Many thanks to **Eben Roux** for code documentation and review.

Chapter 6 Source code.

Chapter 6 : SQL Server Example

In this chapter, we will be focusing on snippets about SQL Server examples. We will discuss queuing in SQL Server. More advanced features for Entity Frameworks will be discussed, as well as MVC-EF examples. This chapter is for those developers working with SQL Server and Entity frameworks with NServiceBus.

The Source code in this section:

In this section, we will be using the **PubSub-SQL** solution:

1) MyMessages – A payment message used for the projects.

2) MyPublisher – A project that publishes EventMessages to the SQLEXPRESS nservicebus tables for pub-sub.

3) Subscriber1 – A project for subscribing to the nservicebus Subscriber1 tables to read and handle EventMessages.

4) Subscriber2 – A project for subscribing to the nservicebus Subscriber2 tables to read and handle IMyEvent messages.

This is a publish-subscibe solution to publishing messages that Subscriber1 handles one type of messages, and Subscriber2 processes a different type of messages. These were ran in VS2012 in Windows Server 2012, with MSMQ, DTC, NServiceBus references, and SQL Server 2012 Express LocalDB installed.

A "nservicebus" database must be present in the SQL Server.

(Worked)

In this section, we will be using the **MVCApp - ModelFirst** solution:

1) MVCApp - ModelFirst – This project shows putting hardcoded messages and creating the MVCApp1.AppContext database to show EF basics with entity models, not entity code.

These were ran in VS2012 in Windows Server 2012, with MSMQ, DTC, NServiceBus references, and SQL Server 2012 Express LocalDB installed.

The **MVCApp1.AppContext** database must exist. The tables are created by running the "Model1.edmx.sql" script to create the **Paymessage** table.

(Worked)

In this section, we will be using the **MVCApp1** solution:

1) MVCApp1 – This project shows creating a table and putting hardcoded messages in it from entity code, not entity models, created from the project's namespace to create MVCApp1.AppContext, where MVCApp1 is the namespace of the project.

These were ran in VS2012 in Windows Server 2012, with MSMQ, DTC, NServiceBus references, and SQL Server 2012 Express LocalDB installed.

This program should create the **MVCApp1.AppContext** database.

(Worked)

In this section, we will be using the **ConsoleDbContext - Config** solution:

1) ConsoleDbContext – This project shows creating a “nservicebus” table and putting Unicast configs into a UnicastBusConfigDBs table and message endpoints in the MessageEndpointMappingDBs table.

This is to demonstrate reading and writing material in the “nservicebus” database tables using code only EF.

2) ReadEFConfig – This project reads the entries from the ReadEFConfig to demonstrate using configs from tables.

These were ran in VS2012 in Windows Server 2012, with MSMQ, DTC, NServiceBus references, and SQL Server 2012 Express LocalDB installed.

Worked.

In this section, we will be using the **/BasicWCF2/MVCApp-WCF** solution. In this section, we are just looking at the UnitTestHandlers.

All other code in this directory was built for scratch pads, please use at your own risk.