

Contents

- [Fix UI problems](#)
- [Fix a latent bug with Add](#)
- [Learn More](#)
- [Prerequisites](#)

Part 4: Tidying up - In Part 4 of this series we'll further refine the work we did in [Part 3](#) to address some infelicities in the user experience and fix a bug or two.

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](#)

Fix UI problems

If you've been following along in this series you've noticed a few user experience issues: the screen is empty at launch and the form appears to be disabled as you navigate from employee to employee.

The first problem was solved with a simple fix to our code: when the *Employees* collection finished loading asynchronously we weren't setting the *CurrentEmployee* property and raising a change notification.

```
var results = await query.ExecuteAsync();
results.ForEach(Employees.Add);
CurrentEmployee = Employees.FirstOrDefault();
private Employee _currentEmployee;
public Employee CurrentEmployee {
    get { return _currentEmployee; }
    set { _currentEmployee = value; RaisePropertyChanged("CurrentEmployee"); }
}
```

```
Dim results = Await query.ExecuteAsync()
results.ForEach(Sub(emp) Employees.Add(emp))
CurrentEmployee = Employees.FirstOrDefault
private Employee _currentEmployee
public Employee CurrentEmployee
Get
Return _currentEmployee
End Get
Set
_currentEmployee = value
RaisePropertyChanged("CurrentEmployee")
End Set
```

The disabled appearance of the *DataForm* buttons was solved with a bit of a hack. Follow the video and see the downloaded code for how a second busy indicator was added while *Employee.Orders* are fetched.

It would also be nice to have grid splitters so that the sections of the screen aren't fixed sizes. Easy with the *GridSplitter* control:

```
<sdk:GridSplitter Grid.Row="1" Height="5" HorizontalAlignment="Stretch"
    VerticalAlignment="Top" ShowsPreview="True"/>
<sdk:GridSplitter Grid.Row="2" Height="5" HorizontalAlignment="Stretch"
    VerticalAlignment="Top" ShowsPreview="True"/>
```

```
<sdk:GridSplitter Grid.Row="1" Height="5" HorizontalAlignment="Stretch"
    VerticalAlignment="Top" ShowsPreview="True"/>
<sdk:GridSplitter Grid.Row="2" Height="5" HorizontalAlignment="Stretch"
    VerticalAlignment="Top" ShowsPreview="True"/>
```

The next usability improvement would be a vertical scroll bar for the logged messages. Let's do that by swapping the *ItemsControl* for a *ListBox*.

```
<ListBox ItemsSource="{Binding Log}" Grid.Row="2" />
```

```
<ListBox ItemsSource="{Binding Log}" Grid.Row="2" />
```

Fix a latent bug with *Add*

There's also a bug here with our *Add* logic. You may not have run into it yet, but the *CollectionChanged* add logic is also firing when the *Employees* collection is first loaded. We obviously don't need to add the queried employees to the [EntityManager](#), since they're already in the [entity cache](#). We can fix this by checking the [EntityState](#) of the new entity: if it's detached that tells us the *EntityManager* doesn't yet know about it and we can safely add it to the manager.

```
private void Add(System.Collections.Generic.IEnumerable<Employee> employees) {
    if (employees.Any(emp => emp.EntityAspect.EntityState.IsDetached())) {
        _mgr.AddEntities(employees); // simply add to manager in bare state
    }
}

Private Sub Add(ByVal employees As System.Collections.Generic.IEnumerable(Of Employee))
If employees.Any(Function(emp) emp.EntityAspect.EntityState.IsDetached()) Then
    _mgr.AddEntities(employees) ' simply add to manager in bare state
End If
End Sub
```

Learn More

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

- [EntityManager Overview](#)
- [Asynchronous queries and saves](#)
- [DevForce entities and the UI](#)
- [Entity internals](#)
- [The entity cache](#)
- [Validation overview](#)
- [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/>