WCF Tips and Tricks
[a selection] from the field

Christian Weyer, thinktecture

christian.weyer@thinktecture.com

thinktecture and Christian Weyer

- Support & consulting for Windows and .NET software developers and architects
  - Developer coaching and mentoring
  - Architecture consulting and prototyping
  - Architecture and code reviews
  - Application optimization, troubleshooting, debugging
- Focus on distributed applications, service orientation, workflows, cloud computing, interoperability, security, end-to-end solutions
  - Windows Server, WCF, WF, MSMQ, Azure Services, Windows Azure
- http://www.thinktecture.com
- christian.weyer@thinktecture.com

Microsoft tech·days 09

For starters: **Super-duper TIP**

- WCF is not just Web Services !
- WCF is not always about SOA !

- WCF does not stand for
  - Web Services Consumption Foundation
  - Windows Cool-SOA Foundation

- It is the
  Windows Communication Foundation !

Agenda – Dealing with…

- Consuming
- Hosting
- Bindings
- Quotas & throttles
- Metadata/WSDL
- Performance & throughput
- Large data
- Tracing

Problems →
Solutions/Tips →
Samples

Microsoft tech·days 09

Not covered, some…

- Contract modelling
- REST
- Security
- Asynchronous processing
- Fault handling
- Deep extensibility
- WF integration
- NATs & firewalls

- *… and surely more …*

Consuming – Problems

- Do I always need to create a proxy class from WSDL/MEX?
- How can I make consuming services more robust?
- Is there a way to improve performance when calling services?
- How can I call in-process ‘services’ and WCF services in the same way?

Consuming – Solutions I

- For non-interop no need to use *svcutil.exe* or *'Add Service Reference'*
  - shared contracts approach works good in WCF-to-WCF scenarios
  - **`ChannelFactory<T>`** and **`DuplexChannelFactory<T>`** are powerful means
  - use custom interface extending service contract & **`IClientChannel`**
- Avoid **`using`** statement when dealing with proxies (**`ICommunicationObject`**-derived objects)
  - can still throw at end of using block, e.g. for network errors
  - explicit exception handling; can be dealt with e.g. in extension method

Microsoft tech·days 09

Consuming – Solutions II

- Try to cache proxy or **`ChannelFactory`** in high-throughput applications
  - creating them can mean significant overhead
  - ASP.NET client applications should not create **`ChannelFactory`** on each page call
- Abstract away channel/proxy creation details with **`Activator.CreateInstance`** & **`ChannelFactory<T>`**
  - Service Agent pattern
  - for local and remote services
  - no WCF-isms available, like sessions etc.
  - WCF 4.0 will offer in-process, intra-AppDomain channel

Consuming – Samples

```
interface IMyContractChannel : IMyContract,
          System.ServiceModel.IClientChannel {}
```

Client side

```
ChannelFactory<IMyContractChannel> cf =
          new ChannelFactory<IMyContractChannel>(binding, address);
IMyContractChannel client = cf.CreateChannel();

client.DoIt(…);
client.Close();
```

Client side

```
try
{
     …     client.Close();
}
catch (CommunicationException e)
{
     …     client.Abort();
}
catch (TimeoutException e)
{
     …     client.Abort();
}
catch (Exception e)
{
     …     client.Abort();
     throw;
}
```

Client side

```
IMyContractChannel channel =
          ChannelFactoryManager<IMyContractChannel>
                    .GetChannel("BasicHttpBinding");
```

Client side

Hosting - Problems

- Which host to use?
  IIS/WAS or a self-host?
- How do I inject logic in IIS/WAS hosting?

