

Application note

spaceLYnk and SmartStruxure™ Lite integration via Web Services

A smart configuration only note to share data between a spaceLYnk & a SmartStruxure™ Lite manager using Web Services



Safety Information

Important Information



Read these instructions carefully before trying to install, configure, or operate this software. The following special messages may appear throughout this bulletin or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of either symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, can result in death or serious injury.

CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, can result in minor or moderate injury.

NOTICE


NOTICE is used to address practices not related to physical injury. The safety alert symbol shall not be used with this signal word.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction, installation, and operation of electrical equipment and has received safety training to recognize and avoid the hazards involved.

Safety Precautions

 WARNING
<p>HAZARD OF INCORRECT INFORMATION</p> <ul style="list-style-type: none">• Do not incorrectly configure the software, as this can lead to incorrect reports and/or data results.• Do not base your maintenance or service actions solely on messages and information displayed by the software.• Do not rely solely on software messages and reports to determine if the system is functioning correctly or meeting all applicable standards and requirements.• Consider the implications of unanticipated transmission delays or failures of communications links. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information that is contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2014 Schneider Electric. All rights reserved

Table of Contents

- 1 Introduction 6
- 2 Design 8
 - 2.1 What is Web Service 8
 - 2.2 General architecture 8
 - 2.3 Solution software architecture 9
- 3 Configuration 10
 - 3.1 SmartStruxure™ Lite MPM script configuration 10
 - 3.1.1 Script installation 11
 - 3.1.2 Script installation verification 17
 - 3.2 spaceLYnk script configuration 19
 - 3.2.1 Script installation 19
 - 3.2.2 Script installation verification 24
 - 3.3 Share data between spaceLYnk & SmartStruxure™ Lite MPM..... 25
 - 3.3.1 Share a spaceLYnk object with a SmartStruxure™ Lite MPM one..... 26
 - 3.3.2 Verification of spaceLYnk => SmartStruxure™ Lite MPM objects synchronization 34
 - 3.3.3 Share a SmartStruxure™ Lite MPM object with a spaceLYnk one..... 37
 - 3.3.4 Verification of SmartStruxure™ Lite MPM => spaceLYnk objects synchronization 44
- 4 Conclusion..... 47
- 5 Appendix 48
 - 5.1 Glossary 48

1 Introduction

This application note describes a method to deploy a data synchronisation solution between a spaceLYnk (SL) and a SmartStruxure™ Lite (SSL) Multi-Purpose Manager (MPM) using Web Service technology.

A glossary is available in the appendix chapter 5 of this document. Please refer to it whenever necessary.

Customer value propositions

The customer value propositions correspond to real use cases to connect a spaceLYnk and a SmartStruxure™ Lite Multi-Purpose Manager.

SmartStruxure™ Lite typically manages HVAC (Heating, Ventilation, Air Conditionning) and spaceLYnk manages lighting and shutters, blinders systems in KNX, power metering with Modbus; furthermore a graphical user interface may be built with SmartStruxure™ Lite Multi-Purpose Manager or spaceLYnk.

In this context here are some identified possible use cases which can be implemented with this application note:

- Use case 1: the graphical user interface is built with SmartStruxure™ Lite Multi-Purpose Manager
 - this user interface manages lighting, blinders metering from spaceLYnk
- Use case 2: the graphical user interface is built with spaceLYnk
 - this user interface manages temperature setpoints and metering for SmartStruxure™ Lite
- Use case 3: the whole rooms control devices are KNX ones:
 - for lighting, shutters control with presence management
 - for temperature setpoint and measurements

The temperature data and presence information are transferred by the spaceLYnk to SmartStruxure™ Lite Multi-Purpose Manager which manages HVAC

Competencies

This document is intended for readers who have been trained on spaceLYnk and SmartStruxure™ Lite products. The integration should not be attempted by someone who is new to the installation of each product.

In addition we recommend that you be familiar with:

- The concepts of KNX technology
- Basic technical knowledge on software technologies such as: Web Services (XML, JSON), LUA scripting

System prerequisites

SmartStruxure™ Lite solution includes different SmartStruxure™ Lite Multi-Purpose Managers, this application note is compliant with whole managers of MPM series (SSL-MPM):

- MPM-UN Multi-Purpose Manager
- MPM-VA VAV Manager
- MPM-GW Wireless Manager

Software / Product name	version
spaceLYnk	1.0
SmartStruxure™ Lite MPM software version	2.10.1
Mozilla FireFox	27.0

Table 1: System prerequisites

2 Design

2.1 What is Web Service

A Web service is a method of communications between two electronic devices over a network. It is a software function provided at a network address over the web.

A Web service is defined as:

a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

We can identify two major classes of Web services:

- REST-compliant Web services: in which the primary purpose of the service is to manipulate XML representations of Web resources using a uniform set of stateless operations; and
- Arbitrary Web services: in which the service may expose an arbitrary set of operations.

2.2 General architecture

The architecture for SSL-MPM - SL Web Service exchanges consists on connecting SSL-MPM and SL through their Ethernet/IP/HTTP interfaces within a Local Area Network (LAN) as following:

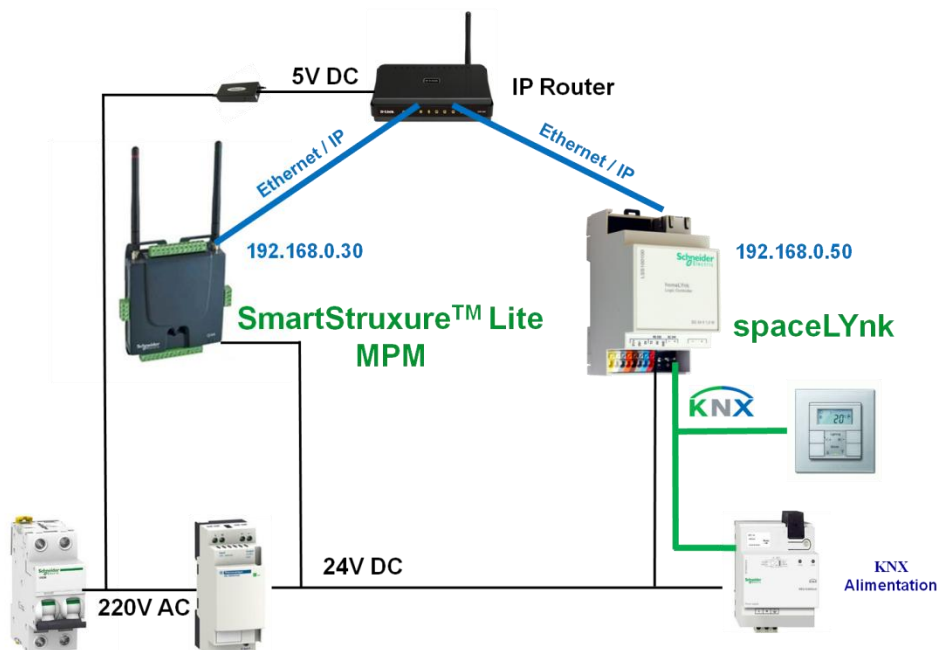


Figure 1: General architecture

2.3 Solution software architecture

The solution principle is either SSL-MPM or SL has to notify value changes to the other entity. The solution is designed to share data between only one SSL-MPM and one SL.

Here are the general overview of the implementation software blocks for the solution to offer bidirectional data sharing between spaceLynk and SmartStruxure™ Lite MPM:

- green software blocks are especially added software blocks for the solution
- grey software blocks are already native spaceLynk and SmartStruxure™ Lite MPM software blocks

