

COBEMAIL

Versión 3.36

Manual del usuario

INDICE

Introducción.....	3
Instalación.....	4
Parámetros que acepta	7
Instrucciones para generar en el archivo de Email	9
Registros de COBEMAIL.WSS	10
EMAIL-FROM.....	10
EMAIL-TO.....	11
EMAIL-CC.....	12
EMAIL-BCC	13
EMAIL-SUBJECT	14
EMAIL-MESSAGE.....	15
EMAIL-ATTACHMENT.....	16
EMAIL-TYPE-TEXT.....	17
EMAIL-TYPE-HTML.....	18
EMAIL-CONFIRM	19
EMAIL-TEMPLATE	20
EMAIL-VAR.....	21
MEMOS.....	22
MEMOS EN VARIABLES	23
EMAIL-BLOCK.....	24
CFGEMAIL.....	25
CFGEMAIL-SERVER.....	25
CFGEMAIL-AUTH	26
CFGEMAIL-SSL.....	27
Registro de Licencia para COBEMAIL	28

Introducción

COBEMAIL, es un programa que nos auxilia en el envío de correos electrónicos (Email) desde nuestras aplicaciones COBOL.

Únicamente hay que hacer una pequeña configuración para indicarle a COBEMAIL por que vía serán enviados los correos, una vez echo esto solo hay que crea un pequeño archivo con nuestro programa COBOL para enviar un email a otra cuenta de correo.

COBEMAIL puede enviar tanto correos en formato Texto como correos en formato HTML gracias al manejo fácil y sencillo de Templates.

Un Template es un archivo en formato HTML que puede ser diseñado con cualquier editor de Paginas Web como por ejemplo Front Page, aunque podemos utilizar el editor de html de nuestra preferencia.

También gracias a el formato de HTML podemos incluir imágenes incrustadas en el email, y estas imágenes son enviadas de manera automática al indicar el archivo HTML de Template, ya que COBEMAIL analiza el archivo de Template y todos los vínculos hacia imágenes son incluidas también en el correo, sin la necesidad de tener que especificar dichos archivos.

El formato de nuestro archivo de email es muy similar al código HTML, el cual también es muy común en otros programas de COBTOOLS como por ejemplo COBVIEW.

Para enviar un email entonces solo debemos de saber como generar dicho archivo texto con las instrucciones para el envío del email, si podemos crear un archivo texto desde COBOL podremos enviar correos sin meternos en complicaciones de tener que conocer algún API especial.

Instalación

La instalación es de lo más sencillo, ya que solo hay que ejecutar el programa de instalación de COBEMAIL, y seguir las instrucciones del asistente.

Una vez instalado el programa en la carpeta donde nosotros hayamos especificado, obtendremos los siguientes archivos:

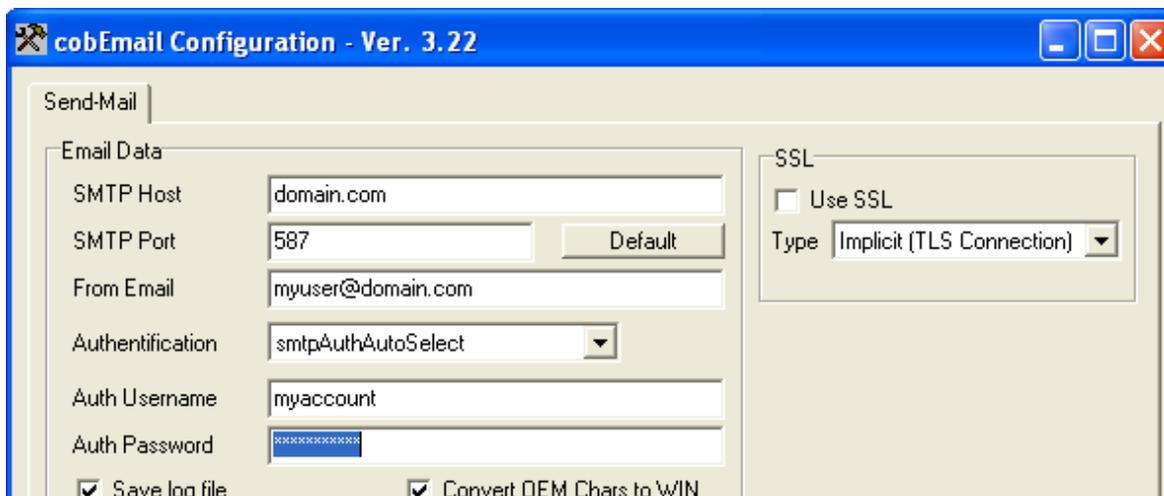
cfgCobEmail.EXE	Programa de configuración de COBEMAIL
CobSendM.EXE	Programa que se encarga de el Envío de Emails
CobEmail.DAT	Archivo que almacena la configuración del programa