Hosting – Solutions

- Use IIS/WAS for robust, highly scalable services
  - beware of the process & AppDomain lifecycle features
  - when using non-HTTP (TCP, Named Pipes, MSMQ) with WAS hosting AppDomain recycling still comes into your way
- Use self-hosting in Windows Service to have full control and light-weight hosting environment
- Custom **`ServiceHost`** & **`ServiceHostFactory`** implementations to provide custom initialization code
  - hook up factory in **`.svc`** file for IIS/WAS

Hosting – Samples

```
class MyServiceHost : System.ServiceModel.ServiceHost
{
    public MyServiceHost(Type serviceType,
        params Uri[] baseAddresses)
                : base(serviceType, baseAddresses)
    {
        ...
    }

    protected override void ApplyConfiguration()
    {
        ...
    }
}
```

Custom ServiceHost

```
class MyServiceHostFactory :
        System.ServiceModel.Activation.ServiceHostFactory
{
    protected override ServiceHost CreateServiceHost(
        Type serviceType, Uri[] baseAddresses)
    {
        return new MyServiceHost(serviceType, baseAddresses);
    }
}
```

Custom ServiceHostFactory

```
<%@ ServiceHost Language="C#" Debug="true"
     Service="MediaService"
         Factory="MyServiceHostFactory" %>
```

.svc file

Bindings - Problems

- My WCF service is slow, what is happening?
- I want to use HTTP but not necessarily angle brackets (aka ‚XML‘)
- How can I choose from the best communication options?

Bindings – Solutions I

- Beware of using the wrong bindings
  - e.g. Visual Studio WCF wizards use **`WsHttpBinding`** (heavy with message security & session-based)
  - only use features you really need
- Think about the real need for session-bound channels/bindings
  - sessions change the game of fault and error handling
  - use sessions when you need session semantics in your service
- But: sessions *can* give performance improvements
  - e.g. security token hand-shake happens only once with *SecureConversation*

Bindings – Solutions II

- Custom bindings will save your day
  - e.g. binary over HTTP often a good trade-off for WCF-to-WCF communication scenarios
  - build custom binding in config or code
- Create user-defined binding for easier re-usage
  - bake common custom binding setups into re-usable code and config implementations
- Use a custom encoder for providing encoding-level tweaks & optimizations
  - e.g. enhanced text encoder in SDK or FastInfoSet encoder from 3rd party

Bindings - Samples

```
<extensions>
  <bindingExtensions>
    <add name="netHttpBinding"
         type="NetHttpBindingCollectionElement,
         Thinktecture.ServiceModel, Version=…" />
  </bindingExtensions>
</extensions>
<bindings>
  <netHttpBinding>
    <binding name="unsecureNetHttp" securityMode="None" />
  </netHttpBinding>

  <customBinding>
    <binding name="binaryHttp">
      <binaryMessageEncoding />
      <httpTransport />
    </binding>
  </customBinding>
</bindings>
```

app/web.config

```
public class NetHttpBinding :
         System.ServiceModels.Channels.Binding,
         ISecurityCapabilities
{
  HttpTransportBindingElement httpTransport;
  HttpsTransportBindingElement httpsTransport;
  BinaryMessageEncodingBindingElement binaryEncoding;
  NetHttpSecurityMode securityMode;

  ...
}
```

User-defined binding

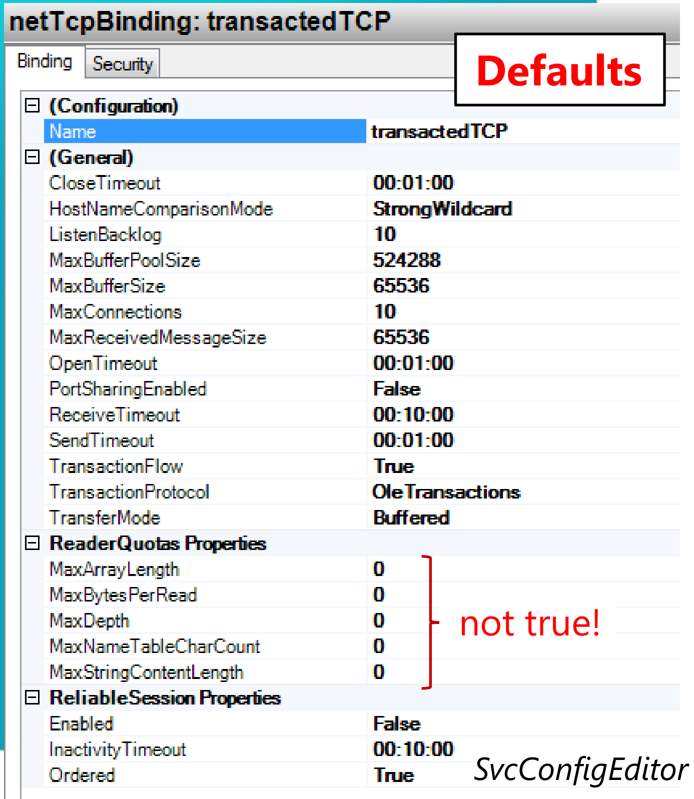
Quotas & Throttles - Problems

- Beyond *Hello World*, all my services and consumers fail with strange exceptions
- My services do not perform the way they are supposed to
- How can I teach WCF to be less ‘conservative’ in Intranet environments?

