

Fwk.Configuration

BLOQUE DE CONFIGURACIÓN

El bloque 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 o bases de datos 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 ni tampoco el archivo .config.

Básicamente el sistema de configuración consiste en una serie de componentes del framework fwk que proveen de los archivos o base de datos de configuración.

La estructura de una configuración vista como xml podría ser:

Estructura:

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

Código 1.0

Grupos

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 “*FuntionalExceptions*” donde los elementos que la compongan sean todos mensajes de errores 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.

Keys

Los distintos grupos están formados por uno o más elementos llamados “**Keys**” y están identificados por un atributo único dentro del grupo llamado “**name**”.

Valor

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.

Aplicaciones Clientes o consumidoras de configuracion

Los consumidores del servicio de configuración pueden ser alguno de los siguientes:

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

Estructura del bloque de Configuración

La ubicación del bloque de configuración se encuentra en el namespace *Fwk.configuration* y su componente más importante es el **ConfigurationManager**. Este es el único que utilizara el desarrollador para interactuar con el bloque de configuración.

- Interfaz publica de ConfigurationMannager:

Método	Descripción
GetConfigurationFile	Obtiene un ConfigurationFile con la estructura vista en Codigo 10 Parámetros: <ul style="list-style-type: none">▪ providerName: Nombre del proveedor de configuración utilizado
GetProperty	Devuelve un grupo de un BaseConfigFile o archivo de configuración del catalogo. Parámetros: <ul style="list-style-type: none">▪ providerName: Nombre del proveedor de configuración utilizado▪ 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">▪ providerName: Nombre del proveedor de configuración utilizado▪ groupName: Nombre del grupo donde se encuentra la propiedad a obtener.

Tabla 1.0

Proveedores de configuración

Los proveedores de configuración son los que dan origen a la configuración propiamente dicha. Estos tienen por objetivo principal Identificar el origen de dato de las configuraciones.

Propiedades de un proveedor

Propiedad	Descripción y valores
name	Nombre que identifica el proveedor. Este valor es que utiliza el ConfigurationManager para trabajar con una configuración determinada
type	Indica el tipo de origen de almacén de la información. Hasta el momento existen : <ul style="list-style-type: none">▪ xml La información está contenida en un archivo xml▪ sqldatabase: La información está contenida en una tabla de sql cuya estructura esta dado por los script de instalación de bases de datos del framework▪ servicewrapper La información proviene de un servicio dispatcher del framework. Este servicio puede estar hosteado en un Web service o en un Remoting Win Service
sourceinfo	Identifica diferentes orígenes de información según el tipo de proveedor <ul style="list-style-type: none">• Si type: = "xml" → "ignore sourceinfo"• Si type: = "sqldatabase" → nombre de cadena de conexión• Si type: = "servicewrapper" → Apuntara a un WrapperConfig en el archivo de configuración y este wrapper a un Dispatcher web o win service
baseConfigFile	Si type = "xml" Nombre de archivo físico de configuración XML si Si type = "sqldatabase" identificador de configuración en la base de datos.

Tabla 2.0

Almacén según el tipo de proveedor xml y database

El almacenamiento de la información de configuración depende del tipo de proveedor. Según el atributo **type** en la tabla 2.0. puede ser en un archivo o una base de datos.

A continuación se detallan ambos casos.

- **Archivos de Configuración provider type xml**

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 2.0 nos muestra un ejemplo práctico de cómo se podría ver un archivo xml de estas características:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ConfigurationFile>
<Groups>
  <Group name="FuntionalsExceptions">
    <Keys>
      <Key name="Excepcion1" encrypted="false">
        <Value><![CDATA[Error 2]]></Value>
      </Key>
      <Key name="Excepcion2" encrypted="false">
        <Value><![CDATA[Error 2]]></Value>
      </Key>
    </Keys>
  </Group>
  <Group name="ClientMessages">
    <Keys>
      <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>
    </Keys>
  </Group>
</Groups>
</ConfigurationFile>
```

Código 2.0

▪ Tablas SQL de Configuración (provider type sqldatabase)

Esta es otra opción válida para almacenar todas las configuraciones que deseemos. Si nuestro proveedor es sqldatabase entonces podremos cargar y recuperar nuestras configuraciones en una tabla de SQL Server

Toda la información será almacenada en una tabla llamada fwk_ConfigMannager

La estructura de la tabla es la siguiente.

Nombre: fwk_ConfigMannager

ConfigurationFileName	nvarchar(50)
[group]	nvarchar(50)
[key]	nvarchar(50)
encrypted	bit
value	nvarchar(1000)

Para este tipo de proveedores es necesario que exista una base de datos con una tabla cuya estructura este representada por el siguiente script.

%SYSTEMDRIVE%/Program Files/Fwk_10.3/scripts/Fwk_ConfigMannager.sql

Procedimiento de configuración en la aplicación

A continuación se detalla el procedimiento para configurar una la aplicación que utilice el sistema de configuración.

1. Abra el archivo App.config o NombreEjecutable.exe.Config con cualquier editor de texto.
2. Agregue al archivo la sección **FwkConfigProvider** en **configSections**

