NXA functions for task handling

Contenido:

- Documentación de BMS Server (4.6.2 NXA functions for task handling)
- Tutorial de uso de las funciones

Documentación de BMS Server

4.6.2 Funciones para manejar tareas

Éste conjunto de funciones son usadas en combinación con las tareas. Algunas funciones proveen funcionalidad de ayuda que sólo pueden ser llamadas dentro de las funciones LUA que son ejecutadas por las tareas (Sección 4.5.1). Además, hay funciones incluidas que pueden ser usadas para añadir o eliminar definiciones de tareas directamente desde scripts LUA. Usarlas nos ofrece la oportunidad de crear dinámicamente tareas que no estás listadas en el archivo de definición de tareas. NOTA: la agregación o la eliminación de tareas puede ser sólo realizado durante el arranque del servidor (en el evento "OnInitEvent").

nxa.SourceItemID | nxa.SourceVar

Esta función puede ser utilizada para preguntar el ID del objeto de origen que lanzó la tarea correspondiente. Sólo puede ser usada en las funciones LUA que son invocadas por tareas.

Devuelve:

- string – ID del objeto de origen que lanzó la tarea.

nxa.DestinationItemID

Esta función puede ser utilizada para preguntar el ID del objeto de destino que está configurado para actuar con la tarea correspondiente. Sólo puede ser usada en las funciones LUA que son invocadas por tareas.

Devuelve:

- string – ID del objeto de destino relacionado con la tarea.

nxa.SetDestinationValue

Esta función puede ser usada para establecer un nuevo valor para el objeto de destino que está configurado en la correspondiente tarea. Sólo puede ser usada en funciones LUA invocadas por tareas. NOTA: el nuevo valor sólo será modificado en el servidor (no llega al dispositivo).

Devuelve:

- variant – nuevo valor del objeto de destino.

nxa.WriteDestinationValue

Esta función puede ser usada para escribir un nuevo valor para el objeto de destino que está

configurado en la correspondiente tarea. Comparado a poner un valor, un valor escrito se propagará también al campo del dispositivo. Sólo puede ser usada en funciones LUA invocadas por tareas.

Devuelve:

- variant – nuevo valor del objeto de destino.

nxa.ReadDestinationValue

Esta función puede ser usada para leer el valor actual del objeto de destino que está configurado en la correspondiente tarea. Esta función puede ser usada sólo en funciones LUA que son invocadas por tareas.

Devuelve:

- variant – valor actual del objeto de destino

nxa.InputValue | nxa.SourceValue

Esta función puede ser usada para consultar el valor del objeto de origen que ha lanzado la tarea correspondiente. Esta función sólo puede ser usada por funciones LUA que son lanzadas por tareas.

Devuelve:

- variant – valor actual del objeto de origen

nxa.ExecuteDelayedScript

Esta función ejecuta un script LUA especificado como un parámetro String. La ejecución puede ser opcionalmente retrasada.

Parameters:

- string – nombre del script LUA
- number – retraso en milisegundos (opcional)

NOTA: la agregación o la eliminación de tareas puede ser sólo realizado durante el arranque del servidor (en la función OnInitEvent()).

nxa.AddWriteTask

Esta función añade una nueva definición para una tarea de 'escritura' durante la ejecución.

Parámetros:

- string – ID del objeto de origen del servidor que lanza la tarea.
- string – ID del objeto de destino del servidor que es cambiado por la tarea.
- bool – corresponde al parámetro “OnReceive” de la definición de la tarea.
- bool – corresponde al parámetro “OnSend” de la definición de la tarea.
- bool – corresponde al parámetro “OnSet” de la definición de la tarea.
- number – intervalo de tiempo en milisegundos que la tarea será retrasada cuando se lance.
- variant – si es usado, este valor será usado en vez del valor del objeto de origen (opcional).

nxa.AddReadTask

Esta función añade una nueva definición para una tarea de 'lectura' durante el periodo ejecución.

Parámetros:

- string – ID del objeto de origen del servidor que lanza la tarea.
- string – ID del objeto de destino del servidor que es cambiado por la tarea.
- bool – corresponde al parámetro “OnReceive” de la definición de la tarea.
- bool – corresponde al parámetro “OnSend” de la definición de la tarea.
- bool – corresponde al parámetro “OnSet” de la definición de la tarea.
- number – intervalo de tiempo en milisegundos que la tarea será retrasada cuando se lance.

nxa.AddSetTask

Esta función añade una nueva definición para una tarea de 'establecer' durante el periodo ejecución.