Microsoft tech·days 09

Quotas & Throttles - Solutions

- Bindings
  - adjust buffers, connection limits, timeouts
- Behaviors
  - configure throttling service behavior
- Serializers
  - check maximum items in object graph value
- Custom **`ChannelFactory`** and **`ServiceHost`** can automate all this
  - e.g. through profiles


netTcpBinding: transactedTCP

Binding | Security

Defaults

(Configuration)	
Name	transactedTCP
(General)	
CloseTimeout	00:01:00
HostNameComparisonMode	StrongWildcard
ListenBacklog	10
MaxBufferPoolSize	524288
MaxBufferSize	65536
MaxConnections	10
MaxReceivedMessageSize	65536
OpenTimeout	00:01:00
PortSharingEnabled	False
ReceiveTimeout	00:10:00
SendTimeout	00:01:00
TransactionFlow	True
TransactionProtocol	OleTransactions
TransferMode	Buffered
ReaderQuotas Properties	
MaxArrayLength	0
MaxBytesPerRead	0
MaxDepth	0
MaxNameTableCharCount	0
MaxStringContentLength	0
ReliableSession Properties	
Enabled	False
InactivityTimeout	00:10:00
Ordered	True

not true!

SvcConfigEditor

Quotas & Throttles - Samples

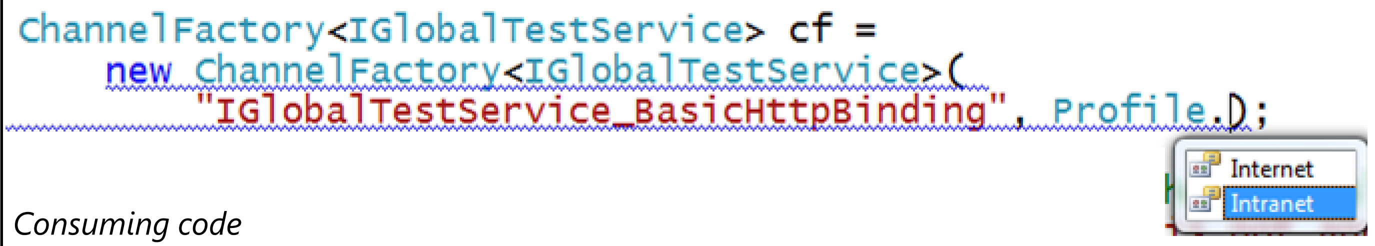
```
<bindings>
  <webHttpBinding>
    <binding name="rawStreamingWeb" transferMode="StreamedResponse">
      <readerQuotas maxArrayLength="999999999"/>
    </binding>
  </webHttpBinding>

  <customBinding>
    <binding name="httpStreaming" sendTimeout="Infinite">
      <binaryMessageEncoding />
      <httpTransport transferMode="Streamed" />
    </binding>
  </customBinding>
<bindings>
```

app/web.config

```
<behaviors>
  <serviceBehaviors>
    <behavior name="MyServiceBehavior">
      <serviceThrottling
        maxConcurrentCalls="1500"
        maxConcurrentInstances="1500"
        maxConcurrentSessions="1500" />
    </behavior>
  </serviceBehaviors>
</behaviors>
```

app/web.config



```
ChannelFactory<IGlobalTestService> cf =
    new ChannelFactory<IGlobalTestService>(
        "IGlobalTestService_BasicHttpBinding", Profile.);
```

Consuming code

WSDL & Metadata - Problems

- Some non-WCF consumers cannot understand the WSDL WCF produces
- My WSDL contains the wrong host name
- I cannot use multiple IIS web site bindings with my WCF services

