

Action Line Framework Libraries

Framework de Configuración

El sistema de configuración que provee el Framework de Fwk permite a las aplicaciones (Back-End o Front-End) disponer un conjunto de mecanismos que facilitan el acceso a archivos de configuración.

Existen casos en que las distintas aplicaciones necesitaran persistir cierta información en algún lugar de acceso rápido que no sea la base de datos del sistema.

Básicamente el sistema de configuración consiste de un Servicio Web publico que provee de los archivos de configuración y dispone de un catalogo con todos los archivos físicos. Estos archivos son simples XMLs con una estructura especial como se muestra a continuación:

Estructura:

```
<Groups>
  <Group name="NombreGrupo_Uno">
    <Key name="NombrePropiedad_Uno" >
      <Value><![CDATA[ValorPropiedad_Uno]]</Value>
    </Key>
    <Key name="NombrePropiedad_Dos" >
      <Value><![CDATA[ValorPropiedad_Dos]]</Value>
    </Key>
  </Group>
  <Group name="NombreGrupo_Dos">
    .
    .
  </Group>
</Groups>
```

Código 1.0

Nota: El explicación de esta estructura se detallara en la sección de "Especificaciones técnicas" más abajo.

Por otro lado cada host cliente dispone de un Servicio de Windows que conoce la ubicación del Web Service recién nombrado e interactúa con los clientes o las distintas aplicaciones que utilizan realmente la información contenida en los archivos de configuración.

De acuerdo lo dicho mas arriba podemos identificar los siguientes elementos en la figura:

