

# LUA scripting

## NXA standard functions

Las Funciones estándar NXA proporcionan las Funcionalidades básicas que cualquier función LUA puede utilizar en cualquier momento durante la ejecución del BMS Server. En general, estas Funciones suelen cambiar valores de ítems en el servidor o Funciones que, cuando invocadas, recuperan el estado actual del workspace.

Las Funciones del estándar NXA son, a saber:

● **`nxa.IsInitialized( )`**

Determina el estado del servidor, indicando si está encendido o no.

Parámetros: (No aplicable)

Devuelve: boolean es true si lo está inicializado o a false si no.

Ej: `nxa.LogInfo(nxa.IsInitialized())`

● **`nxa.IsRunning( )`**

Indica si el servidor está Funcionando

Parámetros: (No aplicable)

Devuelve: boolean a true si está funcionando o a false si no.

Ej: `nxa.LogInfo(nxa.IsRunning())`

● **`nxa.IsSimulation( )`**

Indica si el servidor está en modo simulación o no.

Parámetros: (No aplicable)

Devuelve: boolean a true si se cumple, o a false si no.

Ej: `nxa.LogInfo(nxa.IsSimulation())`

● **`nxa.IsActiveServer( )`**

Determina si el servidor está activo.

Parámetros: (No aplicable)

Devuelve: boolean a true si está activo, o a false si no.

Ej: `nxa.LogInfo(nxa.IsActiveServer())`

● **`nxa.IsMainServer( )`**

Determina si el servidor está definido como servidor principal.

Parámetros: (No aplicable)

Devuelve: boolean a true si se cumple, o a false si no.

Ej: `nxa.LogInfo(nxa.IsMainServer())`

### ● **nxa.IsBackupServer( )**

Sirve para determinar si el servidor está definido como servidor backup o no.

Parámetros: (No aplicable)

Devuelve: boolean a true si es cierto, o a false si no.

Ej: nxa.LogInfo(nxa.IsBackupServer())

The screenshot shows the 'System Messages' window in SQL Server Enterprise Manager. The table below represents the data shown in the messages window:

Type	Date/Time	Module	Message
INFO	29/05/15 17:03:54.667	LUA_ENGINE	Server inicializado:
INFO	29/05/15 17:03:54.667	LUA_ENGINE	TRUE
INFO	29/05/15 17:09:12.373	LUA_ENGINE	Servidor inicializado:
INFO	29/05/15 17:09:12.373	LUA_ENGINE	TRUE
INFO	29/05/15 17:09:12.373	LUA_ENGINE	Servidor corriendo:
INFO	29/05/15 17:09:12.373	LUA_ENGINE	TRUE
INFO	29/05/15 17:09:12.373	LUA_ENGINE	Servidor en modo simulación:
INFO	29/05/15 17:09:12.373	LUA_ENGINE	FALSE
INFO	29/05/15 17:09:12.373	LUA_ENGINE	Servidor activo:
INFO	29/05/15 17:09:12.373	LUA_ENGINE	TRUE
INFO	29/05/15 17:09:12.373	LUA_ENGINE	Servidor principal:
INFO	29/05/15 17:09:12.373	LUA_ENGINE	FALSE
INFO	29/05/15 17:09:12.373	LUA_ENGINE	Servidor backup:
INFO	29/05/15 17:09:12.373	LUA_ENGINE	FALSE

Overlaid on the right is the 'Execute Script' dialog box with the following script:

```
1 nxa.LogInfo("Servidor inicializado:")
2 nxa.LogInfo(nxa.IsInitialized())
3 nxa.LogInfo("Servidor corriendo:")
4 nxa.LogInfo(nxa.IsRunning())
5 nxa.LogInfo("Servidor en modo simulación:")
6 nxa.LogInfo(nxa.IsSimulation())
7 nxa.LogInfo("Servidor activo:")
8 nxa.LogInfo(nxa.IsActiveServer())
9 nxa.LogInfo("Servidor principal:")
10 nxa.LogInfo(nxa.IsMainServer())
11 nxa.LogInfo("Servidor backup:")
12 nxa.LogInfo(nxa.IsBackupServer())
13
```

--Se puede observar el uso de la función LogInfo, explicada más adelante--

### ● **nxa.WorkspaceName( )**

Esta función devuelve el nombre actual del workspace.

Parámetros: (No aplicable)

Devuelve: string con el nombre del workspace.

Ej: nxa.LogInfo(nxa.WorkspaceName())

### ● **nxa.RootPath( )**

La función devuelve la ruta donde se ha instalado el server.

