

Enviado por Luis María Guayán en Jueves, 23 Octubre, 2003



El modelo de objetos de Outlook es muy rico y poderoso. Esta interfaz está disponible como un servidor de automatización, o sea, que todo lo podemos automatizar mediante programación desde Visual FoxPro.

Enviar y leer correo con Outlook desde Visual FoxPro



Por Luis María Guayán, Tucumán, Argentina

El modelo de objetos de Outlook es muy rico y poderoso. Esta interfaz está disponible como un servidor de automatización, o sea, que todo lo podemos automatizar mediante programación desde Visual FoxPro.

Un breve ejemplo

Una de las tareas más fácil de automatizar en Outlook es el envío de un correo. Veremos un ejemplo de solamente unas pocas líneas.

Lo primero que debemos hacer para automatizar Outlook, es crear un objeto Outlook. Una vez creado el objeto, debemos acceder al origen de los datos, pero esto no lo logramos en forma directa, debemos crear un objeto "NameSpace" apropiado que actuará como entrada (en este ejemplo MAPI). El objeto NameSpace proporciona entre otros, los métodos Logon y Logoff.

```
LOCAL lcPerfil AS CHARACTER, lcContrasenia AS CHARACTER , ;
lcDestinatario AS CHARACTER, lcTema AS CHARACTER , ;
lcCuerpo AS CHARACTER
LOCAL loOutlook AS "Outlook.Application", ;
loNameSpace AS OBJECT, loMailItem AS OBJECT
#DEFINE LF CR CHR(10)+CHR(13)

*-- Datos del Mail
lcPerfil = "Prueba"
lcContrasenia = "prueba"
lcDestinatario = "prueba@portalfox.com"
lcTema = "Prueba: " + TTOC(DATETIME())
lcCuerpo = "Prueba enviando un mail desde Visual FoxPro." + LF_CR
lcCuerpo = lcCuerpo + "Saludos." + LF_CR

*-- Creo objetos Outlook y NameSpace
loOutlook = CREATEOBJECT("Outlook.Application")
loNameSpace = loOutlook.GetNameSpace("MAPI")

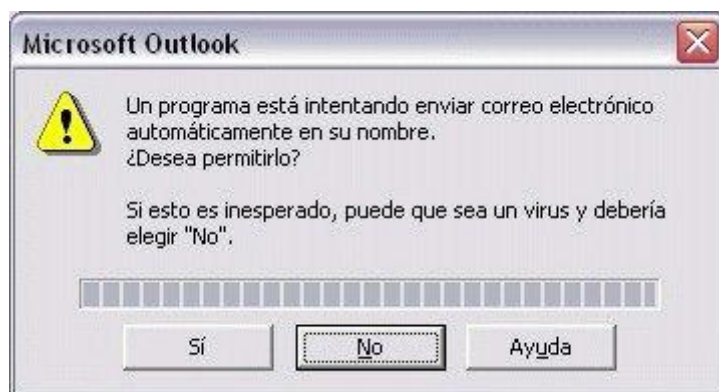
*-- Ejecuto los métodos
loNameSpace.Logon(lcPerfil , lcContrasenia)
loMailItem = loOutlook.CreateItem(0)
loMailItem.Recipients.ADD(lcDestinatario)
loMailItem.Subject = lcTema
loMailItem.Body = lcCuerpo
loMailItem.Send
loNameSpace.Logoff
```

```
loNameSpace = .NULL.  
loOutlook = .NULL.
```

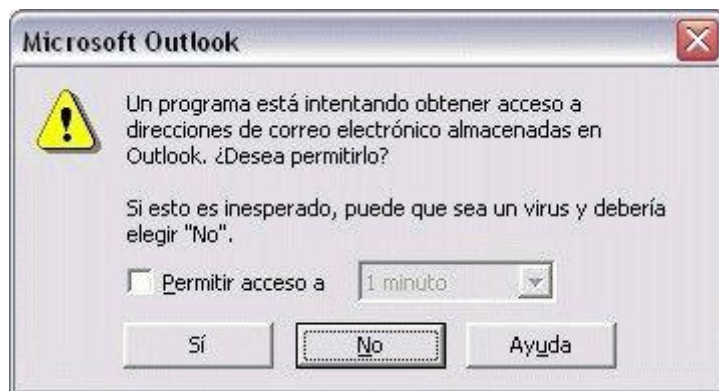
Problemas de seguridad

Outlook XP y Outlook 2000 SP2, incluyen los parches de seguridad de Microsoft. Estos parches restringen, entre otras cosas, el acceso a la libreta de direcciones y el envío de correo mediante automatización, con el fin de evitar códigos maliciosos que toman los datos de nuestra libreta de direcciones y envían correo sin nuestro consentimiento.