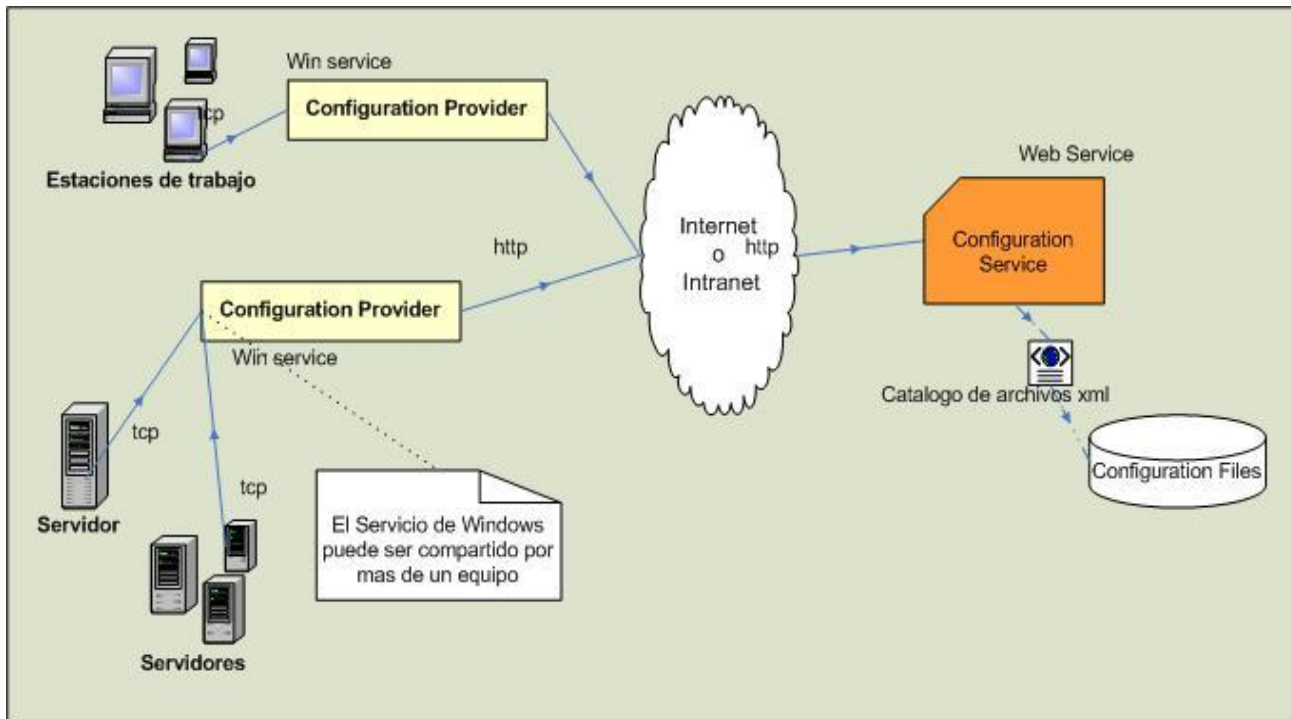


Figura 1.0

Configuration Service

Se trata de un Servicio Web que conoce el catalogo "**CatalogEntries**" de archivos XML y sabe donde se ubica cada uno de ellos en algún repositorio de disco.

Este catalogo provee, además de la información de ubicación de cada uno de los archivos, cierta información de extra sobre estado del archivo o si es cacheable o no, etc.

El Configuration Service provee de ciertos mecanismos que permiten la obtención del archivo de configuración y la información de la versión del mismo.

- Interfaz publica del servicio *Configuration Service*:

Método	Descripción
GetConfig	Devuelve el contenido completo de un archivo de configuración dado el nombre de archivo. Parámetros: <ul style="list-style-type: none">▪ FileName: Nombre de archivo *.xml que contiene los grupos y claves de configuración
GetFileVersionStatus	Compara versión del archivo especificado para determinar si es necesario actualizar. Parámetros:

- **FileName:** Nombre de archivo
- **ClientVersion:** Versión del archivo en el cliente

Tabla 2.0

Configuration Provider

Es un servicio de windows que tiene por objetivo principal escuchar peticiones de las aplicaciones que requieran informacion de alguno de los archivos de configuración. Este servicio puede ser instalado en cada una de las maquinas cliente o host o bien puede ser compartido por mas de una.

Este servicio tiene dos puntos de conexión bien definido como se muestra en la siguiente figura:

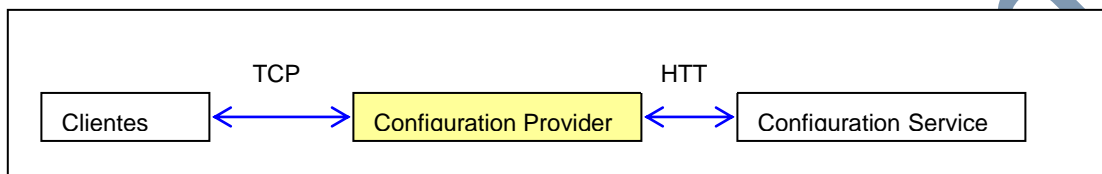


Figura 2.0

Por un lado a través de un puerto tcp los cliente se conectan al servicio. Esta conexión se define en un archivo de configuración de Remoting.

Ej:

<tcp://localhost:8989/ConfigurationHolder.rem>

En el otro extremo el servicio “Configuration Provider” establece en su configuración file una clave que apunta a la **url** donde se ubica el Servicio Web Configuration Service :

EJ:

<http://localhost/Fwk.Configuration.ConfigurationService/ConfigurationService.aspx>

- Interfaz publica del servicio *Configuration Provider*:

Método	Descripción
GetConfig	Ya descrito en la tabla 1.0
GetFileVersionStatus	Ya descrito en la tabla 1.0
GetProperty	Devuelve un grupo de un BaseConfigFile o archivo de configuración del catalogo. Parámetros: <ul style="list-style-type: none">▪ BaseConfigFileName: Nombre de archivo▪ GroupName: Nombre del grupo donde se encuentra la propiedad a obtener▪ PropertyName: Nombre de la propiedad que se desea obtener

GetGroup	Devuelve un grupo de un BaseConfigFile Parámetros: <ul style="list-style-type: none">▪ BaseConfigFileName: Nombre de archivo▪ GroupName: Nombre del grupo donde se encuentra la propiedad a obtener.
SetConfigurationFile	Configura un ConfigurationFile. Parámetros: <ul style="list-style-type: none">▪ ConfigurationFile: Objeto ConfigurationFile. Este objeto contiene información tal como:<ul style="list-style-type: none">✓ Fecha y hora de la obtención de la configuración.✓ Contenido del archivo descriptado.✓ Estado del ConfigurationFile (Ok, Expired ,Inconsistent)▪ FileName: Nombre de archivo *.xml de configuración.
InitializeLifetimeService	Redefinición de InitializeLifetimeService de MarshalByRefObject para configuración personalizada. El InitializeLifetimeService obtiene un objeto de servicio de duración para controlar la directiva de duración de esta instancia.