Para los ejemplos plasmados en este manual, así como los ejemplos que descargue de Internet, asumiremos que el directorio de Instalación es C:\COBEMAIL>

Antes de utilizar COBEMAIL deberemos de configurarlo, de lo contrario no podremos enviar correos de ninguna manera.

Deberemos de ejecutar primero el programa **cfgCobEmail**, el cual nos sirve para realizar esta tarea.

Al ejecutarlo nos aparecerá una ventana como la siguiente:



Los datos que necesitamos configurar son los siguientes:

SMTP Host – Esta puede ser un nombre de dominio o la dirección IP del servidor de SMTP que se utilizara para el envío de nuestro email. Regularmente se registra el de nuestro proveedor de Internet, a menos de que la empresa donde vayamos a instalar este software cuente con el suyo propio.

SMTP Port – Es el puerto TCP donde se encuentra nuestro servidor SMTP escuchando las peticiones, por default podemos usar el puerto **smtp**, a menos de que nuestro servidor escuche en alguno diferente.

From Email – Aquí podemos especificar la cuenta de email desde la cual serán enviados los correos, sirve además para que las respuestas a nuestros emails sean enviados hacia esta cuenta. La cuenta FROM también puede ser especificada desde el archivo de envío del email, si esto sucede entonces no será usada la cuenta FROM de la configuración del programa si no la especificada en el archivo de email.

Authentication – Sirve para indicar si el servidor SMTP por el cual será enviado el correo necesita Autenticación, de ser así, deberemos de escoger el tipo de autenticación que utiliza, las opciones son:

- smtpAuthNone
- smtpAuthPlain
- smtpAuthLogin
- smtpAuthCramMD5
- smtpAuthCramSha1
- smtpAuthNtlm
- smtpAuthAutoSelect

La opción que debemos de seleccionar dependerá de nuestro servidor SMTP al que nos conectaremos.

Si desconocemos el método de autenticación, podemos utilizar smtpAuthAuthSelect y dejar que el COBEMAIL intente con distintas opciones.

Auth Username – Es el nombre de usuario que será usado para hacer la autenticación.

Auth Password – El password del usuario con el que será autenticado en el servidor SMTP. Desde la versión 3.10 este password ya se guarda encriptado en el archivo de configuración.

Save Log File – Este parámetro es nuevo, incluido en desde la version1.10, si lo marcamos, le indicamos que nos lleve un registro de los emails que se han enviado a travez de COBEMAIL, con fines de llevar un monitoreo.

Convert OEM Chars to WIN – Esta opción activada permite, que cuando un archivo texto trae caracteres de acentuación o como la Ñ, y el archivo tiene formato de MSDOS, estos caracteres serán convertidos a caracteres Unicode de Windows, para que sean enviados correctamente en el email.

Language – Esta es una lista, en la que podemos seleccionar el idioma en que deseamos que se muestren los mensajes o advertencias al momento de enviar un email. Así como los títulos de mensajes que se almacenan en el Log.

SSL – Con este checkbox, podemos indicar que el servidor SMTP hace obligatorio el uso de certificados de seguridad SSL. Esto depende de cómo funcione el servidor SMTP. Es muy útil si vamos a utilizar cuentas de GMAIL para enviar correos a través de smtp.

SSL Type – Con esta opción, podemos indicar el tipo de seguridad SSL que utiliza el servidor SMTP.

- None
- Implicit (TLS Connection)
- Explicit (STARTTLS)

El más común es: Implicit (TLS Connection), como es el caso de gmail.com, aunque como todos los parámetros de conexión, este depende de cual tipo utiliza nuestro servidor SMTP.