Parámetros: (No aplicable)

Devuelve: string con la ruta.

Ej: nxa.LogInfo(nxa.RootPath())

### ● **nxa.WorkspacePath( )**

La función devuelve la ruta donde se encuentra el workspace.

Parámetros: (No aplicable)

Devuelve: string con la ruta.

Ej: nxa.LogInfo(nxa.WorkspacePath())

### **nxa.ScriptFilesPath( )**

Esta función retorna la ruta donde encuentra el directorio del script en el workspace actual.

Parámetros: (No aplicable)

Devuelve: string con la ruta.

Ej: nxa.LogInfo(nxa.ScriptFilesPath())

### **nxa.ConfigFilesPath( )**

Esta función devuelve la ruta al fichero de configuración del actual workspace.

Parámetros: (No aplicable)

Devuelve: string con la ruta.

Ej: `nxa.LogInfo(nxa.ConfigFilesPath())`

### **nxa.DataFilesPath( )**

La función devuelve la ruta al directorio del fichero de datos del actual workspace.

Parámetros: (No aplicable)

Devuelve: string con la ruta.

Ej: `nxa.LogInfo(nxa.DataFilesPath())`

- **nxa.LogFilesPath( )**

La función devuelve la ruta del directorio de historial del actual workspace.

Parámetros: (No aplicable)

Devuelve: string con la ruta.

Ej: `nxa.LogInfo(nxa.LogFilesPath())`

- **nxa.ProjectFilesPath( )**

Devuelve la ruta al directorio del proyecto del actual workspace que contenga los archivos de proyecto Smart Voyager (\*.vxf).

Parámetros: (No aplicable)

Devuelve: string con la ruta.

Ej: `nxa.LogInfo(nxa.ProjectFilesPath())`

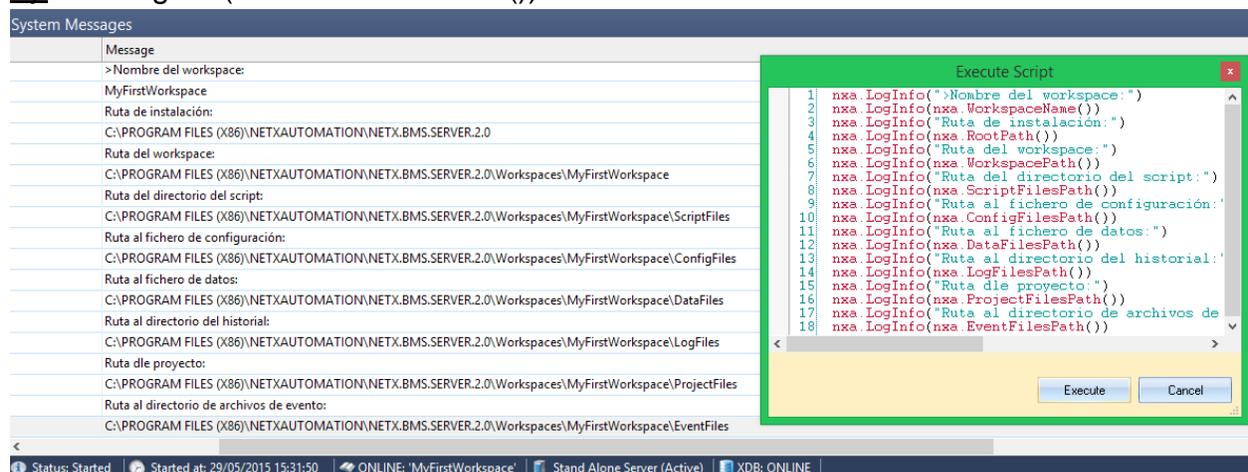
- **nxa.EventFilesPath( )**

La función devuelve la ruta al directorio del archivo de evento del actual workspace.

Parámetros: (No aplicable)

Devuelve: string con la ruta.

Ej: `nxa.LogInfo(nxa.EventFilesPath())`



- **nxa.LogInfo( )**

La función genera un mensaje de información almacenado en el historial del sistema.

Parámetros: string con el mensaje.

Devuelve: (No aplicable)

Ej: `nxa.LogInfo("Mensaje de prueba")`

- **nxalogs.LogWarning( )**

Genera un mensaje de warning que es almacenado en el historial del sistema.

Parámetros: string con el mensaje.