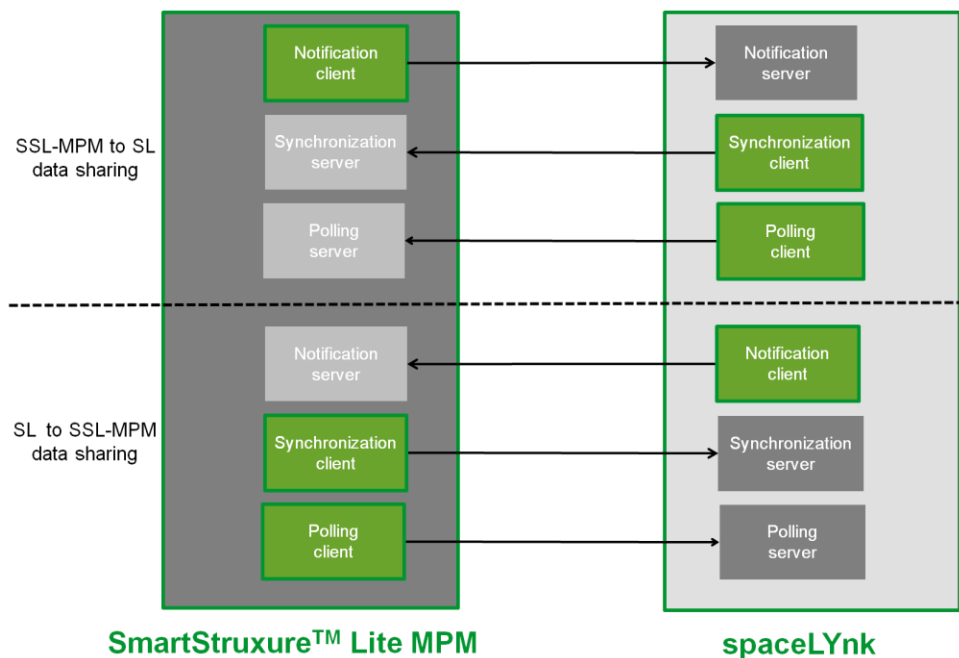


Figure 2: Solution software architecture

3 Configuration

3.1 SmartStruxure™ Lite MPM script configuration

Run Mozilla Firefox (with Windows, OS X, Linux) or Safari (if iOS) with SSL-MPM URL containing SSL-MPM IP address (in our example: <http://192.168.0.30>) to access the SSL-MPM manager; you get a Login page:

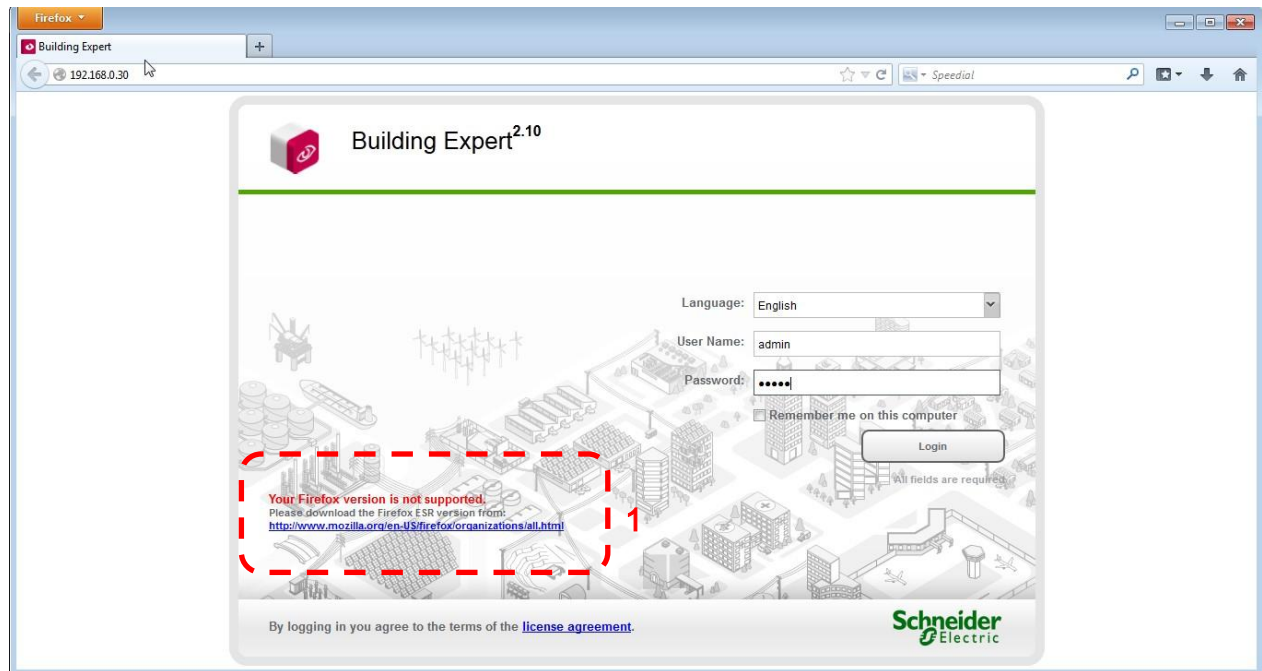


Figure 3: SSL-MPM manager home page

Note: check on this page you have not a **message**¹ (if it is the case you have to update your Firefox browser to the right version, even if it is possible to use a different version but you may have some visualization issues).

Now, you can log to enter the SSL-MPM manager, if you have not changed default administrator login & password, it is:

- Login: “admin”
- Password : “admin”

3.1.1 Script installation

In SSL-MPM manager home page:

- Right click on **SmartStruxure™ Controller**¹
- Left click on **Add Objects** ²