Microsoft tech·days 09

WSDL & Metadata – Solutions I

- Use custom extension to flatten WSDL into one file
  - need to use same namespace values for **`ServiceContract`**, **`ServiceBehavior`**, **`BindingNamespace`**
  - eliminates **`wsdl:import`** and **`xsd:import`**
- Register host headers in IIS to reflect names into WSDL
  - for HTTP and HTTPS
- Specify different URIs for listening and exposing in WSDL

```
<endpoint
   address="https://www.tt.com/TheUriIWantInWSDL"
   listenUri="http://localhost/
     TheActualUriServiceListensToOnThisBox" ...>
```

- Consider exposing a static WSDL which documents your published interface version

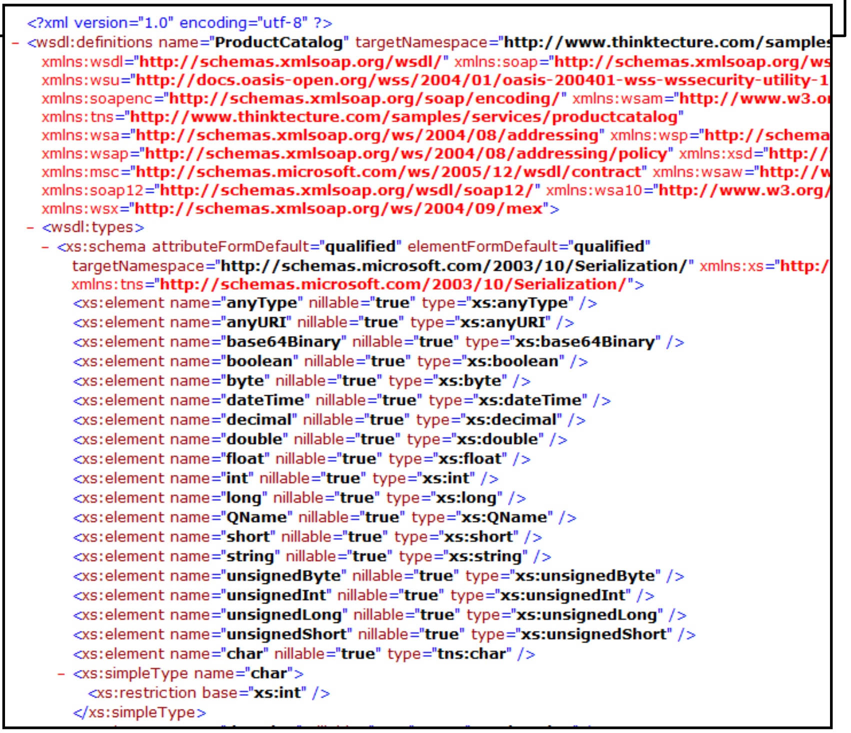
WSDL & Metadata – Solutions II

- Multiple IIS site bindings result in multiple base addresses
  - WCF only supports a single base address in this scenario
  - fix yourself in .NET 3.0 with custom **`ServiceHostFactory`**
- .NET 3.5 supports **`<baseAddressPrefixFilters>`**
  - pass-through filter which provides a mechanism to pick the appropriate IIS bindings

WSDL & Metadata - Samples

```
<%@ ServiceHost Language= "C#" Service="ProductCatalog"
  Factory="Thinktecture.ServiceModel.Activation.FlatWsdlServiceHostFactory"
%>
```

