Einstieg in WCF
Tipps, Tricks und Tools

27.04.2010

Melanie Eibl

EMail: melanie@eibl-winterscheid.de
Website: http://www.eibl-winterscheid.de

Windows Communication Foundation

- Teil des .NET Frameworks ab Version 3.0
- Ziele
  - Vereinheitlichung
  - Kompatibilität
  - Unterstützung service-orientierter Entwicklung
- In .NET 4.0 gibt es nur wenig Neuerungen

Vereinheitlichung

	ASMX	.NET Remoting	Enterprise Services	WSE	System. Messaging	System. Net	WCF
Interoperable Web Services	X						X
Binary .NET –.NET Communication		X					X
Distributed Transactions, etc.			X				X
Support for WS-* Specifications				X			X
Queued Messaging					X		X
RESTful Communication						X	X

Kompatibilität

- Kompatibilität mit Anwendungen auf Basis anderer Technologien
- Interoperable Kommunikation über SOAP

Service-orientierte Entwicklung

- Teile ein Schema, keine Klasse
- Services sind autonom
- Grenzen sind eindeutig
- Richtlinien-basierte Kompatibilität

Wesentliche Bausteine

- [ServiceContract]
- [OperationContract]
- [DataContract]
- [DataMember]

ServiceContract

- Namespace: System.ServiceModel
- Gibt an, dass ein Interface oder eine Klasse einen Dienstvertrag in einer Anwendung definiert

OperationContract

- Namespace: System.ServiceModel
- Gibt an, dass eine Methode einen Vorgang definiert, der Bestandteil eines Dienstvertrags in einer Anwendung ist

DataContract

- Namespace: System.Runtime.Serialization
- Gibt an, dass der Typ einen Datenvertrag definiert oder implementiert und mit einem Serialisierer wie dem DataContractSerializer serialisierbar ist
- Um ihren Typ serialisierbar zu machen, müssen die Autoren hierfür einen Datenvertrag definieren

DataMember

- Namespace: System.Runtime.Serialization
- Gibt bei Anwendung auf den Member eines Typs an, dass der Member Teil eines Datenvertrags ist und über den DataContractSerializer serialisierbar ist

Hostingmöglichkeiten

- Konsolenanwendung
- Windows-Forms-Anwendung
- Windows-Dienst
- COM+
- IIS/ASP.NET
- Windows Activation Service (WAS)

Endpunkte

- Endpunkte ermöglichen Clients den Zugriff auf die Funktionalität, die ein WCF-Dienst anbietet
- Nachrichten werden zwischen Client- und Server-Endpunkten ausgetauscht
- Endpunkt-ABC: Address – Binding – Contract (Mindestbestandteile eines Endpunktes)
- Ein Service kann über mehrere Endpunkte erreichbar sein

Address

- URI
- Kann absolut oder relativ angegeben werden
- Bei relativer Adressierung muss der Host eine Basisadresse bereitstellen
- Standardwert ist eine leere Zeichenfolge
- z. B. "http://www.fabrikam.com:322/mathservice.svc/secureEndpoint"

Binding

- Beispiele für vom System bereitgestellte Bindungen
  - BasicHttpBinding
  - WSHttpBinding
  - NetNamedPipeBinding
  - NetMsmqBinding
- Es können auch individuelle Bindungstypen erzeugt werden

Contract

- Gibt an, welche Verträge von diesem Endpunkt verfügbar gemacht werden
- Die Assembly muss den/die Vertragstypen implementieren
- Der Standardwert ist eine leere Zeichenfolge

Metadaten bereitstellen

- Durch Metadaten beschreibt sich der Service gegenüber dem Client selbst
- Vor dem Deployment sollte dieser Endpunkt entfernt werden

```
<endpoint
      address="mex"
      binding="mexHttpBinding"
      contract="IMetadataExchange"/>
```

Weitere Endpunkt Attribute

- name
- behaviorConfiguration
- bindingName
- bindingNamespace
- usw.

WcfTestClient.exe

- Tool zum Testen von Services
  - C:\Program Files\Microsoft Visual Studio 10.0\Common7\IDE\WcfTestClient.exe
- Ermöglicht die Eingabe von Testparametern
- Testparameter werden dem Service übermittelt
- Zeigt die Antwort des Service an

Demo WcfTestClient.exe

Svcutil.exe

- ServiceModel Metadata Utility-Tool
  - C:\Program Files\Microsoft SDKs\Windows\v7.0A\bin\SvcUtil.exe
- Ruft Metadaten von einem WCF-Dienst ab
- Erstellt einen WCF-Clientproxy, der auf den Dienst zugreifen kann
- Erstellt Konfigurationsdatei

Demo SvcUtil.exe

SvcConfigEditor.exe

- Configuration Editor-Tool
  - C:\Program Files\Microsoft SDKs\Windows\v7.0A\bin\SvcConfigEditor.exe
- Konfiguration von WCF-Diensten mit einer grafischen Benutzeroberfläche
- Einstellungen für Bindungen, Verhalten, Dienste und Diagnosen

Demo SvcConfigEditor.exe

Instrumentierung von WCF-Diensten

- Trace-Listener hinzufügen
- Message-Logging einstellen
- Service behaviorConfiguration

Trace-Listener hinzufügen

```
<system.diagnostics>
  <sources>
    <source name="System.ServiceModel"
            switchValue="Information, ActivityTracing">
      <listeners>
        <add name="log"
             type="System.Diagnostics.XmlWriterTraceListener"
             initializeData="c:\temp\Traces.svclog"/>
      </listeners>
    </source>
  </sources>
  <trace autoflush="true"/>
</system.diagnostics>
```

Message-Logging einstellen

```
<system.serviceModel>
  <diagnostics>
    <messageLogging
        logEntireMessage="true"
        logMalformedMessages="false"
        logMessagesAtServiceLevel="false"
        logMessagesAtTransportLevel="true"
        maxMessagesToLog="3000"
        maxSizeOfMessageToLog="2000">
    </messageLogging>
  </diagnostics>
</system.serviceModel>
```

ServiceDebugBehavior

- Namespace: System.ServiceModel.Description
- Aktiviert Debugging- und Hilfeinformationsfeatures für einen Dienst

```
<service
    name="BonnToCodeWcfServiceLibrary.ExceptionService"
    behaviorConfiguration="metadataAndDebug">
```

```
<behaviors>
  <serviceBehaviors>
    <behavior name="metadataAndDebug">
      <serviceMetadata
          httpGetEnabled="True"/>
      <serviceDebug
          httpHelpPageEnabled="true"
          includeExceptionDetailInFaults="True" />
    </behavior>
    <behavior name="metadataOnly">
      <serviceMetadata
          httpGetEnabled="True"/>
    </behavior>
  </serviceBehaviors>
</behaviors>
```

SvcTraceViewer.exe

- Service Trace Viewer Tool
  - C:\Program Files\Microsoft SDKs\Windows\v7.0A\bin\SvcTraceViewer.exe
- Hilft beim Analysieren von Diagnoseablaufverfolgungen
- Kann im Config-File konfiguriert werden

Demo SvcTraceViewer.exe