

DLL NEO - Versión de Integración – v1.0.0.0

Índice:

1. Requerimientos
2. Ficheros
3. Espacio de Nombres
4. Constructor
5. Destructor
6. Inicialización / Conexión
7. Eventos
8. Parámetros Generales y de Red del Equipo
9. Lectura de Estados del Sistema
10. Información equipos FlexNet incluidos en el sistema (NEO Extensions)
11. Lectura de Parámetros y Estados de Zonas
12. Lectura de Parámetros y Estados de Fuentes
13. Control y Monitorización de Zonas
14. Control y Monitorización de Fuentes
15. Inicio / Detención de la lectura de vómetros
16. Fecha y Hora del Equipo
17. Supervisión de Líneas de Altavoces
18. Supervisión de Alimentación
19. Ejemplos de Uso

1. Requerimientos

- NET Framework 4.0

2. Ficheros

Ficheros que habrá que incluir como referencia en proyectos de integración.

NeoCtrlLib.dll	Librería para control y monitorización de sistemas LDA NEO
Socket.dll	Librería auxiliar.

3. Espacio de Nombres

```
LDA.Device.NEO240EN
```

4. Constructor

```
Dim WithEvents neo As NEO240EN  
neo = New NEO240EN()
```

5. Destructor

```
neo.Dispose() ó neo.pClose()
```

- Ambos métodos son equivalentes, emplear sólo uno de ellos.
- Llamar siempre que deje de utilizarse cualquier instancia de la librería para liberar recursos y finalizar subprocesos.

6. Inicialización / Conexión

```
Public Function foConnect(  
    ByVal IP As String,  
    Optional ByVal puerto As Integer = 60000,  
    Optional ByVal enableBroadcastSocket As Boolean = False  
) As Collection
```

- Lo habitual es especificar únicamente la IP del equipo remoto, por ejemplo:

```
res = neo.foConnect("192.168.3.110")
```

- Posibles valores de retorno:

- res("state") = 1
Ok.
- res("state") = 0
Dirección IP no válida.

7. Eventos

`Public Event data_On_Date()`

Se lanza este evento cuando los datos de configuración del equipo están actualizados en la DLL. Puede emplearse como confirmación de fiabilidad en la lectura de parámetros de configuración del sistema.

`Public Event evEMGStatusChanged(ByVal senderIp As String, ByVal emgIsActive As Boolean)`

Se lanza este evento cuando el sistema NEO al que estamos conectados entra o sale del estado de Emergencia. Nos da como parámetros una cadena con la IP del equipo y un booleano indicando si la emergencia está activa (True) o no (False).

`Public Event evAcsiModeChanged(ByVal senderIp As String, ByVal new_status As Boolean)`

Evento que indica un cambio en el modo ACSI del sistema NEO. Nos da como parámetros una cadena con la IP del equipo y un booleano que indica si el modo ACSI está activo (True) o no (False). El modo ACSI se activa cuando se conectan dispositivos ACSI (Micrófonos MPS y VAP) en el puerto ACSI. Cuando esto sucede la entrada de audio 5 deja de estar disponible como fuente de audio del sistema y es empleada internamente para distribuir el audio de los dispositivos ACSI.

`Public Event evInstallationVersionChanged(ByVal senderIp As String, ByVal new_value As UInt32)`

El parámetro 'InstallationVersion' es un valor numérico vinculado a la actual configuración de zonas del sistema (número de zonas y asignación de canales de salida a las mismas). Cuando se detecta algún cambio en esta configuración se lanzará este evento que nos dará como parámetros la IP del equipo y el nuevo valor numérico asignado a la nueva configuración zonal.

`Public Event device_reset_occurred()`

Este evento se lanza si se detecta un reset o reinicio en el equipo.

8. Parámetros Generales y de Red del Equipo

PARÁMETRO	TIPO	DESCRIPCIÓN
<code>neo.NeoConfig.deviceInfo.modelo</code>	String	Nombre de Modelo NEO
<code>neo.NeoConfig.deviceInfo.installationVersion</code>	UInteger	Numero de instalación - Cambia siempre que se altere el número de zonas o la asignación de canales a éstas.
<code>neo.NeoConfig.deviceInfo.serial</code>	String	Número de Serie del Equipo
<code>neo.NeoConfig.deviceInfo.place</code>	String	Cadena de texto informativa sobre ubicación del equipo
<code>neo.NeoConfig.deviceInfo.ap_ver</code>	String	Versión de firmware del equipo
<code>neo.NeoConfig.deviceInfo.master</code>	Boolean	Indica si el equipo es el NEO controlador del sistema (true) o no (false).
<code>neo.NeoConfig.deviceInfo.extension</code>	Boolean	Indica si el equipo es un NEO Extension (true) o no (false).
<code>neo.NeoConfig.deviceInfo.mac</code>	Byte array	MAC del interfaz Ethernet
<code>neo.NeoConfig.ethConfig.ip</code>	Byte Array	IP del equipo
<code>neo.NeoConfig.ethConfig.gateway</code>	Byte Array	Puerta de enlace del equipo
<code>neo.NeoConfig.ethConfig.mask</code>	Byte Array	Máscara de red del equipo

