ELECTRONIC DOCUMENT LIBRARY

Es conjunto de clases que permiten la generación de comprobantes fiscales digitales en sus diferentes versiones y las cuales están desarrolladas en código nativo para:

Delphi

- Se utilizo la versión 7 de Borlando Delphi.
- Todo el código de las librerías es 100 % nativo, no se hace uso de componentes externos (dll, ocx, etc).

Dot Net

- Como lenguaje se usa C# en su versión 2.0.
- Todo el código de las librerías es 100 % nativo, no se hace uso de componentes externos (dll, ocx, etc).

Características Generales

- Generación de CFD versión 1.0, 2.0 y 3.0
- Generación de todo tipo de complementos
- Generación de diferentes tipos de adendas y en diferentes formatos (XML, EDI, Texto, etc.).
- Diferentes tipos de pruebas para asegurar la validez del documento.
- Validación de comprobantes generados por terceros, no importando la versión.
- Generación del código de barras bidimensional para la impresión del CFDI.

A continuación describo a detalle cada una de las clases que son usadas para la generación y validación de los comprobantes.

Electronic Certificate

Es la clase encargada de manejar todo lo relacionado con los certificados de sello, entre sus características tenemos:

- Lectura de certificados y llaves privadas del SAT (X509 y PKCS7)
- Lectura de archivos PKCS12
- Lectura de certificados de partir de una cadena de texto en Base 64.
- Almacenar el certificado en una cadena de texto codificándolo en Base 64.
- Capacidad para validar un certificado contra una autoridad certificadora.
- Expone información del certificado:
    - Vigencia del certificado
    - Numero de serie
    - Otros.

No se hace uso intensivo de esta clase, ya que solo se debe crear para cargar el certificado con el que se desea firmar el documento.

Manage Electronic Document

Esta clase es la encargada de administrar todos los recursos necesarios para la generación y validación de un CFD. Entre estos recursos tenemos:

Listado de folios autorizados.

- Esta clase permite verificar que el folio asignado al documento fue previamente autorizado por el SAT; para esto se utiliza el archivo de folios publicado en el sitio FTP del SAT.
- Lo único que debe hacer el desarrollador es definir la ruta física del archivo; ya que su uso es de forma automática cuando se esté generando o validando un documento.

Listado de certificados revocados.

- Esta clase permite verificar que el certificado que se va a usar para firmar el documento o el que se usó, según sea el caso, no estuviera revocado o vencido al momento en que se genera o se generó el documento.
- Lo único que debe hacer el desarrollador es definir la ruta física del archivo; ya que su uso es de forma automática cuando se esté generando o validando un documento.
- Este archivo es proporcionado por el SAT a través de su sitio FTP.

Listado de las autoridades certificadoras.

- Esta clase es usada para verificar que el certificado que se va a usar para firmar o que se usó para el proceso de firmado, fue emitido por una autoridad certificadora, en este caso el SAT.
- Esta clase lee un archivo XML que contiene todas las autoridades. Dicho archivo es generado y mantenido manualmente, en cada liberación se entregará precargado este archivo.
- Lo único que debe hacer el desarrollador es definir la ruta física del archivo; ya que su uso es de forma automática cuando se esté generando o validando un documento.

Schema

- Esta clase es la encargada de validar el documento contra los diferentes esquemas publicados por el SAT, esto es, el del comprobante y los complementos según sea el caso.
- Su uso es de forma automática y lo único que debe de hace el desarrollador es definir a través de propiedades cada uno de los esquemas del CFD y de los complementos.

Electronic Certificate List

- Esta clase es una lista que contiene certificados electrónicos y se creó debido a que en la versión 1.0 Y 2.0 del CFD el certificado no es un dato obligatorio; su objetivo es ir almacenando los certificados de los emisores de CFD para ser usados en el momento de la validación. Cada vez que se valida un CFD se puede configurar para que el certificado de dicho CFD – si lo trae incorporado – se agregue a la lista.
- Lee un archivo XML que contiene todos los certificados y los usa al momento de la validación de un CFD.
- Su uso es de forma automática y lo único que debe de hace el desarrollador es definir la ruta de dicho archivo.

