

CONTPAQi SDK

Emisión del complemento de Carta Porte 2.0 mediante SDK

CONTPAQi Comercial Premium (v7.3.0 o superior)



Introducción

Introducción

Nuestra autoridad fiscal ha realizado algunos cambios en la emisión de documentos que afecta directamente al uso del complemento Carta Porte el cual ahora debe contener una estructura fija dentro del XML distinta a como se manejaba en pasado, posteriormente a este cambio le surgen algunos inconvenientes que son ajustados en una segunda versión de dicho complemento (Carta Porte 2.0) y que nuestro aplicativo CONTPAQi Comercial Premium te da la oportunidad de manejar.

En este documento se abordará exclusivamente la emisión de dicho complemento haciendo uso de una herramienta que se incluye para los desarrolladores "CONTPAQi SDK Comercial Premium".

En este documento solo tratará el tema de la Emisión de documentos generados en el sistema con la incorporación del complemento Carta Porte 2.0. Cabe mencionar que la manipulación de dicho complemento (Carta Porte 2.0) en cuanto a su llenado y/o modificación dentro del sistema no es una funcionalidad que actualmente esté cubierta mediante SDK.

Objetivo

Dar conocer el proceso de emisión de CFDI cuando este requiere incorporar el complemento de Carta Porte 2.0.

Contenido

Requerimientos.

Recomendaciones.

Implementación.

- Escenario 1: Documento con Complemento ya creado en el sistema.
- Escenario 2: Documento creado en el sistema, pero sin complemento.

Requerimientos:

- Versión mínima del sistema CONTRPAQí Comercial Premium 7.3.0 SP3.
- Versión mínima de CONTRPAQí SDK 14.0.1.
- Tener una instalación estable y funcional de CONTRPAQí Comercial Premium y CONTRPAQí SDK, para el caso de SDK un desarrollo previo donde ya se pueda acceder a una empresa por este medio.
- Conocer las reglas de llenado del complemento Carta Porte 2.0 estipuladas por la autoridad fiscal (SAT), se puede consultar en [Instructivo de Llenado del CFDI](#) .
- Contar con un documento dentro de CONTRPAQí Comercial Premium al que se quiera adjuntar el complemento Carta Porte 2.0.
- Tener ya un CSD junto con su contraseña, estos deberán estar vigentes para una emisión de CFDI (pueden ser DEMO los CSDs).
- Haber agregado el CSD en la configuración del concepto que se utilizara para la emisión.

Recomendaciones:

- Licencia Comercial con un mínimo de 5 usuarios (recomendación para un correcto funcionamiento al momento de emitir documentos).
- Haber utilizado la función `fEmitirDocumento` que se encuentra en CONTRPAQí SDK.

Implementación.

La emisión de documentos con el complemento de carta porte 2.0 mediante SDK se puede ejecutar de dos maneras:

Escenario 1: Ya se cuenta con el documento generado en el sistema y este a su vez también cuenta con el complemento Cata Porte 2.0 generado y correctamente llenado desde el aplicativo CONTPAQi Comercial Premium.

Escenario 2: Existe un documento ya creado dentro del sistema, pero no con su complemento Carta Porte 2.0.

Para poder emitir cualquier documento utilizando el SDK Comercial es necesario utilizar la función `fEmitirDocumento` y será por medio de la cual se realizará la emisión de documentos en los dos escenarios antes planteados. Un ejemplo de cómo declarar dicha función en sus implementaciones es la siguiente:

```
[DllImport("MGWServicios.DLL")]
public static extern Int32 fEmitirDocumento(
    [MarshalAs(UnmanagedType.LPStr)] string aCodConcepto,
    [MarshalAs(UnmanagedType.LPStr)] string aSerie, double aFolio,
    [MarshalAs(UnmanagedType.LPStr)] string aPassword,
    [MarshalAs(UnmanagedType.LPStr)] string aArchivoAdicional);
```

Nota: Esta función se utilizará de la misma manera que en versiones anteriores, no ha sufrido ningún cambio en su declaración.

Escenario 1 “Documento con complemento ya creado en el sistema”.

Lo primero que hay que hacer será conocer los datos que corresponden al documento que vamos a emitir, ya sea que se consulten directamente en el sistema o que mediante una implementación personalizada consulten esa información mediante el SDK.

Nota: La contraseña del CSD y el Archivo Adicional son datos personalizados y por lo tanto no se pueden consultar por medio de SDK.

