

Contents

- [Problem](#)
- [Solution](#)
 - [The standard configuration](#)
 - [Service names](#)
- [Example](#)
 - [Use Windows credentials](#)
 - [Configure TCP bindings](#)
- [Additional resources](#)

You can take control of the configuration of WCF communications in your [EntityServer](#) by directly working with the WCF `<system.serviceModel>` configuration section.

Problem

The `<system.serviceModel>` section is defined in the server's config file. When the *EntityServer* is hosted by IIS this will be the web.config, otherwise it will be either ServerService.exe.config or ServerConsole.exe.config depending on the host.

When you define the `<system.serviceModel>` in your configuration file, you're telling DevForce that you're taking control over the configuration of the WCF services comprising the *EntityServer*.

You can use the WCF Service Configuration Editor available from the Visual Studio **Tools** menu to edit the `<system.serviceModel>` information.

Solution

We can't explain all the nearly endless complexities of WCF configuration, but here we'll show you a few samples.

The standard configuration

DevForce will [automatically](#) determine the WCF configuration based on your environment. The sample below is what the standard configuration would look like if instead defined in config.

Here we define two services, "EntityService" and "EntityServer", and each has several endpoints - one for each type of client application environment supported. The endpoints configured are based on the *supportedClientApplicationType* value on the `<serverSettings>`. Here we're assuming an Enterprise license. All endpoints use compressed binary formatted messages over http.

```
<configuration>
<system.serviceModel>
<services>
<service name="IdeaBlade.EntityModel.Server.RemoteEntityService">
    <!-- Endpoint for Windows clients -->
    <endpoint address="" binding="customBinding" bindingConfiguration="compressedBinaryBinding"
        contract="IdeaBlade.EntityModel.Server.IEntityServiceContract" />

    <!-- Endpoint for Silverlight clients -->
    <endpoint address="sl" binding="customBinding" bindingConfiguration="compressedBinaryBinding"
        contract="IdeaBlade.EntityModel.Server.IEntityServiceContract" />
    <!-- Endpoint for Windows Store and Universal clients -->
    <endpoint address="store" binding="customBinding" bindingConfiguration="compressedBinaryBinding"
        contract="IdeaBlade.EntityModel.Server.IEntityServiceContract" />
    <!-- Endpoint for Windows Phone clients -->
    <endpoint address="wp" binding="customBinding" bindingConfiguration="compressedBinaryBinding"
        contract="IdeaBlade.EntityModel.Server.IEntityServiceContract" />
</service>
<service name="IdeaBlade.EntityModel.Server.EntityServer">
    <!-- Endpoint for Windows clients -->
    <endpoint address="" binding="customBinding" bindingConfiguration="compressedBinaryBinding"
        contract="IdeaBlade.EntityModel.Server.IEntityServerContract" />
    <!-- Endpoint for Silverlight clients -->
    <endpoint address="sl"
```

```

        binding="customBinding" bindingConfiguration="compressedBinaryBinding"
        contract="IdeaBlade.EntityModel.Server.IEntityServerContract" />
<!-- Endpoint for Windows Store and Universal clients -->
<endpoint address="store"
        binding="customBinding" bindingConfiguration="compressedBinaryBinding"
        contract="IdeaBlade.EntityModel.Server.IEntityServerContract" />
<!-- Endpoint for Windows Phone clients -->
<endpoint address="wp"
        binding="customBinding" bindingConfiguration="compressedBinaryBinding"
        contract="IdeaBlade.EntityModel.Server.IEntityServerContract" />
</service>

</services>
<bindings>

<customBinding>

    <binding name="compressedBinaryBinding">
        <gzipMessageEncoding>
            <readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647" maxArrayLength="2147483647" />
        </gzipMessageEncoding>
        <httpTransport maxReceivedMessageSize="2147483647" />
    </binding>
</customBinding>

</bindings>
<extensions>
    <bindingElementExtensions>
        <add name="gzipMessageEncoding" type="IdeaBlade.Core.Wcf.Extensions.GZipMessageEncodingElement, IdeaBlade.Core"/>
    </bindingElementExtensions>
</extensions>
</system.serviceModel>
</configuration>

```

Service names

With .NET 4.5, WCF now validates the service name definitions in your .config file and wants to see the full type name of the service class. You see these full type names in the samples above. You can use "friendly" names too, however, and DevForce will still find the services. Note you'll see an error indicator from Intellisense for these service names when editing your .config, but these "errors" are harmless and do not affect the runtime functioning of your application.

