

---

## Connect Smart Client Applications with Windows Communication Foundation

Brian Noyes

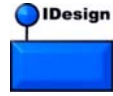
Chief Architect  
IDesign, Inc. ([www.idesign.net](http://www.idesign.net))



---

### About Brian

- Chief Architect, IDesign Inc. ([www.idesign.net](http://www.idesign.net))
- Microsoft Regional Director / MVP Solution Architect
- Writing
  - Data Binding with Windows Forms 2.0, Addison Wesley, January 2006
  - Smart Client Deployment with ClickOnce, Addison Wesley, Summer 2006
  - MSDN Magazine, MSDN Online, CoDe Magazine, The Server Side .NET, asp.netPRO, Visual Studio Magazine
- Speaking
  - Microsoft TechEd US, Europe, Malaysia, Visual Studio Connections, DevTeach, INETA Speakers Bureau, MSDN Webcasts
- Participates in Microsoft Design Reviews
- E-mail: [brian.noyes@idesign.net](mailto:brian.noyes@idesign.net)
- Blog: <http://www.softinsight.com/bnoyes>



## Agenda

---

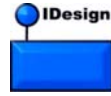
- [WCF Overview](#)
- Connecting Applications with WCF
- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer



## WCF Overview

---

- Next generation distributed communications API / Infrastructure
- Empowers development of service-oriented applications
- Part of WinFx
  - WCF - Communications
  - WPF - Presentation
  - WinWF - Workflow
- Scheduled for delivery in ??? (Guestimate early 2007)
- Ships with Windows Vista
- Available for WinXP, Windows Server 2003



## WCF Overview

---

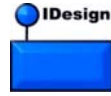
- Replaces the programming model of (but does not make obsolete):
  - Web services
  - .NET Remoting
  - Enterprise Services
  - MSMQ
- Uses the existing OS protocols and capabilities of those technologies under the covers
  - DCOM, TCP, HTTP, SOAP, DTC, MSMQ, etc.



## Agenda

---

- WCF Overview
- [Connecting Applications with WCF](#)
- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer



## Connecting Applications with WCF

---

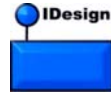
- Design, develop, deploy services
  - IIS hosted (IIS 5.1, 6, 7)
  - Self-hosted (Any exe)
  - Windows Activation Service (IIS 7)
- Design, develop, deploy consumers (clients)
  - Windows clients
  - Web clients
  - Other services
- Choose programming model:
  - Declarative (config file)
  - Programmatic (System.ServiceModel classes)
  - Mixed



## Keys to WCF – A-B-C

---

- Address
  - The place to find the service
- Binding
  - The protocols that the service exposes for connecting
  - Policy for connecting (security, sessions, etc.)
- Contract
  - The operations and data that the service exposes



## WCF Service By The Numbers

---

- Create a service contract (interface)
- Create data contracts for custom types
- Create a service implementation that implements the contract
- Create config file entries for declarative aspects
- Host it
  - Svc file in IIS
  - ServiceHost instance in exe



## WCF Client By The Numbers

---

- Generate proxy to the service
  - svcutil.exe <http://serviceserver/service.svc>
  - svcutil.exe net.tcp://serviceserver:2112/
- Add proxy and config file to client
- Create instance of proxy and call methods through service contract
- Implement callback object if appropriate

## Agenda

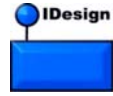
---

- WCF Overview
- Connecting Applications with WCF
- [Channel Selection](#)
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer

## Standard Bindings

---

Binding	Description
BasicHttpBinding	HTTP(S), text encoding, security, not much else
WSHttpBinding / WSDualHttpBinding	HTTP(S), text or MTOM encoding, security, sessions, reliable, transactions
NetTcpBinding	TCP, binary, security, sessions, reliable, transactions
NetNamedPipeBinding	Named pipe, binary, security, sessions, reliable, transactions
NetMsmqBinding	MSMQ, binary, security, sessions, reliable, transactions
MsmqIntegrationBinding	MSMQ, text, sessions, reliable, transactions
NetPeerTcpBinding	Peer-to-Peer communications



## Agenda

---

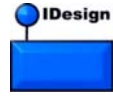
- WCF Overview
- Connecting Applications with WCF
- Channel Selection
- **Asynchronous Calls**
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer



## Asynchronous calls

---

- Service calls can:
  - Take a "long" time
  - Be one-way
  - Call back into client
  - Be re-entrant
- Need to make calls asynchronously
- Handle completion / error handling
- Display results in UI
  - Completion events/callbacks on background thread