Para guardar nuestros cambios, solo damos clic en el botón OK, y podremos empezar a usar nuestro programa para enviar correos.

Parámetros que acepta

El programa que se encarga del envío de EMAILS es: COBSENDM.EXE
El formato para ejecutarlo es el siguiente:

COBSENDM archivo-de-email opciones

/NODEL

Este parámetro sirve para indicar que el archivo de email siga existiendo después de que sea enviado el email.

Por default el archivo de email es eliminado después de que el correo es enviado.

Si deseamos que este archivo siga existiendo, solo incluimos el parámetro /NODEL en la lista de opciones.

Únicamente el archivo del envío del email es eliminado en forma automática, cualquier otro archivo que se adjunte NO será eliminado, tampoco se eliminan los templates usados ni tampoco las imágenes que necesite un template de HTML.

Ejemplos de ejecución:

1. COBSENDM ENVIO.TXT
2. COBSENDM.EXE EMAIL.TXT /NODEL
3. COBSENDM.EXE C:\TEMP\EMAIL.TXT

En el ejemplo 1, el email es enviado con las instrucciones indicadas en el archivo ENVIO.TXT, este archivo será eliminado después de enviado el email.

En el ejemplo 2, el email se envía con las instrucciones en el archivo EMAIL.TXT, y este archivo no es eliminado después de enviar el correo.

Por ultimo, el ejemplo 3 nos sirve para ejemplificar que también podemos enviar un email con un archivo que este en algún directorio en específico, podemos colocar nuestro archivo de email en cualquier directorio de nuestro disco duro.

/LOG archivo.log

Este parámetro sirve para indicar que archivo usar como registro o log de sucesos.

Util, si queremos tener un log por terminal o usuario.

Si no se especifica, usara el archivo de log de default: cobemail.log, y se ubica donde este el cobsendm.exe

/CFG archivo.dat

Con este parámetro podemos indicar un archivo de configuración distinto al de default. El archivo de default es COBEMAIL.DAT

Esto sirve para cuando necesitamos utilizar una configuración del servidor SMTP distinta a la de default, o personalizada por departamento o usuario.

También se puede indicar el mismo parámetro /CFG al programa CFGCOBEMAIL.EXE

/NOERROR

/NOERRORS

/NOERRORES

Sirve para indicar que no muestre por pantalla la ventana de errores, para que en caso de algún proceso por lotes, no se detenga a esperar la intervención del usuario.

Todo el error quedaría registrado en el archivo de log.

/STAT

Muestra una ventana con el estatus del resultado del envío.

Si fue exitoso o si existió algún error al enviar.

A diferencia de solo mostrar una pausa en caso de errores /STAT muestra siempre el resultado del envío.

/LOGEMAIL cuenta-de-correo

Este parámetro nos sirve para indicar una cuenta de email, a la que se le enviara un correo con el resultado de todo el log del envío actual.

El mismo LOG que se registra en el archivo de log, se envía a esta cuenta de email.

Instrucciones para generar en el archivo de Email

Nuestro archivo de email, como se dijo anteriormente, no es más que un archivo texto generado desde nuestro programa COBOL.

Para generar un archivo de este tipo, en COBOL 85 primero declararíamos una instrucción SELECT de la siguiente manera:

```
SELECT EMAILDATA
      ASSIGN TO PRINT "EMAIL.TXT"
      FILE STATUS IS STAT-EMAIL.
```

Esta declaración es similar a declarar un archive de impresión, con la diferencia de que en lugar de enviarlo a una impresora lo enviamos a un archivo, el cual es generado como texto.

Y para nuestro FD, podríamos declarar algo como lo siguiente:

```
FD  EMAILDATA
    LABEL RECORDS ARE OMITTED.
01  R-EMAIL          PIC X(500) .
```

Ahora bien, junto con COBEMAIL va una pieza importante de código, este es un archivo llamado: COBEMAIL.WSS

Este archivo contiene la declaración de todas las variables para generar las diferentes Marcas con las que podemos especificar todos los datos necesarios para enviar un email.

Este archivo también lo podemos obtener junto con los ejemplos de COBEMAIL, dichos ejemplos también aplican todo lo que se explica en este manual.

Registros de COBEMAIL.WSS

EMAIL-FROM

Este registro sirve para especificar el nombre de la persona y la cuenta desde donde se esta enviando el Email.