9. Lectura de Estados del Sistema

<u>PARÁMETRO</u>	<u>TIPO</u>	<u>DESCRIPCIÓN</u>
<code>neo.Connected</code>	Boolean	<p>True: Hay conexión establecida con el equipo.</p> <p>False: No hay conexión establecida con el equipo.</p>
<code>neo.OnDate</code>	Boolean	<p>True: La DLL tiene actualizados los parámetros del equipo (se puede comprobar este flag antes de leer algún parámetro del equipo en la DLL o bien hacer uso del evento asociado descrito en el apartado anterior)</p> <p>False: Hay parámetros del sistema no sincronizados con el equipo.</p>
<code>neo.NeoConfig.systemStatus.EMG_Active</code>	Boolean	<p>True: EMG Activo. Sistema en estado de Emergencia.</p> <p>False: EMG No Activo.</p>
<code>neo.NeoConfig.systemStatus.FLT_Active</code>	Boolean	<p>True: FLT Activo. Sistema en estado de Fallo</p> <p>False: FLT No Activo. No hay fallos detectados en el sistema.</p>
<code>neo.NeoConfig.systemStatus.DIS_Active</code>	Boolean	<p>True: DIS Activo. Alguna zona del sistema está desarmada.</p> <p>False: DIS No Activo – Ninguna zona desarmada.</p>
<code>neo.NeoConfig.systemStatus.ACSI_Mode</code>	Boolean	<p>True: Modo ACSI Activo. Entrada 5 no disponible como entrada de audio, empleada internamente para distribución de audio de micrófonos MPS y VAP.</p> <p>False: Modo ACSI No Activo.</p>

10. Información equipos FlexNet incluidos en el sistema (NEO Extensions)

La información sobre los equipos FlexNet (NEO Extensions) que forman parte del sistema la podemos obtener del controlador del sistema accediendo a:

```
neo.NeoConfig.flexnetSlaves
```

Este campo es un array donde cada posición representa un NEO Extension que forma parte del sistema LDA NEO. Cada equipo tiene un identificador FlexNext. El NEO controlador del sistema tiene reservado el identificador 1, el resto de equipos (NEO Extensions) tendrán identificadores a partir del 2 en adelante, siempre consecutivos. El controlador del sistema no aparece en este array, por tanto el primer equipo listado tendrá el identificador 2. Si no hay NEO Extensions en el sistema el array aparecerá vacío. La información que podemos encontrar aquí sobre cada NEO Extension es la siguiente:

<u>PARÁMETRO</u>	<u>TIPO</u>	<u>DESCRIPCIÓN</u>
<code>neo.NeoConfig.flexnetSlaves(i).id</code>	Byte	Identificador FlexNet del NEO Extension
<code>neo.NeoConfig.flexnetSlaves(i).modelo</code>	String	Modelo de NEO
<code>neo.NeoConfig.flexnetSlaves(i).IP</code>	Byte Array	IP del equipo
<code>neo.NeoConfig.flexnetSlaves(i).Gateway</code>	Byte Array	Puerta de enlace del equipo
<code>neo.NeoConfig.flexnetSlaves(i).Submask</code>	Byte Array	Máscara de red
<code>neo.NeoConfig.flexnetSlaves(i).ap_ver</code>	String	Versión de firmware
<code>neo.NeoConfig.flexnetSlaves(i).firstOutputId</code>	Ushort	Identificador global en el sistema del primer canal de salida de este equipo
<code>neo.NeoConfig.flexnetSlaves(i).outputsCount</code>	Byte	Número de canales de salida de este equipo
<code>neo.NeoConfig.flexnetSlaves(i).missing</code>	Boolean	True: Indica que el equipo esperado no se encuentra o se corresponde con un modelo distinto al configurado

11. Lectura de Parámetros y Estados de Zonas

La información sobre el estado de las zonas del sistema NEO la leeremos del controlador del sistema. Sus valores se guardan en:

```
neo.NeoConfig.zm.zones
```

Este campo es un array donde cada posición representa una zona. Estas se encuentran ordenadas por su identificador. La primera zona tendrá siempre el identificador 1 y el resto serán consecutivos. Es posible que este array tenga valor nulo si aun no se han sincronizado los valores de la DLL con los del equipo (aun no estamos OnDate). Estos parámetros son los que podemos encontrar en cada posición del array de zonas del sistema:

PARÁMETRO	TIPO	DESCRIPCIÓN
<code>neo.NeoConfig.zm.zones(i).id</code>	UShort	Identificador de zona
<code>neo.NeoConfig.zm.zones(i).name</code>	String	Nombre de la zona
<code>neo.NeoConfig.zm.zones(i).desc</code>	String	Descripción de la zona
<code>neo.NeoConfig.zm.zones(i).gain</code>	Sbyte	Volumen de la zona
<code>neo.NeoConfig.zm.zones(i).mute</code>	Boolean	Indica si la zona tiene Mute activo (true) o no (false)
<code>neo.NeoConfig.zm.zones(i).source</code>	Byte	Identificador de la fuente rutada a esta zona o 0 (cero) si no hay ninguna fuente rutada.
<code>neo.NeoConfig.zm.zones(i).effectiveSource</code>	Byte	Fuente efectiva rutada. Normalmente coincide con el campo anterior, salvo que se ruten fuentes temporales (micrófonos PTT o ACSI, reproducción de mensajes desde Eventos del sistema...)
<code>neo.NeoConfig.zm.zones(i).fltActive</code>	Boolean	La zona se encuentra en estado de Fallo (si true).
<code>neo.NeoConfig.zm.zones(i).emgActive</code>	Boolean	La zona se encuentra en estado de Emergencia (si true).
<code>neo.NeoConfig.zm.zones(i).disarmed</code>	Boolean	La zona se encuentra en estado de Desarme (si true).
<code>neo.NeoConfig.zm.zones(i).eventId</code>	Byte	Si es mayor que 0 indica el identificador de un evento del sistema que está haciendo uso de la zona.

