

# Building End-to-End Multi-Client Service Oriented Applications

Module 14

The Car Rental Desktop – Part 2

*Development of a complete view and unit-testing*



# Highlights

- **The “Maintain Cars” view/viewmodel**
  - Will use a viewmodel-first approach
  - Deterministic initialization
  - Incorporate service factory to obtain proxies on-demand
  - Events to fire messages to view
  - Contain a “child” viewmodel for single car-edit
- **Unit testing of a viewmodel**

# Summary

- **Every view had a viewmodel – strict adherence**
- **Viewmodels offered deterministic initialization**
  - Remember, views get reset but the viewmodel does not
- **The service factory offers service proxies on-demand**
  - ViewModelBase pattern offers easy disposability wrapping
- **View base class allows easy-to-access methods for wiring/unwiring viewmodel events**
  - Eliminates possibilities of memory leaks
- **The other views/viewmodels use identical patterns**
  - Are significantly simpler than MaintainCarsViewModel/View

**End of Module**