

# Platform Cache In Salesforce

Biswajeet July 6, 2018 4 Comments

### What is Cache?

Cache is a temporary data storing technique. We use Cache to store static data, that is being used frequently in our application for faster performance and better reliability.

### What is Platform Cache?

Platform Cache is a memory layer that stores Salesforce session and org data for later access. When we use Platform Cache, the application can run faster because they store reusable data in memory. Application can quickly access this data; they don't need to duplicate calculations and requests to the database on subsequent transactions. The Platform Cache is like the RAM for the cloud application.

There are two types of Platform Cache:

**Session Cache:** It stores data for individual user sessions and lives alongside a user session. The maximum life of a session is 8 hours. Session cache expires when its specified time-to-live is reached or when the session expires after eight hours, whichever comes first.

**Org Cache:** It stores data that any user in an org reuses. Unlike session cache, org cache is accessible across sessions, requests, and org users and profiles. Org cache expires when its specified time-to-live is reached.

### When can we use Platform Cache?

We can use Platform Cache, when we want to store temporary data to use in code, but do not want to save it in database. Using cached data improves the performance of the application and is faster than performing SOQL queries repetitively, making multiple API calls, or computing complex calculations.

### Platform Cache Considerations:

- Cache isn't persisted. There's no guarantee against data loss.
- Some or all cache is invalidated when you modify an Apex class in your org.
- Data in the cache isn't encrypted.
- Org cache supports concurrent reads and writes across multiple simultaneous Apex transactions.
- Session cache doesn't support asynchronous Apex. Like we can't use future methods or batch Apex with session cache.
- The session cache can store values up to eight hours and the org cache can store values up to 48 hours.

### Platform Cache Limits:

Cache Allocations by Edition:

- Developer Edition (0 MB by Default, but can get 10 MB as a trial).
- Enterprise Edition (10 MB by default)
- Unlimited Edition (30 MB by default)
- Performance Edition (30 MB by default)

Session Cache Limits:

- Maximum size of a single cached item for put() methods (100 KB)
- Maximum local cache size for a partition, per-request (500 KB)
- Minimum developer-assigned time-to-live 300 seconds (5 minutes)
- Maximum developer-assigned time-to-live 28,800 seconds (8 hours)
- Maximum session cache time-to-live 28,800 seconds (8 hours)

Org Cache Limits:

- Maximum size of a single cached item for put() methods (100 KB)
- Maximum local cache size for a partition, per-request (1,000 KB)
- Minimum developer-assigned time-to-live 300 seconds (5 minutes)
- Maximum developer-assigned time-to-live 172,800 seconds (48 hours)
- Default org cache time-to-live 86,400 seconds (24 hours)

### Create Partition of Platform Cache:

Go to Setup | Search "Platform Cache" in Quick Find Box | Click Platform Cache

In developer edition, it is 0 MB space default. But we can get 10 MB on click of "Request Trial Capacity" button.

## Platform Cache Partition

[Help for this Page](#) ?

Platform cache partitions let you segment the org's available cache space. Each partition's capacity is managed independently.

Request temporary Platform Cache trial capacity. Once your trial request is approved, you can allocate capacity to partitions. The maximum number of trial requests is 10 and orgs must wait 90 days between requests.

[Request Trial Capacity](#)

### Org Has Zero Available Platform Cache

New Platform Cache Partition

Namespace Prefix ↑	Name	Label	Default Partition	Allocated Capacity	Created By	Created Date	Last Modified By	Last Modified Date
No records to display.								

Now Click on "New Platform Cache Partition" to create Partition.

▼ Detail

Namespace Prefix	local
Label	myPartition
Name	myPartition
Default Partition	<input checked="" type="checkbox"/>
Description	

▼ Capacity (MB)

Total Available	
Trial	0
Organization	0
Total	0

Session Cache Allocation

Trial	5
Organization	0
Total	5

Org Cache Allocation

Trial	5
Organization	0
Total	5

Total Allocation

Trial	10
Organization	0
Total	10

Save

Cancel

Each partition has one session cache and one org cache segment and you can allocate separate capacity to each segment. Session cache can be used to store data for individual user sessions, and org cache is for data that any users in an org can access.

Partition Has Zero Used Capacity

Recalculate

(Last: 7/6/2018 3:20 PM)

Edit

▼ Details

Namespace Prefix	local	Created By	Biswajeet Samal, 4/14/2018 11:58 AM
Label	myPartition	Created Date	4/14/2018 11:58 AM
Name	myPartition	Last Modified By	Biswajeet Samal, 7/6/2018 3:20 PM
Default Partition	<input checked="" type="checkbox"/>	Last Modified Date	7/6/2018 3:20 PM
Description			

▼ Capacity (MB)

Session Cache Allocation

Trial	5
Organization	0
Total	5

Org Cache Allocation

Trial	5
Organization	0
Total	5

Total Allocation

Trial	10
Organization	0
Total	10

Note: We cannot delete default Partition. If you want to delete partition, you have to uncheck the default partition option.

Put and retrieve data using Platform cache:

To call Partition in apex code we use `namespace.partitionName` `local` as namespace can be used in both scenario if you have namespace enabled or not enable in your org. In below example I'm using "myPartition" partition.

Store and Retrieve Values from the Session Cache:

Session Cache in Apex Class:

```
1 //Add a value to the cache
2 Integer cartValue = 22;
3 Cache.Session.put('local.myPartition.CartValue', cartValue);
4 if (Cache.Session.contains('local.myPartition.CartValue')) {
5     Integer cachedCartValue = (Integer)Cache.Session.get('local.myPartition.CartValue');
6 }
```

To refer to the default partition and the namespace of the invoking class, we can omit the namespace.partition prefix and specify the key name.

```
1 //Use of default partition without namespace and partition
2 Cache.Session.put('CartValue', 22);
3 if (Cache.Session.contains('CartValue')) {
4     Integer cachedCartValue = (DateTime)Cache.Session.get('CartValue');
5 }
```

Session Cache in Visualforce Page:

```
1 <apex:outputText value="{!$Cache.Session.local.myPartition.CartValue}"/>
```

Store and Retrieve Values from the Org Cache:

Org Cache in Apex Class:

```
1
```

```
1 //Add a value to the cache
2 Integer cartValue = 22;
3 Cache.Org.put('local.myPartition.CartValue', cartValue);
4 if (Cache.Org.contains('local.myPartition.CartValue')) {
5     Integer cachedCartValue = (Integer)Cache.Org.get('local.myPartition.CartValue');
6 }

1 //Use of default partition without namespace and partition
2 Cache.Org.put('CartValue', 22);
3 if (Cache.Org.contains('CartValue')) {
4     Integer cachedCartValue = (DateTime)Cache.Org.get('CartValue');
5 }
```

Org Cache in Visualforce Page:

```
1 <apex:outputText value="{!$Cache.Org.local.myPartition.CartValue}"/>
```