12. Lectura de Parámetros y Estados de Fuentes

La información sobre el estado de las fuentes del sistema NEO la leeremos del controlador del sistema. Sus valores se guardan en:

```
neo.NeoConfig.zm.sources
```

Este campo es un array donde cada posición representa una fuente. Estas se encuentran ordenadas por su identificador. La primera zona tendrá siempre el identificador 1 y el resto serán consecutivos.

Identificadores de las fuentes del sistema:

- **1 a 5** : Se refieren a las 5 **entradas físicas** ubicadas en la trasera del equipo NEO controlador del sistema.
- **6** : Se refiere al **micrófono PTT** ubicado en el frontal del equipo NEO controlador del sistema.
- **7 y 8** : Se refieren a los dos **reproductores de mensajes de audio** internos del equipo NEO controlador del sistema.

Estos parámetros son los que podemos encontrar en cada posición del array de fuentes de audio del sistema:

<u>PARÁMETRO</u>	<u>TIPO</u>	<u>DESCRIPCIÓN</u>
<code>neo.NeoConfig.zm.sources(i).id</code>	Byte	Identificador de la fuente
<code>neo.NeoConfig.zm.sources(i).name</code>	String	Nombre de la fuente
<code>neo.NeoConfig.zm.sources(i).desc</code>	String	Descripción
<code>neo.NeoConfig.zm.sources(i).gain</code>	Sbyte	Ganancia
<code>neo.NeoConfig.zm.sources(i).mute</code>	Boolean	Mute activo (true) o no (false)
<code>neo.NeoConfig.zm.sources(i).eventId</code>	Byte	Si es mayor que 0 indica el identificador de un evento del sistema que está haciendo uso de la fuente.

13. Control y Monitorización de Zonas

```
Public Function foGetZonesCount(
    ByRef zones As Ushort
) As Collection
```

- Nos devuelve en el parámetro pasado por referencia el número de zonas configuradas en el sistema NEO:

```
res = neo.foGetZonesCount(num_zonas)
```

- Posibles valores de retorno:
 - res("state") = NEO_RESULT.OK
 - res("state") = NEO_RESULT.ERROR_NOT_CONNECTED : Sin conexión con el equipo.
 - res("state") = NEO_RESULT.ERROR_SYSTEM_BUSY : Aun no se ha determinado el número de zonas del sistema.

```
Public Function foGetZoneVumeter(
    ByVal zoneId As Ushort,
    ByRef vum As Sbyte
) As Collection
```

- Nos devuelve en el parámetro pasado por referencia el vumetro de una zona. En el caso de una zona con múltiples canales de salida asignados, se devolverá el vumetro del nivel más alto leído. La lectura periódica de los vumetros debe haberse inicializado con la función startVumetersRead.
- Función válida con el NEO controlador del sistema, no con NEO Extensions.
- Por ejemplo, para leer en la variable zVum el vumetro de la zona 1:

```
res = neo.foGetZoneVumeter(1, zVum)
```

- Posibles valores de retorno:
 - res("state") = NEO_RESULT.OK
 - res("state") = NEO_RESULT.ERROR_NOT_CONNECTED : Sin conexión con el equipo.
 - res("state") = NEO_RESULT.ERROR_FORBIDDEN_OPERATION : El equipo es un NEO Extension.
 - res("state") = NEO_RESULT.ERROR_INVALID_OPERANDS : Identificador de zona no válido.
 - res("state") = NEO_RESULT.UNKNOWN_ERROR : Error no especificado.

```
Public Function foSetZoneVolume(
    ByVal zoneId As Ushort,
    ByVal gain As Sbyte,
    Optional ByVal waitDataOnDate As Boolean = False
) As Collection
```

- Establece el volumen de una zona del sistema determinada por su identificador.
- Podemos también cambiar simultáneamente el volumen de todas las zonas estableciendo el parámetro `zoneId = 0`.
- El parámetro `waitDataOnDate` indica si se quiere que la función realice una espera hasta que los cambios de rutado se hayan realizado en el equipo y se hayan sincronizado en la DLL.
- Los valores máximo y mínimo teóricos que puede tener el volumen de una zona son:

```
NEO240EN.ZONE_MANAGER_VOLUME_LIMITS.ZONE_MAX
NEO240EN.ZONE_MANAGER_VOLUME_LIMITS.ZONE_MIN
```

- El volumen máximo y mínimo que se puede asignar a una zona se ve también limitado por el volumen de los canales de salida asignados a dicha zona.
- Por ejemplo, si queremos establecer un volumen de -25 para la zona 4:

```
res = neo.foSetZoneVolume(4, -25)
```