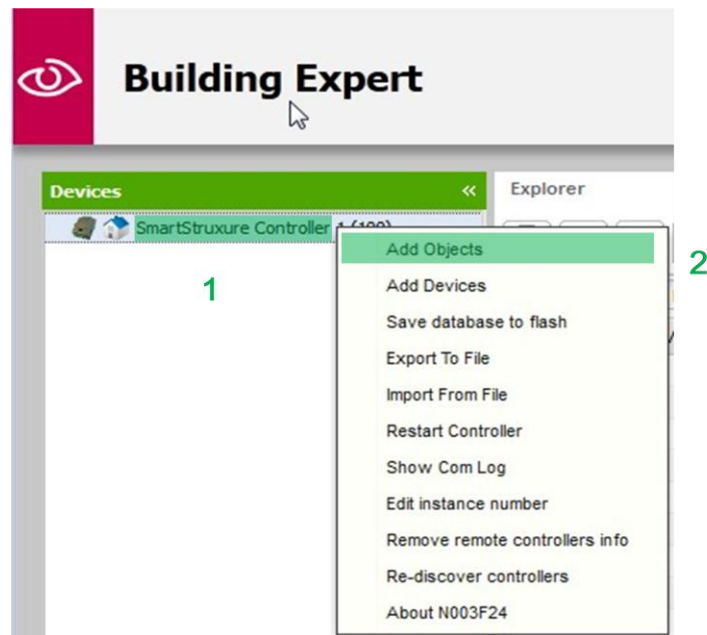


Figure 4: SSL-MPM manager home page

- Select **Lua Program**³ filter

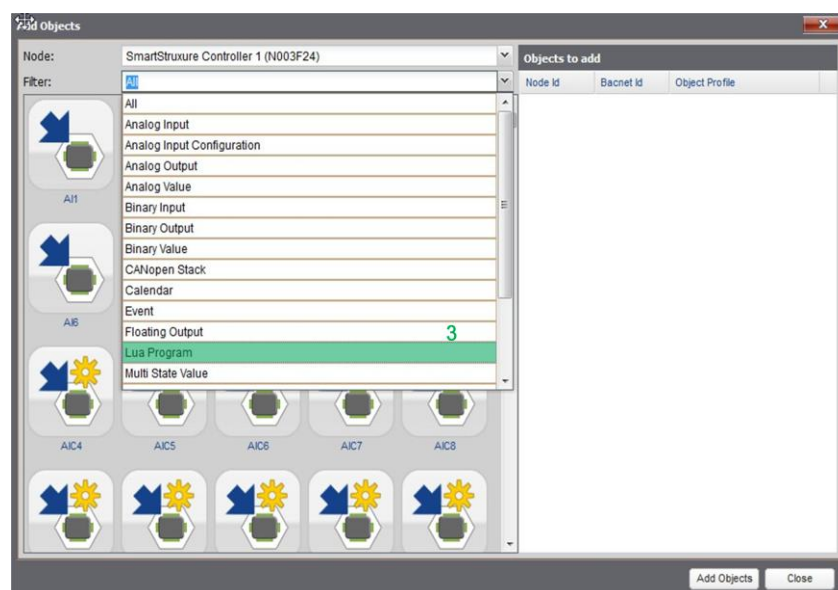


Figure 5: Create a LUA script in SSL-MPM, step 1

- Left click on a free PGx⁴ (in our example PG1)
- Left click on Add Objects⁵

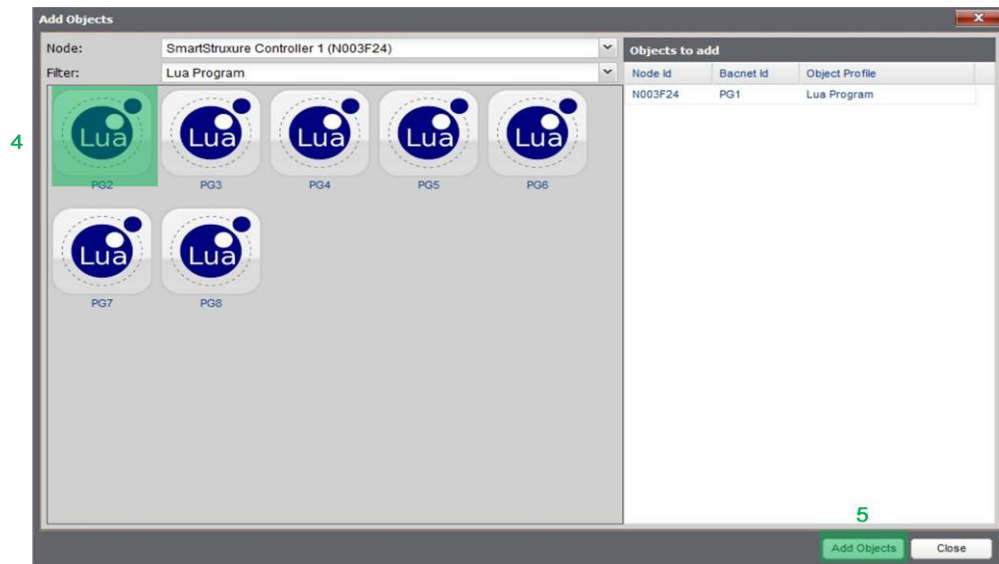


Figure 6: Create a LUA script in SSL, step 2

A void script has been created in the SSL-MPM named 100.PG1.

- Left click on 100.PG1⁶ (scroll window with the scroll bar⁷ if necessary)

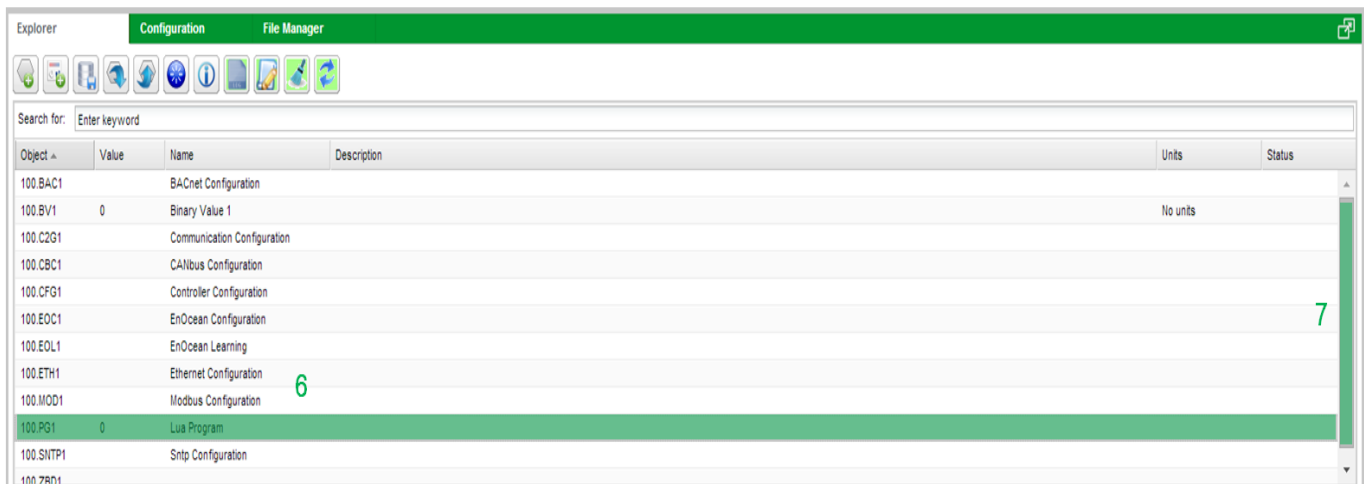


Figure 7: SSL home page with new 100.PG1 script

- Select **Script⁸** Tab in Lua Program frame
- Left click on **Import⁹**



Figure 8: Import a script in 100.PG1

An “Import from File” window opens:

- Left click on **File icon¹⁰**



Figure 9: Import LUA script window: step 1

A new window opens:

- Navigate in your PC folders to retrieve *SSL-MPM_script* folder of the Application Note unzipped package: go in SSL-MPM_Script folder
- Select *AN004_v1-0_WS_SL-SSL_MPM_script.c2g¹¹*
- Left click on **Open¹²**

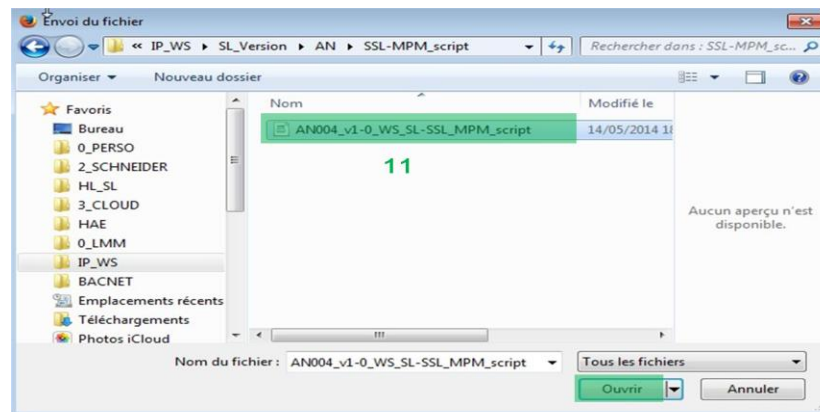


Figure 10: Select LUA script to import

In “Import from File” window:

- Select **PG1**¹³
- Left click on **Import**¹⁴

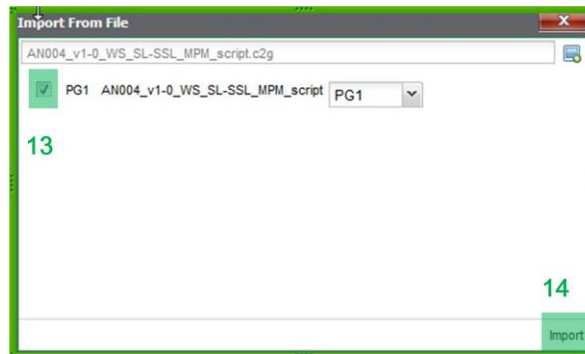


Figure 11: Import LUA script window: step 2

The AN004_v1-0_WS_SL-SSL_MPM_script is now installed.

This script has not to be modified:

- Don't modify script timer
- Script is by default active and has to be permanently active: don't stop it

Only possible modifications (in script USER MODIFIABLE CONFIGURATION part) are detailed in following part.

- In **Script**¹⁵ Tab in Lua Program frame
- Left click on **Edit**¹⁶



Figure 12: Edit 100.PG1 script

In the 100.PG1 script text zone:

- Go to **"USER MODIFIABLE CONFIGURATION"**¹⁷ part

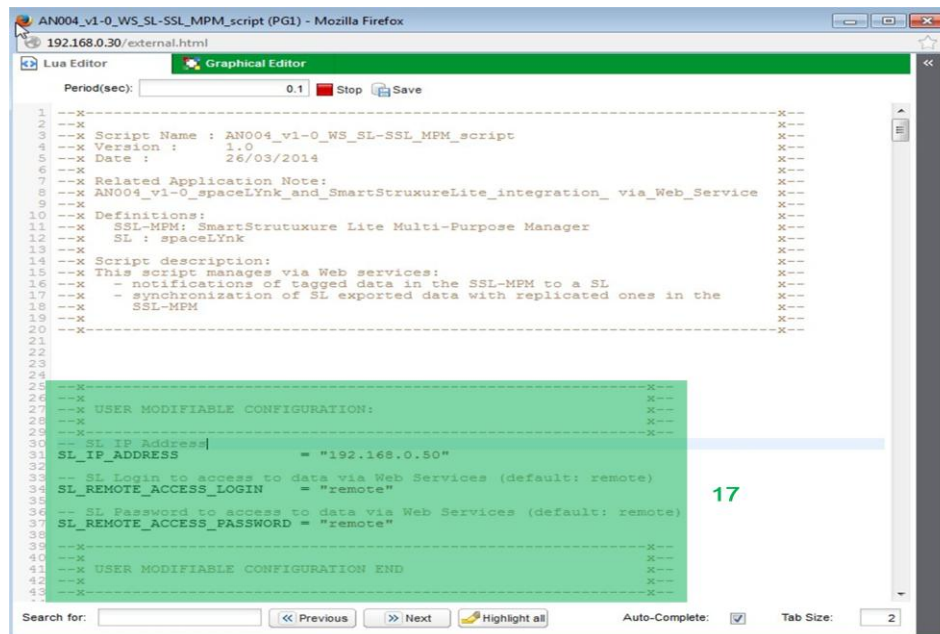


Figure 13: 100.PG1 script edition

- Update the configuration parameters according the SL you use:
 - SL_IP_ADDRESS** → set the IP address of the spaceLYnk you want the SSL-MPM will push data
 - Example: **SL_IP_ADDRESS = "192.168.0.50"**
- Following configuration parameters are default SL ones, update them only if you modified them on SL:
 - SL_REMOTE_ACCESS_LOGIN** → set the remote user login to access to SL via web services ("remote" by default)
 - Example: **SL_REMOTE_ACCESS_LOGIN = "remote"**
 - SL_REMOTE_ACCESS_PASSWORD** → set the remote user password to access to SL via web services ("remote" by default)
 - Example: **SL_REMOTE_ACCESS_PASSWORD = "remote"**

```
--X-----X--
--X-----X--
--X USER MODIFIABLE CONFIGURATION: X--
--X-----X--
--X-----X--
-- SL IP Address X--
SL_IP_ADDRESS = "192.168.0.50" X--
-- SL Login to access to data via Web Services (default: remote) X--
SL_REMOTE_ACCESS_LOGIN = "remote" X--
-- SL Password to access to data via Web Services (default: remote) X--
SL_REMOTE_ACCESS_PASSWORD = "remote" X--
--X-----X--
--X-----X--
--X USER MODIFIABLE CONFIGURATION END X--
--X-----X--
```

Figure 14: Set new SL configuration in 100.PG1 script

- Left click on **Save**¹⁸
- And **Close**¹⁹ window

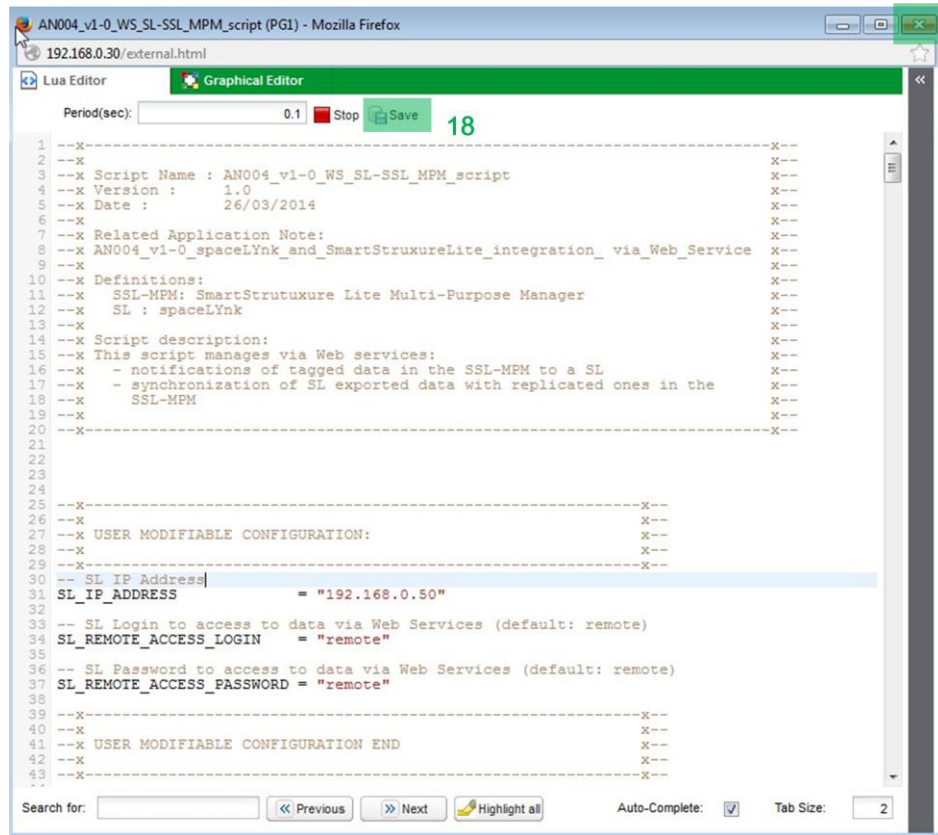


Figure 15: Save and close 100.PG1 script

The SSL-MPM script is loaded with SL configuration: SSL-MPM configuration is finished.

3.1.2 Script installation verification

To verify the configuration, you have to check in script logs you have a log notifying the connection with spaceLYnk is OK.

To view SSL-MPM logs, go in SSL-MPM manager home page:

- Left click on **100.PGx¹** corresponding to script named “*AN004_v1-0_WS_SL-SSL_MPM_script*”

(Scroll window with the **scroll bar⁷** if necessary)

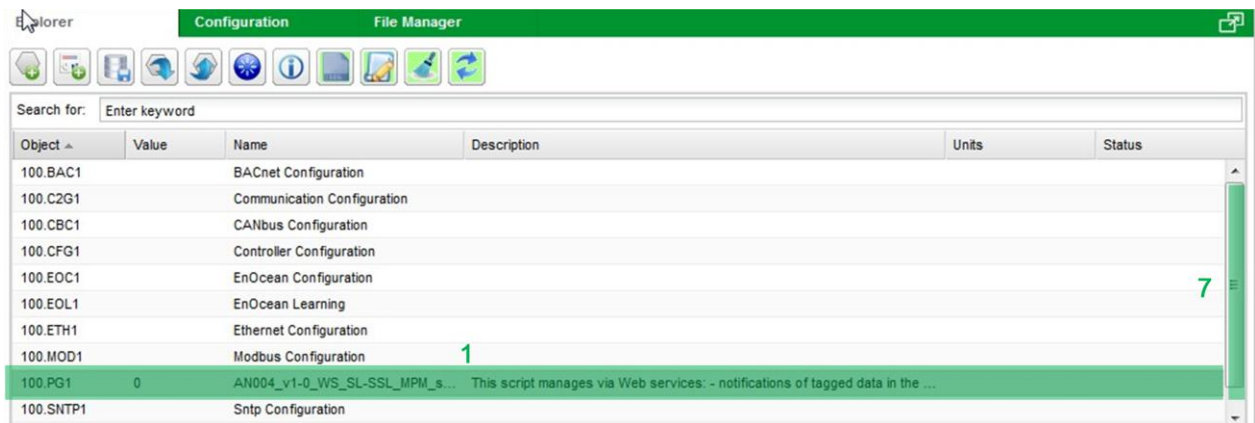


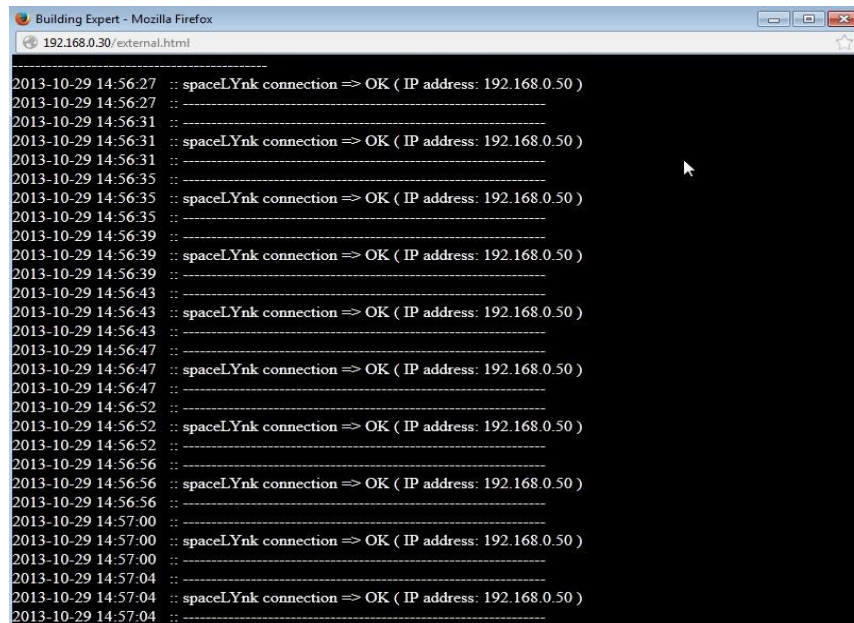
Figure 16: SSL-MPM manager home page with 100.PG1 script