Este marca también sustituye la cuenta FROM (desde) que se indica en la configuración global del programa.

Esta marca es opcional, ya que si no se especifica nada en cada correo que quisiéramos mandar utilizaría la del programa.

Para utilizar esta marca, necesitamos enviar dos valores en las siguientes variables:

FROM-NAME – Nos sirve para especificar el nombre de la persona a quien pertenece la cuenta desde donde se envía el email.

FROM-EMAIL – es la cuenta de correo desde donde se envía el email.

Ejemplo:

```
MOVE "ALBERTO ORTEGA" TO FROM-NAME.  
MOVE "aortega@cobtools.com" TO FROM-EMAIL.  
WRITE R-EMAIL FROM EMAIL-FROM.
```

Con estas instrucciones, en nuestro archive de email, generaríamos la siguiente marca:

```
<FROM NAME =' ALBERTO ORTEGA' EMAIL=' aortega@cobtools.com' >
```

EMAIL-TO

Con este registro podemos especificar a que persona deseamos enviar el correo electrónico.

Necesitamos especificar el nombre de la persona, así como la cuenta de Email hacia donde debe llegar el correo.

Los datos que debemos de mandar así como las variables que debemos de utilizar son las siguientes:

TO-NAME – Aquí especificamos el nombre de la persona a quien va dirigido el email.

TO-EMAIL – Es la cuenta de correo electrónica a donde debe de llegar el correo.

Ejemplo:

```
MOVE "JUAN PEREZ" TO TO-NAME.  
MOVE "jperez@dominio.com" TO TO-EMAIL.  
WRITE R-EMAIL FROM EMAIL-TO.
```

Con estas instrucciones, en nuestro archive de email, generaríamos la siguiente marca:

```
<TO NAME =' JUAN PEREZ' EMAIL=' jperez@dominio.com' >
```

Podemos especificar varias veces este mismo registro si deseáramos enviar el mismo email a diferentes cuentas de correo como destinatario.

EMAIL-CC

Este registro se utiliza para enviar copias del mismo email.
Las cuentas de email que se registren con este registro serán reportadas en el dato “Con copia para”.

Al momento de enviar un email, no hay diferencia entre usar un EMAIL-TO y un EMAIL-CC, para enviar copias del mismo email a diferentes personas, lo único es que a todos les llegara el email y las cuentas adicionales serán reportadas en el dato “Con copia para”.

Los datos que se necesitan son los siguientes:

CC-NAME – Es el nombre de la persona a quien va dirigida la copia del email.

CC-EMAIL – Es la cuenta de correo para quien va dirigida la copia del mismo email.

Ejemplo:

```
MOVE "MIGUEL HERNANDEZ" TO CC-NAME.  
MOVE "mhernandez@dominio.com" TO CC-EMAIL.  
WRITE R-EMAIL FROM EMAIL-CC.
```

Con estas instrucciones, en nuestro archive de email, generaríamos la siguiente marca:

```
<CC NAME ='MIGUEL HERNANDEZ' EMAIL='mhernandez@dominio.com' >
```

EMAIL-BCC

Este registro se utiliza para enviar copias del mismo email pero con Copia Oculta. Las cuentas de email que se registren con este dato recibirán una copia del correo, pero las cuentas que van en el TO o en CC no podrán ver estas cuentas.

Los datos que se necesitan son los siguientes:

BCC-NAME – Es el nombre de la persona a quien va dirigida la copia del email.

BCC-EMAIL – Es la cuenta de correo para quien va dirigida la copia del mismo email.

Ejemplo:

```
MOVE "MIGUEL HERNAN" TO BCC-NAME.  
MOVE "mhernan@dominio.com" TO BCC-EMAIL.  
WRITE R-EMAIL FROM EMAIL-BCC.
```

Con estas instrucciones, en nuestro archive de email, generaríamos la siguiente marca:

```
<BCC NAME ='MIGUEL HERNAN' EMAIL='mhernan@dominio.com'>
```

EMAIL-SUBJECT

Este registro es utilizado para enviar el Asunto del email, es un solo registro el que se envía, y si envía varios de estos registros solo el último será el que se quede como Asunto.