Devuelve: (No aplicable)

Ej: nxalogs.LogWarning("Mensaje de warning")

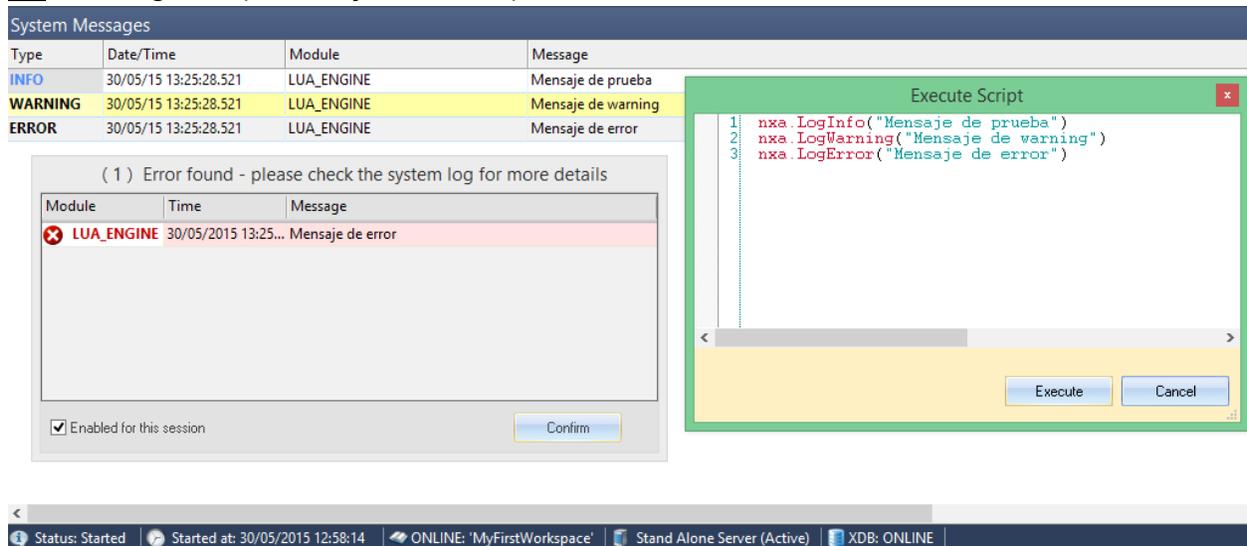
- **nxalogs.LogError( )**

Esta función genera un mensaje de error que es almacenado en el historial del sistema.

Parámetros: string con el mensaje.

Devuelve: (No aplicable)

Ej: nxalogs.LogError("Mensaje de error")



- **nxalogs.SpinTelegramLog( )**

Esta función fuerza a almacenar el historial de mensajes actual y crea uno nuevo.

Parámetros: (No aplicable)

Devuelve: boolean que indica el éxito o fracaso de la operación.

Ej: if nxalogs.SpinTelegramLog() then  
    nxalogs.LogInfo("Exito")  
end

- **nxalogs.SpinSystemLog( )**

Esta función fuerza a archivar el actual documento de historial y crea uno nuevo.

Parámetros: (No aplicable)

Devuelve: boolean que indica el éxito o fracaso de la operación.

Ej: if nxalogs.SpinSystemLog() then  
    nxalogs.LogInfo("Exito")  
end

System Messages			
Type	Date/Time	Module	Message
INFO	30/05/15 13:40:34.572	LUA_ENGINE	Éxito
INFO	30/05/15 13:40:34.572	LUA_ENGINE	Éxito

Execute Script

```

1  if nxa.SpinTelegramLog() then
2  nxa.LogInfo("Éxito")
3  end
4
5  if nxa.SpinSystemLog() then
6  nxa.LogInfo("Éxito")
7  end

```

Status: Started | Started at: 30/05/2015 12:58:14 | ONLINE: 'MyFirstWorkspace' | Stand Alone Server (Active) | XDB: ONLINE

- **nxa.GetLastErrorText()**

La función devuelve el último texto de error como string.

Parámetros: (No aplicable)

Devuelve: string con el error.

Ej: nxa.LogInfo(nxa.GetLastErrorText())

- **nxa.GetLastErrorCode()**

La función devuelve el código del último error como integer.

Parámetros: (No aplicable)

Devuelve: string con el error.