- Select **Script²** Tab in Lua Program frame
- Left click on **Output³**



Figure 17: View 100.PG1 script logs

- Read the text and check if there is a log with “spaceLYnk connection => OK”



```

Building Expert - Mozilla Firefox
192.168.0.30/external.html
-----
2013-10-29 14:56:27 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-29 14:56:27 :: -----
2013-10-29 14:56:31 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-29 14:56:31 :: -----
2013-10-29 14:56:35 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-29 14:56:35 :: -----
2013-10-29 14:56:39 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-29 14:56:39 :: -----
2013-10-29 14:56:43 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-29 14:56:43 :: -----
2013-10-29 14:56:47 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-29 14:56:47 :: -----
2013-10-29 14:56:52 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-29 14:56:52 :: -----
2013-10-29 14:56:56 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-29 14:56:56 :: -----
2013-10-29 14:57:00 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-29 14:57:00 :: -----
2013-10-29 14:57:04 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-29 14:57:04 :: -----

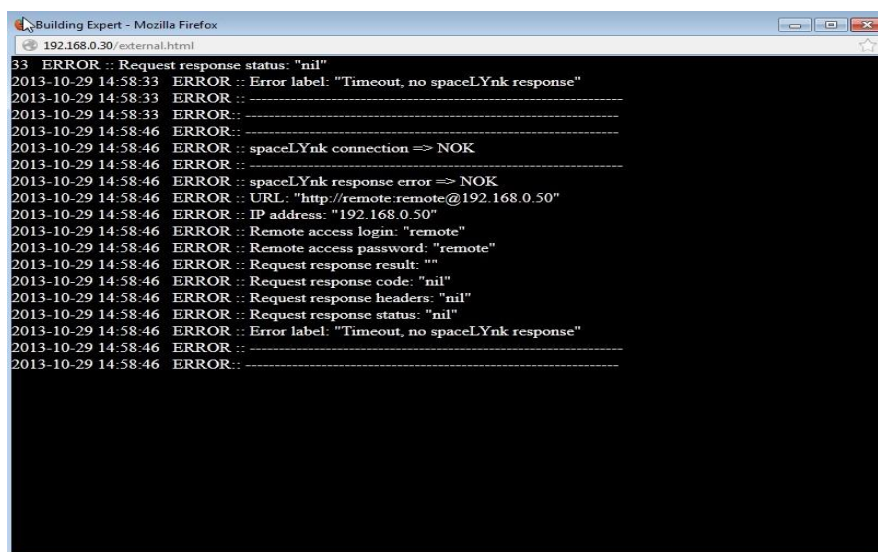
```

Figure 18: 100.PG1 script console: SL connection OK

Error cases:

If you have logs with ‘ERROR ::’, here are some possible error cases:

- SL IP address is bad or SL & SSL are not well IP connected:
 - ⇒ Check SL & SSL are connected over Ethernet
 - ⇒ Change IP address at the beginning of the script
- SL remote access login is bad
- SL remote access password is bad



```

Building Expert - Mozilla Firefox
192.168.0.30/external.html
33 ERROR :: Request response status: "nil"
2013-10-29 14:58:33 ERROR :: Error label: "Timeout, no spaceLYnk response"
2013-10-29 14:58:33 ERROR :: -----
2013-10-29 14:58:33 ERROR :: -----
2013-10-29 14:58:46 ERROR :: spaceLYnk connection => NOK
2013-10-29 14:58:46 ERROR :: -----
2013-10-29 14:58:46 ERROR :: spaceLYnk response error => NOK
2013-10-29 14:58:46 ERROR :: URL: "http://remote:remote@192.168.0.50"
2013-10-29 14:58:46 ERROR :: IP address: "192.168.0.50"
2013-10-29 14:58:46 ERROR :: Remote access login: "remote"
2013-10-29 14:58:46 ERROR :: Remote access password: "remote"
2013-10-29 14:58:46 ERROR :: Request response result: ""
2013-10-29 14:58:46 ERROR :: Request response code: "nil"
2013-10-29 14:58:46 ERROR :: Request response headers: "nil"
2013-10-29 14:58:46 ERROR :: Request response status: "nil"
2013-10-29 14:58:46 ERROR :: Error label: "Timeout, no spaceLYnk response"
2013-10-29 14:58:46 ERROR :: -----
2013-10-29 14:58:46 ERROR :: -----

```

Figure 19: 100.PG1 script console: SL connection NOK

3.2 spaceLYnk script configuration

3.2.1 Script installation

Run Mozilla Firefox (with Windows, OS X, Linux) or Safari (if iOS) with SL URL containing SL IP address (in our example: <http://192.168.0.50>) to access the SL start page:

- Left click on **Configurator**¹

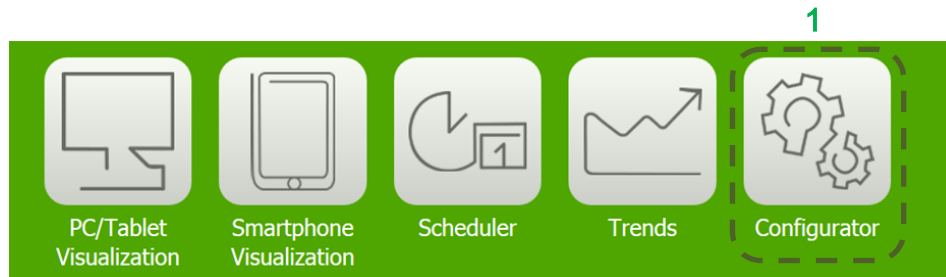


Figure 20: spaceLYnk starting page

You get a Login page:

If you have not changed default administrator login & password, it is:

- Login: “admin”
- Password : “admin”

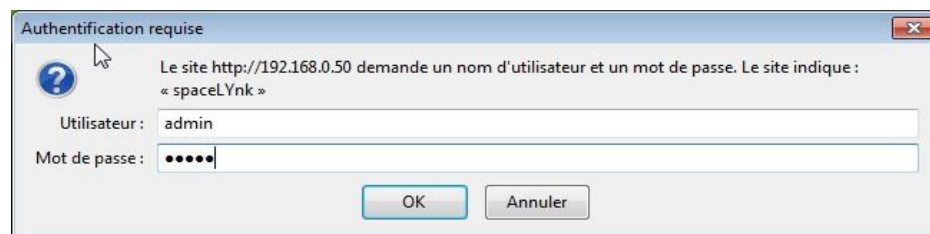


Figure 21: spaceLYnk configurator login page

- Left click on **Scripting²**
- Left click on **Tools³**
- Left click on **Restore scripts⁴**

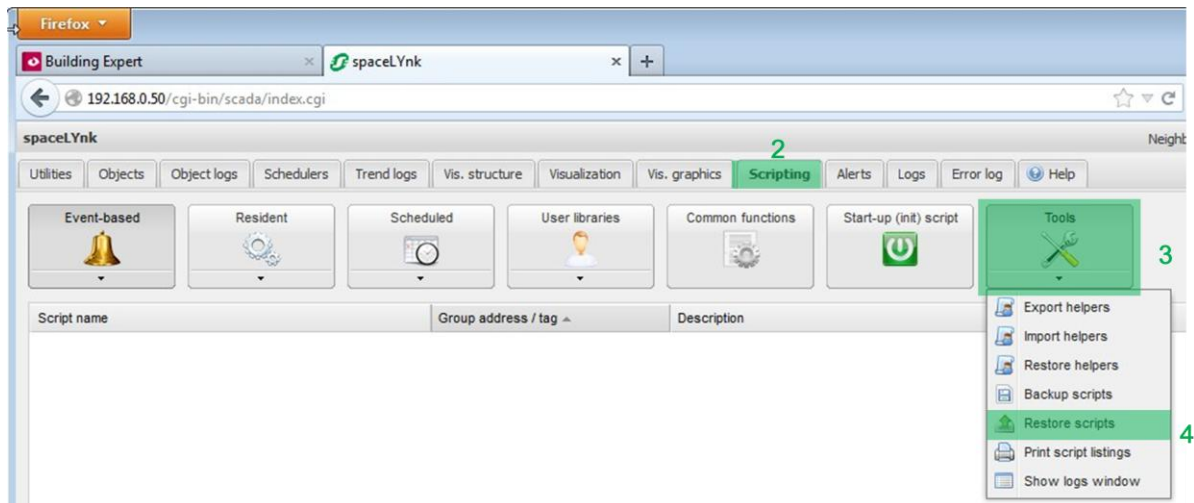


Figure 22: spaceLYnk configurator home page

- Select **Append keeping existing scripts⁵**.
- Left click on **Browse...⁶** and select “AN004_v1-0_WS_SL_SSL-SL_scripts.tar” file stored on your PC in unzipped Application Note package (in SL_scripts folder)
- Left click then on **Save⁷**

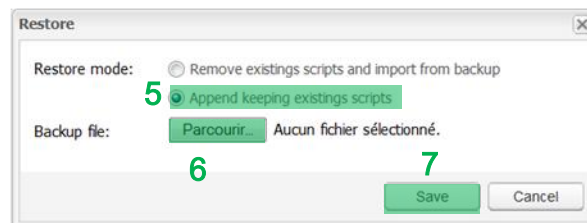


Figure 23: spaceLYnk scripts restore window

- You must get following window notifying you the restore has been well carried out; SL will reboot to apply scripts:



Figure 24: spaceLYnk scripts restore success window

You can check the following scripts have been installed:

- In Event-based scripting part:
 - AN004_v1-0_WS_SSL-SL_SL_Notification_Event_Client
- In Resident scripting part:
 - AN004_v1-0_WS_SSL-SL_SL_Notification_Cyclic_Client
 - AN004_v1-0_WS_SSL-SL_SL_Polling_Client
- In Users libraries scripting part:
 - AN004_v1-0_WS_SSL-SL_SL_Lib

Those scripts and library have not to be modified:

- Don't modify script timers
- Scripts and library are by default active and they have to be permanently active

Only possible modifications (USER MODIFIABLE CONFIGURATION script part) are detailed in following part.

