Mas sobre el envio de mensajes de correo electrónico desde Visual FoxPro lecturas 13962

Enviado por LuisMaria en Lunes, 21 Julio, 2008

Un artículo mas sobre el tema de enviar mensajes de correo electrónico desde Visual FoxPro.

Mas sobre el envio de mensajes de correo electrónico desde Visual FoxPro

Por Luis María Guayán, Tucumán, Argentina

Existen muchas herramientas y formas de enviar mensajes de correo electrónico desde Visual FoxPro, y seguramente por ello, existen muchos artículos y códigos escritos sobre este tema. La razón de escribir mas de los mismo, es comentarles sobre mi experiencia y elección personal, de la herramienta que utilizo actualmente, la cual me satisface y cubre todas mis necesidades. Estoy convencido que lo mismo le sucede a muchos desarrolladores y espero que al terminar de leer estas líneas, esto ayude a muchos mas.

The winner is ...

Como indico arriba, esto es *"mi experiencia y elección personal"*, obviamente no digo que sea *"la mejor"*, ni digo tampoco *"la peor"*, por lo tanto no entraré en comparación con otras herramientas, solo hablaré de **CDO**.

¿Qué es CDO?

CDO *(**Collaboration Data Objects**)* es un componente COM *(**Component Object Model**)* que simplifica la escritura de código para crear y manipular mensajes de Internet y es parte integrante (Cdosys.dll) de los sistemas operativos Windows 2000 y superiores, siendo esta la primer gran ventaja, ya que no necesitamos descargar, comprar, ni licenciar, ninguna otra herramienta extra. CDO no necesita que tengamos un servidor SMTP local, solo necesita acceso a la Web, a algún servidor SMTP que nos permita enviar los mensajes de correo electrónico.

Usando VFP

Otra ventaja adicional, es que se puede utilizar CDO con lenguajes que soporten COM y Automation, y Visual FoxPro lo soporta. A continuación un sencillo código para enviar un mensaje de correo electrónico con CDO desde VFP. Recuerde configurar correctamente el nombre del servidor SMTP, el puerto SMTP, el nombre de usuario y contraseña.

```
loCfg = CREATEOBJECT("CDO.Configuration")
WITH loCfg.Fields
    .Item("http://schemas.microsoft.com/cdo/configuration/smtpserver") = "smtp.mail.com"
    .Item("http://schemas.microsoft.com/cdo/configuration/smtpserverport") = 25
    .Item("http://schemas.microsoft.com/cdo/configuration/sendusing") = 2
    .Item("http://schemas.microsoft.com/cdo/configuration/sendusername") = "user@mail.com"
    .Item("http://schemas.microsoft.com/cdo/configuration/sendpassword") = "password"
    .Update
ENDWITH

loMsg = CREATEOBJECT ("CDO.Message")
WITH loMsg
    .Configuration = loCfg
    .From = "user@mail.com"
    .To = "user1@mail.com.ar"
    .Subject = "Prueba desde VFP"
    .TextBody = "Este es un mensaje de prueba con CDO desde Visual FoxPro."
    .Send()
ENDWITH
```

Adjuntando archivos en nuestros mensajes

Para enviar archivos adjuntos en un mensaje de correo electrónico, solo debemos llamar al método AddAttachment() del objeto Message, tantas veces como archivos deseamos adjuntar, con la ruta y el nombre del archivo como parámetros:

```
WITH loMsg
  .AddAttachment("C:\Imagenes\Foto1.jpg")
  .AddAttachment("C:\Imagenes\Foto2.jpg")
  .AddAttachment("C:\Imagenes\Foto3.jpg")
ENDWITH
```

Autenficación y cifrado SSL

Algunos servidores SMTP, necesitan autentificación y cifrado SSL para iniciar sesión. Para ello debemos configurar las propiedadadessmtpauthenticate y smtpusessl del objeto Configuration con el valor .T..

```
WITH loCfg.Fields
    .Item("http://schemas.microsoft.com/cdo/configuration/smtpauthenticate") = .T.
    .Item("http://schemas.microsoft.com/cdo/configuration/smtpusessl") = .T.
ENDWITH
```

Mensajes a múltiples destinatarios

Para enviar el mensaje de correo a mas de un destinatario, solo debemos separar las direcciones de correos electrónicos, con el caracter "," (coma) en la propiedad To. También podemos agregar una copia para otro destinatario, como así también una copia oculta, configurando las propiedades Cc y Bcc respectivamente.

```
WITH loMsg
  .To = "user1@mail.com, user2@mail.com"
  .Cc = "user3@mail.com"
  .Bcc = "user4@mail.com"
ENDWITH
```

