

# DCM 2.0 setup guide

This guide explains how to set up DCM 2.0. DCM 2.0 is an ongoing effort to separate the Process Engine from the Runtime and to support more DCM features natively in Blueriq. If you wish to set up DCM 2.0 for a Blueriq instance that is installed with the installer, you need to enable the DCM Lists feature during installation. It is enabled by default. Installing the DCM Lists feature installs and configures some tools that are necessary for DCM 2.0.

DCM 2.0 is not compatible with the current DCM (1.0) implementation. When you setup DCM 2.0, you can not run DCM 1.0 project anymore, and vice versa.

## Set up message broker

If Blueriq is installed with the installer with the DCM Lists feature enabled, a message broker has already been installed and all necessary queues and exchanges have already been configured. Otherwise, please install a supported message broker. More information on how to install a supported message broker can be found on the [Queue](#) page. For instructions on how to configure the message broker for DCM 2.0, please refer to the [Configuration for the Case Engine \[deprecated\]](#) page.

## Set up MongoDB

If Blueriq is installed with the installer with the DCM Lists feature enabled, MongoDB has already been installed. Otherwise, please install a supported version of MongoDB and add a user and a database for the Case Engine. More information on how to install MongoDB can be found on the [Document database MongoDB](#) page.

## Runtime configuration

After all necessary tools have been installed and configured, some Runtime configuration needs to be added. As the Case Engine is still part of the Runtime for now, all configuration files described below should be placed in the Runtime configuration folder.

### Case Engine user

In order for the Runtime to be able to communicate with the Case Engine, a user needs to be configured for the Case Engine. If the development-tools profile is enabled, a user can be added in the Authorisation settings in the Runtime dashboard. Otherwise, a user can be created manually by adding it to the users.properties file:

#### users.properties

```
# User for Case Engine
caseengine={noop}caseengine,case-engine
```

### Case Engine

For the Case Engine, it is required to set some properties for the message broker and for MongoDB. Below is an example of all required properties. This example assumes that Blueriq has been installed with the installer with default settings and the DCM Lists feature enabled. Otherwise, please adapt the example to your environment.

## application-case-engine.properties

```
# RabbitMQ - Case Engine
blueriq.case.engine.rabbitmq.host=localhost
blueriq.case.engine.rabbitmq.port=13089
blueriq.case.engine.rabbitmq.username=blueriq
blueriq.case.engine.rabbitmq.password=welcome
blueriq.case.engine.rabbitmq.virtualHost=blueriq
blueriq.case.engine.rabbitmq.queueNames=dcmEventsQueue

# MongoDB - Case Engine
blueriq.case.engine.data.mongodb.host=localhost
blueriq.case.engine.data.mongodb.port=13088
blueriq.case.engine.data.mongodb.database=caseEngine
blueriq.case.engine.data.mongodb.username=blueriq
blueriq.case.engine.data.mongodb.password=welcome
blueriq.case.engine.data.mongodb.authenticationDatabase=caseEngine

# RabbitMQ - Process Engine
blueriq.process.engine.v2.rabbitmq.host=localhost
blueriq.process.engine.v2.rabbitmq.port=13089
blueriq.process.engine.v2.rabbitmq.username=blueriq
blueriq.process.engine.v2.rabbitmq.password=welcome
blueriq.process.engine.v2.rabbitmq.virtualHost=blueriq
blueriq.process.engine.v2.rabbitmq.exchangeName=dcmTasksEvents

# RabbitMQ - Case Engine Scheduler
blueriq.case.engine.scheduler.rabbitmq.host=localhost
blueriq.case.engine.scheduler.rabbitmq.port=13089
blueriq.case.engine.scheduler.rabbitmq.username=blueriq
blueriq.case.engine.scheduler.rabbitmq.password=welcome
blueriq.case.engine.scheduler.rabbitmq.virtualHost=/
blueriq.case.engine.scheduler.rabbitmq.ssl.enabled=false
blueriq.case.engine.scheduler.rabbitmq.exchangeName=dcmScheduledEvents

# MongoDB - Caselocking
blueriq.locking.mongodb.host=localhost
blueriq.locking.mongodb.port=13088
blueriq.locking.mongodb.database=caseEngineLocks
blueriq.locking.mongodb.username=blueriq
blueriq.locking.mongodb.password=welcome
blueriq.locking.mongodb.authenticationDatabase=caseEngine
```



For a detailed overview of all Case Engine properties, please refer to the [Case Engine Properties](#) page.

## Case Engine Client

For the Case Engine Client, it is required to set properties for the message broker and the case engine (which can be the same runtime). Below is an example of all required properties.

This example assumes that Blueriq has been installed with the installer with default settings and the DCM Lists feature enabled. Otherwise, please adapt the example to your environment.

#### application-case-engine-client.properties

```
# RabbitMQ - Case Engine Client
blueriq.dcm.rabbitmq.host=localhost
blueriq.dcm.rabbitmq.port=13089
blueriq.dcm.rabbitmq.username=blueriq
blueriq.dcm.rabbitmq.password=welcome
blueriq.dcm.rabbitmq.exchangeName=dcmEvents
blueriq.dcm.rabbitmq.queueNames=dcmTasksEventsQueue
blueriq.dcm.rabbitmq.virtualHost=blueriq

# Case Engine
# As the Case Engine is still part of the Runtime, this URL currently simply points to the Runtime
blueriq.dcm.case-engine.url=http://localhost:94/Runtime/server
# The user and password of a user with the case-engine role
blueriq.dcm.case-engine.username=caseengine
blueriq.dcm.case-engine.password=caseengine
```



For a detailed overview of all Case Engine Client properties, please refer to the [Case Engine Client Properties](#) page.

## Profiles

Once all properties have been set, the final step for setting up DCM 2.0 is to enable the `case-engine` profile and the `case-engine-client` profile in the `bootstrap.properties` file. After that, restart your Runtime and you should be good to go!

## Next steps

DCM 2.0 comes with a set of new services and containers that communicate with the case engine. These services and containers have the `DCM_` prefix. You can find them in the [Container types](#) and [Service call types](#), or see the [DCM library](#). You need to use these in your DCM 2.0 project. For more information, please contact support.

## Other resources

[Case Engine](#)

[Case Engine Client Component](#)