

# Caching

Normally, when a workflow executes, Flux will routinely need to access the database to obtain information about the workflow. If a workflow runs in a loop, for example, Flux may need to obtain a copy of the workflow from the database each time the loop executes. This can cause performance problems if the workflows are very large or run very frequently, causing Flux to query the database often and for large amounts of data.

Caching is designed to increase engine performance in these situations. With caching enabled, the first time a workflow executes, a copy of that workflow is stored in memory. The next time Flux needs to obtain the workflow, it can use this in-memory copy, allowing the engine to minimize database queries and improve runtime performance.

**NOTE:** Caching is configured separately on all engines in the cluster. Setting a cache type of "LOCAL" or "NONE" on a single engine will not update any other engine's configuration within the cluster, so be sure to update each engine's configuration appropriately when changing the cache type on any one engine.

- [Local Caching](#)
- [Enabling Caching](#)
- [When Should I Use Caching?](#)

## Local Caching

With local caching, every engine uses its own cache. If the engine is participating in a cluster, other engines in the cluster cannot access this cache. For this reason, the local cache is best used when the engine is not running in a cluster, or when you do not want clustered engines to share information across the network.

To enable the local cache, set the `CACHE_TYPE` engine configuration option like so:

```
CACHE_TYPE=LOCAL
```

## Enabling Caching

Local caching is disabled on all engines by default. To enable caching, set the `CACHE_TYPE` engine configuration option to "LOCAL":

```
CACHE_TYPE=LOCAL
```

## When Should I Use Caching?

Caching is most effective when you are using large, complex workflows that run frequently in a loop. Small workflows and workflows that run infrequently (or run only once) are not typically affected by caching – for these cases, we suggest disabling caching to reduce memory usage on the system.

Additionally, if you are noticing a large number of queries to the database for your workflows – especially to the `FLUX_VARIABLE` table – enabling caching may help increase the performance for these workflows and reduce the load on the database. For this reason, we recommend enabling caching if you see a heavy load on the `FLUX_VARIABLE` table, or other queries related to your workflows.