

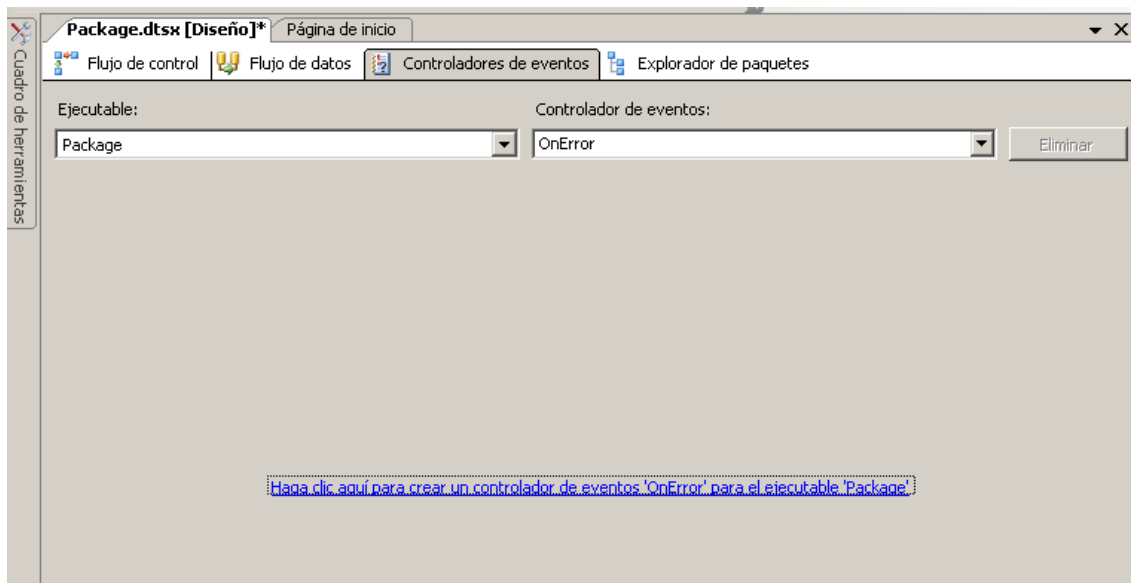


La mayoría de las categorías de eventos (con excepción de aquellos de componentes específicos) existen por paquete, es decir, que en un paquete puede haber múltiples manejadores de eventos llamados por un solo tipo de evento. Los eventos llamados por un subcomponente son procesados por default por su propio manejador (event handler) y por el manejador del mismo evento de su padre o antecesores hasta el contenedor de mayor nivel. Es decir, los eventos se procesan jerárquicamente.

Es posible evitar el procesamiento de eventos en los subcomponentes por sus antecesores o padres si se asigna "falso" a la propiedad "Propagate" (el default es "verdadero").

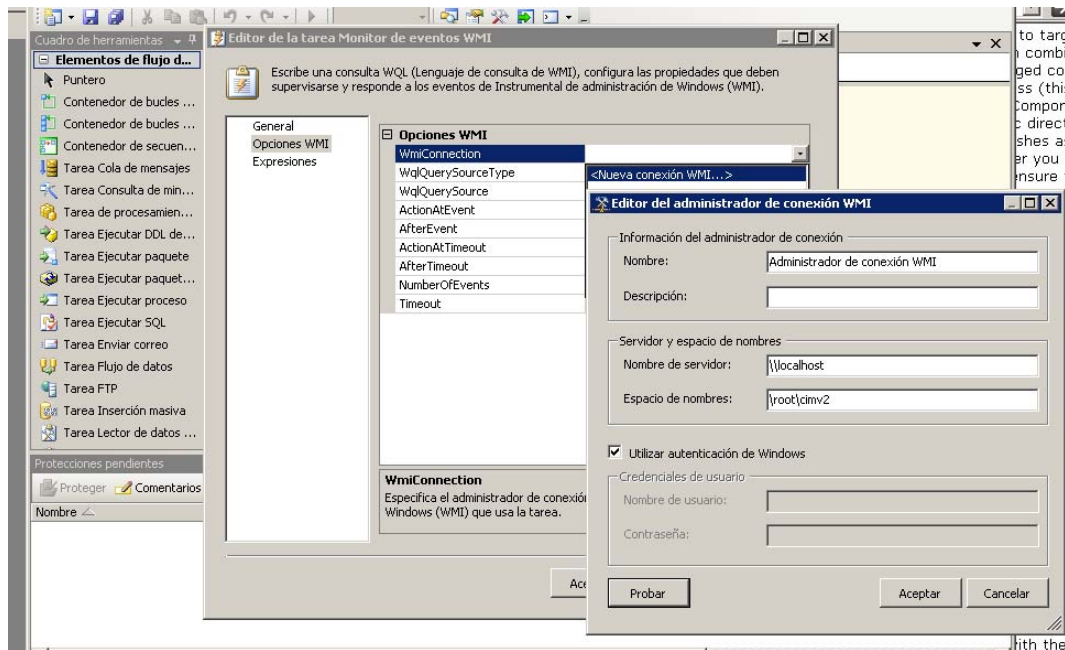
Los event handlers comparten diversas variables del sistema como EventHandlerStartTime, LocaleID, SourceDescription, SourceID y SourceName y variables específicas de control como **ErrorCode** y **ErrorDescription**, que están asociadas al evento OnError.

