW7.0. For more details see Cache Modes (BW73) and SAP note 1026944.

As of BW7.4, the global cache objects are always stored in the data base, cache modes referring to the main memory do not exist any longer. This mode is called “Cache in database”, it is basically identical with the former mode ‘Query Aggregate’. The following modes are available (in particular it is now possible to switch off the usage of the local cache, see also 2017248):

- Cache mode I: Cache is inactive New data is read during every single navigation step.
- Cache mode 0: Local cache The database cache is deactivated. In this case, the system works exclusively with the local cache.
- Cache mode D: Cache in database
- Cache mode P: Every provider as set in the InfoProvider definition This mode is only available for queries on MultiProviders and CompositeProviders.

More details can be found in the online document under Cache Modes (BW74).

Cache Parameters

The cache parameters are set in Customizing during implementation, you can find the current parameters in the Cache Monitor (transaction RSR_CACHE). The Global Cache Size should normally not be smaller than 300-500 MB. The corresponding basis parameter is rsdb/esm/buffersize_kb (which has to be larger/equal than the OLAP cache). Note that these parameters only relate to the main memory cache and not to the database caches. Please see

- BW73: OLAP: Cache Monitor
- BW75: OLAP Cache Monitor

for further details.
Cache Objects / Benefit

As described in note 1681396, there are two main OT processing blocks during the query runtime: the OLAP engine and the DataManager (DM). The corresponding main statistic events are the following:

- 9000 Data Manager
- 3110 OLAP: Data Selection
- 3200 OLAP: Data Transfer

The cache objects are created in the OLAP engine before most of the OLAP calculations and process units are carried out. In terms of the statistics events, between 'Data Selection' and 'Data Transfer' (see the picture of the page Query Memory Consumption). This means that the usage of the OLAP cache only helps to improve the query runtime when most of the time is spent in the 'DataManager' and/or 'DataSelection'. It also means that all features (e.g. currency conversion, virtual key figures) that are processed before 'Data Transfer' have an impact on the data saved in the cache objects. Hence, if there are changes with this features, the cache objects need to get invalidated.

Cache Invalidation

It's very important to assure that the usage of the cache always leads to the same query result in comparison to a data base access. That implies that cache entries can not be used if there were any changes to the data that is already stored in the cache. For instance, if the query definition has changed decisively, then the cache has to be invalidated. Here a list of some other points:

1. Changes to metadata (e.g. definition of Cube or InfoObjects)
2. Certain changes to the query definition (like adding a new basis key figure)
3. Activation of master data if navigation attributes (or formula variables with substitution from the attribute) are used in the query
4. Activation of hierarchies that are used as a selection or presentation hierarchy in the query
5. Changing or adding exchange rates if currency translation is active in the query.
6. Changes to the InfoProvider data itself, that is, insert, update or delete. Basis cubes can use the feature Update Cache Objects in Delta Process where new transaction data (requests) are added to the cache object.

All of these changes lead to an invalidation of a cache entry. The old cache entry is deleted during the first execution of the query. The system collects all timestamps of all changes and compares the maximum of them with the createtimestamp of the cache. For further details see notes 822302 and 1138864 and Delta Cache.

Limitation of cache objects (Report RSR_CACHE_RSRV_CHECK_ENTRIES does not work any longer as of 7.4)

Since too many entries in the cache administration tables (e.g. RSR_CACHE_DATA_B) can lead to performance problems with Bex Queries, reducing the cache objects - or keeping it into a a certain limit - is somewhat of an important housekeeping activity. The way to regulate this has changed between former Releases and BW >= 74. The details can be found here.

When does the query use an existing cache object?

First the OLAP Engine checks whether valid objects exists, see paragraph 'Cache Invalidation' above. If yes, it is checked whether there is a cache object which contains the requested data. Basically, this means that the navigational state of the query (all filters and exact drilldown) exactly fits to the cache object or only needs subset of the object (which can be derived). Please note that the OLAP Engine only can take cache objects assigned to a certain query. So, one prerequisite for taking a cache object is that the technical name of the query is identical. Hence, even a copy of a query (which reads exactly the same data from the db) cannot take the cache object from the original query.

- If the existing cache objects cannot be used, the query creates a new object. If a subset is derived of an existing cache object, the system also creates a new corresponding object.
- The cache objects depend on the Query Read Mode (see Query Read Mode & Cache). The default mode is the mode H where only the data which is really necessary is retrieved from the database. Hence, also the cache object contains only the minimum of data for a specific navigational state. In the following discussion we assume that the mode H is used(regarding mode A see Query Read Mode & Cache).
- Query property 'Use Selection of Structure Elements'(KIDSEL): If this setting is active, the cache object only contains the data for all key figures displayed in the query. In case a hidden key figure is added to the drilldown, the system has to retrieve the data from the database and creates a new cache object.
- Query contains Variables: the behaviour depends on whether the variable is a Space or State Variable, see SPACE and STATE Variables and also note 1105139.
  - Different values for Space Variables always lead to separate cache objects (derivation not possible).
  - A subset of an cache object can only be taken (derived) when the variable is a State variable and the corresponding characteristic is in the drilldown(in case a Display Hierarchy is active, the hierarchy has to be expanded to the level of this characteristic).
  - Variables in structure elements (Restricted Key Figures) are handled like 'static' variables(space variables). No 'derivation' from an existing cache object is possible, new variable values generate new cache objects.
When a query cannot use the OLAP cache

Basically, a query can use the OLAP cache when the OLAP Engine gets 'notified' when the data, the query is based on, has changed. This is very important since otherwise the query wouldn't display the correct/expected results.