Tabla 3.0

Aplicaciones Clientes

Estos son los consumidores del servicio de configuración. Los clientes se comunican con el servicio de Windows para tal propósito este servicio de Windows puede ser Local o Remoto depende del diseño de arquitectura de la solución que se establezca.

Las distintas aplicaciones pueden ser alguno de los siguientes:

- Servicios de Windows
- Aplicaciones cliente
- Despachadores de servicios
- Etc.

Nota: Las especificaciones técnicas sobre como configurar cada uno de los distintos puntos anteriormente

Especificaciones técnicas

A continuación daremos los pasos a tener en cuenta al momento de configurar la aplicación en el **Cliente**, **Configuration Provider** y **Configuration Service**.

En el cliente o aplicación

1. Ante todo en el ejecutable de la aplicación debe tener la siguiente dll:

Fwk.Configuration.ConfigurationService

2. Abra el archivo App.config o NombreEjecutable.exe.Config con cualquier editor de texto.

En la sección `applicationSettings` asegúrese de contar con un nodo `"Fwk.Configuration.Properties.Settings"` y que quede configurado de la siguiente manera de la siguiente manera como se muestra en código 2.0

```
<<applicationSettings>
  <Fwk.Configuration.Properties.Settings>
    <setting name="BaseConfigFile" serializeAs="String">
      <value>ConfigurationSample.xml</value>
    </setting>
    <setting name="RemotingConfigFile" serializeAs="String">
      <value>RemotingConfigFile.config</value>
    </setting>
  </ Fwk.Configuration.Properties.Settings>
</applicationSettings>
```

Código 2.0

Donde:

- **RemotingConfigFile:** Es el nombre del archivo que apunta al servicio de Windows Configuration Provider.
- **BaseConfigFile:** es el nombre del archivo de configuración que la aplicación utilizara.

Nota: Este archivo es independiente al archivo App.config que pueda tener cualquier configuración.

3. Verifique que el archivo de configuración remota **RemotingConfigFile** al que apunta la configuración anterior (Código 2.0) exista físicamente en el cliente.

4. Verifique que el archivo [BaseConfigFile](#) al que apunta la configuración anterior exista en el repositorio de archivos xml, generalmente ubicado donde está el **WS ConfigurationService.asmx**.

Configuración de los archivos RemotingConfigFile y BaseConfigFile

- **Configurar el archivo RemotingConfigFile**

Este archivo contiene la información de remoting que permite conectarse al servicio de configuración "Configuration Provider".

Se trata de un archivo xml con cierta estructura Remoting que contiene información sobre canales y objetos remotos, por lo tanto su nodo root será `<system.runtime.remoting>`, la cual es necesaria para que el framework de .net (RemotingConfiguration) entienda que se trata de un cliente que quiere acceder a un objeto publicado "type" en cierta dirección "url".

A continuación mostraremos un ejemplo de archivo **RemotingConfigFile** típico que apunta a la mismo Host Local y al puerto "8989"

```
<system.runtime.remoting>
  <application>

    <client>
      <wellknown type="Fwk.Configuration.Common.ConfigurationHolder,
Fwk.Configuration.Common"
url="tcp://localhost:8989/ConfigurationHolder.rem" />
    </client>

    <channels>
      <channel name="tcp config" ref="tcp" port="0">
        <serverProviders>
          <formatter ref="soap" typeFilterLevel="Full" />
          <formatter ref="binary" typeFilterLevel="Full" />
        </serverProviders>
      </channel>
    </channels>

  </application>
  <customErrors mode="Off" />
</system.runtime.remoting>
```

Código 2.1

Nota: Si no dispone de este archivo lo puede crear usted mismo con cualquier editor de texto agregarle la información detallada en **Código 2.1** y sávelo con la extensión ***.config**.

