You may want to create **an offline *EntityManager*** during testing and development or to run the application when disconnected from the network and database.

The most frequently used *EntityManager* constructor is declared like this:

```
public EntityManager(
  bool shouldConnect=true, // Whether to start connecting to the server immediately
  string dataSourceExtension=null,
  IbEm.EntityServiceOption entityServiceOption=IbEm.EntityServiceOption.UseDefaultService,
  string compositionContextName=null);
```

```
Public Sub New( _
 Optional ByVal shouldConnect As Boolean =True, _
 Optional ByVal dataSourceExtension As String =Nothing, _
 Optional ByVal entityServiceOption As IbEm.EntityServiceOption =IbEm.EntityServiceOption.UseDefaultService, _
 Optional ByVal compositionContextName As String =Nothing)
End Sub
```

All of the parameters are optional. The parameter of interest in this topic is *shouldConnect* which is *true* by default. That means that the *manager* automatically starts connecting to the *EntityServer*.

You'll probably do just that in production. But you might want to set it false.

Maybe your application doesn't have a connection to the server at right now. Maybe the user has launched the application while several miles high on an airplane. You could design the application to run disconnected until a network connection can physically be established.

Maybe the application shouldn't connect at all during this session. Many automated tests require the presence of an *EntityManager*, perhaps one populated with test entities. They don't need - and don't want - to talk to the server or the database. That would only make the tests slower and vulnerable to interrupted connections and disabled databases.

Many programmers run disconnected when running the application in development for the same reasons; there might not even be a database in the early stages of development.

Of course you won't be able to save changes while disconnected. Fortunately, most automated tests don't need to save and the savvy developer can easily divert save requests while in development mode.

You can create a disconnected *EntityManager* like this:

```
manager = new NorthwindEntities(shouldConnect: false);
// ...
manager.ConnectAsync(connectCallback); // Time to connect
```

```
manager = New NorthwindEntities(shouldConnect:= False)
' ...
manager.ConnectAsync(connectCallback) ' Time to connect
```