- Posibles valores de retorno:
 - `res("state") = NEO_RESULT.OK`
 - `res("state") = NEO_RESULT.ERROR_NOT_CONNECTED`
Sin conexión con el equipo.
 - `res("state") = NEO_RESULT.ERROR_FORBIDDEN_OPERATION_IN_EMG_STATE`
Sistema en estado de Emergencia, operación no permitida.
 - `res("state") = NEO_RESULT.ERROR_ITEM_DOESNT_EXIST`
Identificador de zona no válido.
 - `res("state") = NEO_RESULT.ERROR_INVALID_VOLUME`
El volumen especificado supera los máximos teóricos para un volumen de zona.
 - `res("state") = NEO_RESULT.ERROR_VOLUME_CONFLICT_BETWEEN_OUTPUT_AND_ZONE`
Conflicto de volumen entre la zona y alguno de los canales de salida asignados a ella.
 - `res("state") = NEO_RESULT.ERROR_TIMEOUT`
Se ha superado el timeout de espera para la sincronización de datos de la función cuando se le pasa el parámetro `waitDataOnDate` con el valor `True`.

```
Public Function foSetZoneRoute(  
    ByVal zoneId As Ushort,  
    ByVal sourceId As Byte,  
    Optional ByVal waitDataOnDate As Boolean = False  
) As Collection
```

- Establece el rutado de una zona del sistema.
- Podemos también cambiar simultáneamente el rutado de todas las zonas estableciendo el parámetro `zoneId = 0`.
- Para dejar una zona sin rutado se establecerá el parámetro `sourceId = 0`
- El parámetro `waitDataOnDate` indica si se quiere que la función realice una espera hasta que los cambios de rutado se hayan realizado en el equipo y se hayan sincronizado en la DLL.
- Por ejemplo, rutar a la zona 3 la fuente 2 sin realizar espera de sincronización de datos:

```
res = neo.foSetZoneRoute(3, 2)
```

- Posibles valores de retorno:
 - `res("state") = NEO_RESULT.OK`
 - `res("state") = NEO_RESULT.ERROR_NOT_CONNECTED`
Sin conexión con el equipo.
 - `res("state") = NEO_RESULT.ERROR_FORBIDDEN_OPERATION_IN_EMG_STATE`
Sistema en estado de Emergencia, operación no permitida.
 - `res("state") = NEO_RESULT.ERROR_ITEM_DOESNT_EXIST`
Identificador de zona no válido.
 - `res("state") = NEO_RESULT.ERROR_INVALID_OPERANDS`
Identificador de fuente no válido.
 - `res("state") = NEO_RESULT.ERROR_TIMEOUT`
Se ha superado el timeout de espera para la sincronización de datos de la función cuando se le pasa el parámetro `waitDataOnDate` con el valor `True`.

```
Public Function foSetZoneMultiRoute(  
    ByVal zonesId() As Ushort,  
    ByVal sourcesId() As Byte,  
    Optional ByVal waitDataOnDate As Boolean = False  
) As Collection
```

- Establece simultáneamente el rutado de múltiples zonas del sistema.
- El parámetro `waitDataOnDate` indica si se quiere que la función realice una espera hasta que los cambios de rutado se hayan realizado en el equipo y se hayan sincronizado en la DLL.
- Por ejemplo, rutar a las zonas 1, 2 y 3 las fuentes 2,4 y 1 respectivamente sin realizar espera de sincronización de datos:

```
res = neo.foSetZoneMultiRoute({1,2,3}, {2,4,1})
```

- Posibles valores de retorno:
 - `res("state") = NEO_RESULT.OK`
 - `res("state") = NEO_RESULT.ERROR_INVALID_OPERANDS`:
Los arrays son nulos, están vacíos, son de distinta longitud o ésta supera el número máximo de zonas permitidas en el sistema (`NEO240EN.ZM_MAX_ZONES_COUNT`)
 - `res("state") = NEO_RESULT.ERROR_NOT_CONNECTED`
Sin conexión con el equipo.
 - `res("state") = NEO_RESULT.ERROR_FORBIDDEN_OPERATION_IN_EMG_STATE`
Sistema en estado de Emergencia, operación no permitida.
 - `res("state") = NEO_RESULT.ERROR_UNSUPPORTED_OPERATION_IN_FW_VER`
Operación no soportada por la versión de firmware del equipo.
 - `res("state") = NEO_RESULT.ERROR_ITEM_DOESNT_EXIST`
Algún identificador de zona no es válido.
 - `res("state") = NEO_RESULT.ERROR_TIMEOUT`
Se ha superado el timeout de espera para la sincronización de datos de la función cuando se le pasa el parámetro `waitDataOnDate` con el valor `True`.

```
Public Function foSetZoneMute(  
    ByVal zoneId As Ushort,  
    ByVal mute As Boolean,  
    Optional ByVal waitDataOnDate As Boolean = False  
) As Collection
```

- Configura si se activa o no el 'mute' de una zona determinada por su identificador.
- El parámetro `waitDataOnDate` indica si se quiere que la función realice una espera hasta que los cambios de rutado se hayan realizado en el equipo y se hayan sincronizado en la DLL.
- Por ejemplo, si queremos activar el 'mute' de la zona del sistema con identificador 12 y realizar una espera hasta que el cambio tenga efecto:

```
res = neo.foSetZoneMute(12, True, True)
```