Save

- Esta clase fue creada para definir como sería el proceso de generación del CFD, esto es, a través de ella se define las características que se ejecutaran durante el proceso de generación; esta clase en sí misma no ejecuta el proceso de generación solo contiene la definición del mismo.
- A continuación se describen dichas características:
    - ***Validar contra el esquema:*** Con esta opción estamos definiendo que durante el proceso de generación del documento, esté debe ser validado contra el esquema (ó esquemas si tiene complementos) antes de ser generado, con la finalidad de verificar lo definido por el SAT.
    - ***Validar información:*** Con esta opción indicamos que se lleve a cabo una validación de la información, esto es por ejemplo, que el RFC del emisor y el receptor cumpla con ciertas características o que el subtotal más impuestos sea igual al total, etc.
    - ***Validar el folio:*** Aquí indicamos que antes de generar el CFD se debe de validar que el folio que se va asignar al documento haya sido autorizado por el SAT.
    - ***Validar el certificado con la autoridad certificadora.*** En este caso se verifica que el certificado con el que se va a firmar el documento haya sido emitido por una autoridad certificadora válida.
    - ***Validar la vigencia del certificado:*** En este caso estamos indicando que se debe verificar que el certificado no esté revocado o vencido al momento de generar el documento.
    - ***Validar el RFC del emisor.*** En este caso se indica que el RFC del emisor debe coincidir con la información contenida en el certificado usado para firmar.
    - ***Agregar el certificado:*** Indica que el certificado debe ser convertido en Base 64 y almacenado en el documento.
    - ***Firmar:*** En este punto estamos indicando que se genere el documento junto con su sello.
- Esta clase ha sido creada de esta forma para darle versatilidad al desarrollador al momento de generar el archivo, ya que éste podría :
    - Querer generar el XML sin firmarlo,
    - Generar en CFD sin incluir el certificado

    - Generar el CFD sin validar si el folio fue autorizado o no por el SAT.
    - Otros…
- Mi recomendación es activar todas las opciones para que el proceso sea mas completo y seguro; aunque no sea estrictamente necesario.
- Certificado es una propiedad que permite almacenar el certificado creado anteriormente y el cual será usado durante el proceso de generación.

Load

- Esta clase es la contra parte de la clase anterior y comparte muchas de sus características, a continuación se van enumerar, pero sólo se explican aquellas que son nuevas:
    - Validar contra el esquema
    - Validar información
    - Validar el folio
    - Validar el certificado con la autoridad certificadora
    - Validar la vigencia del certificado
    - Validar el RFC del emisor
    - ***Agregar el certificado al listado de certificados.*** Como recordaran líneas arriba comente acerca del listado de certificados, activando esta opción se va agregar el certificado cuando se valide un CFD, con esto si en un futuro un documento no trae certificado se buscará en este listado y será usado para el proceso de validación.
    - Validar el sello del documento.
- Igualmente, se ha creado de esta forma para dar versatilidad al desarrollador, mi recomendación es activar todas las opciones.
- Contiene una propiedad certificado en donde se tiene el certificado que fue usado por el emisor para generar el documento, es accesible al desarrollador por si desea obtener información del mismo.

Electronic Document

- Esta clase es la que tiene todo el conocimiento para generar y validar un documento.
- Hace uso de las clases anteriormente definidas.
- Contiene la clase a la cual se le asignan todos los datos que serán usados para generar el documento.
- Contiene los métodos que permite ejecutar las operaciones como tal:
    - LoadFromFile
    - SaveToFile
    - Otros…

Existen 3 clases que son la base para todos los objetos en la librería, a continuación se describen dichas clases, sus métodos y propiedades.

TFIELDBASE

Esta clases es la base para todos los campos dentro de la librería; no se hace uso directo de dicha clase, lo que se hace es heredar otras clases que implementan métodos abstractos de esta clase.

Métodos

- *function IsAssigned() : Boolean;*

  Este método permite saber si el campo tiene un valor asignado, es muy usado para los datos opcionales. Cuando a un campo se le asigna un valor esta función devuelve verdadero; no importando si el valor asignado fue vacio, esto para el caso de que el campo sea de tipo string. La única forma de que este método devuelve false es cuando se crea el campo o cuando se inicializa con el método clear().

- *procedure Clear();*

  Procedimiento que permite limpiar el valor de un campo, si el campo es de tipo numérico o fecha, el valor que será asignado al campo una vez sea ejecutado dicho método es 0 y si el campo es de tipo string será vacio.