Con el Business Intelligence Development Studio vayamos al tab "Controladores de eventos" que se encuentra en el diseñador de paquetes. El tab contiene 2 listbox: uno con todos los paquetes ejecutables y el otro con los eventos que corresponden a cada uno y abajo un link para crear un controlador de eventos para el paquete abierto:



Cuando presionamos clic en el link se presentará una pantalla para añadir elementos de control de flujo si ocurre un evento como un error.

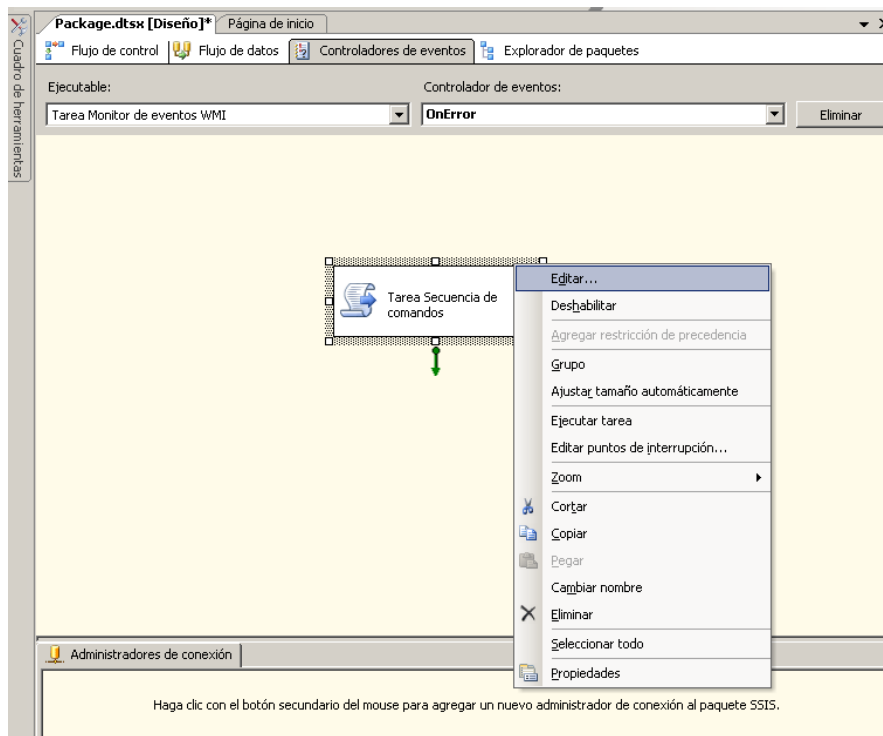
Por ejemplo, para el evento OnError, se puede disparar una condición de error usando la tarea "WMI Event Watcher" o "Tarea Monitor de eventos WMI". Colocamos esta tarea en el paquete y vayamos a editar sus propiedades en la opción "Opciones WMI" y creamos una conexión WMI con el valor por default como se muestra a continuación:



En WqlQuerySource colocamos cualquier consulta WMI que genere un error:

```
SELECT * FROM __InstanceCreationEvent WITHIN 10
WHERE TargetInstance ISA "Win32_NTLogEvent"
AND TargetInstance.Logfile = "System"
```

