







This repository

Search or type a command


[Explore](#) [Gist](#) [Blog](#) [Help](#)

 rofr   


 [DevrexLabs / OrigoDB.Workshop](#)

 Unwatch

1

 Star

0

 Fork

0

## 05 Configuration

rofr edited this page an hour ago · 4 commits

### Materials

- [Engine Documentation](#)
- [Slides](#)
- [Protobuf module repo](#)

### Exercise 1

In this exercise you will modify an existing model to use ProtoBuf formatting for the journal and snapshots. The current configuration uses BinaryFormatter serialization. For each operation, take a note of the duration printed to the console.

The console app relies on command line arguments. If you're uncomfortable with the command line, you can enter arguments under Project/Properties/Debug or just hardcode args into the console app. Better yet why not just use scripts?

#### Task 1 - BinaryFormatter

1. Open and examine the solution `Ex01\FreeDb.sln`
2. Build the model by running the console application using the compressed bz2 file found under Resources as input.
3. Quit the application and run again, this time loading the from journal previously built.
4. Take a snapshot
5. Quit and run again, this time loading from snapshot
6. Take a note of the size of the journal and snapshot

#### Task 2 - Configure with protobuf

1. Add a nuget reference to origodb.protobuf in the FreeDb.App project
2. Create a method that builds an EngineConfiguration instance configured for protobuf snapshotting using code only configuration. It could look like this:

```
static EngineConfiguration CreateProtobufConfig()
{
    var config = new EngineConfiguration();
    var typeModel = TypeModel.Create();

    typeModel.Add(typeof(FreeDbModel), false)
        .Add(1, "_tracks");

    typeModel.Add(typeof(Track), false)
        .Add(1, "Artist")
        .Add(2, "Title")
        .Add(3, "Album")
        .Add(4, "Genre");

    ProtoBufFormatter.ConfigureSnapshots<FreeDbModel>(config, typeModel);
    return config;
}
```

3. Modify the Build and Load methods to use the protobuf configuration
4. Repeat the runs and compare sizes / timings

▼ Pages 8

[01 Introduction](#)

[02 Modeling](#)

[03 Testing](#)

[04 Hosting](#)

[05 Configuration](#)

[06 Immutability](#)

[07 Server](#)

[Home](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/DevrexLabs/OrigoDB.Workshop/wiki/05-Configuration> 

 Clone in Desktop

+ Add a custom footer