- Left click on **Scripting**⁸
- Left click on **Resident**⁹
- Left click on **Editor icon**¹⁰ for “AN004_v1-0_WS_SSL-SL_SL_Polling_Client” script

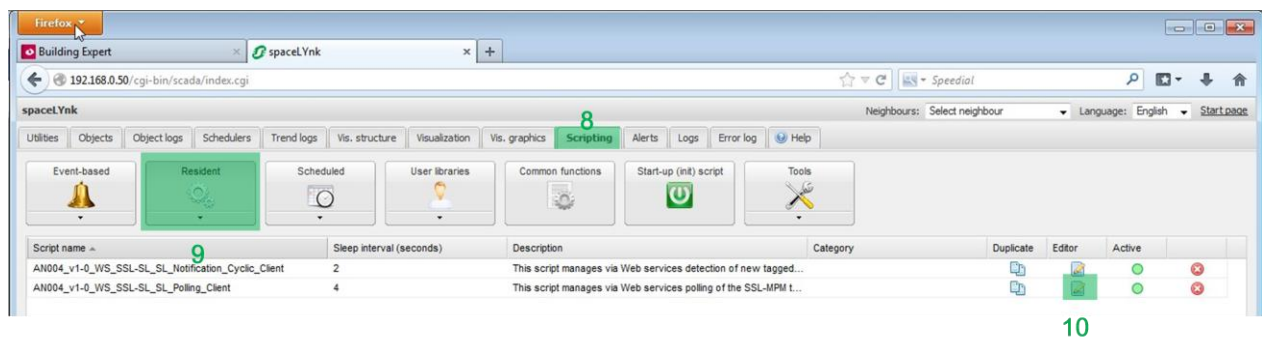


Figure 25: AN004_v1-0_WS_SSL-SL_SL_Polling_Client script edition

In the “AN004_v1-0_WS_SSL-SL_SL_Polling_Client” script text zone:

- Go to **“USER MODIFIABLE CONFIGURATION”**¹¹ part

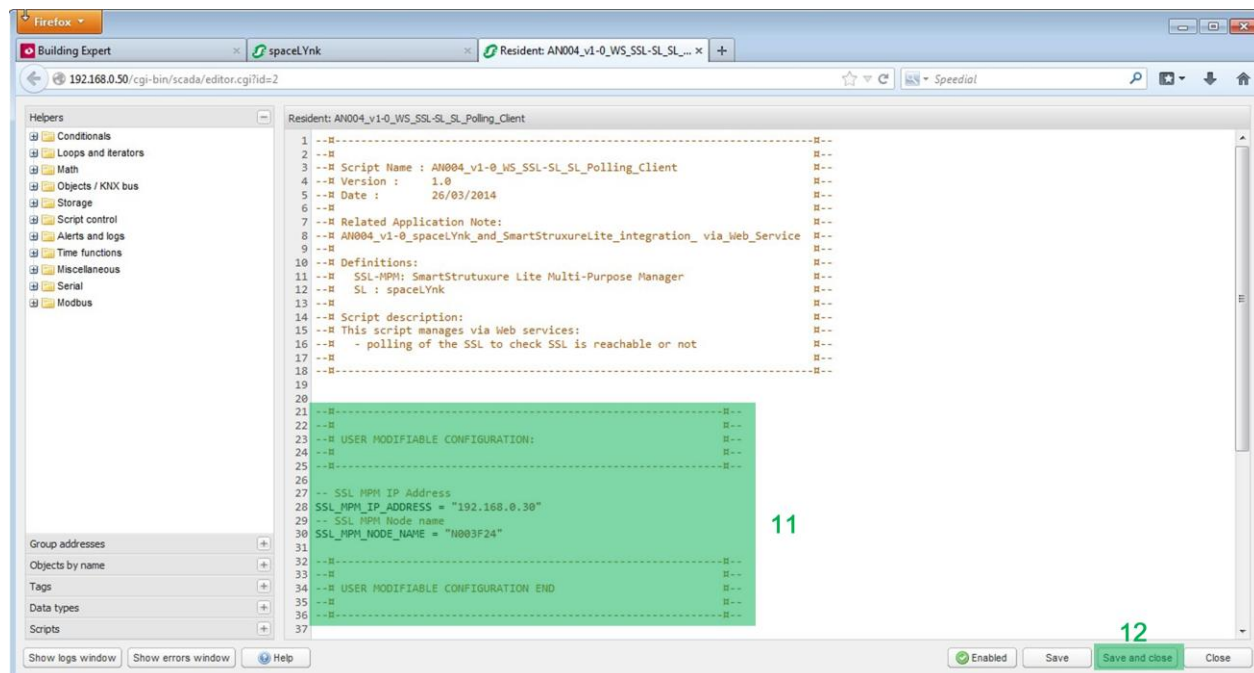


Figure 26: Set SSL configuration in “AN004_v1-0_WS_SSL-SL_SL_Polling_Client” script

- Update the configuration parameters according the SSL-MPM you use:
 - SSL_MPM_IP_ADDRESS → the SSL-MPM IP address you want the SL will push data and get data from
 - Ex: **SSL_MPM_IP_ADDRESS = "192.168.0.30"**
 - SSL_MPM_NODE_NAME → the SSL-MPM node name corresponding to the SSL-MPM IP address
 - Ex: **SSL_MPM_NODE_NAME = "N003F24"**
 - To get **SSL_MPM_NODE_NAME**: go in SSL-MPM manager home page:
 - ⇒ Right click on **SmartStruxure™ Controller¹**
 - ⇒ Read last attribute **Node name²**

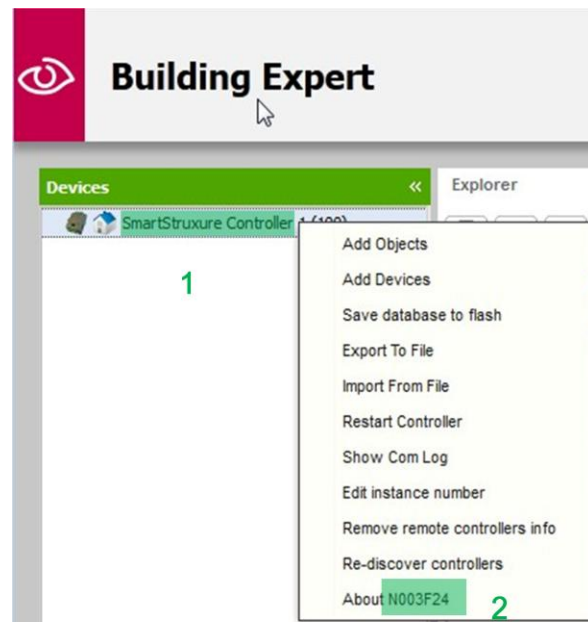


Figure 27: SSL-MPM manager home page

- Left click on **Save and Close¹²**

```

21  --H-----H--
22  --H-----H--
23  --H USER MODIFIABLE CONFIGURATION: H--
24  --H-----H--
25  --H-----H--
26
27  -- SSL MPM IP Address
28  SSL_MPM_IP_ADDRESS = "192.168.0.30"
29  -- SSL MPM Node name
30  SSL_MPM_NODE_NAME = "N003F24"
31
32  --H-----H--
33  --H-----H--
34  --H USER MODIFIABLE CONFIGURATION END H--
35  --H-----H--
36  --H-----H--

```

Figure 28: Set SSL-MPM IP address and Node name in SL script

The SL scripts are loaded with SSL-MPM configuration: SL configuration is finished.

