WCF 4.0 Discovery

Scott Seely
http://www.pluralsight.com/

Outline

- Service Discovery
- Ad-hoc Discovery
- Managed Discovery

Service Discovery

- **Service discovery makes it possible to "discover" endpoints dynamically**
  - Good for scenarios where services come and go at runtime (e.g. devices)
  - The ideas have been around for a while but are now formalizing (OASIS)
- **WCF 4.0 provides two types of service discovery**

Adhoc	Managed
Clients can discover services on a local subnet (UDP-based)	Clients can discover services on a larger "managed" network (beyond the local subnet) through a discovery proxy

Simple ad-hoc discovery

- **Services must first expose a discovery endpoint**
  - You can use the "udpDiscoveryEndpoint" standard endpoint
  - Once it's active, clients will be able to discover it over UDP
- **Client can then use the standard "dynamicEndpoint" to find it**
  - In code, the client uses the DiscoveryClient class
  - It's also possible to provide FindCriteria (scopes) to narrow results
  - If the service exposes a MetadataExchange endpoint, the client can discover that too and dynamically bind to the endpoint.

Recommendations to speed locating services

- **Default: Time based scope resolution of 20 seconds**
- **Just find first service that responds**
  - FindCriteria.MaxResults = 1
- **Find all services that can respond 'quickly'**
  - FindCriteria.Duration = TimeSpan.FromSeconds(1)
- **FindCriteria may also appear in configuration**

Binding Dynamically

- **Different services exposed on different protocols.**
  - May be any WCF transport
  - On local network, all authentication done using network credentials (safe assumption)
- **Create FindCriteria that uses MetadataExchange to discover binding**
  - FindCriteria.CreateMetadataExchangeEndpointCriteria(Type)
  - Finds MetadataExchange endpoint that expose an endpoint of the named type.
- **MetadataResolver to find the metadata and Binding**
- **ChannelFactory to create the proxy**

Discovery scopes

- **Clients can narrow discovery results by searching based on "scopes"**
  - A scope is a URI that has meaning to the caller and service
  - Services associate themselves with scopes while announcing
  - Clients can then discover based on scope information
  - All of this can be specified in configuration (both service & client)

Discovery scopes

- **Clients can narrow discovery results by searching based on "scopes"**
  - Services associate themselves with scopes while announcing
  - Clients can then discover based on scope information
  - All of this can be specified in configuration (both service & client)

```
<system.serviceModel>
    <client>
      <endpoint
          name="discovery"
          kind="dynamicEndpoint"
          binding="wsHttpBinding"
          contract="IHelloWorld">
      </endpoint>
    </client>
</system.serviceModel>
```

Configure Scopes at Service

```
<endpointBehaviors>
   <behavior>
     <endpointDiscovery>
       <scopes>
         <add scope="ldap:///ou=teachers,o=pluralsight.com"/>
       </scopes>
     </endpointDiscovery>
   </behavior>
 </endpointBehaviors>
```

Configure scopes at client

```
<standardEndpoint name="dynamicEndpointConfiguration">
  <discoveryClientSettings>
    <endpoint kind="udpDiscoveryEndpoint" />
    <findCriteria maxResults="1">
      <scopes>
        <add scope="ldap:///ou=teachers,o=pluralsight.com"/>
      </scopes>
    </findCriteria>
  </discoveryClientSettings>
</standardEndpoint>
```

Service Announcements

- **WCF 4 also makes it easy for services to "announce" themselves**
  - This allows clients who are "listening" to learn about new services
  - Ultimately this reduces the amount of probing/multicast messaging
- **The <serviceDiscovery> behavior allows you to define a set of announcement endpoints to use**
  - Do NOT add announcement to the service endpoints collection: just doesn't work.
- **Clients host an AnnouncementService to listen for announcements**
  - OnlineAnnouncementReceived: Indicates a new endpoint is online
  - OfflineAnnouncementReceived: Indicates an endpoint went offline

Configuring Announcements

```
<behaviors>
  <serviceBehaviors>
    <behavior>
      <serviceMetadata/>
      <serviceDiscovery>
        <announcementEndpoints>
          <endpoint kind="udpAnnouncementEndpoint" />
        </announcementEndpoints>
      </serviceDiscovery>
    </behavior>
  </serviceBehaviors>
  <endpointBehaviors>
    <behavior>
      <endpointDiscovery enabled="true"/>
    </behavior>
  </endpointBehaviors>
</behaviors>
```

Managed service discovery

- **Ad-hoc discovery is limited to the local subnet**
  - To discover across networks, you need managed service discovery
- **Implementing managed service discovery is more involved**
  - You must implement a complete discovery proxy
  - WCF 4 comes with a DiscoveryProxy to provide the structure
- **Your implementation must define how to:**
  - Save (and cache) discovery announcements
  - Respond to incoming discovery probes
- **Use cases:**
  - Allow discovery to span subnets.
  - Limit broadcast on network
  - Make services discoverable on the Internet

Client side Managed Discovery

- No standardEndpoints (yet)
- Easiest path: create clients in code
- Use DiscoveryEndpoint(Binding, EndpointAddress) to talk directly to endpoint.

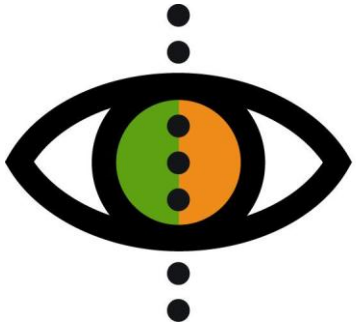
Service Side Managed Discovery

- For announcing new services, service uses AnnouncementEndpoint(Binding, Address) to talk to DiscoveryProxy
- No standard endpoints (yet)
- No need for DiscoveryEndpoint on Service!

Summary

- Discovery allows for reduced configuration
- Discovery reduced management of configuration as services move
- Can create automatically configuring applications
- Use announcement services to learn about services as they enter and leave your subnet.
- To handle Discovery across subnets or to reduce broadcasts, use managed Discovery

For more in-depth online developer training visit



pluralsight

see what you can learn

on-demand content from authors you trust