- Posibles valores de retorno:
 - `res("state") = NEO_RESULT.OK`
 - `res("state") = NEO_RESULT.ERROR_NOT_CONNECTED`
Sin conexión con el equipo.
 - `res("state") = NEO_RESULT.ERROR_FORBIDDEN_OPERATION_IN_EMG_STATE`
El sistema está en estado de Emergencia, operación no permitida.
 - `res("state") = NEO_RESULT.ERROR_ITEM_DOESNT_EXIST`
Identificador de zona incorrecto.
 - `res("state") = NEO_RESULT.ERROR_TIMEOUT`
Se ha superado el timeout de espera para la sincronización de datos de la función cuando se le pasa el parámetro `waitDataOnDate` con el valor `True`.

```
Public Function foSetZoneExtraInfo(  
    ByVal zoneId As UShort,  
    ByVal name As String,  
    ByVal desc As String  
) As Collection
```

- Establece la información extra (nombre y descripción) de una zona del sistema.
- La longitud máxima para ambas cadenas de texto las determinan las constantes:

```
NEO240EN.MAX_NAME_LEN  
NEO240EN.MAX_DESC_LEN
```

- Por ejemplo, para establecer nombre y descripción de la zona del sistema con identificador 1:

```
res = neo.foSetZoneExtraInfo(1, "Zona 1", "Descripción Zona 1")
```

- Posibles valores de retorno:
 - res("state") = NEO_RESULT.OK
 - res("state") = NEO_RESULT.ERROR_NAME_TOO_LONG
Nombre demasiado largo.
 - res("state") = NEO_RESULT.ERROR_DESCRIPTION_TOO_LONG
Descripción demasiado larga.
 - res("state") = NEO_RESULT.ERROR_ITEM_DOESNT_EXIST
Identificador de zona no válido.
 - res("state") = NEO_RESULT.ERROR_NOT_CONNECTED
Sin conexión con el equipo.
 - res("state") = NEO_RESULT.ERROR_FORBIDDEN_OPERATION_IN_EMG_STATE
El sistema está en Emergencia, operación no permitida.
 - res("state") = NEO_RESULT.ERROR_SYSTEM_BUSY
Sistema ocupado procesando otra llamada a esta misma función.
 - res("state") = NEO_RESULT.UNKNOWN_ERROR
Error no especificado.

14. Control y Monitorización de Fuentes

```
Public Function foGetSourceVumeter(
    ByVal sourceId As UShort,
    ByRef vum As Sbyte
) As Collection
```

- Lee el vúmetro de una fuente del sistema. La lectura periódica de los vúmetros debe haberse inicializado con la función `startVumetersRead`.
- Por ejemplo, para guardar en la variable `fVum` el valor del vúmetro de la fuente 1:

```
res = neo.foGetSourceVumeter(1, fVum)
```

- Posibles valores de retorno:
 - `res("state") = NEO_RESULT.OK`
 - `res("state") = NEO_RESULT.ERROR_NOT_CONNECTED`
Sin conexión con el equipo.
 - `res("state") = NEO_RESULT.ERROR_FORBIDDEN_OPERATION`
El equipo es un NEO Extension.
 - `res("state") = NEO_RESULT.ERROR_INVALID_OPERANDS`
Identificador de zona no válido.
 - `res("state") = NEO_RESULT.UNKNOWN_ERROR`
Error no especificado.

```
Public Function foSetSourceVolume(  
    ByVal channel As Byte,  
    ByVal vol As Sbyte,  
    Optional ByVal waitDataOnDate As Boolean = False)  
As Collection
```

- Establece el volumen de una fuente del sistema.
- Los valores máximo y mínimo que puede tener el volumen de una fuente son:

```
NEO240EN.ZONE_MANAGER_VOLUME_LIMITS.INPUT_MAX  
NEO240EN.ZONE_MANAGER_VOLUME_LIMITS.INPUT_MIN
```

- El parámetro `channel` hace referencia al identificador de la fuente de audio del sistema.
- El parámetro `waitDataOnDate` indica si se quiere que la función realice una espera hasta que los cambios de rutado se hayan realizado en el equipo y se hayan sincronizado en la DLL.
- Por ejemplo, para cambiar el volumen de la fuente de audio 3:

```
res = neo.foSetSourceVolume(3, -25)
```

- Posibles valores de retorno:
 - `res("state") = NEO_RESULT.OK`
 - `res("state") = res("state") = NEO_RESULT.ERROR_NOT_CONNECTED`
Sin conexión con el equipo.
 - `res("state") = NEO_RESULT.ERROR_FORBIDDEN_OPERATION_IN_EMG_STATE`
El sistema está en Emergencia, operación no permitida.
 - `res("state") = NEO_RESULT.ERROR_ITEM_DOESNT_EXIST`
La fuente especificada no existe.
 - `res("state") = NEO_RESULT.ERROR_INVALID_VOLUME`
Volumen incorrecto, fuera de límites.
 - `res("state") = NEO_RESULT.ERROR_TIMEOUT`
Se ha superado el timeout de espera para la sincronización de datos de la función cuando se le pasa el parámetro `waitDataOnDate` con el valor `True`.

```
Public Function foSetSourceMute(  
    ByVal channel As Byte,  
    ByVal mute As Boolean,  
    Optional ByVal waitDataOnDate As Boolean = False  
) As Collection
```