.svc file



```
<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions name="ProductCatalog" targetNamespace="http://www.thinktecture.com/samples
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/ws
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsam="http://www.w3.o
    xmlns:tns="http://www.thinktecture.com/samples/services/productcatalog"
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsp="http://schema
    xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy" xmlns:xsd="http://
    xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract" xmlns:wsaw="http://w
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:wsa10="http://www.w3.org/
    xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex">
  - <wsdl:types>
    - <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
        targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/" xmlns:xs="http:/
        xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/">
        <xs:element name="anyType" nillable="true" type="xs:anyType" />
        <xs:element name="anyURI" nillable="true" type="xs:anyURI" />
        <xs:element name="base64Binary" nillable="true" type="xs:base64Binary" />
        <xs:element name="boolean" nillable="true" type="xs:boolean" />
        <xs:element name="byte" nillable="true" type="xs:byte" />
        <xs:element name="dateTime" nillable="true" type="xs:dateTime" />
        <xs:element name="decimal" nillable="true" type="xs:decimal" />
        <xs:element name="double" nillable="true" type="xs:double" />
        <xs:element name="float" nillable="true" type="xs:float" />
        <xs:element name="int" nillable="true" type="xs:int" />
        <xs:element name="long" nillable="true" type="xs:long" />
        <xs:element name="QName" nillable="true" type="xs:QName" />
        <xs:element name="short" nillable="true" type="xs:short" />
        <xs:element name="string" nillable="true" type="xs:string" />
        <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte" />
        <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt" />
        <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong" />
        <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort" />
        <xs:element name="char" nillable="true" type="tns:char" />
      - <xs:simpleType name="char">
          <xs:restriction base="xs:int" />
        </xs:simpleType>
```

app/web config

```
<serviceHostingEnvironment>
  <baseAddressPrefixFilters>
    <add prefix="http://thinktecture.de/"/>
    <add prefix="http://thinktecture.com"/>
  </baseAddressPrefixFilters>
</serviceHostingEnvironment>
```

applicationHost.config (IIS7)

```
<bindings>
  <binding protocol="http”
     bindingInformation="*:80:www.thinktecture.com" />
  <binding protocol="https"
     bindingInformation="*:443:www.thinktecture.com" />
</bindings>
```

Large Data - Problems

- My service eats a lot of memory and chokes the CPU when sending/receiving large data
- Bigger messages are making my communication really slow
- I have arbitrary, non-structured data to transfer

Large Data – Solutions I

- WCF supports MTOM for encoding binary data
  - MTOM especially useful for interop
- Chunking channels available as SDK & community samples
  - enables sending chunks of data instead of one single piece
  - transparently works with different transports as a binding element

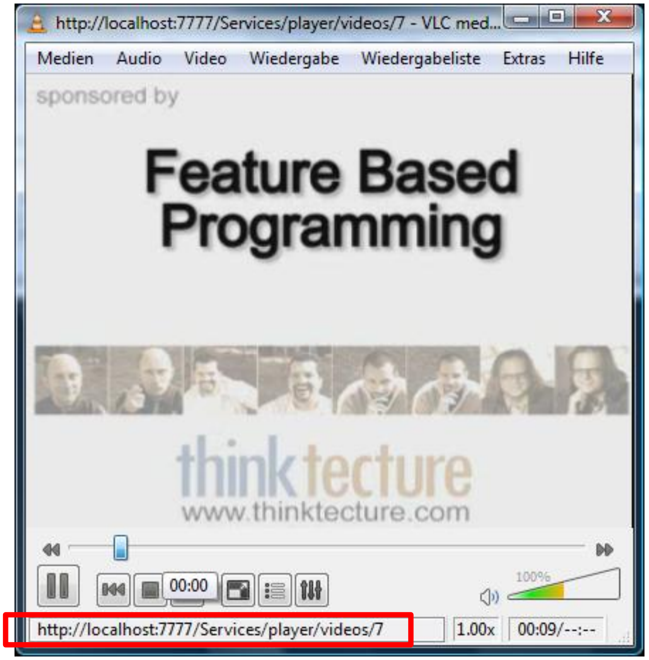
Large Data – Solutions II

- Consider using streaming for transfering abitrary data
  - requires certain contract shape
    - `Stream`
    - `Message`
    - `Stream` as single body in `MessageContract`
  - works over any transport besides MSMQ
  - works with transport and mixed-mode security
  - still watch out for quotas
  - powerful with web programming model

Large Data - Samples

```
[ServiceContract]
public interface IVideoPlayer
{
    [OperationContract]
    [WebGet(UriTemplate = "videos/{videoID}")]
    [WebContentType(MimeType = "video/x-ms-wmv")]
    Stream Play(string videoID);
}
```