Ej: nxa.LogInfo(nxa.GetLastErrorCode())

System Messages			
Type	Date/Time	Module	Message
INFO	30/05/15 13:45:10.586	LUA_ENGINE	
INFO	30/05/15 13:45:10.586	LUA_ENGINE	0

Execute Script

```

1  nxa.LogInfo(nxa.GetLastErrorText())
2  nxa.LogInfo(nxa.GetLastErrorCode())

```

Status: Started | Started at: 30/05/2015 12:58:14 | ONLINE: 'MyFirstWorkspace' | Stand Alone Server (Active) | XDB: ONLINE

--Debido a la falta de errores, el primer método devuelve un string vacío y el segundo un 0--

- **nxa.GetValue( ) | nxa.Value( )**

Esta función solicita el valor actual de un Server Item. Si el estado del ítem es 'UNCERTAIN' (incierto), devolverá el valor por defecto.

Hay que destacar que la función lee el valor actual del ítem almacenado en el servidor (no enviará una petición de lectura para el campo del dispositivo requerido).

Parámetros: string con el itemID del Server Item.  
tipo con valor por defecto.

Devuelve: tipo con el actual valor del ítem.

Ej: nxa.LogInfo(nxa.GetValue("NETx\\VAR\\Boolean\\Item1",0))

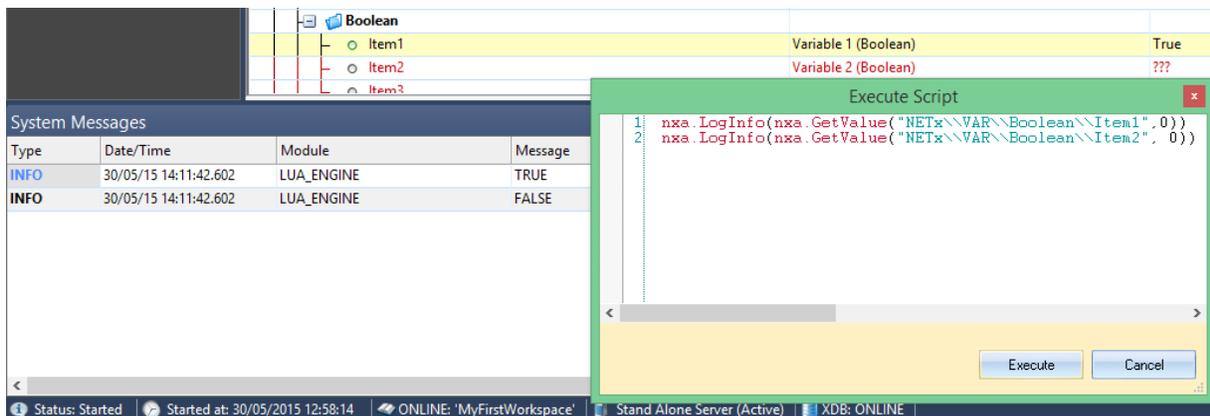
- **nxa.SetValue( )**

La función actualiza el valor del Server Item especificado. Se puede especificar un tiempo de retraso. El valor sólo se actualizará en el servidor (no lo hará en el campo del dispositivo requerido).

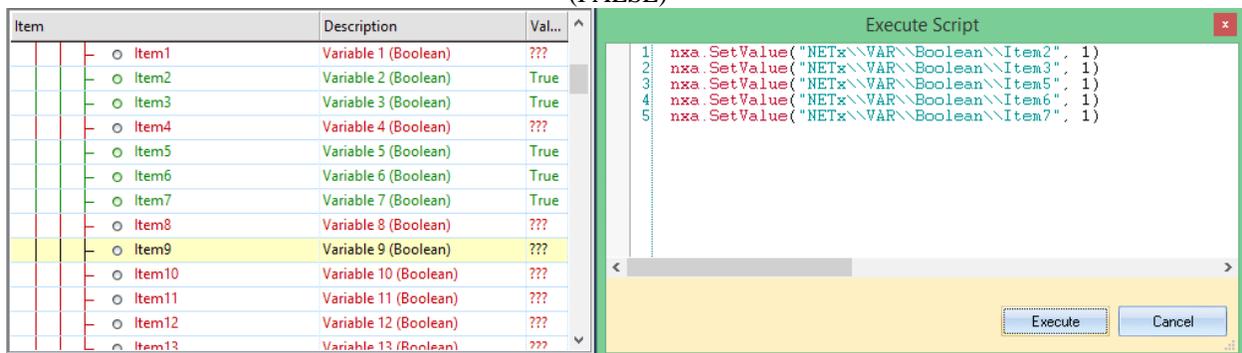
Parámetros: string con el itemID del ítem a actualizar.  
tipo con el valor que actualizará el ítem.  
number con el retraso (en milisegundos).