```
<configSections>
  <section name="FwkConfigProvider" type="Fwk.ConfigSection.ConfigProviderSection, Fwk.Bases" />
</configSections>
```

Código 3.0

3. Agregue la sección con el/los providers de configuración y establezca un proveedor por defecto

```
<!--Configuracion del los proveedores de configuracion del framework -->
<FwkConfigProvider defaultProviderName="p1">
  <Providers>
    <add name="p1" type="local" baseConfigFile="ConfigurationManager_test.xml" />
  </Providers>
</FwkConfigProvider>
```

Código 3.1

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

```
<configuration>
  <configSections>
    <section name="FwkConfigProvider" type="Fwk.ConfigSection.ConfigProviderSection,
Fwk.Bases" />
  </configSections>

  <connectionStrings>

    <add name="test"
      connectionString="Database=fwktest;Server=(local)\SQLEXPRESS;Integrated
      Security=SSPI" providerName="System.Data.SqlClient" />
  </connectionStrings>

  <!--Configuracion del los proveedores de configuracion del framework -->
  <FwkConfigProvider defaultProviderName="p1">
    <Providers>
      <add name="p1" type="local" baseConfigFile="ConfigurationManager.xml"
        sourceInfo="" />
      <add name="p2" type="sqldatabase " sourceInfo ="test" baseConfigFile =
        "MyApp1" />
    </Providers>
  </FwkConfigProvider>
</configuration>
```

Código 3.2

Sistema de configuración a través de servicios.

En ocasiones la será necesario que todo lo relativo a configuración sea accedida por servicios y que los clientes que la utilizan no tengan acceso real al origen de esta información. Es decir que no conozcan si los datos provienen de un archivo o alguna base de datos determinada..

Para tal propósito existe una tercera configuración llamada “servicewrapper”

Este tipo de configuración requiere que en algún lugar exista un service dispatcher ejecutando servicios soa del framework configurado para tener acceso al bloque de configuración bajo alguna o la combinación de las dos formas mencionada anteriormente. XML o SQL

Por otro lado requerirá clientes (aplicaciones de cualquier tipo) que cuenten con un provider de este tipo y un fwkwrapper que apunte a tal dispatcher.

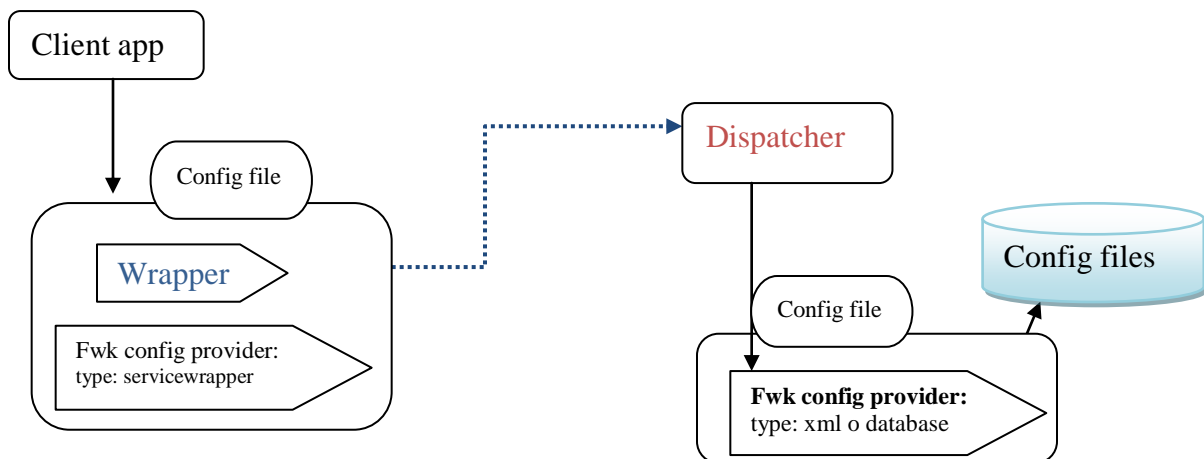


Imagen 2.0

Procedimiento para configurar aplicaciones y tipo de configuración type = “servicewrapper”

- 1- Edite el .config del dispatcher (web o win service remoting o wcf)
 - 2- Agregue uno o mas proveedores de configuración de tipo sql o xml
Ver cuadro de código código 3.0
 - 3- Agregue una configuración de Service Metadata para los servicios de configuración del framework. Este servicemetadata debe apuntar al Metadata.xml o base dedatos que tenga la información del ensamblado de servicios del framework.
- 3.2 Si no cuenta aun con la metadata, dirijase al archivo *Fwk_Metadata.xml* y alójela en un lugar accesible para el dispatcher. EJ el /bin del servicio de Windows o web service.

%SYSTEMDRIVE%/Program Files/Fwk_10.3/scripts/Fwk_Metadata.xml

Nota: para entender como configurar la metadata de servicios y como como manejar la herramienta de metadatos de servicios puede leer el documento:

%SYSTEMDRIVE%/Program Files/Fwk_10.3/docs/ServiceManagement.docx