- Establece el mute de una fuente del sistema.
- El parámetro `waitDataOnDate` indica si se quiere que la función realice una espera hasta que los cambios de rutado se hayan realizado en el equipo y se hayan sincronizado en la DLL.
- Por ejemplo, queremos quitar “desmutear” la fuente 2 y “mutear” la fuente 3:

```
res = neo.foSetSourceMute(2, False)  
res = neo.foSetSourceMute(3, True)
```

- Posibles valores de retorno:
 - `res("state") = NEO_RESULT.OK`
 - `res("state") = res("state") = NEO_RESULT.ERROR_NOT_CONNECTED`
Sin conexión con el equipo.
 - `res("state") = NEO_RESULT.ERROR_FORBIDDEN_OPERATION_IN_EMG_STATE`
El sistema está en Emergencia, operación no permitida.
 - `res("state") = NEO_RESULT.ERROR_ITEM_DOESNT_EXIST`
La fuente especificada no existe.
 - `res("state") = NEO_RESULT.ERROR_TIMEOUT`
Se ha superado el timeout de espera para la sincronización de datos de la función cuando se le pasa el parámetro `waitDataOnDate` con el valor `True`.

```
Public Function foSetSourceExtraInfo(  
    ByVal sourceId As Byte,  
    ByVal name As String,  
    ByVal desc As String  
) As Collection
```

- Estable nombre y descripción de una fuente.
- La longitud máxima para ambas cadenas de texto las determinan las constantes:

```
NEO240EN.MAX_NAME_LEN  
NEO240EN.MAX_DESC_LEN
```

- Por ejemplo, para establecer nombre y descripción de la fuente del sistema con identificador 1:

```
res = neo.foSetSourceExtraInfo(1, "Fuente 1", "Descripción Fuente 1")
```

- Posibles valores de retorno:

- res("state") = NEO_RESULT.OK
- res("state") = NEO_RESULT.ERROR_NAME_TOO_LONG
Nombre demasiado largo.
- res("state") = NEO_RESULT.ERROR_DESCRIPTION_TOO_LONG
Descripción demasiado larga.
- res("state") = NEO_RESULT.ERROR_ITEM_DOESNT_EXIST
La fuente especificada no existe.
- res("state") = NEO_RESULT.ERROR_NOT_CONNECTED
Sin conexión con el equipo.
- res("state") = NEO_RESULT.ERROR_FORBIDDEN_OPERATION_IN_EMG_STATE
El sistema está en Emergencia, operación no permitida.
- res("state") = NEO_RESULT.ERROR_SYSTEM_BUSY
Sistema ocupado procesando otra llamada a esta misma función.
- res("state") = NEO_RESULT.UNKNOWN_ERROR
Error no especificado.

15. Inicio / Detención de la lectura de vúmetros

`Public Sub startVumetersRead(ByVal read_mode_local As Boolean)`

- Inicia la lectura periódica de vúmetros.
- Los valores del parámetro `read_mode_local` indican:
 - True = Se leerán todos los vúmetros locales del equipo, pero no los del sistema distribuido.
 - False = Se leerán los vúmetros de las entradas y salidas a nivel de sistema.

`Public Sub stopVumetersRead()`

- Detiene la lectura periódica de vúmetros.

16. Fecha y Hora del Equipo

`Public Function foSetTimeAutoUpdateEnable(ByVal enable As Boolean) As Collection`

- Habilita o deshabilita la actualización automática de fecha y hora del equipo a partir de la fecha y hora del ordenador local.
- Esta opción está habilitada por defecto. Si no queremos modificar fecha y hora del equipo, deberemos deshabilitar esta característica antes de conectarnos:

```
neo = New NEO240EN()
neo.foSetTimeAutoUpdateEnable(False)
neo.foConnect(neoIP)
```

- Posibles valores de retorno:
 - `res("state") = NEO_RESULT.OK`

`Public Function foGetDeviceTime(ByRef deviceDateTime As DateTime) As Collection`

- Obtiene la fecha y hora actual del equipo.
- Posibles valores de retorno:
 - `res("state") = NEO_RESULT.OK`
 - `res("state") = NEO_RESULT.ERROR_NOT_CONNECTED`
Sin conexión con el equipo.
 - `res("state") = NEO_RESULT.ERROR_UNSUPPORTED_OPERATION_IN_FW_VER`
Función no compatible con la versión de firmware del equipo.
 - `res("state") = NEO_RESULT.ERROR_TIMEOUT`
No se ha recibido respuesta del equipo.

`Public Function foSetTime(ByVal dateTime As DateTime) As Collection`

- Establece la fecha y hora del equipo.
- Posibles valores de retorno:
 - `res("state") = NEO_RESULT.OK`
 - `res("state") = NEO_RESULT.ERROR_NOT_CONNECTED`
Sin conexión con el equipo.

17. Supervisión de Líneas de Altavoces

No es posible leer los parámetros de supervisión de líneas de altavoces de un NEO Extension a través del NEO Controlador del Sistema, por lo que en este caso será necesario emplear una nueva instancia de la DLL para cada NEO Extension perteneciente al sistema.

Los valores de supervisión EOL los encontramos en:

```
neo.NeoConfig.lineSupervisor.eol
```

Este campo es un array con tantas posiciones como salidas tenga el equipo referenciado. El primer elemento se corresponde con la línea 1 y así sucesivamente.

