

# Quarkus - Infinispan Embedded

Infinispan is an elastically scalable in-memory data store that you can embed directly in your application.

Check out the [Infinispan documentation](#) to find out more about the Infinispan project.



This technology is considered preview.

In *preview*, backward compatibility and presence in the ecosystem is not guaranteed. Specific improvements might require to change configuration or APIs and plans to become *stable* are under way. Feedback is welcome on our [mailing list](#) or as issues in our [GitHub issue tracker](#).

For a full list of possible extension statuses, check our [FAQ entry](#).

## Adding the Infinispan Embedded Extension

After you set up your Quarkus project, run the following command from the base directory:

```
./mvnw quarkus:add-extension -Dextensions="infinispan-embedded"
```

The command adds the following dependency to your `pom.xml`:

```
<dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-infinispan-embedded</artifactId>
</dependency>
```

## Feature Support

The Infinispan embedded extension offers core caching functionality that includes clustered caches, off-heap memory, data persistence, and transactions.



The Infinispan embedded extension does not currently support indexing capabilities.

## Transactions

Configure Infinispan caches for transactional operations with a specific `TransactionManagerLookup`, as follows:

```
<local-cache name="quarkus-transaction">
  <transaction transaction-manager-lookup=
"org.infinispan.transaction.lookup.JBossStandaloneJTAManagerLookup"
/>
</local-cache>
```

For more information, see the Quarkus [Transaction Guide](#).

## Native Limitations

When running Infinispan in native mode, some limitations apply:

- JMX management is not supported. + GraalVM does not allow native VM interfaces.
- **UDP/Multicast** is not supported. + You must use **TCP** transport and a non-UDP based membership protocol with JGroups.

## Injection (CDI)

The Infinispan embedded extension provides injection capabilities so you do not have to configure and start caches manually.



Additional injection functionality will be available in future versions.

### EmbeddedCacheManager

This is the main entry point to configure and obtain caches.

## Configuration Reference

Configuration property fixed at build time - All other configuration properties are overridable at runtime

Configuration property	Type	Default
<code>quarkus.infinispan-embedded.xml-config</code>		
The configured Infinispan embedded xml file which is used by the managed EmbeddedCacheManager and its Caches	string	



JMX elements are disabled during native runtime so you can use the same configuration file in JVM and native modes.