Cuando intentamos enviar un correo desde Visual FoxPro, se nos presenta el siguiente cuadro de dialogo, que luego de 5 segundos habilita el botón "Si".



Cuando intentamos acceder a la libreta de direcciones aparece el cuadro de dialogo el cual nos permitirá un acceso inmediato, o de 1, 2, 5, ó 10 minutos que debemos seleccionar.



¿Cómo solucionamos este problema?

Estas son algunas de las opciones que disponemos nosotros para trabajar con estos parches de seguridad:

- Mantener la versión de Office 2000 SR-1 y no actualizarla ni instalarle parches de seguridad, con los peligros que esto significa.
- Si se tienen Outlook y Exchange instalados, el administrador de Exchange, puede disminuir las alertas o registrar algunas aplicaciones como seguras.
- Outlook Redemption: Es un objeto COM que se adapta fácilmente a la automatización y utiliza la MAPI extendida. Esta DLL fue escrita por Dmitry Streblechenko (MS Outlook MVP) y esta disponible en <http://www.dimastr.com/redemption>. Este es un producto comercial con un valor de U\$S 200 aproximadamente. Existe para descarga una versión libre con fines de desarrollo.
- Express ClickYes: Es un pequeño programa residente que se maneja mediante la API de

Windows. Este "presionará" el botón "Si" antes de que el dialogo aparezca. Este programa es gratis y esta disponible en <http://www.express-soft.com/mailmate/clickyes.html>. En el mismo sitio existe un ejemplo para Visual FoxPro.

El formulario de ejemplo

En este ejemplo utilizaremos un formulario con un objeto PageFrame con dos Páginas, una para enviar correo y la otra para leer los correos desde la Bandeja de Entrada.

La siguiente figura es la página para el envío de un correo.

Enviar y leer correo

Para: sysops@portalfox.com

Con Copia:

Tema: Prueba de mail desde Visual FoxPro

Solo una prueba automatizando Outlook XP desde Visual FoxPro 8.

Saludos

Luis Maria Guayan
Tucuman, Argentina

SysOp de www.PortalFox.com
Microsoft Visual FoxPro MVP

Adjuntos: C:\BOOT.INI

Aviso: Mensaje enviado con éxito.

Enviar Salir

Esta otra figura nos muestra la página para la lectura de la bandeja de entrada.

Enviar y leer correo

Enviado por	Tema	Recibido
Pablo Roca	RV: Contacto	19/03/2003 16:10:36
Diego M. Romero	Re: Capacitación CEview	10/05/2003 18:41:01
Antonio Muñoz de Burgos	Nueva publicación	06/09/2003 09:57:45
Nestor Portillo	Question	10/09/2003 16:24:16
Pablo Roca	Sobre el artículo publicado en MSDN	25/09/2003 20:25:25
Ana Maria Bisbé York	Articulos traducidos de Jim Booth	30/09/2003 13:47:45

FYI

Saludos,

Pablo Roca MS - VFP MVP
La Coruña - España
<http://www.portalfox.com>

Todos los mensajes Solo los no leídos Leer mensajes Salir

En el método Init() creamos una instancia de la clase cOutlook con la sentencia:

```
THISFORM.oCorreo = NEWOBJECT('cOutlook','cOutlook.prg')
```

Creamos el servidor de automatización con:

```
THISFORM.oCorreo.CrearServidor()
```

El método CrearServidor() establece una referencia a la instancia de Outlook en la propiedad oOutlook. En este método también creamos un objeto NameSpace que nos permitirá acceder a las carpetas especiales de Outlook.

También en el Init() del formulario, llamamos a otro formulario para el inicio de sesión de Outlook:

```
*-- Formulario de inicio de sesión
DO FORM Inicio WITH THISFORM.oCorreo TO llAceptar
IF NOT (llAceptar AND THISFORM.oCorreo.IniciarSesion())
MESSAGEBOX('Falló el inicio sesión', 48, 'Inicio de sesión')
RETURN .F.
ENDIF
```

Enviar un correo

Antes de invocar el método EnviarCorreo(), configuramos todas las propiedades necesarias para el envío de correo. Esto lo hacemos en el método Click() del botón "Enviar".

```
*-- Armo el mensaje
WITH THISFORM.oCorreo
    .CargarVector(THIS.PARENT.txtTo.VALUE, 'aTO')
    .CargarVector(THIS.PARENT.txtCC.VALUE, 'aCC')
    .CargarVector(THIS.PARENT.txtAdjunto.VALUE, 'aAdjuntos')
    .cTema = ALLTRIM(THIS.PARENT.txtTema.VALUE)
    .cCuerpo = ALLTRIM(THIS.PARENT.edtCuerpo.VALUE)
    IF .EnviarCorreo()
        MESSAGEBOX('Mensaje enviado con éxito.', 64, 'Aviso')
        THISFORM.LimpiarPagina()
    ELSE
        MESSAGEBOX('No se pudo enviar el mensaje.', 48, 'Problemas')
    ENDIF
ENDWITH
RETURN
```

En el llamado del método EnviarCorreo() de la clase cOutlook es donde se crea un nuevo mensaje y se arma según las propiedades anteriormente configuradas.