Parámetros:

- string – ID del objeto de origen del servidor que lanza la tarea.
- string – ID del objeto de destino del servidor que es cambiado por la tarea.
- bool – corresponde al parámetro “OnReceive” de la definición de la tarea.
- bool – corresponde al parámetro “OnSend” de la definición de la tarea.
- bool – corresponde al parámetro “OnSet” de la definición de la tarea.
- number – intervalo de tiempo en milisegundos que la tarea será retrasada cuando se lance.
- variant – si es usado, este valor será usado en vez del valor del objeto de origen (opcional).

nxa.AddScriptTask

Esta función añade una nueva definición para una tarea de 'script' durante el periodo ejecución.

Parámetros:

- string – ID del objeto de origen del servidor.
- string – ID del objeto de destino del servidor.
- bool – corresponde al parámetro “OnReceive” de la definición de la tarea.
- bool – corresponde al parámetro “OnSend” de la definición de la tarea.
- bool – corresponde al parámetro “OnSet” de la definición de la tarea.
- number – intervalo de tiempo en milisegundos que la tarea será retrasada cuando se lance.
- variant – si es usado, este valor será usado en vez del valor del objeto de origen (opcional).

nxa.AddVarTask

Esta función puede ser usada para añadir una tarea que es lanzada si una variable cambia su valor.

Parámetros:

- string – nombre de la variable que será el lanzador de la tarea.
- string – nombre de la función LUA que será lanzada.
- number – retraso opcional para la función.

Devuelve:

- bool – cierto en caso de éxito, falso en cualquier otro caso.

Tutorial de uso

Se pueden añadir tareas de distintas formas, la principal es a través del menú Extensions > [Live] Task definitions. La alternativa a esto es mediante la programación de scripts mediante LUA, a través de la opción Edit Script, seleccionando 'nxaDefinitions.lua'.
Para añadir las distintas tareas (nxa.AddWriteTask, nxa.AddReadTask, nxa. AddSetTask…) se debe hacer en la función 'OnInitEvent', es la función que se ejecuta al iniciar el servidor.

```
function OnInitEvent()
  InitializeSysXCON()

  -- función que cada vez cambia los segundos del programa, se lee el Item1 de las variables String
  nxa.AddReadTask((“NETx\\Today\\Seconds”,”NETx\\VAR\\String\\Item1”, true, true, true, 0)

  --  función que cada vez que cambia los segundos del sistema, se guardan en el Item1 de las variables String el valor de origen (que lanza la tarea)
  nxa.AddWriteTask(“NETx\\Today\\Seconds”,”NETx\\VAR\\String\\Item1”, true, true, true, 0, nxa.SourceValue)

--función idéntica al write pero la variación no llega al dispositivo
 nxa.AddSetTask("NETx\\Today\\Seconds","NETx\\VAR\\String\\Item3", true, true, true, 100, nxa.SourceValue)
end
```

Además, se pueden crear scripts .lua aparte donde podemos crear nuestras funciones y poder hacer un tratamiento de datos más exhaustivos, que luego invocaremos con la función nxa.AddScriptTask dentro de la función 'OnInitEvent'.

```
require ‘script’

function OnInitEvent()
```

```
  InitializeSysXCON()

  nxa.AddScriptTask(“NETx\\String\\Item1”, “NETx\\String\\Item2”, true, true, true, 0,
scriptPrint())

end
```

Con la misma idea, se puede hacer el mismo procedimiento pero dependiendo de variables, lo que hace que cada vez que cambie la variable se ejecuta el script.

```
require ‘script’

function OnInitEvent()
  InitializeSysXCON()

  nxa.AddVarTask(“NETx\\String\\Item1”, scriptPrint(), 0)

end
```