Para configurar el cliente para que apunte al servicio de configuración modifique el valor de **url** con el nombre de Host y puerto donde esta corriendo el mismo.

```
url="tcp://[ServerName]:[Port]/ConfigurationHolder.rem"
```

La siguiente tabla muestra los valores posible a modificar y su significado

Valor	Descripción
ServerName	Reemplace este valor por el valor del host donde esta corriendo el servicio Configuration Provider.
Port	Reemplace este valor por el valor del puerto que tiene configurado el servicio de Windows Configuration Provider.
Elemento customErrors	Proporciona información acerca de los mensajes de error personalizados de una aplicación. El elemento customErrors se puede definir en cualquier nivel de la jerarquía de archivos de aplicación. <code><customErrors mode="Off On RemoteOnly" /></code> Ver Tabla 4.1 con el detalle de estos valores.-

Tabla 4.0

Elemento **customErrors**

Valor	Descripción
On	Todos los llamadores reciben información completa sobre excepciones.
Off	Todos los llamadores reciben información filtrada sobre excepciones.
RemoteOnly	Los llamadores locales reciben información completa sobre excepciones; los llamadores locales reciben información filtrada sobre excepciones.

Tabla 4.1

▪ Configuración del archivo BaseConfigFile

Desde el punto de vista del cliente lo que interesa aquí es que el archivo de configuración realmente exista en el servicio de Windows y de que este registrado en el catálogo de archivos.

Además también asegúrese de poner en el App.config de la aplicación el nombre correcto del archivo de configuración.

Por ejemplo: ConfigurationFileCliente.xml o ConfigurationFileRemotingService.xml

Configuration Provider

1. Verifique que este Servicio de Windows esté instalado e iniciado. El mismo puede encontrarse en la PC local o en algún Host remoto en el que se decida que se comparta el mismo.
2. Si el servicio está instalado vaya a la consola de Servicios de Windows y busque nombre "Prominente.Framework - Configuration Provider Service" selecciónelo e inícielo.
3. Verifique que el Configuration Provider Service tenga bien configurado el archivo de configuración:

Fwk.Configuration.ConfigurationProvider.exe.config

El siguiente código muestra un ejemplo práctico de cómo debería verse el archivo de configuración:


```
<applicationSettings>
  <Fwk.Configuration.Common.Properties.Settings>
    <setting
name="Fwk_Configuration_Common_ConfigurationService_ConfigurationService"
serializeAs="String">
      <value>
http://localhost:1282/Fwk.Configuration.ConfigurationService/ConfigurationService.aspx
      </value>
    </setting>
    <setting name="Lifetime" serializeAs="String">
      <value>3600</value>
    </setting>
  </Fwk.Configuration.Common.Properties.Settings>
</applicationSettings>

<system.runtime.remoting>
  <application name="ConfigurationProvider">
    <service>
      <wellknown mode="Singleton"
type="Fwk.Configuration.Common.ConfigurationHolder,
Fwk.Configuration.Common" objectUri="ConfigurationHolder.rem" />
    </service>
    <channels>
      <channel ref="tcp" port="8989"/>
    </channels>
  </application>
</system.runtime.remoting>
```

Código 3.0

Donde:

- **Fwk_Configuration_Common_ConfigurationService_ConfigurationService:**
Es la dirección donde está publicado el Servicio Web **ConfigurationService**.
- **Lifetime:**
Es el tiempo de vida del Servicio Web.
- **ConfigurationProvider**
Es el nombre de la clase remota, proveedora de configuración, hospedada en el Servicio Windows.

Configuration Service

1. Verifique que el Servicio Web (**ConfigurationService.asmx**) esté instalado y corriendo en IIS. Una forma sencilla de probar esto es copiando la **url** donde apunta el servicio de configuración y pegarla a un explorador.

