### Fwk Framework Libreries

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 en una serie de componentes del framwwork que proveen de los archivos o base de datos de configuración.

La estructura de una configuración determinada vista como xml podría ser:

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.

Utilización

El componente principal y único que utilizara el desarrollador para interactuar con el bloque de configuración será

ConfigurationManager *(Fwk.Configuration.ConfigurationManager)*

* + Interfaz publica del servicio *ConfigurationMannager*:

|  |  |
| --- | --- |
| **Método** | **Descripción** |
| **GetConfigurationFile** | Obtiene un ConfigurationFile con la estructura vista en Codigo 10  Parámetros:   * ***providerName***: Nombre del proveedor de configuración utilizado |
| **GetFileVersionStatus** |  |
| **GetProperty** | Devuelve un grupo de un BaseConfigFile o archivo de configuración del catalogo.  Parámetros:   * ***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:   * ***providerName***: Nombre del proveedor de configuración utilizado * ***groupName***: Nombre del grupo donde se encuentra la propiedad a obtener. |
|  |  |
|  |  |

**Tabla 3.0**

**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.

**Almacenamiento de la información de configuración**

El almacenamiento puede se un archivo XML o una base de datos de sql cualquiera.

**Aplicaciones** **Clientes**

Estos son los consumidores del servicio de configuración. Los clientes se utilizan *ConfigurationMannager* para tal propósito este servicio de Windows puede ser Local o Remoto depende del diceño 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 - Abra el archivo App.config o NombreEjecutable.exe.Config con cualquier editor de texto y agregue:

1. Agregar la configuración de sección **FwkConfigProvider** en **configSections**

<configSections>

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

</configSections>

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

<!--Configuracion del los proveedores de configuracion del frameworck -->

<FwkConfigProvider defaultProviderName="p1">

<Providers>

<add name="p1" type="local" sourceInfo="ConfigurationManager.xml" appid=""/>

<add name="p2" type=" sqldatabase " sourceInfo ="test" appid="bgbang" />

</Providers>

</FwkConfigProvider>

**Código 2.0**

Donde:

* **sourceInfo**: depende del tipo de proveedor

Si type: = “xml” = Archivo de configuracion XML

Si type = “sqldatabase” 🡺 nombre de cadena de conexion

* **appid*:***  identificador útil cuando estamos nuestra configuración no es única en una base de datos a la cual

apunta nuestro provider.

* **type**: Indica el tipo de origen de almacén de la información.

Hasta el momento existen

* + **local** 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 instalacion de bases de datos del framework

El siguiente código muestra un ejemplo practico de como deberia 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 frameworck -->

<FwkConfigProvider defaultProviderName="p1">

<Providers>

<add name="p1" type="local" sourceInfo="ConfigurationManager.xml" appid=""/>

<add name="p2" type=" sqldatabase " sourceInfo ="test" appid="bgbang" />

</Providers>

</FwkConfigProvider>

</configuration>

**Código 3.0**

* ***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 *“FuntionalsExceptions”* 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.

* ***Tablas sql de Configuración provider tye sqldatabase***

Esta es otra opcion valida para almacenar todas las configuraciones que deseemos. Si nuestro porveedor es sqldatabase entonces podremos cargar y recuperar nuestraas configuraciones en una tabla de SQL Server

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)

* **ConfigurationFileName**: Vendría a ser lo que es para el tipo xml el nombre del archivo
* **Group** tiene el mismo significado que groupname
* **Key:** reprecenta property name
* **Value:** es el valor de la propiedad
* **Encrypted** determina si la información esta encriptado o no.

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

*Fwk\_ConfigMannager.sql*