

{**Energize**} your brain

WCF: Best Practices and Pitfalls Discovered



Brian Camping, Avanade

brianc@avanade.com

29 May 2009

{**Energize**} your brain

Expectations

- Know your ABC's
- Best practices are not rocket science.
- What's in your toolkit?



{**Energize**} your brain

Agenda

➤ **Service Design**

- Hosting
- Consuming
- Contracts
- Fault Handling
- Instance Management
- Performance



Service Design

- Basic goals
 - Consistent, Reusable, Reliable, Secure
- Maintain thread safety
- Garbage in, garbage out
 - Leave your service in a consistent state

Service Design

- Minimize the impact of change
 - Defined development architecture
 - Greater consistency and easier maintenance
 - Keep cross cutting concern logic separate
 - Fault Handling, Logging, Transactions, Caching

{**Energize**} your brain

Agenda

✓ Service Design

➤ **Hosting**

- Consuming
- Contracts
- Fault Handling
- Instance Management
- Performance



Hosting

- Choosing the right host
 - When to use IIS, WAS, Self hosting
- Enable reusability
 - Custom ServiceHost
 - Custom ServiceHostFactory
 - ServiceHost<T>

{Energize} your brain

```
public class CustomSvcHost : ServiceHost
public class CustomSvcHostFactory : ServiceHostFactory
public class ServiceHost<T> : ServiceHost
{
    public ServiceHost() : base(typeof(T)) { }

    public ServiceHost(params string[] baseAddresses) :
        base(typeof(T), Convert(baseAddresses)) { }

    public ServiceHost(params Uri[] baseAddresses) :
        base(typeof(T), baseAddresses) { }

    static Uri[] Convert(string[] baseAddresses)
    {
        Converter<string, Uri> convert =
            delegate(string address)
            {
                return new Uri(address);
            };
        return Array.ConvertAll(baseAddresses, convert);
    }
}
```



{**Energize**} your brain

Agenda

- ✓ Service Design
- ✓ Hosting
- **Consuming**
 - Contracts
 - Fault Handling
 - Instance Management
 - Performance



Consuming

- Client/Proxy Management
 - ClientBase<T>
 - Dispose properly - why and how?
 - SafeProxyWrapper
 - ChannelFactory
 - Still have to close the CF
 - ChannelFactory<T>

{**Energize**} your brain

DEMO - Consuming



{**Energize**} your brain

Agenda

- ✓ Service Design
- ✓ Hosting
- ✓ Consuming
- **Contracts**
 - Fault Handling
 - Instance Management
 - Performance



Contracts

- Service Contract
 - Keep your Operation count low
 - Ease maintenance pain
 - Contract Factoring, Interfaces
 - Contract First Development?
 - Understand WSDL to Contract Relationship
 - Interoperability

Contracts

- Data Contract
 - Decouple your services from clients
 - Maintain version independence
 - Be explicit with what's in and what's out
 - Be implicit with [**DataMember**] Named Parameters

Contracts

- Message Contract
 - When to employ?
 - SOA?
 - Request/Response

{**Energize**} your brain

```
[MessageContract]
P
{
  [MessageContract]
  {
    public class ProcessResponse
    {
      public ProcessResponse() { }

      public ProcessResponse(GenericPair<int, string>[] valuePairs)
      {
        this.ValuePairs = valuePairs;
      }

      [MessageBodyMember]
      public GenericPair<int, string>[] ValuePairs { get; set; }
    }
  }

  public string[] Values { get; set; }
}
```



{**Energize**} your brain

Agenda

- ✓ Service Design
- ✓ Hosting
- ✓ Consuming
- ✓ Contracts
- **Fault Handling**
 - Instance Management
 - Performance



Fault Handling

- Ensure:
 - Separation of concerns
 - Consistency in development of services
- How?
 - ServiceBase,
 - Single point of handling
 - ServiceFault
 - Foundational member of FaultContract
 - FaultException<T>

{**Energize**} your brain

Demo – Fault Handling



{**Energize**} your brain

Agenda

- ✓ Service Design
- ✓ Hosting
- ✓ Consuming
- ✓ Contracts
- ✓ Fault Handling
- **Instance Management**
 - Performance



Instance Management

- What are my options?
- InstanceContextMode & SessionMode
 - Binding: choose wisely
 - Be explicit with Named Parameters
- Throttling
 - Adjust accordingly

{Energize} your brain

```
public class ServiceHost<T> : ServiceHost
{
    public void SetThrottleBehavior(ServiceThrottlingBehavior stb, bool
    public void SetThrottleBehavior(int maxSessions, int maxInstances, int
    { maxCalls)
    {
        ServiceThrottlingBehavior stb = new ServiceThrottlingBehavior();
        stb.MaxConcurrentSessions = maxSessions;
        stb.MaxConcurrentInstances = maxInstances;
        stb.MaxConcurrentCalls = maxCalls;
        ConfigureThrottlingBehavior(stb, true);
    }

    Description.Behaviors.Find<ServiceThrottlingBehavior>(),

    if (currentSTB != null && overrideConfig)
    {
        Description.Behaviors.Remove<ServiceThrottlingBehavior>();
    }

    Description.Behaviors.Add(stb);
}
}
```



{**Energize**} your brain

Agenda

- ✓ Service Design
- ✓ Hosting
- ✓ Consuming
- ✓ Contracts
- ✓ Fault Handling
- ✓ Instance Management
- **Performance**



Performance

- Message
 - Keep your message size small
 - Break up large data
 - Compression
- Server Configuration
 - Employ Throttling
 - Assess security
- Binding
 - Use binary encoding on enterprise services

{**Energize**} your brain

Wrap It Up

Topics:

- ✓ Hosting
- ✓ Consuming
- ✓ Contracts
- ✓ Fault Handling
- ✓ Instance Management
- ✓ Performance

Toolkit:

1. Custom ServiceHost/Factory
2. Custom ServiceHost<T>
3. SafeProxyWrapper
4. ChannelFactory<T>
5. ServiceBase, ServiceFault
6. Message Contracts



{**Energize**} your brain

Resources

- WCF Samples
 - <http://go.microsoft.com/fwlink/?LinkId=87352>
 - Stock Trader - <http://msdn.microsoft.com/stocktrader>
 - Dinner Now - <http://www.codeplex.com/DinnerNow>
- Helpful Links
 - WCF Security Guide – wcfsecurityguide.codeplex.com
 - WCFTestClient.exe - <http://msdn.microsoft.com/en-us/library/bb552364.aspx>
 - WCF Reference - <http://msdn.microsoft.com/en-us/library/ms735119.aspx>



{**Energize**} your brain

The End

Please complete your evaluation forms!



www.avanade.com

www.avanade.nl

brianc@avanade.com

<http://briancamping.net/>

Introductions

- Avanade
 - The leading global technology integrator specializing in the Microsoft Enterprise Platform.
 - Over 9000 professionals in 22 countries
- Our consultants
 - Deliver value according to each customer's requirements, timeline and budget by combining insight, innovation and the talent of our global workforce, which includes an extensive offshore community of Microsoft technology experts.
- Power of three
 - Our unique connections with Accenture and Microsoft enhance our ability to deliver effective mission-critical solutions for organizations in all industries. A joint venture between Accenture and Microsoft, Avanade was formed to merge the strengths of both companies – Accenture's consulting and Microsoft's technology expertise – into a single company.



<http://www.avanade.nl>

<http://www.avanade.com>