

WCF Basic Routing

Scott Seely

<http://www.pluralsight.com/>



Outline

- Introduction to message routing
- Configuring Routes
- Demonstration

Introduction to Message Routing

- **What is routing?**
 - Based on properties of a message, send to appropriate destination through intermediary
- **What properties to route on?**
 - Action header
 - Receive address
 - Address prefix
 - Message header contents
- **WS-* messaging allows for transport neutral messaging**
 - Security, transactions, addressing all in headers

Use Cases for Routing

- Deploy server in DMZ, route to infrastructure within enterprise.
- Bridge protocols (ex. http→net.tcp)
- Well known public face for services. Abstract deployment of other services.
- Load balancing of service calls.
- Handle failover.
- BizTalk seems to be too heavy of a solution.
 - Watch for when BizTalk starts making sense!

RoutingService

- Just a Microsoft provided WCF class
- Hosted in a ServiceHost- available in EXEs, Windows Service, or IIS
- Completely configurable
 - <system.serviceModel>/<routing>
- Imperative (code) model matches configuration

Routing Service Endpoints

- **Configure like every other service**
 - Address
 - Binding
 - Contract
- **Listens with MEPs.**
 - One-way: `ISimplexDatagramRouter`
 - One-way with session: `ISimplexSessionRouter`
 - Request-reply: `IRequestReplyRouter`
 - Duplex: `IDuplexSessionRouter`

Configuring the Service

```
<services>
  <service behaviorConfiguration="routingData"
    name="System.ServiceModel.Routing.RoutingService">
    <endpoint address="/helloWorld" binding="wsHttpBinding" name="helloWorld"
      contract="System.ServiceModel.Routing.IRequestReplyRouter"/>
    <endpoint address="/oneway" binding="basicHttpBinding" name="oneWay"
      contract="System.ServiceModel.Routing.ISimplexDatagramRouter"/>
    <endpoint address="/duplex" binding="wsDualHttpBinding" name="duplex"
      contract="System.ServiceModel.Routing.IDuplexSessionRouter"/>
  </service>
</services>
```

Enabling Routing

- Endpoint needs a <routing> behavior on the endpoint
`<routing filterTableName="routingTable" />`
- Points to a filter table in the <routing> section of config

Routing Config

```
<routing>
  <filters>
    <filter name="HelloWorld" filterType="EndpointName" filterData="helloWorld" />
    <filter name="OneWay" filterType="EndpointName" filterData="oneWay" />
    <filter name="Duplex" filterType="EndpointName" filterData="duplex" />
  </filters>
  <filterTables>
    <filterTable name="routingTable">
      <add filterName="HelloWorld" endpointName="HelloWorldService" />
      <add filterName="OneWay" endpointName="OnewayService" />
      <add filterName="Duplex" endpointName="DuplexService" />
    </filterTable>
  </filterTables>
</routing>
```

- endpointName points to client configuration
- Other filters exist: Action, XPath, EndpointAddress, PrefixEndpointAddress, MatchAll
- Any filter you wrote for MessageLogging works here.

Forwarding Endpoint Configuration

- **Note: Contract is always "*" for all messages.**
 - Allows routing to be 'blind' to contents.

```
<client>
  <endpoint name="HelloWorldService"
    address="http://localhost:58941/HelloWorld.svc"
    binding="basicHttpBinding" contract="*" />
  <endpoint name="OnewayService"
    address="net.msmq://localhost/private/OnewayService"
    binding="netMsmqBinding" bindingConfiguration="onewaydemo"
    contract="*" />
  <endpoint name="DuplexService"
    address="net.tcp://localhost/DuplexService"
    binding="netTcpBinding" contract="*" />
</client>
```

Summary

- Look out for cases where your routes get too complex. It might be time to buy BizTalk.
- Message routing provides a way to bridge protocols
- The feature is highly configurable, allowing for easy updates
- Routing focuses on the MEP behind the contract, not the contract itself.

For more in-depth **online** developer **training** visit



on-demand content from authors you **trust**