Devuelve: (No aplicable)

Ej: nxa.LogInfo(nxa.SetValue("NETx\\VAR\\Boolean\\Item1", 1))



-Debido a que el estado de la segunda variable es UNCERTAIN, el resultado obtenido es el valor por defecto, 0 (FALSE)--



- **nxa.SetItemData( )**

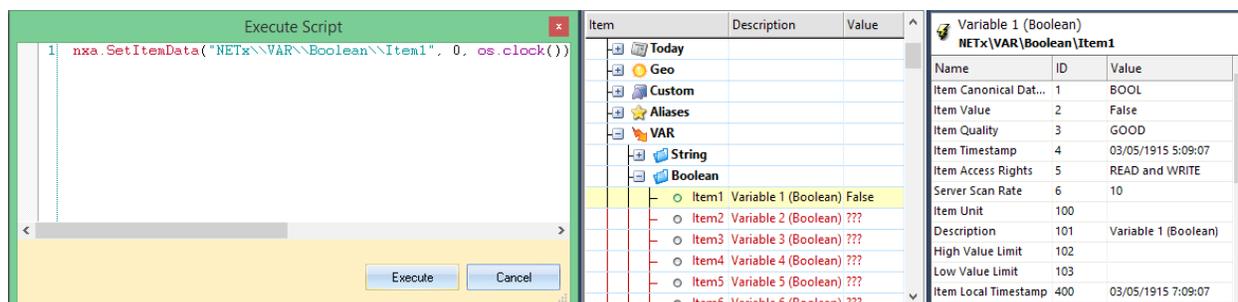
nxa Función responsable por poner un valor específico a un ítem en el servidor.

Similar a la Función `nxa.SetValue`, donde se puede configurar la información de origen y del tiempo.

Parámetros: string - ID del ítem que debe ser inicializado  
variable - nuevo valor que se debe inicializar  
date - información temporal del valor. Grabado en el histórico de la BD  
string - origen de la información

Devuelve: (No aplicable)

Ej: `nxa.SetItemData("NETx\\VAR\\Boolean\\Item1", 0, os.clock(), "SYS:LUA;SRC:MyDevice")`



--os.clock() devuelve el tiempo que el programa lleva en ejecución, de ahí esta marca temporal extraña--

### ● `nxa.ResetItemValue( )`

Función responsable de borrar un valor específico de un ítem en el servidor.

Pone el valor a "UNCERTAIN".

Parámetros: string - ID del ítem que debe ser reiniciado  
date - información temporal del valor. Grabado en el histórico de la BD  
string - origen de la información

Devuelve: (No aplicable)

Ej: `nxa.LogInfo(nxa.ResetItemValue("NETx\\VAR\\Boolean\\Item1", os.clock(), "SYS:LUA ; SRC:MyDevice"))`

### ● `nxa.ReadValue( )`

Función que hace activamente la lectura del valor de un ítem específico.

El valor leído solo envía la lectura del valor requerido para el dispositivo del campo.

Se puede añadir un retraso opcional elegible.

Parámetros: string - ID del ítem que debe ser leído  
number - retraso en milisegundos

Devuelve: variable - valor actual del ítem

Ej: `nxa.LogInfo(nxa.ReadValue("NETx\\VAR\\Numeric\\Item1", 500))`

### ● `nxa.WriteValue( )`

Función que hace activamente la escritura del valor de un ítem específico.

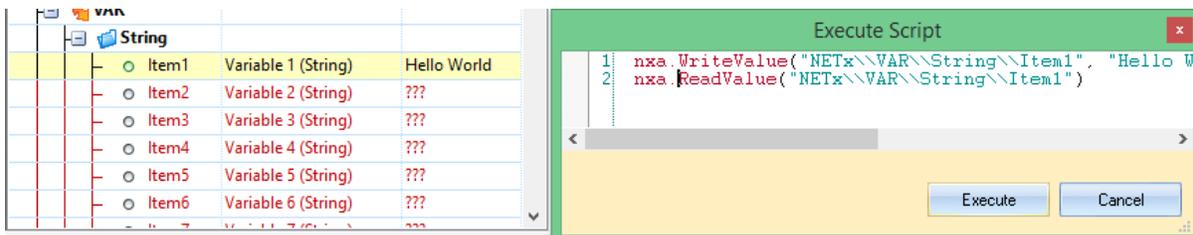
El valor leído solo propaga la petición para el dispositivo de campo.

Se puede añadir un retraso opcional elegible.