PARÁMETRO	TIPO	DESCRIPCIÓN
<code>neo.NeoConfig.lineSupervisor.eol(i).enableA</code>	Boolean	Nos dice si la supervisión EOL A está habilitada
<code>neo.NeoConfig.lineSupervisor.eol(i).enableB</code>	Boolean	Nos dice si la supervisión EOL B está habilitada
<code>neo.NeoConfig.lineSupervisor.eol(i).statusA</code>	EOL_STATUS_CODES	Estado de supervisión EOL A
<code>neo.NeoConfig.lineSupervisor.eol(i).statusB</code>	EOL_STATUS_CODES	Estado de supervisión EOL A

EOL_STATUS_CODES es tipo enumerado que puede tomar los siguientes valores:

<code>NEO240EN.EOL_STATUS_CODES.OK</code>	Estado de línea correcto.
<code>NEO240EN.EOL_STATUS_CODES.OPEN_CIRCUIT</code>	Circuito Abierto
<code>NEO240EN.EOL_STATUS_CODES.SHORT_CIRCUIT</code>	Cortocircuito
<code>NEO240EN.EOL_STATUS_CODES.NOT_INSTALLED</code>	Supervisión no habilitada
<code>NEO240EN.EOL_STATUS_CODES.LINE_FAILURE</code>	Fallo de línea.

Para los valores de supervisión de impedancia de líneas de altavoces tenemos:

- `neo.NeoConfig.ampSupervisionConfig.lineZmeasEnable`

Array de booleanos de longitud igual al número de salidas del equipo donde cada posición indica si está o no habilitada la supervisión de impedancia para las líneas de altavoces asociadas.

- `neo.ampSupervisionData.ch(i).statusFlags`

Valor de tipo `AMP_CH_SUPERVISION_FLAGS` que nos indica el estado de la línea en cuanto a la supervisión de impedancia de la misma. El array `neo.ampSupervisionData.ch` tiene tantos elementos como salidas tenga el equipo. El tipo `AMP_CH_SUPERVISION_FLAGS` es un tipo enumerado que puede tomar los siguientes valores:

<code>NEO240EN.AMP_CH_SUPERVISION_FLAGS.AMP_CH_STATUS_MASK</code>	Valor de lectura de impedancia desconocido.
<code>NEO240EN.AMP_CH_SUPERVISION_FLAGS.AMP_CH_STATUS_INVALID</code>	Valor de lectura de impedancia no válido.
<code>NEO240EN.AMP_CH_SUPERVISION_FLAGS.AMP_CH_STATUS_CA</code>	Circuito Abierto
<code>NEO240EN.AMP_CH_SUPERVISION_FLAGS.AMP_CH_STATUS_CC</code>	Cortocircuito
<code>NEO240EN.AMP_CH_SUPERVISION_FLAGS.AMP_CH_STATUS_OK</code>	Valor de lectura de impedancia correcto.

18. Supervisión de Alimentación

PARÁMETRO	TIPO	DESCRIPCIÓN
<code>neo.NeoConfig.powerConfig.main_supervision</code>	Boolean	Nos dice si la supervisión de Alimentación Principal está habilitada True → Sí
<code>neo.NeoConfig.powerConfig.batt_supervision</code>	Boolean	Nos dice si la supervisión de Batería está habilitada True → Sí
<code>neo.NeoConfig.powerConfig.chg_supervision</code>	Boolean	Nos dice si la supervisión de Cargador está habilitada True → Sí
<code>neo.NeoConfig.powerConfig.main_error</code>	Boolean	Indica error en Alimentación Principal True → Error
<code>neo.NeoConfig.powerConfig.batt_error</code>	Boolean	Indica error en Batería True → Error
<code>neo.NeoConfig.powerConfig.chg_error</code>	Boolean	Indica error en Cargador True → Error

19. Ejemplos de Uso

Creación de objeto NEO:

```
Dim WithEvents neo As NEO240EN
neo = New NEO240EN()
```

Secuencia de establecimiento de conexión con el sistema NEO:

1. Conectar con equipo con la dirección ip 192.168.0.100:

```
res = neo.foConnect("192.168.0.100")
```

- Si la IP es válida: `res("state") = 1`

2. Esperar a que el flag `neo.Connected` sea igual a `True`.

- Para detectar si se produce una desconexión con el equipo habrá que monitorizar este flag.

3. Esperar a que el flag `neo.OnDate` sea `True` para confirmar que en la DLL ya tenemos los parámetros del equipo actualizados. Para esto también podemos capturar el evento `neo.data_On_Date` que se lanza cuando la DLL actualiza los parámetros del equipo por primera vez y posteriormente cada vez que se modifique algún parámetro del sistema.

4. Ya podemos empezar a monitorizar y controlar el sistema NEO.

Rutado de fuentes a zonas en un sistema NEO:

Supongamos que se emplea la entrada de audio física número 1 de un sistema NEO para nuestras fuentes de audio y queremos emitir ese audio por la zona 3.