InfoProvider Level

E.g. when running a query based on a InfoCube, the system checks whether new request were loaded into the Provider(table RSDINFOPROVDA TA). If yes, either the old cache objects are invalidated or the so called delta cache mechanism(see Delta Cache) is used. In case e.g. a query is accessing a Virtual InfoProvider based on a Function, the OLAP Engine does not 'know' whether a cache object is still up-to-date and hence, by default, the cache is switched off(but there is a way/workaround(cache validity in transaction RSDIPROP) to make the cache available for such a query, see note 822302 and Example). If BW is based on Hana, properties of InfoProviders can prevent the query of using the cache. E.g. in case an HCPR is base on a HanaView and there is no InfoObject association, a query cannot use the OLAP cache since there is no stable SID processing possible(see also note 2271658 - Design Considerations for Composite Provider).

Overview: Provider which do not use the Cache (by default)

- **Classic BW Providers**
- **Virtual Provider**: see Example
- **ALVL**: planing queries based on an AggregationLevel do not use the cache in contrast to the associated plan buffer query "P!!!1P", see note 1136163 for details.
- **BWonHana/BW4Hana**
  - **HCPR**
    - based on HanaView: caching only possible if all fields are assigned to Infoobjects(used in query), in addition cache validity has to be defined. (NOTE: This feature is not yet released, but still under discussion by SAP Developers.)
    - based on BW Providers, at least one is joined: before BW75, the cache could only be taken in case the right operand of the join does support the Delta Cache (so, e.g. when there is a join with an Infoobject or an ADSO which is not 'cube-like', the query cannot use the cache). As of BW75, this restriction does not exists any longer(it is sufficient when the providers support the cache like an Infoobject which can use the cache(validation timestamp based) but not the delta cache mechanism).
    - **OpenODS View**: similar to an HCPR based on HanaView
    - **ADSO**: if fields(without associated Infoobject) are used in query, the OLAP cache cannot be used(no SID processing possible, see note 2271658).
    - InfoProvider which contain characteristics with Virtual Master Data based on SAP HANA View: in this case, the OLAP cache must be switched off since a query(where e.g. a corresponding navigation attribute is in the drilldown) would otherwise not realize that master data has changed and possibly use an outdated cache objects.

Query Level

There is also a query feature which has an impact on the validity of stored cache objects, see Virtual Key Figures and Characteristics. Since the data can be changed in this user exit, existing cache objects may get outdated(if the logic of the exit somehow depends on time) without a possibility to notify the OLAP Engine. Hence, by default, the cache is switched off(but can be switched on as explained in Cache&Virtual KeyFigues and 2201882).

Query Property: Data Integrity Profile

Basically the Data Integrity Profile defines how the query should behave when the underlying data is changed during query navigation. The data integrity profile provides you with an convenient way to choose a certain behavior without the need to make separate expert settings in the read mode, cache mode, cache usage mode and the InfoProvider grouping. See Query Property: Data Integrity Profile for further details.

Other Topics

- **Query Read Mode & Cache**
- **Query Property: Data Integrity Profile**
- **Delta Cache**
- **Virtual Providers & Query Refresh & Local Cache**
- **BWonHana**: Basically, the database the BW system is based on has no impact on the usage and maintenance of the OLAP cache. The benefit of using the cache always depends on the time the query needs to retrieve the data from the database, see Cache Objects / Benefit.

Documentation

SAP Consulting Notes
- **2529038** Cache cleanup via RSR_CACHE_RSRV_CHECK_ENTRIES fails
- **2017248** Disabling internal OLAP cache buffers (local cache)
- **1924424** BW740 RSRCACHE: Enhanced functionality to disengage Cache Entries
- **2140135** Report RSR_CACHE_RSRV_CHECK_ENTRIES does not work anymore
- **2261297** Delimiting the number of cache entries per single query
- **2760262** Slight differences in results of non-cumulative queries with and without OLAP cache usage
- **2002599** How to proceed in case of Enqueue-Locks caused by the OLAP-Cache
- **1996277** Cache inconsistencies after deleting requests from InfoProvider
- **1136163** Die Query-Einstellungen RSRT -> Eigenschaften
- **822302** Olap-CACHE bei Remote-Provdeern
- **2201882** No Query cache generated when using Virtual Characteristic/Key Figure
- **935656** (BW3x) Cache enqueue attempts cause performance problems
- **959457** Cache enqueue attempt causes performance problem in BI 7.0
- **1107434** (BW7x) - OLAP cache consultation: Long wait times due to locks
- **1026944** New cache mode with no directory for BI 7.0
- **1836616** Shared memory overflow when writing the cache - BRAIN 895
- **1711747** Delimiting the Cache Objects for Plan Data Queries
- **1138864** Delta procedure of the OLAP cache
- **1022589** The query settings RSRT -> Query Mass Maintenance

**SAP Online Documentation**

OLAP: Cache Monitor (BW73 Release)

OLAP Cache Monitor (BW74 Release)

**SAP Support Troubleshooting Guideline**

SSG BW-BEX-OT-CCH