Service contract



http://localhost:7777/Services/player/videos/7

Client

Host

```
WebServiceHost webHost = new WebServiceHost(
  typeof(VideoPlayerService));

WebHttpBinding binding = new WebHttpBinding();
binding.TransferMode = TransferMode.Streamed;

webHost.AddServiceEndpoint(
  typeof(IVideoPlayer),
  binding,
  "http://localhost:7777/Services/player");
```

tech·days

Performance/Throughput - Problems

- Somehow my entire WCF-based application is ‚slow'
- Hosting my WCF service in IIS seems not to perform well under high load
- I cannot seem to get a high throughput when clients talk to my service via HTTP
- All that data is being transferred again and again, it makes my system slow

Performance/Throughput – Solutions

- Configuring throttling can heal a lot (look there!)
- .NET 3.5 SP1 provides asynchronous HTTP module & handler for hosting WCF in IIS for better behavior
- Client-side HTTP communication is limited to 2 concurrent connections to a server
  - configurable through **`System.Net`**
- Cache, cache, cache!
  - try to use caching intensively (but wisely) to save unnecessary round-trips

Microsoft tech·days 09

Performance/Throughput - Samples

```xml
<system.net>
  <connectionManagement>
    <add address="*" maxconnection="20"/>
  </connectionManagement>
</system.net>
```

app/web.config

```csharp
public List<Episode> ListEpisodes()
{
    IDataCache cache = DataCacheFactory.CreateInstance();
    List<Episode> episodes =
      cache.Get<List<Episode>>(CacheConstants.AllEpisodes);

    if (episodes == null)
    {
        var episodeList = mediaLogic.ListAllEpisodes();
        episodes = EpisodeDataMapper.MapAllEpisodes(episodeList);

        cache.Add(CacheConstants.AllEpisodes, episodes);
    }

    return episodes;
}
```

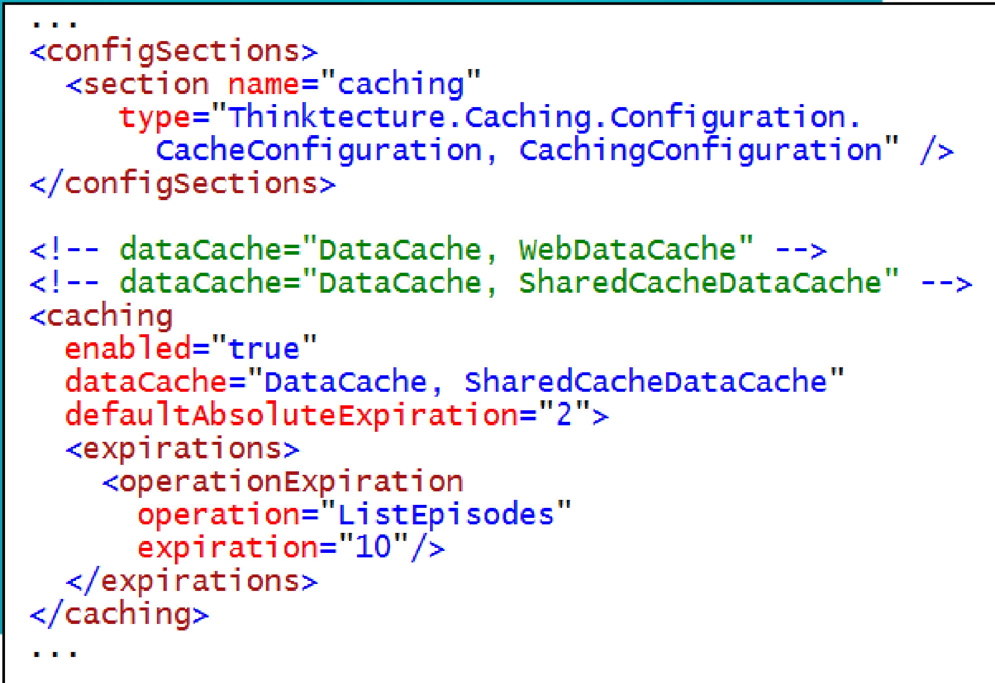
E.g. service facade

```csharp
public interface IDataCache
{
    void Add(string key, object cacheItem);
    TCacheItem Get<TCacheItem>(string key);
    void Remove(string key);
}
```

