

Contents

- [The DebugLog](#)
- [Add error handling](#)
- [Learn More](#)
- [Prerequisites](#)

Part 5: Coping with Trouble - In Part 5 of this series we'll work with the application from [Part 4](#) to add error handling.

This video was recorded using DevForce 2010. Download the sample code below for updated techniques in DevForce 2012.

- **Platform:** Silverlight
- **Language:** C#, VB
- **Download:** [Tour of DevForce Silverlight](#)

The DebugLog

When stuff happens one of the first things you should do is look at the DevForce *DebugLog*. DevForce always generates [tracing messages](#) as it runs, and by default writes those messages on the [EntityServer](#) to a file named DevForceDebugLog.xml. When a problem occurs in your application, the messages in the log can often help to diagnose the problem.

As shown during the video, if we introduce an error (here we modified the database connection string) a message will be recorded in the log file indicating the problem:

```
DataSourceKey: NorthwindIBEntities found in assembly: SimpleSteps, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=null
CompositionContext: '-IbDefault-' - Probed for any 'EntityServerErrorInterceptor' and found
'IdeaBlade.EntityModel.Server.EntityServerErrorInterceptor'.
Caught exception: System.Data.EntityException: The underlying provider failed on Open. --->
System.Data.SqlClient.SqlException: Cannot open database "NorthwindIBxxx" requested by the login. The
```

Add error handling

Every application should have error handling, and we've been remiss in not adding it earlier. When using [asynchronous queries and saves](#) in the task-based asynchronous API, we can add a try/catch to handle the problem, just as we would with synchronous operations.

```
try {
    var results = await query.ExecuteAsync();
    results.ForEach(Employees.Add);
    CurrentEmployee = Employees.FirstOrDefault();
} catch (Exception e) {
    throw new InvalidOperationException("Query failed: " + e.Message);
}

Try
Dim results = Await query.ExecuteAsync()
results.ForEach(Sub(emp) Employees.Add(emp))
CurrentEmployee = Employees.FirstOrDefault
Catch e As Exception
Throw New InvalidOperationException("Query failed: " + Convert.ToString(e.Message))
End Try
```

Learn More

Here are some links for more detail on what we've covered here, and additional information we hope you find useful.

- [Error handling with asynchronous operations](#)
- [Trace logging](#)
- [Understand logged info](#)
- [Troubleshooting](#)
- [More Silverlight resources](#)

Prerequisites

The user interface for the application built during this tour uses a DataForm component supplied by the Silverlight 5 Toolkit (different from the Silverlight 5 Tools!). You can download the Toolkit here:

<http://silverlight.codeplex.com/>