Parámetros: string - ID del ítem que debe ser leído  
variable - valor del ítem a ser escrito  
number - retraso en milisegundos

Devuelve: (No aplicable)

Ej: `nxa.LogInfo(nxa.WriteValue("NETx\\VAR\\Numeric\\Item1", 123, 500))`



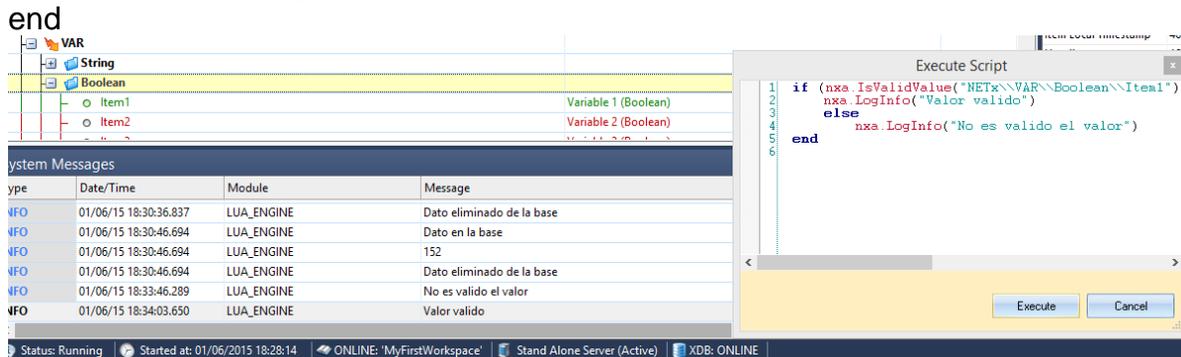
- **nxa.IsValidValue( )**

Función que verifica si el ítem tiene un valor válido.

Parámetros: string - ID del ítem que debe ser testado

Devuelve: bool - indica si el valor es correcto o no

Ej: if (nxa.IsValidValue("NETx\\VAR\\Numeric\\Item1")) then  
     nxa.LogInfo("Valor valido")  
 else  
     nxa.LogInfo("El valor no es correcto.")  
 end



- **nxa.GetPropertyValue( ) | nxa.PropertyValue( ) | nxa.Property( )**

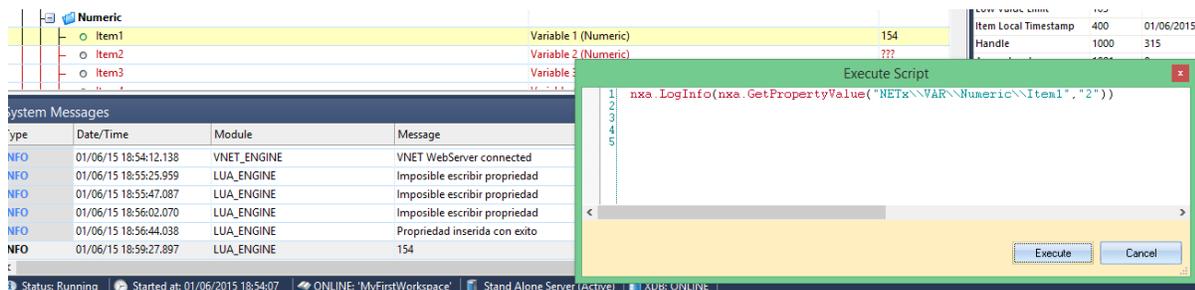
Función que lee las propiedades de un ítem en el servidor.

Parámetros: string - ID del ítem que contiene las propiedades

string - ID de la propiedad de que se lee

Devuelve: variable - valor actual de la propiedad

Ej: nxa.LogInfo(nxa.GetPropertyValue("NETx\\VAR\\Numeric\\Item1", "2"))



- **nxa.SetPropertyValue( ) | nxa.SetProperty( )**

Función que escribe un nuevo valor a las propiedades de un ítem en el servidor.  
Si no es posible escribir, el resultado es false.

Parámetros: string - ID del ítem que contiene las propiedades  
string - ID de la propiedad de que se desea escribir  
variable - nuevo valor de la propiedad

Devuelve: bool - verdadero si la operación es exitosa

Ej: `if (nxa.SetPropertyValue("NETx\\VAR\\Numeric\\Item1","2", 154)) then  
nxa.LogInfo("Propiedad inserida con éxito")  
else  
nxa.LogInfo("Imposible escribir propiedad")  
end`

The screenshot shows a software interface with a tree view on the left, a central script execution window, and a system messages table at the bottom.