3.2.2 Script installation verification

To verify the configuration, see into the SL **Logs¹** tab:

- Read the text and check if there is a log with “SSL MPM connection => OK”



Figure 29: spaceLYnk Log Tab: SSL-MPM connection OK

Error cases:

If you have logs with ‘ERROR ::’, here are some possible error cases:

- SSL-MPM IP address is bad or SL & SSL-MPM are not well IP connected:
 - ⇒ Check SL & SSL-MPM are connected over Ethernet
 - ⇒ Change IP address in AN004_v1-0_WS_SSL-SL_SL_Polling_Client resident script
- SSL-MPM Node Name is bad:
 - Change Node Name in AN004_v1-0_WS_SSL-SL_SL_Polling_Client resident script

If you see no error in the SL **Logs¹** tab and the configuration is always not operational: you can have a look in

Error log² tab:

- If you have logs in this tab for one of AN004_v1-0_WS_SSL-SL_xxxx scripts
 - we recommend you to remove all AN004_v1-0_WS_SSL-SL_xxxx scripts and install again the SL configuration
 - if after the new configuration installation, the configuration is always not operational, please contact your country Schneider Electric support

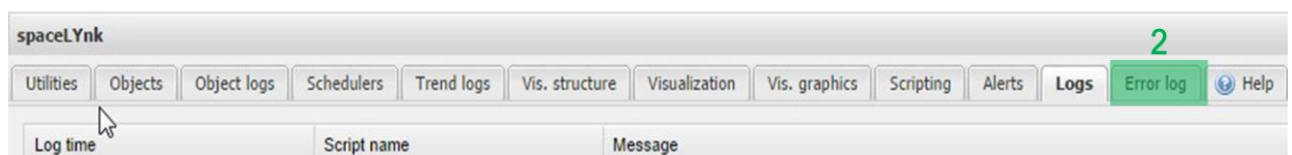


Figure 30: spaceLYnk Error Log Tab

3.3 Share data between spaceLYnk & SmartStruxure™ Lite MPM

The principle to share data in both directions between a spaceLYnk and a SmartStruxure™ Lite MPM is:

- 1) To share from spaceLYnk to SmartStruxure™ Lite MPM:
 - a. **to select** a SL existing object **OR to create** a new object in SL
 - b. **to select** a SSL-MPM existing object **OR to create** a new object in SSL-MPM
 - c. **to tag** SSL-MPM object to link it with SL one
 - d. **to tag** SL object to link it with SSL-MPM one

- 2) To share from SmartStruxure™ Lite MPM to spaceLYnk:
 - a. **to select** a SSL-MPM existing object **OR to create** a new object in SSL-MPM
 - b. **to select** a SL existing object **OR to create** a new object in SL
 - c. **to tag** SL object to link it with SSL-MPM one
 - d. **to tag** SSL-MPM object to link it with SL one

Next parts describe how to share **one object** from spaceLYnk to SmartStruxure™ Lite MPM and from SmartStruxure™ Lite MPM to spaceLYnk:

If you want to share several objects you have to carry out the same procedure
as many times as you want to link objects
between the spaceLYnk and the SmartStruxure™ Lite MPM.

3.3.1 Share a spaceLYnk object with a SmartStruxure™ Lite MPM one

It is possible to share SL objects in SSL-MPM with following types {"AO", "AV", "BO", "BV"}.

Here is the mapping table between SL and SSL-MPM data types:

SL data types	SSL-MPM data types
01. 1 bit (boolean)	BO, BV
05. 1 byte unsigned integer	AO, AV
06. 1 byte signed integer	AO, AV
07. 2 byte unsigned integer	AO, AV
08. 2 byte signed integer	AO, AV
09. 2 byte floating point	AO, AV
3 byte unsigned integer	AO, AV
10. 3 byte time / day	<i>mapping is not possible</i>
11. 3 byte date	<i>mapping is not possible</i>
12. 4 byte unsigned integer	AO, AV
13. 4 byte signed integer	AO, AV
14. 4 byte floating point	AO, AV
15. 4 byte access control	AO, AV

In addition to tag principle, the created or existing object in SSL-MPM will have a second tag corresponding to the object name in SL:

For example:

- SL object name = **mySLObject**
=> SSL-MPM object
 - name = mySSLObject
 - TAG: **mySLObject**

Here is the general process to implement a SL to SSL-MPM object sharing:

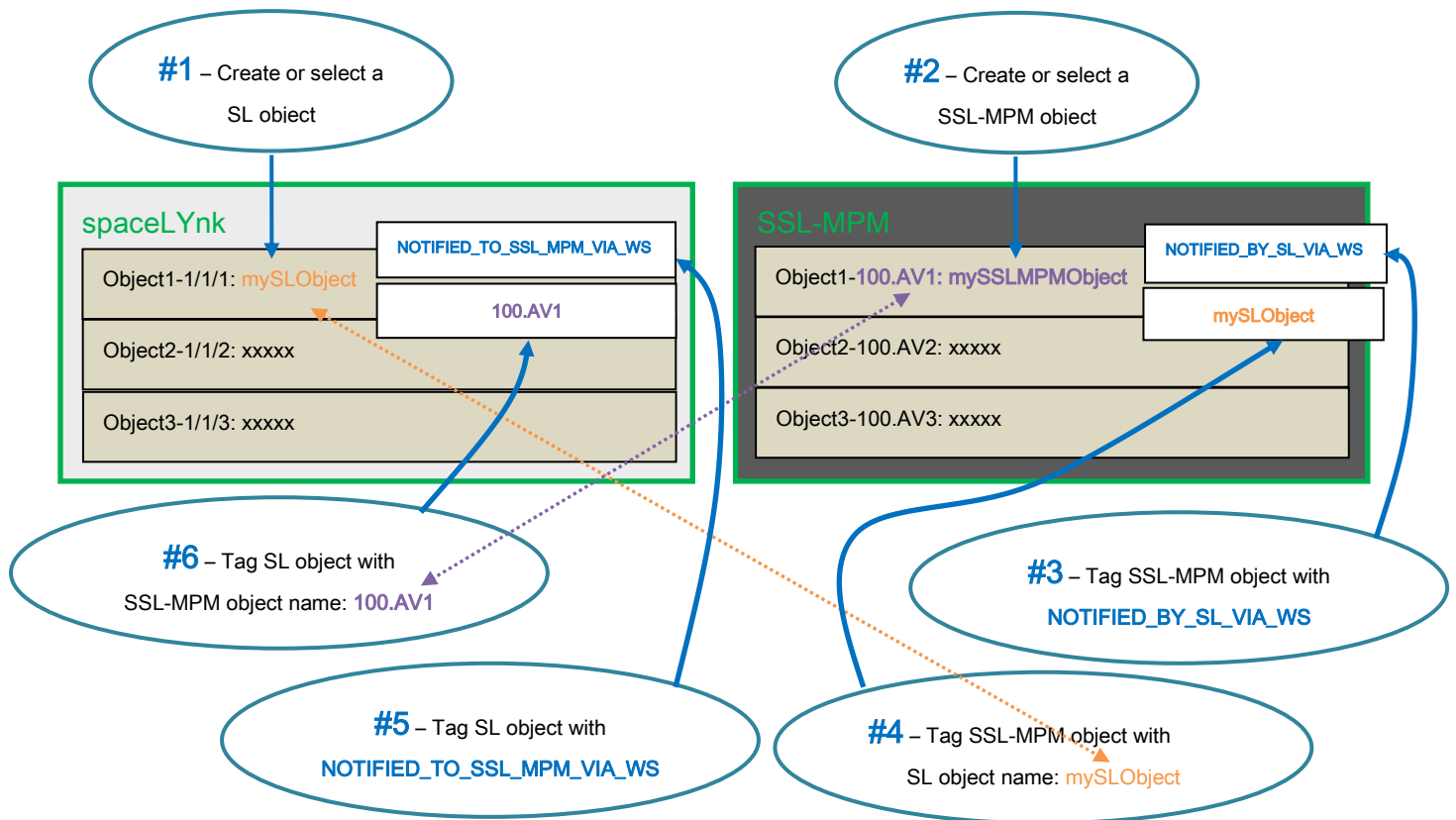


Figure 31: General process to implement a SL to SSL-MPM object sharing

Following parts will detail how to configure SL and SSL-MPM to set those different steps.