Caching lib

Performance/Throughput – Cache Solution

- Cache should be abstracted from actual cache product/implementation
- Generic interface with different implementations
  - local, in-proc cache
  - distributed cache
- ASP.NET's web cache can also be used outside of ASP.NET
- Distributed caches necessary for farm and scale-out scenarios
  - MemcacheD
  - SharedCache
  - ‚Velocity'



```
...
<configSections>
  <section name="caching"
     type="Thinktecture.Caching.Configuration.
       CacheConfiguration, CachingConfiguration" />
</configSections>

<!-- dataCache="DataCache, WebDataCache" -->
<!-- dataCache="DataCache, SharedCacheDataCache" -->
<caching
  enabled="true"
  dataCache="DataCache, SharedCacheDataCache"
  defaultAbsoluteExpiration="2">
  <expirations>
    <operationExpiration
      operation="ListEpisodes"
      expiration="10"/>
  </expirations>
</caching>
...
```

Special Case: **Tracing**

- Use it! It can save your… ☺
- If things go wrong and you have no clue why: trace!
- But do not overuse it when in production
  - wrong usage can mean severe overhead
- Configured via config file
  - can be manipulated via code, but only through WMI

- Did I already say tracing can save your …?

Tracing - Sample

```
<system.diagnostics>
   <sources>
     <source name="System.ServiceModel" switchValue="Warning"
                   propagateActivity="true">
       <listeners>
         <add type="System.Diagnostics.DefaultTraceListener"
                   name="Default" />
         <add name="ServiceModelTraceListener" />
       </listeners>
     </source>
   </sources>

   <sharedListeners>
     <add initializeData=„MyService_Trace.svclog"
       type="System.Diagnostics.XmlWriterTraceListener, …"
       name="ServiceModelTraceListener"
         traceOutputOptions="Timestamp" />
  </sharedListeners>

   <trace autoflush="true" />
 </system.diagnostics>
```

app/web.config

PS script

```
$ms = get-wmiobject -class "AppDomainInfo"
  -namespace "root\servicemodel" -computername "." |
  where {$_.Name -eq "MyWCFHost.exe"}
$ms.TraceLevel = "Warning, ActivityTracing"
$ms.Put()
```

```
<system.serviceModel>
  <diagnostics wmiProviderEnabled="true"/>
…
```

app/web.config

Resources

- Avoiding problems with the using statement
  - http://msdn.microsoft.com/en-us/library/aa355056.aspx
- Custom encoders
  - http://msdn.microsoft.com/en-us/library/ms751486.aspx
  - http://blogs.msdn.com/drnick/archive/2006/05/16/598420.aspx
- Tracing
  - http://msdn2.microsoft.com/en-us/library/ms732023.aspx
  - http://msdn2.microsoft.com/en-us/library/aa751795.aspx
  - http://msdn2.microsoft.com/en-us/library/ms733025.aspx
  - http://msdn2.microsoft.com/en-us/library/aa751917.aspx

Resources

- Setting up IIS SSL host headers
  - http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/596b9108-b1a7-494d-885d-f8941b07554c.mspx
  - http://blogs.iis.net/thomad/archive/2008/01/25/ssl-certificates-on-sites-with-host-headers.aspx
- baseAddressPrefixFilter
  - http://msdn.microsoft.com/en-us/library/bb924492.aspx
- Chunking channel
  - http://code.msdn.microsoft.com/WCFResources/Release/ProjectReleases.aspx?ReleaseId=1546

Resources

- Asynchronous WCF HTTP Module/Handler for IIS7 for Better Server Scalability
  - http://blogs.msdn.com/wenlong/archive/2008/08/13/orcas-sp1-improvement-asynchronous-wcf-http-module-handler-for-iis7-for-better-server-scalability.aspx
- WCF bindings & more
  - http://www.noemax.com

Resources

- We have code solutions for some WCF problems, for free of course
  - Thinktecture.ServiceModel
- → Email Christian Weyer
  - christian.weyer@thinktecture.com
- Weblog Christian Weyer
  - http://blogs.thinktecture.com/cweyer
- thinktecture
  - http://www.thinktecture.com

thinktecture

{In-depth support and consulting for software architects and developers}

http://www.thinktecture.com/

christian.weyer@thinktecture.com

http://blogs.thinktecture.com/cweyer/

Microsoft® tech·days 09