## Asynchronous calls

---

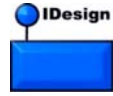
- BackgroundWorker component
  - Clean event driven model
  - Executes designated method on thread from thread pool
  - Handles marshaling completion/progress to UI thread



## Asynchronous Calls

---

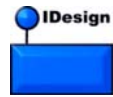
- Manual Async Calls
  - Delegate based
  - BeginXXX/EndXXX method variants
    - ▲ Client side: Svcutil /a switch
    - ▲ Service side: OperationContract.AsyncPattern = true
  - Register callback method
  - Gets called on background thread, must marshal results yourself to UI thread (ISynchronizeInvoke)



## Avoiding Deadlocks

---

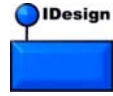
- Run ServiceHost on non-UI thread
  - Start from async method
  - ServiceBehavior.RunOnUIThread = false
- Invoke service methods that may cause callback from worker thread



## Agenda

---

- WCF Overview
- Connecting Applications with WCF
- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer



## Client Initiated Transactions

---

- Indigo transactions built on top of System.Transactions + WS-\*
- Initiate a transaction in client code
- Transaction is flowed to all services called that support transactions

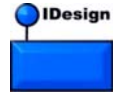
```
using (TransactionScope scope =
    new TransactionScope())
{
    // Call Service1 - transaction flowed to service1
    // Call Service2 - transaction flowed to service2
    scope.Complete();
} // All commit or all rollback
```



## Enabling Transactions

---

- Transactions do not flow by default
- Requirements:
  - Enclose service calls in transaction scope on client
  - transactionFlow = true on binding
  - [TransactionFlow] attribute on service contract methods
    - ▲ Optional – specify transaction scoping options
  - OperationBehavior attribute on service methods
    - ▲ TransactionFlowRequired = true
    - ▲ TransactionAutoComplete = true



## Enabling Transactions

---

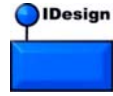
- Enable WS-Atomic Transactions on DTC:
  - `xws_reg -wsat+` at command line



## Agenda

---

- WCF Overview
- Connecting Applications with WCF
- Channel Selection
- Asynchronous Calls
- Transactions
- [Sessions](#)
- Callbacks
- Security
- Peer-to-Peer



## Sessions

---

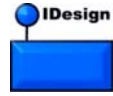
- Often service calls are related to one another
  - Start Order
  - Add Order Items
  - Submit Order
- Individual service calls are just messages
  - Underlying protocol may not support sessions
- Need to relate multiple messages together
- Sessions are maintained through shared service instances on service side



## Reliable Session Channels

---

- Session: Maintain state over a series of service calls
  - [ServiceContract(Session=true)] on contract
- Reliable session: retries, delivery assurance, delivery order
  - Add reliableSession binding element to binding



## Agenda

---

- WCF Overview
- Connecting Applications with WCF
- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer



## Callbacks

---

- Service can call client
  - Events
- Need to register client callback contract on service
  - `ServiceContract.CallbackContract`
- Need to obtain callback channel instance in service
  - `OperationContext.GetCallbackChannel<T>`
- Callbacks happen asynchronously from client perspective

## Agenda

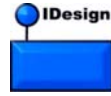
---

- WCF Overview
- Connecting Applications with WCF
- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer

## Security

---

- Three kinds: Transport, Message, Mixed
- Must match the requirements of the service
- Credentials:
  - Windows
  - X.509
  - Username/password
- Message integrity / confidentiality
  - SSL
  - WS-Security
- Authorization
  - .NET Role-based security
  - PrincipalPermission / IsInRole



## Agenda

---

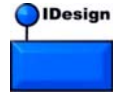
- WCF Overview
- Connecting Applications with WCF
- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer



## Peer-to-Peer Communications

---

- Each client is a also service
- Must call every other (interested) client directly
- Can simulate with duplex channels
- Can use PeerChannel
  - Sets up a single shared channel on the network for multiple clients
  - Multicast within the channel



## Summary

---

- WCF is a powerful and flexible distributed application, service oriented application development platform
- Remember your ABC's
- Think in terms of contracts: service and data



## Resources

---

- *Programming Indigo*, David Pallman, Microsoft Press, 2005.
- *Windows Communications Foundation Jumpstart*, Michele Leroux Bustamante, O'Reilly & Associates, Spring 2006
- PDC Sessions:  
<http://microsoft.sitestream.com/PDC05/>
- *Programming .NET Components 2<sup>nd</sup> Edition*, Juval Löwy, O'Reilly & Associates, 2005.

Email: [brian.noyes@idesign.net](mailto:brian.noyes@idesign.net)

Blog: <http://www.softinsight.com/bnoyes>