1. En primer podemos mirar los parámetros de rutado de la zona accediendo al array que contiene la información de las zonas del sistema, concretamente a la tercera posición del array (zona 3), teniendo en cuenta que los índices del array empiezan desde 0:

```
neo.NeoConfig.zm.zones(2).source
neo.NeoConfig.zm.zones(2).effectiveSource
```

a) Si en ambos campos ya tenemos el valor 1 quiere decir que la fuente 1 ya está asignada a la zona y no será necesario rutarla.

b) Si en '`source`' tenemos la fuente 1 y en '`effectiveSource`' tenemos otra fuente distinta, querrá decir que aun estando rutada la fuente 1 existe un rutado temporal en la zona y que eventualmente se volverá a recuperar el rutado efectivo de la fuente 1. Esto puede deberse a:

- * Un micrófono del sistema NEO tiene concedida 'palabra' sobre la zona.
- * La zona ha entrado en emergencia. En este caso nos puede ser útil el flag:

```
neo.NeoConfig.zm.zones(2).emgActive
```

- * Un evento del sistema NEO, con la opción 'deshacer cambios' activa, realiza una acción de rutado.

c) Si en el campo `source` tenemos una fuente distinta a la fuente 1 deberemos rutarla a la zona para poder emitir audio por ella.

Aspectos a tener en cuenta:

- Si el sistema NEO se encuentra en estado de Emergencia no se permiten cambios de rutado.
- Cambios de rutado sobre zonas desarmadas no tienen efecto.
- Si en la zona hay un rutado temporal correspondiente a un micrófono, es posible rutar nuestra fuente pero el rutado no será efectivo (`source = effectiveSource`) mientras el micrófono tenga concedida palabra sobre la zona.

2. Si tras la comprobación de los parámetros de la zona 3 se hace necesario rutar la fuente 1 lo haremos con:

```
res = neo.foSetZoneRoute(3, 1)
```

- Si `res("state") = NEO_RESULT.OK` quiere decir que no se han detectado errores y la llamada a la función es correcta.

- Esperar a que el flag `neo.OnDate` se ponga a `True` o a que salte el evento `neo.data_On_Date()` y comprobar que el rutado se ha efectuado correctamente.

3. Ahora podemos monitorizar la fuente para detectar cambios de rutado. Para comprobar si se producen cambios de rutado en la zona habrá que monitorizar el flag '`neo.OnDate`' o controlar el evento '`neo.data_On_Date`'.

- Cuando se produzca el evento `neo.data_On_Date` o se detecte una transición `False` → `True` en el flag `neo.OnDate` ya sabremos que se ha producido un cambio en algún parámetro del equipo, momento en el que podemos monitorizar el rutado de las zonas para determinar si es ahí donde se ha producido el cambio.

- Mientras que los valores `source` y `effectiveSource` de la zona sean igual a 1 (la fuente que hemos rutado) querrá decir que nuestro rutado se mantiene, con lo que el audio que enviemos a esa entrada 1 será emitido por la zona 3.

- Si en `effectiveSource` vemos una fuente distinta a la fuente 1 mientras que el campo `source` mantiene dicha fuente podemos determinar que se ha producido un cambio de rutado temporal.

- Si en cambio es en el campo `source` donde detectamos el cambio de fuente rutada es porque se ha realizado un rutado fijo y deberemos realizar explícitamente el rutado de la fuente 1 si queremos recuperarla. Este cambio puede deberse a:

- * Un rutado explícito de otra fuente.
- * Un evento del sistema NEO, con la opción 'deshacer cambios' no activa, realiza una acción de rutado.

Ejemplo básico de rutado de una fuente por la que se desea emitir un mensaje:

1. Si no tenemos la fuente (o fuentes) rutada a la zona (o zonas) seleccionadas deberemos realizar la operación de rutado con:

- Fuente 1 a zona 3:
`res = neo.foSetZoneRoute(3, 1)`

ó

- Fuente 1 a las zonas 1, 2 y 3:
`res = neo.foSetZoneMultiRoute({1,2,3}, {1,1,1})`

* Si el resultado de la llamada a la función es correcto, esperamos por el flag `neo.OnDate` o por el evento `neo.data_On_Date` y a continuación comprobamos si el rutado se ha realizado satisfactoriamente.

2. Si en la zona o zonas tenemos ya rutada la fuente deseada podemos empezar la emisión del mensaje y a continuación empezar a monitorizar las zonas implicadas para comprobar si se producen cambios de rutado y actuar en consecuencia (como dar el mensaje por interrumpido).

3. Si finaliza la emisión del mensaje sin haberse producido cambios de rutado ajenos a nuestra acción podemos determinar que la emisión ha sido correcta. Llegado este caso podríamos:

a) Eliminar el rutado realizado:

- Ninguna fuente a la zona 3:
`res = neo.foSetZoneRoute(3, 0)`

ó

- Ninguna fuente a las zonas 1, 2 y 3:
`res = neo.foSetZoneMultiRoute({1,2,3}, {0,0,0})`

b) Rutar otra fuente distinta que hubiera previamente a la emisión del mensaje:

- Fuente 2 a la zona 3:
`res = neo.foSetZoneRoute(3, 2)`

ó

- Fuente 2 a la zona 1, Fuente 3 a la zona 2 y Fuente 4 a la zona 3
`res = neo.foSetZoneMultiRoute({1,2,3}, {2,3,4})`

c) Mantener el rutado por si queremos emitir otro mensaje, por ejemplo.

