

Connecting Smart Client Applications with Indigo

Brian Noyes

IDesign Inc. (www.idesign.net)

brian.noyes@idesign.net

Visual Studio
CONNECTIONS

About Brian

- Chief Architect, IDesign Inc. (www.idesign.net)
- Microsoft MVP
- Writing
 - Data Binding in 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
- Blog: <http://www.softinsight.com/bnoyes>

Visual Studio
CONNECTIONS

Agenda

- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer

Visual Studio
CONNECTIONS

Channel Selection

- Services can often be called through different channel stacks
- Considerations:
 - Messaging patterns
 - Exposed endpoints
 - Security needs

Visual Studio
CONNECTIONS

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

Visual Studio
CONNECTIONS

Agenda

- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer

Visual Studio
CONNECTIONS

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

Visual Studio
CONNECTIONS

Asynchronous calls

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

Visual Studio
CONNECTIONS

Asynchronous Calls

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

Visual Studio
CONNECTIONS

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**

Visual Studio
CONNECTIONS

Agenda

- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer

Visual Studio
CONNECTIONS

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
```

Visual Studio
CONNECTIONS

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

Visual Studio
CONNECTIONS

Enabling Transactions

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

Visual Studio
CONNECTIONS

Agenda

- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer

Visual Studio
CONNECTIONS

Sessions

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

Visual Studio
CONNECTIONS

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

Visual Studio
CONNECTIONS

Agenda

- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- **Callbacks**
- Security
- Peer-to-Peer

Visual Studio
CONNECTIONS

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

Visual Studio
CONNECTIONS

Agenda

- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer

Visual Studio
CONNECTIONS

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

Visual Studio
CONNECTIONS

Agenda

- Channel Selection
- Asynchronous Calls
- Transactions
- Sessions
- Callbacks
- Security
- Peer-to-Peer

Visual Studio
CONNECTIONS

Peer-to-Peer Communications

- No true multicast in V1
- Each client is a also service
- Must call every other (interested) client directly
- Can simulate with duplex channels
- Can use PeerChannel

Visual Studio
CONNECTIONS

Resources

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

E-mail: brian.noyes@idesign.net

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

Visual Studio
CONNECTIONS