Para usar este registro solo hay que mover el título del Asunto a la variable: SUBJECT

Ejemplo:

```
MOVE "REPORTE DE VENTAS DE ENERO/2006" TO SUBJECT.  
WRITE R-EMAIL FROM EMAIL-SUBJECT.
```

Con estas instrucciones, en nuestro archive de email, generaríamos la siguiente marca:

```
<SUBJECT CONTENT=' REPORTE DE VENTAS DE ENERO/2006' >
```

EMAIL-MESSAGE

Este registro se utiliza para escribir el cuerpo del mensaje a enviar en el email.

Cada línea que se agregue con un registro de estos, será una línea nueva en el cuerpo del mensaje.

Este registro se utiliza solamente cuando el tipo de Email es Texto.

Para usar este registro solo mueva el dato con la línea a insertar en la variable MESSAGE.

Ejemplo:

```
MOVE "LINEA EN EL CUERPO DE MENSAJE" TO MESSAGE.  
WRITE R-EMAIL FROM EMAIL-MESSAGE.  
MOVE "OTRA LINEA DE MENSAJE" TO MESSAGE.  
WRITE R-EMAIL FROM EMAIL-MESSAGE.
```

Con estas instrucciones, en nuestro archive de email, generaríamos la siguiente marca:

```
<MESSAGE CONTENT=' LINEA EN EL CUERPO DE MENSAJE' >  
<MESSAGE CONTENT=' OTRA LINEA DE MENSAJE' >
```

EMAIL-ATTACHMENT

Este registro es uno de los beneficios mas importantes de COBEMAIL, con este registro podemos indicar un archivo adicional que deseamos adjuntar con nuestro Email.

Podemos especificar diferentes archivos en un mismo email, realmente la limitante en cuanto a cantidad de archivos y tamaño de estos dependerá de la capacidad de memoria del equipo que se encargara de enviar el email, así como el ancho de banda de la conexión de Internet.

Lo único que hay que hacer es especificar el archivo a adjuntar a la variable FILE-ATTACH.

Podemos especificar una ruta completa, o una relativa, siempre dependiendo de en que directorio se encuentra ubicado al momento de mandar a llamar el programa COBSENDM.EXE

Ejemplo:

```
MOVE "C:\TEMP\DATOS.ZIP" TO FILE-ATTACH.  
WRITE R-EMAIL FROM EMAIL-ATTACHMENT.  
  
MOVE "F:\DATOS\ENVIO.DAT" TO FILE-ATTACH.  
WRITE R-EMAIL FROM EMAIL-ATTACHMENT.
```

Con estas instrucciones, en nuestro archive de email, generaríamos la siguiente marca:

```
<ATTACHMENT FILE='C:\TEMP\DATOS.ZIP' >  
<ATTACHMENT FILE='F:\DATOS\ENVIO.DAT' >
```

EMAIL-TYPE-TEXT

Este registro se usa para especificar que el email será enviado en formato TEXT.

No es necesario especificar ningún dato en alguna variable en especial.

Ejemplo.

```
WRITE R-EMAIL FROM EMAIL-TYPE-TEXT.
```

Con estas instrucciones, en nuestro archive de email, generaríamos la siguiente marca:

```
<EMAIL TYPE='TEXT' >
```

Cuando se envía un email en formato TEXTO el contenido del mensaje es únicamente en formato Texto.

El texto se puede indicar con el registro EMAIL-MESSAGE, y no se puede usar el registro EMAIL-TEMPLATE para indicar un archivo de Template HTML.

EMAIL-TYPE-HTML

Este registro es para especificar que el email a enviar será en formato HTML.

No se necesita indicar algún dato especial en alguna variable.

Ejemplo:

```
WRITE R-EMAIL FROM EMAIL-TYPE-HTML.
```

Con estas instrucciones, en nuestro archive de email, generaríamos la siguiente marca:

```
<EMAIL TYPE=' HTML' >
```

Cuando se envía un email en formato HTML el cuerpo del mensaje que se registra en el registro EMAIL-MESSAGE puede ser tanto en formato Texto como en Formato HTML.

Aunque lo mas recomendable es utilizar un Template HTML generado con algún editor de Paginas Web.

EMAIL-CONFIRM

Con este registro podemos especificar que deseamos recibir una confirmacion de email recibido, cuando el cliente de correo lea el email en su servidor de correos.