Podemos hacer que aparezca el nombre completo del remitente y/o del destinatario, anteponiendo el nombre a la dirección de correo electrónico, y encerrando ésta última entre "<" y ">"

```
WITH loMsg
  .From = "Jose Fox <jose@fox.org>"
  .To = "Usuario Uno <user1@mail.com>, Usuario Dos <user2@mail.com>"
ENDWITH
```

Mensajes con formato HTML

Con CDO muy fácilmente se puede enviar mensajes de correo electrónico con formato HTML, solamente configurando la propiedadHTMLBody con código HTML válido, en lugar de configurar la propiedad TextBody que es válida para los mensajes con texto sin formato.

```
WITH loMsg
    .HTMLBody = "<p>Este es un texto HTML con <b>negritas</b> o con <i>cursivas</i>.</p>"
ENDWITH
```

Esta propiedad la podemos configurar con el contenido de un archivo HTML con la función FILETOSTR() como se muestra a continuación.

```
WITH loMsg
    .HTMLBody = FILETOSTR("C:\Archivo.htm")
ENDWITH
```

Mas adelante veremos otra manera de dar formato HTML a un mensaje de correo electrónico, a partir de un archivo HTML ubicado en el disco local o la Web.

Solicitando confirmación de lectura

Podemos configurar nuestros mensajes de correo electrónico para solicitar una confirmación de lectura y recibir ésta confirmación en la dirección predeterminada u otra (siempre y cuando el servidor y/o el programa cliente del destinatario tengan habilitada esta opción). El código para solicitar y recibir la confirmación de lectura en la misma dirección de correo del remitente es el siguiente:

```
WITH loMsg
   .From = "user@mail.com"
   .Fields("urn:schemas:mailheader:disposition-notification-to") = .From
   .Fields("urn:schemas:mailheader:return-receipt-to") = .From
   .Fields.Update
ENDWITH
```

Marcar los mensajes con importancia y prioridad

Otra opción que nos brinda CDO es configurar la importancia y prioridad de nuestros mensajes de correo electrónico, marcándolos como Normal (por omision), Alta o Baja. Por ejemplo el código para marcar un mensaje con importancia y prioridad Alta es:

```
WITH loMsg
   *-- Prioridad
   && -1=Low, 0=Normal, 1=High
   .Fields("urn:schemas:httpmail:priority") = 1
   .Fields("urn:schemas:mailheader:X-Priority") = 1
   *-- Importancia
   && 0=Low, 1=Normal, 2=High
   .Fields("urn:schemas:httpmail:importance") = 2
   .Fields.Update
ENDWITH
```

Mensajes con imágenes

Bueno, hasta aquí todo marcha bien sin mayores complicaciones; ahora solo nos faltaría saber como añadir imágenes a nuestros mensajes de correo con formato HTML.

Una opción simple, es que en el código HTML hagamos referencia a una URL de un sitio donde este almacenada la imagen, de esta forma se mostrará, siempre y cuando, se tenga acceso a internet. Esta opción tiene a favor que el mensaje enviado tiene un tamaño pequeño, ya que las imágenes estan almacenadas en algún lugar de la Web.

```
WITH loMsg
   .HTMLBody = "<p><img src='http://www.portalfox.com/logo.gif'></p>" + ;
      "<p>La imagen de arriba esta ubicada en el sitio de PortalFox.</p>"
ENDWITH
```

Otra opción es embebiendo las imágenes en el mensaje de correo. Para ello tenemos direrentes maneras de hacerlo.

Una forma es reemplazando el contenido: SRC='c:\Imagenes\Imagen10.gif' en la etiqueta IMG, por un ID de Contenido, como por ejemplo: SRC='cid:imagen10-gif'. La ruta y nombre del archivo de imagen, como el ID de Contenido, lo pasamos como parámetro al método AddRelatedBodyPart() que agregará un objeto imagen al cuerpo del mensaje.

```
WITH loMsg
   .HTMLBody = "<p><img src='cid:id_imagen10'></p>" + ;
      "<p>La imagen de arriba esta embebida en el mensaje.</p>"
   loBP = .AddRelatedBodyPart("c:\Imagenes\Imagen10.gif", "id_imagen10", 1)
   WITH loBP.Fields
      .Item("urn:schemas:mailheader:Content-ID") = "id_imagen10"
      .Update
   ENDWITH
ENDWITH
```

Esta forma de embeber las imágenes en los mensajes de correo electrónico, se utiliza cuando generamos programáticamente toda la cadena HTML y las imágenes son muy pocas. Cuando la cantidad de imágenes aumenta, este método resulta un poco engorroso.

