

Contents

- [Overview](#)
- [Connection interception](#)
- [Command interception](#)

You can use the command and connection interception capabilities of Entity Framework 6 in your DevForce application.

Overview

Entity Framework 6 introduced new [interception capabilities](#) "to allow external code to observe and potentially intercept EF operations." A common use of interception is to add logging and simple profiling of low-level EF actions. In advanced scenarios you can use it to set connection or command level information, such as SQL Server context_info or application role.

You can use these interception capabilities in any DevForce application using EF 6, regardless of whether the model was written Code First or is EDMX-based. For n-tier applications, the interception must be done on the EntityServer side of your application.

Connection interception

To intercept connection-level operations, add a class implementing *IDbConnectionInterceptor*:

```
public class EFConnectionInterceptor : System.Data.Entity.Infrastructure.Interception.IDbConnectionInterceptor {  
    ...  
}
```

The interface defines a number of interception points for transactions, configuration, and connections.

You must also register your interceptor for EF to use it:

```
System.Data.Entity.Infrastructure.Interception.DbInterception.Add(new EFConnectionInterceptor());
```

Command interception

To intercept command-level operations, you can either add a class implementing *IDbCommandInterceptor* or sub-class the *DbCommandInterceptor* to override specific operations. Each provides before and after interception for queries and non-query commands.

```
public class EFCommandInterceptor : System.Data.Entity.Infrastructure.Interception.IDbCommandInterceptor {  
    ...  
}
```

Be sure to register this interceptor too:

```
System.Data.Entity.Infrastructure.Interception.DbInterception.Add(new EFCommandInterceptor());
```