- *procedure FillWithRandomData();*

  Este procedimiento tiene como finalidad llenar el campo con un valor aleatorio; su funcionalidad no está totalmente completa, por lo tanto no se recomienda su uso; ha sido creado para ser usado como apoyo en el desarrollo; en un futuro se terminara su implementación y será totalmente funcional, por el momento no se recomienda su uso.

- *function AsString() : String;*

  Función que permite obtener el valor del campo como un string; el resultado obtenido ya se encuentra formateado de acuerdo a los requerimientos del SAT.

- *procedure SetData(const value : string);*

  Es un procedimiento para uso interno, NO DEBE DE SER USADO DE FORMA DIRECTA.

Propiedades

- *TagName*

  Esta propiedad es de sólo lectura y contiene el nombre del atributo dentro del XML que representa el campo. El desarrollador nunca tendrá la necesidad de intervenir con esta propiedad, es sólo de uso interno.

De esta clase desciende muchas otras:

- TFieldString
- TFieldInteger
- TFieldDate
- TFieldTime
- TFieldDateTime

Que implementan los métodos abstractos y que tienen una propiedad VALUE que pemirte asignar y obtener el valor del campo.

TFLOAFIELD

Esta clase hereda de la clase TFieldBase, su única peculiaridad es que tiene una propiedad adicional.

Propiedades

- *Decimales*

  Esta propiedad permite definir el número de decimadles que se usaran representar el valor del campos como un string. Su valor permitido es entre 2 y 6. Para la versión 1.0 del CFD se deben manejar 2 decimales, que fue lo que determino el SAT, para las otras versiones puede ser máximo 6.

TObjectBase

Esta clase es la base de todos los objetos contenedores, un ejemplo de esto es TEmisor, TReceptor, TConcepto, etc.

Esta clase nunca es usada de forma directa, solo sirve como clase abstract que implementa cierta funcionalidad.

Métodos

- *function IsAssigned() : Boolean;*

  Este método permite saber si alguna de las propiedades del objeto tiene un valor asignado, es muy usado para los datos opciones. Cuando a algunas de sus propiedades se le asigna un valor esta función devuelve verdadero; no importando si el valor asignado fue vacio, esto para el caso de que la propiedad sea de tipo string. La única forma de que este método devuelve false es cuando se crea el objeto o cuando se inicializa con el método clear().

- *procedure Clear();*

  Procedimiento que permite limpiar de una sola vez todas las propiedades del objeto.

- *procedure FillWithRandomData();*

  Este procedimiento tiene como finalidad llenar cada una de las propiedades del objeto con un valor aleatorio; ha sido creada para ser usado como apoyo en el desarrollo.

Propiedades

- *TagName*

  Esta propiedad es de sólo lectura y contiene el nombre del atributo dentro del XML que representa el objeto. El desarrollador nunca tendrá la necesidad de intervenir con esta propiedad, es sólo de uso interno.

TListBase

Esta clase es la base de las listas de objetos, un ejemplo de esto es TConceptos, TPartes, TRetenciones, etc.

Esta clase nunca es usada de forma directa, solo sirve como clase abstract que implementa cierta funcionalidad.

Es objeto es responsable de todos los objetos que crea, es por eso que se encarga de la creación y liberación de los mismos.

Métodos

- *function IsAssigned() : Boolean;*

  Este método permite saber si la lista está vacía o no.

- *procedure Clear();*

  Elimina todos los objetos de la lista, liberando los de memoria y dejando la lista vacía.

- *procedure FillWithRandomData();*

  Este procedimiento tiene como finalidad llenar cada una de los objetos de la lista con un valor aleatorio; ha sido creada para ser usado como apoyo en el desarrollo.

- *Function Add() : Integer;*

  Con este procedimiento se agregan objetos a la lista. El valor devuelto es la posición que ocupa el nuevo objeto en la lista.

- *procedure Delete(Const index : Integer);*

  Elimina un objeto de lista, el cual es liberado de memoria. Ejecutando este método el tamaño de la lista cambia.

Propiedades

- *TagName*

  Esta propiedad es de sólo lectura y contiene el nombre de la lista dentro del XML. El desarrollador nunca tendrá la necesidad de intervenir con esta propiedad, es sólo de uso interno.

- *Count*

Proporciona el numero de objetos que contiene la lista..