Ejemplo:

<http://localhost:1282/Fwk.Configuration.ConfigurationService/ConfigurationService.asmx>

<http://localhost/ConfigurationService/ConfigurationService.asmx>

2. Verifique que el Servicio Web posea la carpeta "**ConfigurationFiles**" con los siguientes archivos:

- **Catalog.xml**: Que contiene referencias y valores correspondientes a los distintos archivos de configuración.
- El archivo **Catalog.xml** debe tener la siguiente estructura:

```
<CatalogEntries>
  <CatalogEntry name="ConfigurationSample.xml">
    <CurrentVersion>1.0</CurrentVersion>
    <Available>true</Available>
    <ForceUpdate>true</ForceUpdate>
    <TTL>60</TTL>
    <Encrypt>false</Encrypt>
    <BaseConfigFile>true</BaseConfigFile>
    <Cacheable>true</Cacheable>
  </CatalogEntry>
  <CatalogEntry name="OtroConfigurationSample.xml">
    <CurrentVersion>2.0</CurrentVersion>
    <Available>true</Available>
    <ForceUpdate>false</ForceUpdate>
    <TTL>120</TTL>
    <Encrypt>false</Encrypt>
    <BaseConfigFile>true</BaseConfigFile>
    <Cacheable>true</Cacheable>
  </CatalogEntry>
</CatalogEntries>
```

Código 4.0

La siguiente tabla muestra los atributos que se pueden asignar a los archivos de configuración en el Catalogo de archivos.

Valor	Descripción
CurrentVersion	Indica versión actual del archivo. EJ: 1.0 , 2.0
Available	Indica si está disponible o no. Generalmente no se eliminan los archivos físicamente del catalogo y del disco si no que simplemente se marca este atributo como false.

ForceUpdate	Indica si la actualización es obligatorio u opcional
TTL	Tiempo de vida en segundos del archivo en el ConfigurationHolder (cero: no caduca nunca) Es decir determina cuanto durara la información en memoria del host que tiene el Servicio de Windows que pidió la información al Servicio Web.
Encrypt	Indica si el Web Service encripta el contenido
BaseConfigFile	Indica si el archivo tiene estructura grupo-propiedad
Cacheable	Indica si el archivo puede ser almacenado en memoria por el ConfigurationHolder

Tabla 5.0

- **Archivos de Configuración:** Por ejemplo ConfigurationSample.xml. Estos archivos contienen valores de configuración de cada aplicación, organizado en grupos con sus nombres, propiedades para cada uno y sus respectivos valores.

El Código 4.1 nos muestra un ejemplo práctico de cómo se podría ver un archivo xml de estas características:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Groups>
  <Group name="FuntionalsExceptions">
    <Key name="Excepcion1" encrypted="False">
      <Value><![CDATA[Error 2]]></Value>
    </Key>
    <Key name="Excepcion2" encrypted="False">
      <Value><![CDATA[Error 2]]></Value>
    </Key>
  </Group>
  <Group name="ClientMessages">
    <Key name="Message1" encrypted="False">
      <Value><![CDATA[Hola mundo.]]></Value>
    </Key>
    <Key name=" Message2" encrypted="False">
      <Value><![CDATA[Ejemplo de fwk de configuracion.]]></Value>
    </Key>
  </Group>
</Groups>
```

Código 4.1

Los archivos de configuración como podemos apreciar están divididos en “Grupos” donde cada uno de ellos es un repositorio de ciertas configuraciones que pertenecen a un mismo nivel de abstracción. Por ejemplo podemos tener un grupo llamado “*FunctionalExceptions*” donde los elementos que la compongan sean todos mensajes de errores de tipo funcional o de negocio que mostrara una aplicaron en las pantallas de los clientes. Por otro lado podemos tener otro grupo que contenga información de cadenas de conexión a distintas bases de datos o rutas de acceso a archivos de recursos.

Los distintos grupos que conforman una archivo de configuración como vimos están formados por uno o mas elementos estos elementos son “Keys” y están identificados por un atributo único dentro del grupo llamado “name”.

El valor de estas Keys o claves se encuentra el nodo “Value” dentro de un CDATA[]. Este valor es el que realmente le interesa a aquella aplicaron que la solicite.

Versión preliminar