La confirmacion del correo recibido nos llegara a la cuenta de correo que usemos para enviar el email, ya sea la de la configuracion global, o bien la especificada en el registro EMAIL-FROM.

Si el cliente de correo no soporta esta caracteristica de confirmar el email recibido, nunca recibiremos nuestra confirmacion, asi que estaremos sujetos a la configuracion del cliente.

Ejemplo:

```
WRITE R-EMAIL FROM EMAIL-CONFIRM.
```

Este registro ya tiene todo configurado para que al momento en que lo escribamos se configure la peticion de confirmacion, asi que no es necesario especificar ningun parametro adicional.

La escritura de este registro puede ser encualquier parte del email.

EMAIL-TEMPLATE

Sirve para especificar el archivo de Template HTML a utilizar.

Este archivo de Template va en formato HTML, todas las imágenes que se le incluyan en la página Web serán incluidas en el email para enviarse de manera automática en el Email.

Las imágenes que están en la página Web, serán incluidas en los siguientes directorios, y en el siguiente orden:

- 1.- En la ruta de la imagen tal y como viene incluida en el archivo HTML
- 2.- En el directorio donde esta el Template
- 3.- En el directorio actual donde se esta llamando el programa COBSENDM.EXE

Una vez que la encuentra, se inserta la imagen en el cuerpo del email en forma automática, de tal manera que cuando llega el email al destinatario todas las imágenes serán vistas de manera transparente.

Ejemplo:

```
MOVE ".\HTML\EMAIL1.HTM" TO FILE-TEMPLATE TO FILE-TEMPLATE.  
WRITE R-EMAIL FROM EMAIL-TEMPLATE.
```

Con estas instrucciones, en nuestro archive de email, generaríamos la siguiente marca:

```
<TEMPLATE FILE='.\HTML\EMAIL1.HTM' >
```

En este ejemplo, se indica que el archive de template se encuentra en el subdirectorío HTML, con el nombre EMAIL1.HTM

EMAIL-VAR

Sirve para indicarle un valor a una variable que se especifico en el Template.

En muchas ocasiones deseamos escribir un Template HTML, pero con una parte variable, en donde nuestro programa COBOL pudiera especificar que texto deberia de ir en esa parte.

Por ejemplo, si escribieramos el template de una Carta para una lista de CLIENTES, podriamos poner como variable el Nombre del cliente, para que en lugar de un texto fijo escribiera el Nombre de cada cliente y asi personalizar la carta.

Ejemplo:

```
MOVE "NOMBRE" TO EMAIL-VAR-NAME .
MOVE WSS-NOMBRE TO EMAIL-VAR-VALUE .
WRITE R-EMAIL FROM EMAIL-VAR .
```

La variable EMAIL-VAR-VALUE tiene un picture X(250), si deseamos mandar un valor numerico es recomendable formatearlo antes con una variable con mascara.

En nuestro template indicaremos la variable a cambiar encerrada entre <? variable ?>
Para el ejemplo, deberiamos indicarla como: <? nombre ?>

Podemos definir tantas variables como necesitemos en nuestro Template.

Si por alguna razon indicamos una variable dentro del template, pero no le asignamos valor desde nuestro programa, no mostrara ningun valor, es decir, el template tendra un valor vacio o espacio.

El programa TEST3.CBL de los ejemplos de uso de COBEMAIL tiene una muestra de cómo pasar valores a las variables del template.

MEMOS

Estos registros son para manejar información en múltiples líneas.

Este tipo de MEMOS son únicamente para incluir el texto del Contenido, y no pueden ser utilizados en contenido donde se usan Templates HTML.

Son varios registros que se deben de utilizar para este fin.

Ejemplo:

```
WRITE LINEA FROM EMAIL-START-MEMO .

MOVE "LINE 1" TO EMAIL-DAT-MEMO .
WRITE R-EMAIL FROM EMAIL-MEMO .

MOVE "LINE 2" TO EMAIL-DAT-MEMO .
WRITE R-EMAIL FROM EMAIL-MEMO .

WRITE LINEA FROM EMAIL-END-MEMO .
```

El registro EMAIL-START-MEMO sirve para indicar el inicio del contenido MEMO.

El registro EMAIL-END-MEMO se utiliza al finalizar el MEMO.