Leer los correos

Para leer los correos de la bandeja de entrada invocamos el método LeerMensajes() de la clase cOutlook desde el método Click() del botón "Leer".

```
ZAP IN curMsg
IF THISFORM.oCorreo.LeerMensajes(THIS.PARENT.opgTipo.VALUE = 1,
'curMsg')
    GO TOP IN curMsg
    THIS.PARENT.grdMensajes.SETFOCUS
ELSE
    MESSAGEBOX('No existen mensajes para traer', 64, 'Aviso')
ENDIF
THIS.PARENT.edtCuerpo.REFRESH
```

En la llamada al método LeerMensajes() creamos un objeto loInbox y traemos todos los mensajes, o solo los mensajes "No leídos" y recorremos uno a uno para cargarlos en un cursor que luego mostraremos en una Cuadrícula y un Cuadro de Edición.

Definición de la clase cOutlook

En este ejemplo disponemos de una clase definida por el usuario, llamada cOutlook con los distintos métodos para realizar el envío y la lectura de los correos. La definición de esta clase es la siguiente:

```
DEFINE CLASS cOutlook AS CUSTOM
  *-- Propiedades
  oOutlook = .NULL.
  oNameSpace = .NULL.
  cPerfil = ''
  cContrasenia = ''
  cTema = ''
  cCuerpo = ''
  DIMENSION aTo(1) AS CHARACTER
  DIMENSION aCC(1) AS CHARACTER
  DIMENSION aBCC(1) AS CHARACTER
  DIMENSION aAdjuntos(1) AS CHARACTER
  *-----
  * Creo el servidor de automatización
  *-----
  PROCEDURE CrearServidor()
    LOCAL loErr1 AS EXCEPTION, loErr2 AS EXCEPTION, llRet AS Logical
    WITH THIS
      *-- Manejo el error con TRY ... CATCH ... FINALLY
      TRY
        *-- Instancio el objeto
        .oOutlook = GETOBJECT( , 'Outlook.Application')
        WAIT WINDOW 'Ya existe una instancia de Outlook...' TIMEOUT 2
      CATCH TO loErr1
      TRY
        *-- Creo el objeto
        .oOutlook = CREATEOBJECT('Outlook.Application')
        WAIT WINDOW 'Nueva instancia de Outlook...' TIMEOUT 2
      CATCH TO loErr2
        MESSAGEBOX('Microsoft Outlook no está instalado.', 16,
'Problemas!!!')
      FINALLY
        ENDTRY
      FINALLY
        ENDTRY
      IF VARTYPE(.oOutlook) = 'O'
        .oNameSpace = .oOutlook.GetNameSpace('MAPI')
        IF VARTYPE(.oNameSpace) = 'O'
          llRet = .T.
        ENDIF
      ENDIF
    ENDWITH
    RETURN llRet
  ENDPROC
  *-----
  * Cierro el servidor de automatización
  *-----
  PROCEDURE CerrarServidor()
    THIS.oOutlook.QUIT()
```

```

RETURN
ENDPROC
*-----
*  Iniciar sesion
*-----
PROCEDURE IniciarSesion()
    LOCAL llRet AS Logical
    TRY
        THIS.oNameSpace.Logon(THIS.cPerfil, THIS.cContrasenia)
        llRet = .T.
    CATCH
        *-- No pudo iniciar sesión
        llRet = .F.
    FINALLY
    ENDTRY
    RETURN llRet
ENDPROC
*-----
*  Cerrar sesion
*-----
PROCEDURE CerrarSesion()
    THIS.oNameSpace.Logoff()
    RETURN
ENDPROC
*-----
*  Envio el correo
*-----
PROCEDURE EnviarCorreo()
    LOCAL loMensaje AS OBJECT, llRet AS Logical
    LOCAL lnI AS INTEGER, lnIndex AS INTEGER
    *-- Creo un nuevo mensaje
    WITH THIS
        loMensaje = .oOutlook.CreateItem(0)
        IF VARTYPE(loMensaje) = 'O'
            loMensaje.Subject = .cTema
            loMensaje.Body = .cCuerpo
            *-- Recipientes
            lnIndex = 0
            *-- TO
            lnLen = ALEN(.aTO)
            FOR lnI = 1 TO lnLen
                IF NOT EMPTY(.aTO(lnI))
                    lnIndex = lnIndex + 1
                    loMensaje.Recipients.ADD(.aTO(lnI))
                    loMensaje.Recipients(lnIndex).TYPE = 1
                ENDIF
            ENDFOR
            *-- CC
            lnLen = ALEN(.aCC)
            FOR lnI = 1 TO lnLen
                IF NOT EMPTY(.aCC(lnI))
                    lnIndex = lnIndex + 1
                    loMensaje.Recipients.ADD(.aCC(lnI))
                    loMensaje.Recipients(lnIndex).TYPE = 2
                ENDIF
            ENDFOR
            *-- BCC
            lnLen = ALEN(.aBCC)
            FOR lnI = 1 TO lnLen
                IF NOT EMPTY(.aBCC(lnI))
                    lnIndex = lnIndex + 1