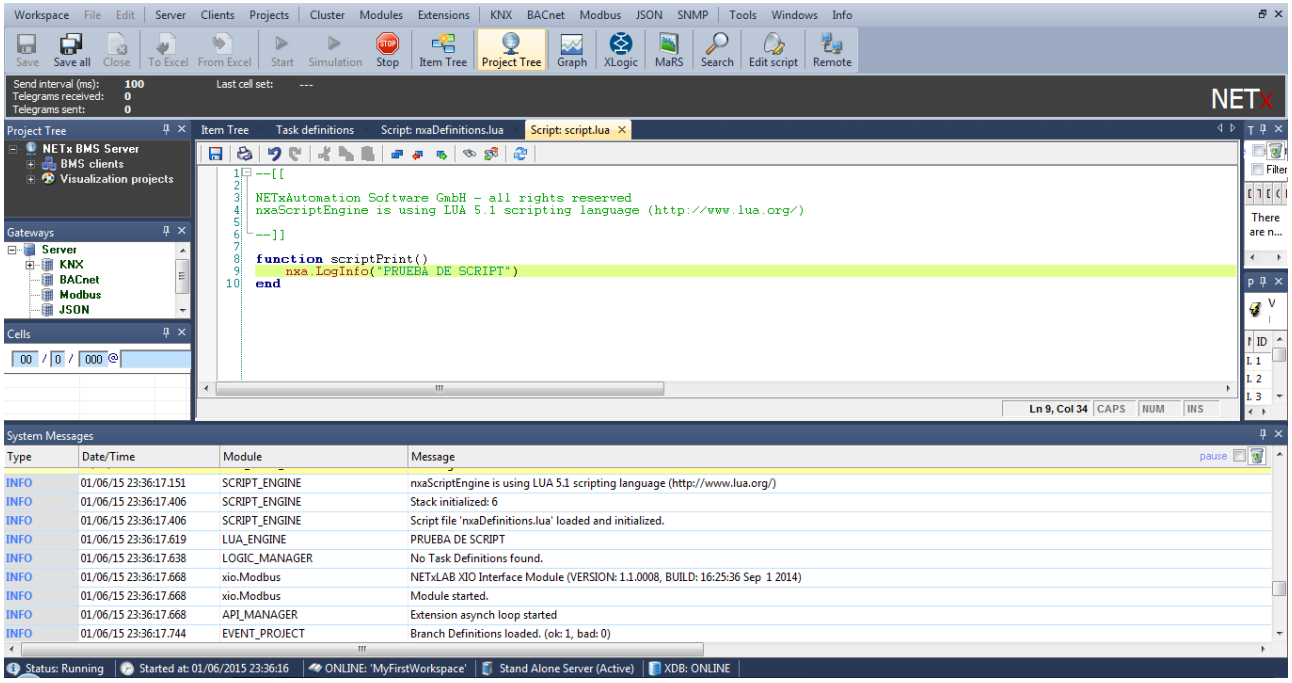
```
-- script.lua

function scriptPrint()
  -- muestra en la consola el ID del objeto de origen, se puede utilizar cualquier función aquí
  nxa.LogInfo(nxa.SourceItemID())
end
```

Las funciones se pueden subdividir en dos grupos:

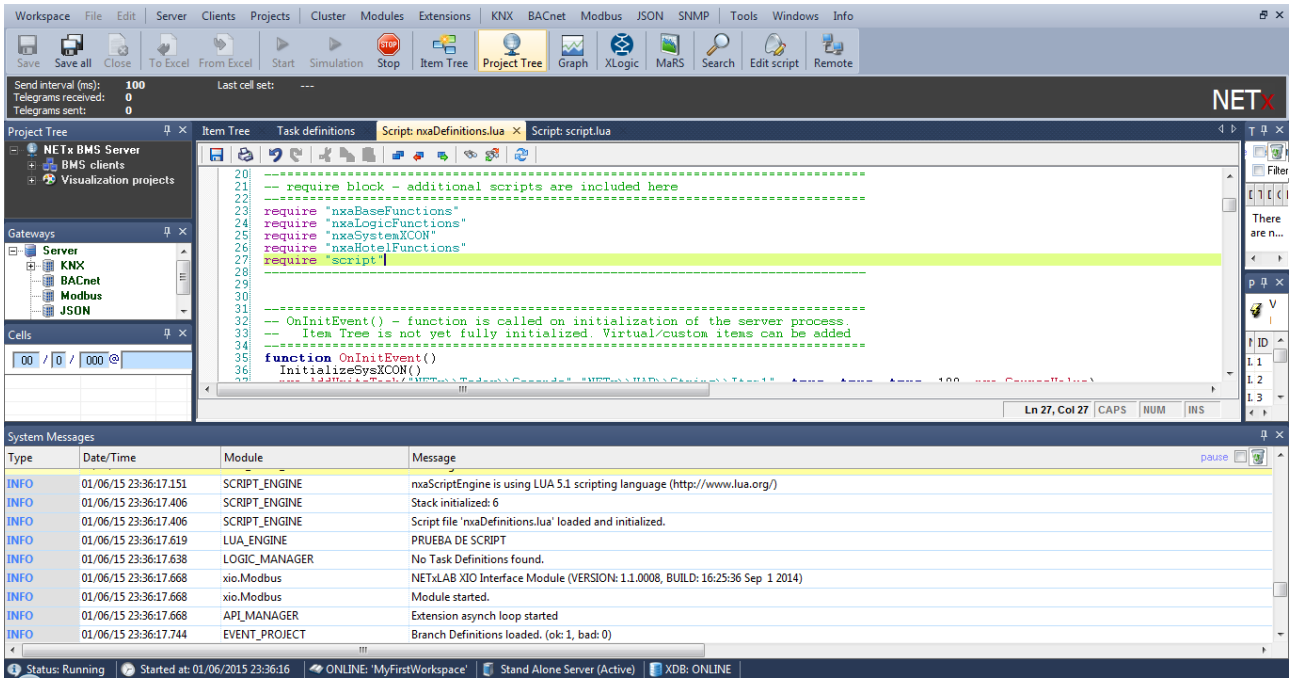
- Funciones para crear tareas
- Funciones que puede leer o modificar los valores de los parámetros de entrada de las tareas.