La otra manera es directamente tomar un archivo HTML, ya sea desde la Web o del disco local, con el método CreateMHTMLBody(), y éste convierte automáticamente los enlaces de las imágenes en un ID de Contenido. A continuación vemos la forma de lograr esto con un ejemplo con la opción de si el archivo está almacenado en la Web o en el disco local:

```
WITH THIS.oMsg
   IF llFileLocal
      *-- Si el archivo está en el disco local
      .CreateMHTMLBody("file://c:\documentos\archivo.htm", 0)
   ELSE
      *-- Si el archivo está en la Web
      .CreateMHTMLBody("http://www.portalfox.com/articulos/archivos/correo.htm", 0)
   ENDIF
ENDWITH
```

Este método es válido, aun si el archivo HTML no contiene imágenes, con lo que lo podemos usar para reemplazar:

```
WITH loMsg
   .HTMLBody = FILETOSTR("C:\Archivo.htm")
ENDWITH
```

con:

```
WITH loMsg
   .CreateMHTMLBody("file://C:\Archivo.htm")
ENDWITH
```

como indicamos mas arriba.

El ejemplo final

Para terminar veremos un ejemplo completo y funcional de como enviar un mensaje de correo electrónico desde Visual FoxPro, con varias de las opciones que mostramos. Vamos a usar una cuenta de Gmail (el servidor SMTP de Gmail requiere autenticación y utiliza cifrado SSL), y enviaremos un mensaje de correo a Usuario Uno, con copia a Usuario Dos, con confirmación de lectura y con formato HTML, tomado de la Web y con imágenes embebidas, y un archivo adjunto que seleccionaremos de nuestro disco.

```
LOCAL loCfg, loMsg, lcFile, loErr
TRY
   loCfg = CREATEOBJECT("CDO.Configuration")
   WITH loCfg.Fields
      .Item("http://schemas.microsoft.com/cdo/configuration/smtpserver") = "smtp.gmail.com"
      .Item("http://schemas.microsoft.com/cdo/configuration/smtpserverport") = 465 && ó 587
      .Item("http://schemas.microsoft.com/cdo/configuration/sendusing") = 2
      .Item("http://schemas.microsoft.com/cdo/configuration/smtpauthenticate") = .T.
      .Item("http://schemas.microsoft.com/cdo/configuration/smtpusessl") = .T.
      .Item("http://schemas.microsoft.com/cdo/configuration/sendusername") = "josefox@gmail.com"
      .Item("http://schemas.microsoft.com/cdo/configuration/sendpassword") = "password"
      .Update
   ENDWITH
   loMsg = CREATEOBJECT ("CDO.Message")
   WITH loMsg
      .Configuration = loCfg
      *-- Remitenete y destinatarios
      .From = "Jose Fox <josefox@gmail.com>"
      .To = "Usuario Uno <user1@gmail.com>"
      .Cc = "Usuario Dos <user2@gmail.com>"
      *- Notificación de lectura
      .Fields("urn:schemas:mailheader:disposition-notification-to") = .From
```

```
        .Fields("urn:schemas:mailheader:return-receipt-to") = .From
        .Fields.Update
        *-- Tema
        .Subject = "Ejemplo del " + TTOC(DATETIME())
        *-- Formato HTML desde la Web
        .CreateMHTMLBody("http://www.portalfox.com/articulos/archivos/correo.htm", 0)
        *-- Archivo adjunto
        lcFile = GETFILE()
        IF NOT EMPTY(lcFile)
            .AddAttachment(lcFile)
        ENDIF
        *-- Envio el mensaje
        .Send()
    ENDWITH
CATCH TO loErr
    MESSAGEBOX("No se pudo enviar el mensaje" + CHR(13) + ;
        "Error: " + TRANSFORM(loErr.ErrorNo) + CHR(13) + ;
        "Mensaje: " + loErr.Message , 16, "Error")
FINALLY
    loMsg = NULL
    loCfg = NULL
ENDTRY
```

Otros enlaces relacionados al uso de CDO con VFP

- **Email and VFP: Part 1c (CDOSYS)**
- **Envío de correo electrónico por el servidor SMTP de Gmail**
- **Incruste imágenes a sus correos electrónicos con CDOSYS**

Notas finales

He tomado la elección de usar CDO para el envio de los mensajes de correo electrónico en mis aplicaciones hace aproximadamente 20 meses. Desde entonces he creado una una clase que encapsula al objeto CDO, y a medida que aparecieron nuevas necesidades, agregaba nuevos métodos y/o propiedades a la definición de la clase, con lo que logre que los cambios no tengan efectos no deseados en mis aplicaciones anteriores. Es por ello que aconsejo a ustedes a que también hagan su propia clase para el envio de mensajes de correo electrónico, con todas sus necesidades, tomandolas de cada ejemplo de este artículo.

Hasta la próxima,

Luis María