```

```

        loMensaje.Recipients.ADD(.aBCC(lnI))
        loMensaje.Recipients(lnIndex).TYPE = 3
    ENDIF
ENDFOR
*-- Adjuntos
lnLen = ALEN(.aAdjuntos)
FOR lnI = 1 TO lnLen
    IF NOT EMPTY(.aAdjuntos(lnI)) AND FILE(.aAdjuntos(lnI))
        loMensaje.Attachments.ADD(.aAdjuntos(lnI))

    ENDIF
ENDFOR
llRet = loMensaje.SEND
ELSE
    llRet = .F.
ENDIF
ENDWITH
RETURN llRet
ENDPROC
*-----
* Lee los mensajes según parámetro
*-----
PROCEDURE LeerMensajes(tlNoLeidos, tcAlias)
    LOCAL loInbox AS 'Outlook.MAPIFolder', loMensajes AS
'Outlook.Items'
    LOCAL loMsg AS OBJECT, lnI AS INTEGER, llRet AS Logical
    IF EMPTY(tcAlias)
        tcAlias = 'curMsg'
    ENDIF
    *-- Inbox
    loInbox = THIS.oNameSpace.GetDefaultFolder(6)
    *-- Mensajes del Inbox
    IF tlNoLeidos
        loMensajes = loInbox.Items.RESTRICT("[Unread] = True")
    ELSE
        loMensajes = loInbox.Items
    ENDIF
    IF VARTYPE(loMensajes) = 'O'
        WITH loMensajes
            IF .COUNT > 0
                *-- Recorro los mensajes
                FOR lnI = 1 TO .COUNT
                    loMsg = .ITEM(lnI)
                    WITH loMsg
                        INSERT INTO (tcAlias) (EnviadoPor, Tema, Recibido,
Cuerpo, NoLeido) ;
                            VALUES (.SenderName, .Subject, .ReceivedTime, .Body,
.UnRead)
                    ENDWITH
                ENDFOR
                llRet = .T.
            ELSE
                llRet = .F.
            ENDIF
        ENDWITH
    ELSE
        llRet = .F.
    ENDIF
    RETURN llRet
ENDPROC
*-----

```

```

* Destroy
*-----
PROCEDURE DESTROY()
    WITH THIS
        .oNameSpace = .NULL.
        .oOutlook = .NULL.
    ENDWITH
ENDPROC
*-----
* Cargo una Matriz con los destinatarios
*-----
PROCEDURE CargarVector(tcTexto, tcVector)
    LOCAL ln AS INTEGER, lnI AS INTEGER, la(1) AS CHARACTER
    IF EMPTY(tcTexto)
        RETURN 0
    ENDIF
    tcTexto = CHRTRAN(tcTexto, ',', ';')
    ln = ALINES(la, tcTexto, .T., ';')
    lcV = "This." + tcVector
    DIMENSION &lcV.(ln)
    FOR lnI = 1 TO ln
        &lcV.(lnI) = la(lnI)
    ENDFOR
    RETURN ln
ENDPROC
ENDDEFINE && Clase cOutlook

```

A la definición de la clase, como así también los formularios, y el ejemplo los podemos descargar del siguiente vínculo: [outlook.zip](#) para su evaluación. Los ejemplos están realizados con Visual FoxPro 8 y Outlook 2002 (Outlook XP).

Hasta la próxima.

Luis María Guayán
Tucumán, Argentina

SysOp de www.PortalFox.com
Microsoft Visual FoxPro MVP