**Execute Script Window:**

```
1 if (nxa.SetPropertyValue("NETx\\VAR\\Numeric\\Item1", "2", 154)) then  
2   nxa.LogInfo("Propiedad inserida con éxito")  
3 else  
4   nxa.LogInfo("Imposible escribir propiedad")  
5 end  
6  
7  
8
```

**System Messages Table:**

Type	Date/Time	Module	Message
NFO	01/06/15 18:54:12.138	VNET_SERVER_SOCKET_BASE	VNET client interface version: 4.2.0
NFO	01/06/15 18:54:12.138	VNET_ENGINE	VNET WebServer connected
NFO	01/06/15 18:55:25.959	LUA_ENGINE	Imposible escribir propiedad
NFO	01/06/15 18:55:47.087	LUA_ENGINE	Imposible escribir propiedad
NFO	01/06/15 18:56:02.070	LUA_ENGINE	Imposible escribir propiedad
NFO	01/06/15 18:56:44.038	LUA_ENGINE	Propiedad inserida con éxito

### ● **nxa.ReadValues( )**

Envía una petición KNX de lectura a todas las direcciones del grupo KNX o aquellas que además tienen activado el flag "ReadOnReconnect".

Parámetros: bool - Si es TRUE, el servidor envía la petición de lectura del KNX para el grupo KNX con flag "ReadOnReconnect"

Devuelve: (No Aplicable)

Ej: `nxa.LogInfo(nxa.ReadValues(true))`

### ● **nxa.GetItemID( ) | nxa.ItemID( )**

Función utilizada para recuperar un itemID de un ítem en el servidor.

Es buscado por la dirección dada en KNX y la dirección de IP por KNX gateway.

Parámetros: string - dirección KNX  
string - dirección IP del KNX gateway

Devuelve: string - itemID del ítem especificado en la dirección KNX y la dirección de IP del gateway de KNX

Ej: `nxa.LogInfo(nxa.GetItemID("NETx\\XIO\\KNX\\192.168.1.2", "GATEWAY"))`

### ● **xdb.SetData( )**

Función utilizada para almacenar cualquier tipo de datos persistentes.

Datos disponibles sólo después del reinicio del servidor.

Parámetros: string - Clave del valor

any - Valor almacenado de modo persistente en la base de datos

Devuelve: bool - Verdadero si lo ejecuta bien, falso si no

Ej: if (xdb.SetData("clave",152)) then  
    nxa.LogInfo("Dato en la base")  
    else  
        nxa.LogInfo("Error")  
end

### ● xdb.ResetData( )

Función utilizada para quitar una pareja clave/valor de una base de datos persistente.

Parámetros: string - Clave del valor

Devuelve: bool - Verdadero si lo elimina bien, falso si no

Ej: if (xdb.ResetData("clave")) then  
    nxa.LogInfo("Dato eliminado de la base")  
    else  
        nxa.LogInfo("Error")  
end

### ● xdb.GetData( )

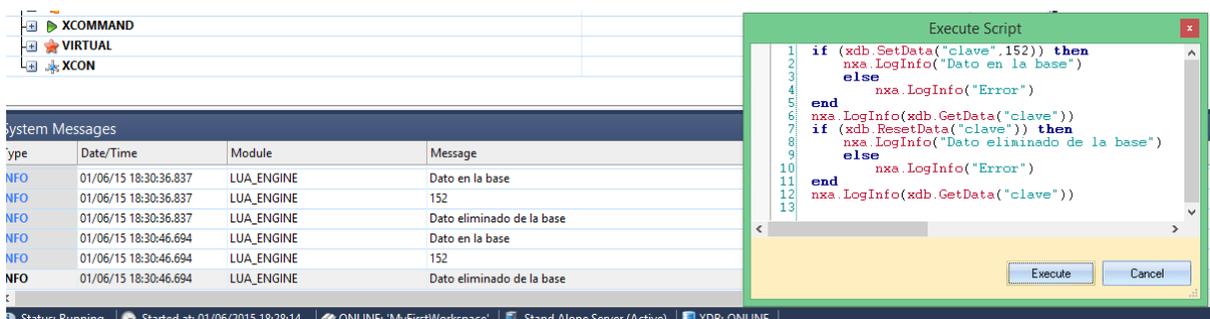
Función utilizada para recuperar un dato almacenado en la base de datos persistente.

Datos disponibles sólo después del reinicio del servidor.

Parámetros: string - Clave del valor que se quiere recuperar

Devuelve: any - Valor almacenado conjuntamente con la clave introducida

Ej: nxa.LogInfo(xdb.GetData("clave"))



### ● nxa.GetVar( )

Función que recupera el dato de la variable global por su nombre.