Una vez que ya tenemos la información de nuestro documento lo siguiente y último paso será emitir con la función `fEmitirDocumento` dicho documento, un ejemplo se comparte a continuación:

```
#region EMITIR DOCUMENTO CARTA PORTE

    public static void EmitirDocumentoCP()
    {
        //variables
        //Datos correspondientes al documento que se va intentar emitir y que deben ser datos existentes dentro de su
        empresa.
        string aCodigoConcepto= "Factura4.0";
        double aFolio = 21;
        string aSerie = "CP";
        string aContraseña = "12345678a";//Contraseña del CSD configurado dentro del concepto asignado en
        aCodigoConcepto
        string aArchivoAdicional = "";

        codigoDeError = MGWServicios.fEmitirDocumento(aCodigoConcepto, aSerie, aFolio, aContraseña,
        aArchivoAdicional);

        //en caso de que la función retorne un código diferente de 0 indicara que no se ejecutó con éxito
        if (codigoDeError != 0)
        {
            Console.WriteLine("Se genero el error " + codigoDeError);
            Console.WriteLine("Descripción: " + MGWServicios.rError(codigoDeError));
        }
        else
        {
            Console.WriteLine("Documento emitido");
        }
    }
}
#endregion
```

Nota: Si bien en el ejemplo los datos están asignados en variables locales, estos también pueden ser compartidos como parámetros de la función `EmitirDocumentoCP`.

Si la información que mandamos como parámetro a la función `fEmitirDocumento` es correcta y no tiene ningún inconveniente en cuanto a su llenado nuestro documento será emitido con éxito, en caso contrario se retornará un mensaje de error que corresponderá al problema por el cual no pudo ser emitido.

Escenario 2 “Documento creado en el sistema, pero sin complemento”.

En este segundo escenario el proceso de emisión también será mediante la función `fEmitirDocumento` solo que al no contar con el complemento Carta Porte incluido en el documento tendremos que adjuntar esta información a la función, para ello utilizaremos el parámetro `aArchivoAdicional`.

¿Qué recibirá el parámetro `aArchivoAdicional` ?

Lo que el documento requiere es adjuntar la información del complemento por lo que serán estos datos los que se enviarán como archivo de texto.

Dentro del aplicativo existe la posibilidad de adjuntar los datos de nuestro complemento a un documento ya sea mediante un archivo `.ini` o un archivo `.xml` y es exactamente esta funcionalidad la que utilizaremos para completar nuestro documento desde SDK.

Nota: Puedes conocer más acerca de este tema en [Agregar archivo XML con complemento](#)

El valor que se le da a la variable `aArchivoAdicional` deberá llevar la siguiente estructura:

Tipo de dato: String Valor: “Complemento:[ruta del archivo xml y/o ini]”

La palabra Complemento indicara el tipo de archivo que intentaremos adjuntar al documento y esta deberá ir seguida del carácter : para indicar en donde comenzara la ruta en la que el SDK posteriormente estaría buscando el archivo `ini` o `xml` de donde tomara la información del complemento.

A continuación, te mostramos un ejemplo de cómo implementar esta funcionalidad:

```
public static void EmitirDocumento( string aCodigoConcepto, double aFolio, string aSerie, string
aContraseña)
{
    //variables
    // Palabra Complemento posteriormente el carácter : y a continuación la ruta donde se encuentra la
    información del complemento seguida del nombre y extensión del archivo

    string aArchivoAdicional = @"Complemento:C:\Compac\Empresas\Esquemas\COMERCIAL\CartaPorte.ini";
    MGWServicios.fEmitirDocumento(aCodigoConcepto, aSerie, aFolio, aContraseña, aArchivoAdicional);

    //hay que recordar que en caso de que la función retorne un código diferente de 0 indicara que no se
    ejecutó con éxito por lo que se puede utilizar una variable para consultar el valor que retorne la
    función.
}
```

Nota: recordar que aun que el ejemplo muestra un archivo con extensión `.ini` también puede ser utilizado un archivo con extensión `.xml`

Podrás consultar la estructura del archivo `.ini` dentro del documento `CartaPorte.ini` que se encuentra en la ruta `C:\Compac\Empresas\Esquemas\COMERCIAL\`.

Importante: Para que el sistema detecte que lo que contiene el archivo .ini es información correspondiente al complemento de carta porte es de suma importancia que la primera línea de información de este archivo sea la etiqueta **[CartaPorte2.0]**, posteriormente las etiquetas que corresponden a trasportes, ubicaciones, figuras, etc., pueden ser actualizadas en cuanto a su acomodo siempre y cuando los valores cumplan con lo estipulado en la guía de llenado del complemento de Carta Porte. Ejemplo:

```
Factura CFDI 3. Traslado_FACTT_281.ini
1  [CartaPorte2.0]
2  TranspInternac=No
3  TotalDistRec=115.
4  [Transporte1]
5  Clave=04
6  PermSCT=TS01
7  NumPermisoSCT=
8  TipoDeServicio=TS02
9  TipoDeTráfico=TT02
10 NombreAseg=Axa
11 NumPolizaSeguro=123567
12 [Transporte1.Derecho1]
13 TipoDerechoDePaso=CDP001
14 KilometrajePagado=10.000
15 [Transporte1.Carrol.Contenedor1]
16 TipoContenedor=TC01
17 PesoContenedorVacio=1.000
18 PesoNetoMercancia=90.000
19 [Transporte1.Carrol]
20 TipoCarro=TC01
21 MatriculaCarro=1234567890
22 GuiaCarro=123
23 ToneladasNetasCarro=90.000
24 [Ubicacion1]
25 TipoUbicacion=Origen
26 IDUbicacion=OR000001
27 TipoEstacion=01
28 NumEstacion=Q0736
29 NombreEstacion=SANTO NINO
30 RFCRemitenteDestinatario=EKU9003173C9
31 NombreRemitenteDestinatario=COM Timbrado AAA 33
32 FechaHoraSalidaLlegada=2022-01-25T09:13:30
33 [Ubicacion1.Domicilio]
34 Pais=MEX
35CodigoPostal=44600
36Estado=JAL
37Municipio=039
38Localidad=03
39Calle=Pablo
40NumeroExterior=2200
41 [Ubicacion2]
42 TipoUbicacion=Origen
43 IDUbicacion=OR000002
44 TipoEstacion=01
45 NumEstacion=Q0736
46 NombreEstacion=SANTO NINO
47 RFCRemitenteDestinatario=EKU9003173C9
48 NombreRemitenteDestinatario=COM Timbrado AAA 33
49 FechaHoraSalidaLlegada=2022-01-25T09:13:30
50 [Ubicacion2.Domicilio]
51 Pais=MEX
52CodigoPostal=44600
53Estado=JAL
```