- You can use "EntityService" instead of "IdeaBlade.EntityModel.Server.RemoteEntityService".
- You can use "EntityServer" instead of "IdeaBlade.EntityModel.Server.EntityServer".
- If you're using data source extensions and require different configuration for each tenant, you can also use a service name starting with "EntityServer_" followed by the extension name, for example "EntityServer_TenantA".

Example

Use Windows credentials

Below is a sample binding adding Windows authentication and message security to the standard DevForce binding. This is not supported for Silverlight clients.

```

<customBinding>
    <binding name="wsBindingCustom">
        <gzipMessageEncoding>
            <readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647" maxArrayLength="2147483647" />
        </gzipMessageEncoding>
        <security authenticationMode="SspiNegotiated"></security>
        <httpTransport maxReceivedMessageSize="2147483647" />
    </binding>
</customBinding>

```

If the EntityServer is hosted in IIS with Windows Authentication enabled for the application, then Windows credentials must be provided with transport security:

```

<customBinding>
    <binding name="wsBindingCustom">
        <gzipMessageEncoding>

```

```

<readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647" maxArrayLength="2147483647" />
</gzipMessageEncoding>
<httpTransport maxReceivedMessageSize="2147483647" authenticationScheme="IntegratedWindowsAuthentication" />
</binding>
</customBinding>

```

Configure TCP bindings

This sample shows several different binding possibilities. You can define any number of binding definitions in your configuration, but it's the *binding* and *bindingConfiguration* attributes on the *endpoint* which determine the binding used. In this sample the "tcpBindingDefault" binding is used. If you try this on your own and wish to use another binding, be sure that you change the *binding* and *bindingConfiguration* on both endpoints.

This sample also shows that you can use either standard or custom WCF bindings. DevForce uses a *CustomBinding*, but this is not a requirement for your application. The standard bindings are better documented by Microsoft and therefore usually easier to use. Standard bindings do not offer compression.

Note that the addresses used here do not use the *.svc extension because the services are not hosted in IIS. You can use *net.tcp* with an IIS-hosted *EntityServer* if you install and configure the Windows Process Activation Service. *net.tcp* is the name for the TCP scheme in WCF.

You can use *net.tcp* with Silverlight clients, but we don't yet have a sample.

```

<configuration>
<system.serviceModel>
<services>
<service name="IdeaBlade.EntityModel.Server.RemoteEntityService">
<endpoint
  address=""
  binding="netTcpBinding" bindingConfiguration="tcpBindingDefault"
  contract="IdeaBlade.EntityModel.Server.IEntityServiceContract" />
</service>
<service name="IdeaBlade.EntityModel.Server.EntityServer">
<endpoint
  address=""
  binding="netTcpBinding" bindingConfiguration="tcpBindingDefault"
  contract="IdeaBlade.EntityModel.Server.IEntityServerContract" />
</service>
</services>
<bindings>
<netTcpBinding>
<!-- Standard tcp binding - uses Windows auth and transport security by default. -->
<binding name="tcpBindingDefault" maxReceivedMessageSize="2147483647" />
<!-- Tcp binding using message security. -->
<binding name="tcpBindingMessage" maxReceivedMessageSize="2147483647">
  <security mode="Message" />
</binding>
</netTcpBinding>
<customBinding>

<!-- DevForce custom binding with TCP and compression. -->
<binding name="tcpBindingCustom">
  <gzipMessageEncoding>
    <readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647" maxArrayLength="2147483647" />
  </gzipMessageEncoding>
  <tcpTransport maxReceivedMessageSize="2147483647" />
</binding>
<!-- Another custom binding, with TCP, compression and transport security. -->
<binding name="tcpBindingCustomSecure">
  <gzipMessageEncoding>
    <readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647" maxArrayLength="2147483647" />
  </gzipMessageEncoding>
  <windowsStreamSecurity />
  <tcpTransport maxReceivedMessageSize="2147483647" />
</binding>
</customBinding>
</bindings>
<extensions>
<bindingElementExtensions>
<add name="gzipMessageEncoding" type="IdeaBlade.Core.Wcf.Extensions.GZipMessageEncodingElement, IdeaBlade.Core"/>
</bindingElementExtensions>

```

```
</extensions>
</system.serviceModel>
</configuration>
```

Additional resources

- [Sample config files](#)
- [Securing Access to Services for Silverlight](#)
- [Custom Bindings](#)