```
<configSections>
  <section name="FwkServiceMetadata" type="Fwk.ConfigSection.ServiceProviderSection,
Fwk.Bases"/>
</configSections>

<FwkServiceMetadata defaultProviderName="met_pelsoft">
  <Providers>
    <add name="met_pelsoft" type="xml" appid="" sourceinfo="PelsoftMetasvc.xml"
securityProviderName=""/>

    <add name="dispatcher_config" type="xml" appid="" sourceinfo="Fwk_Metadata.xml"
securityProviderName=""/>

  </Providers>
</FwkServiceMetadata>

<FwkConfigProvider defaultProviderName="svc">
  <Providers>
    <add name="SVC" type="servicewrapper"
baseConfigFile="ConfigurationManager_Test"
sourceinfo="wrapper_remoto"/>
  </Providers>
</FwkConfigProvider>
```

Código 4.0

4- Ubique el archivo de configuración del cliente y edítelo:

4.1 Agregue un proveedor de configuración:

```
<FwkConfigProvider defaultProviderName="svc">
  <Providers>
    <add name="SVC" type="servicewrapper"
baseConfigFile=""
sourceinfo="wrapper_config"/>
  </Providers>
</FwkConfigProvider>
```

Código 4.1

4.2 Agregue un proveedor de FwkWrapper de tipo WebServiceWrapper

```
<FwkWrapper defaultProviderName="wrapper_servicios">
  <Providers>

    <add name="wrapper_servicios"
      type="Fwk.Bases.Connector.LocalWrapper,Fwk.Bases.Connector"
      appId="pelsoft" serviceMetadataProviderName="dispatcher_metadata" sourceinfo=""/>

    <add name="wrapper_config"
      type="Fwk.Bases.Connector.WebServiceWrapper,Fwk.Bases.Connector" appId="pelsoft"
      serviceMetadataProviderName="dispatcher_config"
      sourceinfo="http://localhost:38091/SingleService.asmx"/>
  </Providers>
</FwkWrapper>
```

Código 4.2

Nota: Es muy importante que el wrapper que este destinado para los servicios de configuración (**wrapper_config**) apunte a un dispatcher que posea un serviceMetadata configurado para ejecutar servicios de configuración. Tal como muestra la **Imagen 2.0**

Cliente: Wrapper name= wrapper_config → serviceMetadataProviderName = "dispatcher_config"
Dispatcher: Metadata name="dispatcher_config" → sourceinfo="Fwk_Metadata.xml"

Ademas el nombre del proveedor de configuración del lado del cliente debe coincidir con un proveedor de configuración en el lado del server.-

Para utilizarlo en el código se hace de manera similar a como se hace en las otras configuraciones. Lo único que tiene q tener en cuenta es que cuando llame a un método de la clase **ConfigurationManager** debe especificar el proveedor de configuración existente en el server.

```
string msg = ConfigurationManager.GetProperty("svc", "ClientMessages", "msg_bienvenida");
```