En estos archivos tanto archivo INI como XML, podrás incluir cualquier información del complemento sin ninguna restricción, sólo deberás verificar el XSD del Complemento de Carta Porte y la Guía de llenado correspondiente al "Medio de transporte".

Ejemplo de XML del complemento:

```
<cartaporte20:CartaPorte Version="2.0" TranspInternac="No" TotalDistRec="50.">
<cartaporte20:Ubicaciones>
  <cartaporte20:Ubicacion TipoUbicacion="Origen" IDUbicacion="OR000001"
RFCRemitenteDestinatario="EKU9003173C9" NombreRemitenteDestinatario="COM Timbrado AAA 33"
FechaHoraSalidaLlegada="2021-12-30T10:27:05">
    <cartaporte20:Domicilio Colonia="0236" Localidad="03" Municipio="039" Estado="JAL" Pais="MEX"
CodigoPostal="44600"/>
  </cartaporte20:Ubicacion>
  <cartaporte20:Ubicacion TipoUbicacion="Destino" IDUbicacion="DE000001"
RFCRemitenteDestinatario="XAXX010101000" NombreRemitenteDestinatario="Carta Porte Ingreso"
FechaHoraSalidaLlegada="2021-12-30T10:27:05" DistanciaRecorrida="50.000">
    <cartaporte20:Domicilio Calle="Pablo" NumeroExterior="2240" Colonia="0236" Localidad="03"
Municipio="039" Estado="JAL" Pais="MEX" CodigoPostal="44600"/>
  </cartaporte20:Ubicacion>
</cartaporte20:Ubicaciones>
<cartaporte20:Mercancias PesoBrutoTotal="3." UnidadPeso="KGM" NumTotalMercancias="000000003">
  <cartaporte20:Mercancia BienesTransp="50101700" Descripcion="Frutos secos" Cantidad="1.00"
ClaveUnidad="KGM" PesoEnKg="1." Unidad="Kilogramo" ValorMercancia="116.00" Moneda="MXN">
    <cartaporte20:CantidadTransporta Cantidad="1.00" IDOrigen="OR000001" IDDestino="DE000001"/>
  </cartaporte20:Mercancia>
  <cartaporte20:Mercancia BienesTransp="50131800" Descripcion="Queso Oaxaca" Cantidad="1.00"
ClaveUnidad="KGM" PesoEnKg="1" Unidad="Kilogramo" ValorMercancia="116.00" Moneda="MXN">
    <cartaporte20:CantidadTransporta Cantidad="1.00" IDOrigen="OR000001" IDDestino="DE000001"/>
  </cartaporte20:Mercancia>
  <cartaporte20:Mercancia BienesTransp="50161900" Descripcion="Azucar" Cantidad="1.00"
ClaveUnidad="XBX" PesoEnKg="1" Unidad="Caja" ValorMercancia="11.60" Moneda="MXN">
    <cartaporte20:CantidadTransporta Cantidad="1.00" IDOrigen="OR000001" IDDestino="DE000001"/>
  </cartaporte20:Mercancia>
  <cartaporte20:Autotransporte PermSCT="TPAF01" NumPermisoSCT="123456789">
    <cartaporte20:IdentificacionVehicular ConfigVehicular="C2" PlacaVM="PLAC892"
AnioModeloVM="2021"/>
  <cartaporte20:Seguros AseguraRespCivil="Axa" PolizaRespCivil="123456789"/>
</cartaporte20:Autotransporte>
</cartaporte20:Mercancias>
<cartaporte20:FiguraTransporte>
  <cartaporte20:TiposFigura TipoFigura="01" RFCFigura="VAAM130719H60" NumLicencia="a234567890"
NombreFigura="Chofer Uber">
  </cartaporte20:TiposFigura>
</cartaporte20:FiguraTransporte>
</cartaporte20:CartaPorte>
```

Nota: con ayuda de esta funcionalidad será posible manejar multi destinos dentro del complemento siempre y cuando se sigan las reglas tanto del XSD como de la Guía de llenado correspondiente, para este escenario no hay un límite de ubicaciones.