La variable EMAIL-DAT-MEMO es donde iremos colocando el contenido a imprimir en varias líneas. En algunos compiladores puede tener contenido con los caracteres de Enter incluidos, y podemos moverlo tal cual.

La variable EMAIL-DATA tiene un tamaño de 1024 caracteres.

Si quisieramos manejar manualmente caracteres de Enter en compiladores donde no se guarda este código, podemos utilizar la secuencia `\r\n` para insertar saltos de línea.

MEMOS EN VARIABLES

Estos registros son para manejar información en múltiples líneas. Y almacenar el contenido en una Variable.

Estos MEMOS los podemos utilizar en Templates HTML, y se manejan iguales que las variables de una sola línea.

Son varios registros que se deben de utilizar para este fin.

Ejemplo:

```
MOVE "NOM-VAR" TO EMAIL-VAR-MEMO .
WRITE LINEA FROM EMAIL-START-VAR-MEMO .

MOVE "LINE 1" TO EMAIL-DAT-MEMO .
WRITE R-EMAIL FROM EMAIL-MEMO .

MOVE "LINE 2" TO EMAIL-DAT-MEMO .
WRITE R-EMAIL FROM EMAIL-MEMO .

WRITE LINEA FROM EMAIL-END-VAR-MEMO .
```

El registro EMAIL-START-VAR-MEMO sirve para indicar el inicio del contenido MEMO. Antes de imprimirlo, debemos de pasar el nombre de la variable en el Template, en este ejemplo fue NOM-VAR.

El registro EMAIL-END-VAR-MEMO se utiliza al finalizar el MEMO.

La variable EMAIL-DAT-MEMO es donde iremos colocando el contenido a imprimir en varias líneas. En algunos compiladores puede tener contenido con los caracteres de Enter incluidos, y podemos moverlo tal cual.

La variable EMAIL-DATA tiene un tamaño de 1024 caracteres.

Si quisiéramos manejar manualmente caracteres de Enter en compiladores donde no se guarda este código, podemos utilizar la secuencia `\r\n` para insertar saltos de línea.

En HTML un salto de línea se debe indicar como la marca `
`.

EMAIL-BLOCK

Este registro sirve para crear bloques repetidos de código HTML en nuestro template. Por ejemplo para cuando deseamos enviar un estado de cuenta de un cliente, con una lista de todas las facturas pendientes de pago.

La lista de facturas es una lista de renglones, en donde vamos a repetir N veces los datos de las facturas. En algún cliente tendrá X número de facturas y otro cliente tendrá un número de facturas diferentes.

Ejemplo:

```
MOVE "FACTURA" TO EMAIL-BLOCK-NAME.  
MOVE "FACT_PASTE" TO EMAIL-BLOCK-PASTE.  
WRITE R-EMAIL FROM EMAIL-BLOCK.
```

Con esto indicamos que deseamos escribir un BLOCK llamado FACTURA que se encuentra definido en el template, y que lo deseamos pegar en el lugar indicado por la marca FACT-PASTE.

Ahora bien, dentro del template HTML que estamos usando, los bloques los definimos de la siguiente manera:

```
<!--? paste fact_paste ?-->  
<!--? block factura ?-->  
<tr>  
    <td><? fecha ?></td>  
    <td><? concepto ?></td>  
    <td><? importe ?></td>  
</tr>  
<!--? endblock factura ?-->
```

Es importante definir los nombres que usaremos en el registro EMAIL-BLOCK, si usamos otros nombres nuestro template no funcionara como lo deseamos.

La única manera de escribir estas marcas, es dentro de los símbolos <!--? ?--> ya que estos son tomados como comentarios por el editor de HTML (FrontPage) o el que sea de nuestra preferencia.

También es muy importante que si dentro del block usamos variables, indiquemos los valores para estas variables en nuestro programa, antes de escribir el registro EMAIL-BLOCK.

En el programa TEST4.CBL de los ejemplos de uso de COBEMAIL tenemos una muestra clara de cómo podemos utilizar este método.

CFGEMAIL

A partir de la versión 3.36 de Cobemail, es posible utilizar diferentes cuentas de correo, incluso de distinto servidor SMTP o dominio.

Esto permite, que desde nuestro programa COBOL podamos alimentar los datos de que servidor o cuenta utilizar para enviar determinados tipos de correo electrónico.