El dato no es persistente y es perdido después del reinicio del servidor.

Parámetros: string - Nombre de la variable

Devuelve: any - Valor almacenado en la variable global según su clave

Ej: nxa.LogInfo(nxa.GetVar("var"))

- **nx.SetVar( )**

Función que añade un valor a una variable global.

El dato no es persistente y es perdido después del reinicio del servidor.

Parámetros: string - Nombre de la variable

any - Valor a ser almacenado en la variable global

Devuelve: bool - Verdadero si añade bien, falso si no

Ej: if (nx.SetVar("var",152)) then  
    nx.LogInfo("Valor en la variable")  
else  
    nx.LogInfo("Error")  
end

- **nx.ClearVar( )**

Función utilizada para quitar una pareja clave/valor de una variable.

Parámetros: string - Nombre de la variable

Devuelve: bool - Verdadero si elimina bien, falso si no

Ej: if (nx.ClearVar("var")) then  
    nx.LogInfo("Variable eliminada")  
else  
    nx.LogInfo("Error")  
end

The screenshot shows a database management interface with a 'System Messages' table and an 'Execute Script' dialog box. The table contains the following data:

Type	Date/Time	Module	Message
NFO	01/06/15 18:14:02.756	LUA_ENGINE	TRUE
NFO	01/06/15 18:14:57.774	LUA_ENGINE	TRUE
NFO	01/06/15 18:17:22.677	LUA_ENGINE	TRUE
NFO	01/06/15 18:18:04.035	LUA_ENGINE	Error
NFO	01/06/15 18:18:28.756	LUA_ENGINE	Valor en la variable
NFO	01/06/15 18:19:03.571	LUA_ENGINE	152
NFO	01/06/15 18:19:47.459	LUA_ENGINE	Valor en la variable
NFO	01/06/15 18:19:47.459	LUA_ENGINE	152
NFO	01/06/15 18:19:47.459	LUA_ENGINE	Variable eliminada

The 'Execute Script' dialog box contains the following Lua code:

```
1 if (nx.SetVar("var",152)) then
2   nx.LogInfo("Valor en la variable")
3 else
4   nx.LogInfo("Error")
5 end
6 nx.LogInfo(nx.GetVar("var"))
7 if (nx.ClearVar("var")) then
8   nx.LogInfo("Variable eliminada")
9 else
10  nx.LogInfo("Error")
11 end
12 nx.LogInfo(nx.GetVar("var"))
```

- **nx.AddDirectLink( )**

Añade un enlace unidireccional entre la fuente y el elemento de destino.

Todas las peticiones (lectura, escritura y peticiones de asignación) se envían desde la fuente hasta el destino.

Parametros: string - Origen del ItemID

string - Destino del ItemID

Devuelve: (No aplicable)

Ej: nx.AddDirectLink("NETx\\VAR\\Boolean\\Item1","NETx\\VAR\\Boolean\\Item2")

- **nx.RemoveDirectLink( )**

Elimina un enlace unidireccional entre dos items.

Parámetros: string - Origen del ItemID

string - Destino del ItemID

Devuelve: (No aplicable)

Ej: nxa.RemoveDirectLink("NETx\\VAR\\Boolean\\Item1","NETx\\VAR\\Boolean\\Item2")

- **nxa.AddDirectBiLink( )**

Añade un enlace bidireccional entre la fuente y el elemento de destino.

Todas las peticiones (lectura, escritura y peticiones de asignación) se envían de un elemento a otro.

Parámetros: string - Item1 del ItemID

string - Item2 del ItemID

Ej: nxa.AddDirectBiLink("NETx\\VAR\\Boolean\\Item1","NETx\\VAR\\Boolean\\Item2")

- **nxa.RemoveDirectBiLink( )**

Elimina un enlace bidireccional entre dos items.

Parámetros: string - Item1 del ItemID

string - Item2 del ItemID

Ej:

nxa.RemoveDirectBiLink("NETx\\VAR\\Boolean\\Item1","NETx\\VAR\\Boolean\\Item2")

- **nxa.MakeItemShadowCopy( )**

Crea un nuevo ítem que se comporta como una copia de la original.

Todas las solicitudes (lectura, escritura y asignación de valor) se reenvían a la copia y viceversa.

Parámetros: string - ItemID de la origen del ítem

string - El nuevo ítem de copia

Ej:

nxa.MakeItemShadowCopy("NETx\\VAR\\Boolean\\Item1","NETx\\VAR\\Boolean\\Item2")