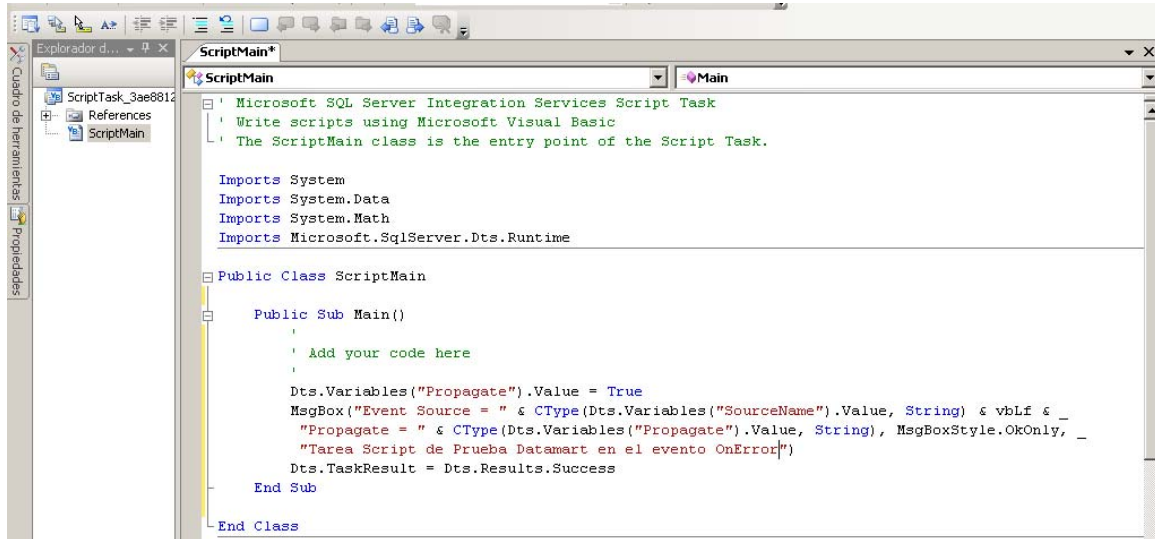
Arrastramos la tarea "Tarea Secuencia de comandos" al diseñador en el área del evento "On error" y presionamos clic derecho sobre esta tarea e ingresamos a la opción "Editar".



Luego seleccionamos la opción "Secuencia de Comandos" y colocamos en "ReadOnlyVariables" el valor "SourceName" y en "ReadWriteVariables" colocamos el valor "Propagate". Estas son variables del sistema del manejador de eventos "On Error" que usaremos para obtener información acerca de la forma en que los eventos son manejados y veremos como modificar su comportamiento por default.

Hacemos Clic en el botón "Diseñador de Secuencia de Comandos" y colocamos el siguiente contenido en Public Sub Main():

```
Public Sub Main()  
    Dts.Variables("Propagate").Value = True  
    MsgBox("Event Source = " & CType(Dts.Variables("SourceName").Value, String) & vbCrLf & _  
        "Propagate = " & CType(Dts.Variables("Propagate").Value, String), MsgBoxStyle.OKOnly, _  
        "Tarea Script de Prueba Datamart en el evento OnError")  
    Dts.TaskResult = Dts.Results.Success  
End Sub
```



En este código se asigna el valor "True" a la variable del sistema "Propagate" y muestra una ventana de mensaje que es el origen del error y el valor actual de la variable Propagate. Luego cerramos "Microsoft Visual Studio for Applications" y presionamos clic en "Aceptar" en el editor de la tarea de secuencia de comandos.

Luego seleccionamos de la lista de paquetes el paquete mismo y usando el mismo procedimiento que el anterior creamos un manejador de eventos para OnError que contenga una tarea de secuencia de comandos. En la opción "Editar" de esta tarea colocamos en "ReadOnlyVariables", el valor "SourceName" y colocamos el siguiente script en la ventana de Microsoft Visual Studio for Applications en el botón de diseño de scripts.

```
Public Sub Main()  
    MsgBox("Evento de origen = " & CType(Dts.Variables("SourceName").Value, String), _  
        MsgBoxStyle.OKOnly, "Tarea script del evento OnError del paquete de prueba ")  
    Dts.TaskResult = Dts.Results.Success  
End Sub
```

Cerramos Microsoft Visual Studio for Applications y el editor de la tarea script.

Luego, ejecutamos el paquete y luego de un segundo de ejecución (el valor de timeout) se dispara el error disparando el evento OnError que se definió para la tarea de monitoreo de eventos generandose la ventana de mensajes que colocamos en VBA para aplicaciones, si presionamos "Aceptar" en la ventana del mensaje del evento del paquete contenedor (sólo para demostrar la jerarquía). Luego, el paquete falla en su ejecución, pero se han capturado los errores correspondientes.

(Consultas: postmaster@softein.com)