Los registros a utilizar son los siguientes, y todos están en el archivo COBEMAIL.WSS a partir de la versión 3.36.

Si no tienes estos registros o variables, descarga los ejemplos de la versión 3.36 en adelante, para poder hacer uso de esos registros y variables.

CFGEMAIL-SERVER.

Permite especificar un servidor de SMTP distinto al que esta en la configuración global.

Usa dos variables:

CFGEMAIL-HOST – se refiere a el nombre del servidor o IP a usar.

CFGEMAIL-PORT – es el numero de puerto TCP a usar, para conectarse a ese servidor de correos.

Ejemplo:

```
MOVE "midominio.com" TO CFGEMAIL-HOST.  
MOVE "587" TO CFGEMAIL-PORT.  
WRITE R-EMAIL FROM CFGEMAIL-SERVER.
```

Nota:

Si se van a usar distintas cuentas, pero todas pertenecen al mismo servidor de correos, o mismo dominio, entonces no es necesario indicar este registro en cada envío de correo.

CFGEMAIL-AUTH

Con este registro, es posible cambiar de Usuario y Contraseña, para enviar el correo con otra cuenta.

Variables del registro:

CFGEMAIL-AUTH-TYPE: Se refiere al tipo de autenticación que necesita nuestro servidor para validar la autenticidad de nuestra cuenta a usar.

Los valores permitidos son:

NONE

PLAIN

LOGIN

CRAMMD5

CRAMSHA1

NTLM

AUTOSELECT

Si sabemos que el servidor no utiliza ningún tipo de autenticación, entonces registramos NONE, y si llegamos a desconocer el tipo exacto podemos poner AUTOSELECT y cobemail utilizara la mas adecuada para el servidor.

CFGEMAIL-AUTH-USER – Aquí indicamos el nombre del usuario o cuenta a utilizar.

CFGEMAIL-AUTH-PASS – Aquí mandamos el password de la cuenta a utilizar.

Ejemplo:

```
MOVE "AUTOSELECT" TO CFGEMAIL-AUTH-TYPE.
MOVE "micuenta@midominio.com" TO CFGEMAIL-AUTH-USER.
MOVE "MiPASSWORD" TO CFGEMAIL-AUTH-PASS.
WRITE R-EMAIL FROM CFGEMAIL-AUTH.
```

CFGEMAIL-SSL

Este registro sirve para indicar si el nuevo servidor (si es que se ha cambiado), utiliza SSL o no.

Variables del registro:

CFGEMAIL-USE-SSL: Indica si el servidor actual pasado por programación utiliza Seguridad SSL. Los valores permitidos son:

Y – para indicar que SI usa SSL

N – para indicar que NO usa SSL

CFGEMAIL-SSL-TYPE – indica el tipo de seguridad SSL que utilice.

NONE – ninguna

IMPLICIT – TLS Connection

EXPLICIT - STARTTLS

El tipo a usar depende más del tipo de SSL que utilice el servidor al que nos vamos a conectar, entonces primero hay que obtener estos datos y en base a ellos indicar el tipo de seguridad SSL que nos indicaron en el servidor.

Ejemplo:

```
MOVE "Y" TO CFGEMAIL-USE-SSL.  
MOVE "IMPLICIT" TO CFGEMAIL-SSL-TYPE.  
WRITE R-EMAIL FROM CFGEMAIL-SSL.
```

Nota:

Con estos tres registros:

CFGEMAIL-SERVER, CFGEMAIL-AUTH y CFGEMAIL-SSL

Se pueden modificar los datos de conexión del cobemail indicados en la configuración global, de tal manera que podemos unos correos enviarlos por la cuenta de ventas y algunos otros por la cuenta de cobranza o contabilidad, etc.

Esto permite que las respuestas lleguen al departamento o cuenta correspondiente que envió el email.

Registro de Licencia para COBEMAIL

Una vez que adquiera la licencia para COBEMAIL, esta le será entregada en forma de archivo.

Dicho archivo es necesario copiarlo al directorio donde se encuentra su instalación de COBEMAIL para que sea activada al instante.

Si desea adquirir su licencia para uso de COBEMAIL, visite el sitio: www.cobtools.com donde podrá encontrar más información.