Las primeras se deben definir en la función ‘OnInitEvent’ cómo ya se ha comentado. El resto puede llamarse dentro de un script que lance una tarea, o incluso como uno de los parámetros de entrada de las funciones de creación.

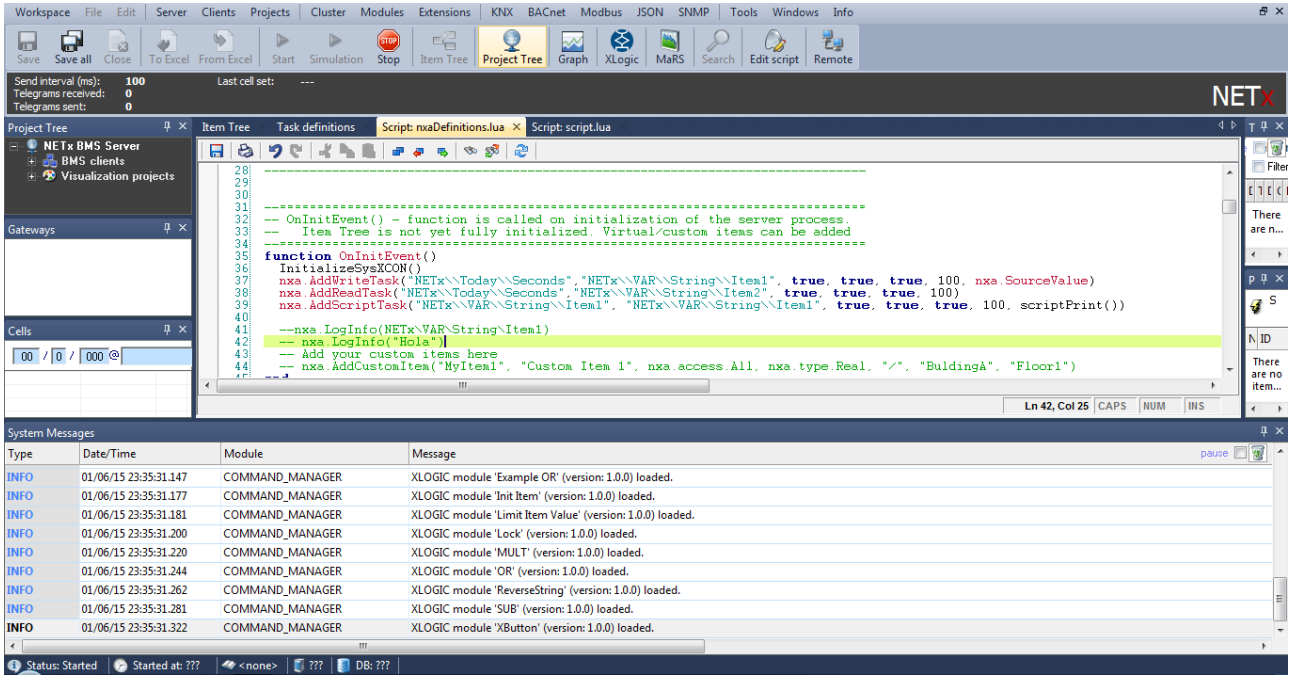
Se escribe un nuevo fichero con las funciones que deseemos.



En el archive ‘nxaDefinitions.lua’ añadimos el script anterior con ‘require’, de ésta manera ya podemos llamar las funciones creadas.



En ‘nxaDefinitions’ definimos la creación de tareas. Tanto en éstas funciones como en el script adicional podemos utilizar el resto de las funciones para obtener los parámetros de las funciones y realizar las operaciones que queramos, además de las funciones del resto del sistema.



Aquí se puede ver el resultado de una función que escribe en el Item1 de String los segundos que cuenta el servidor, el objeto 'Seconds' del sistema lanza la tarea como se ha definido en la tarea y se escribe su valor que se recoge mediante 'nxa.SourceValue'.