In the SL configurator part:

- Left click on **Objects¹** Tab.
- If you want to create a new object:
 - Left click on **Add new object²**.
Set the **Object name³**:
example: **myNewSLObject** (*it is possible to use an object name with space character(s)*)
 - Set the **Group address⁴**.
 - Select the **Data type⁵**.
 - Set the **Export⁶** flag.
 - Set other information if necessary and left click on **Save⁷**
- If you want to use an existing object:
 - Left click on the object you want to notify to SSL-MPM
example: **myExistingSLObject** (*it is possible to use an object name with space character(s)*)
 - Set the **Export⁶** flag.
 - Left click on **Save⁷**

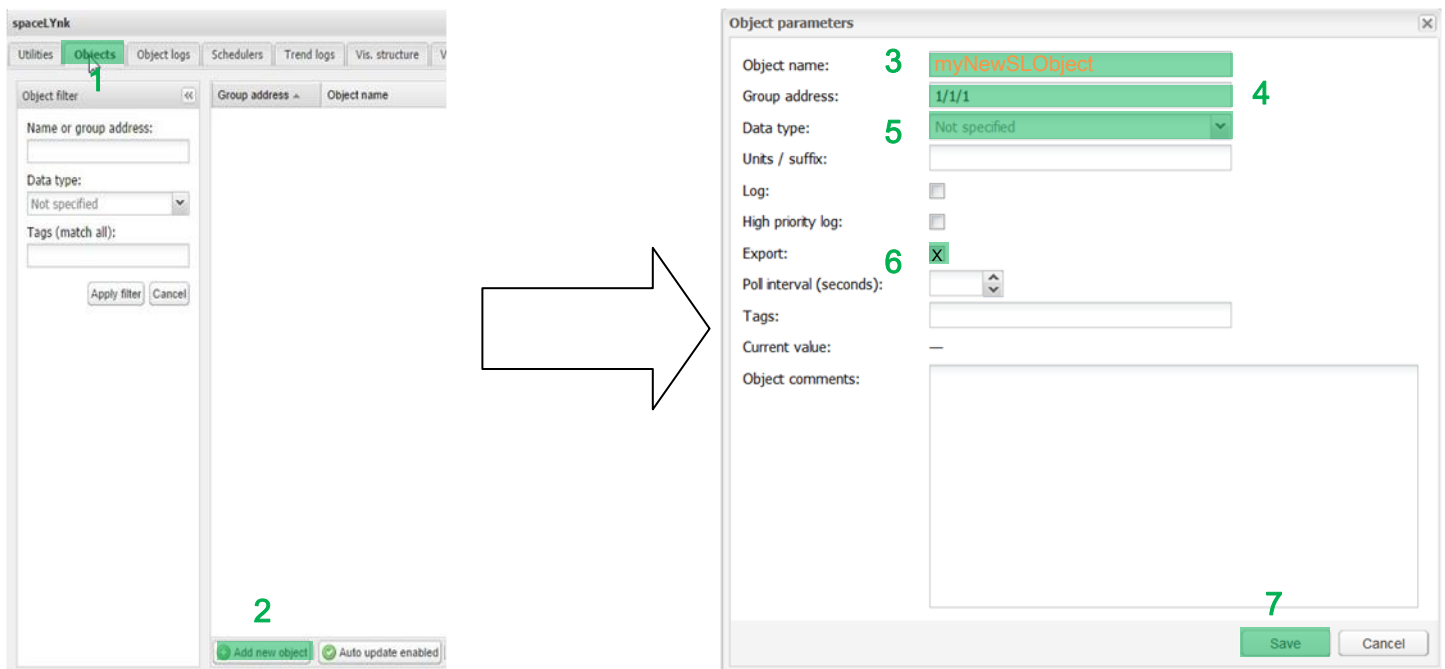


Figure 32: spaceLYnk object Tab

In SSL-MPM manager home page:

- If you want to create a new object:
 - Right click on **SmartStruxure™ Controller¹** and left click on **Add Objects²**

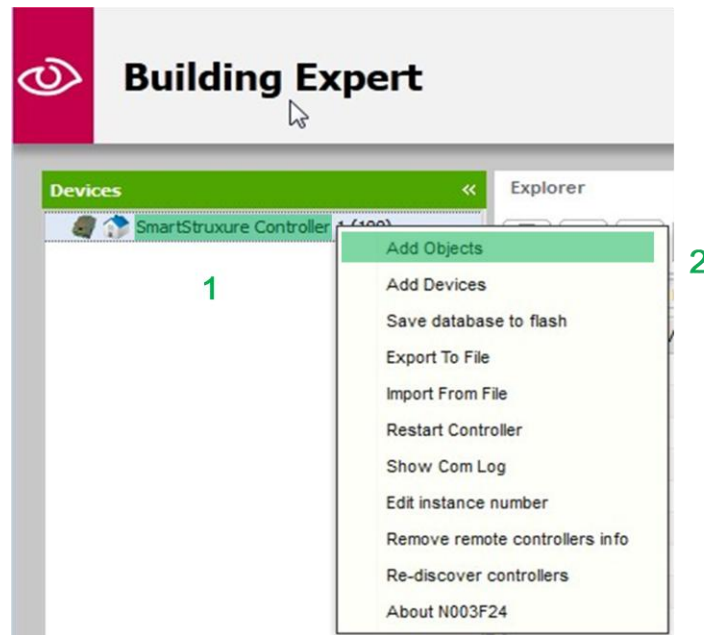


Figure 33: SSL-MPM manager home page

- Select **Filter³** corresponding to a type compliant with SL one
(Possible cases: "AO", "AV", "BO", "BV", see table before for more information)
Example for: **Analog Value** : "AV"

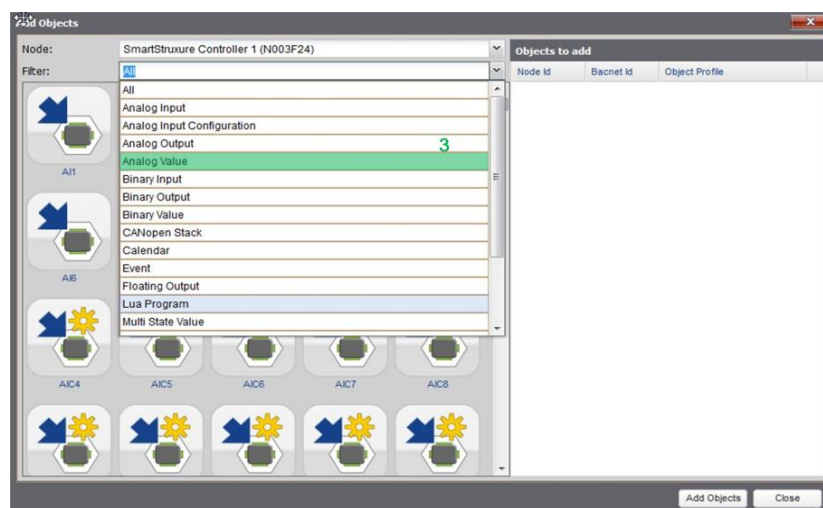


Figure 34: Create an Analog Value object in SSL-MPM, step 1

- Left click on an available **object⁴**: example **AV1**
- Left click on **Add Objects⁵**

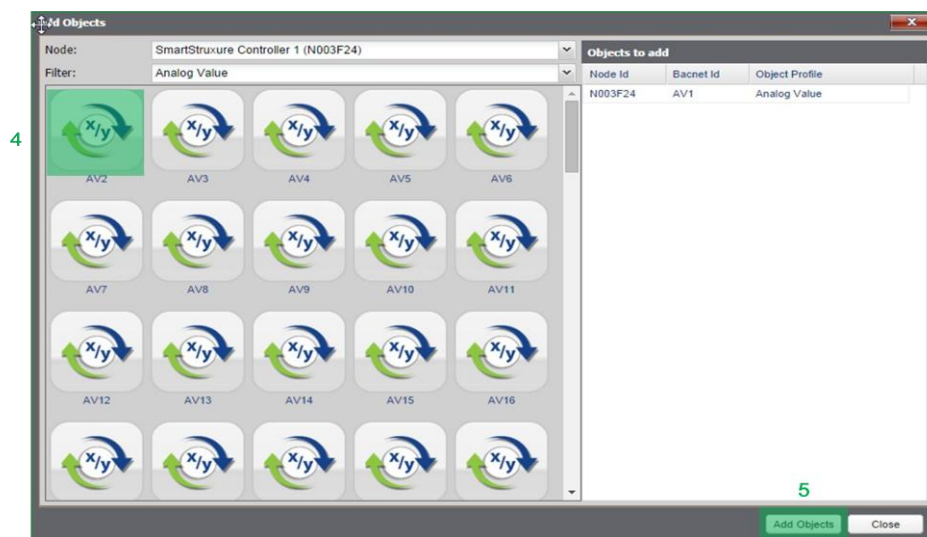


Figure 35: Create an Analog Value object in SSL-MPM, step 2

- Left click on the **object⁶** to select the one you created (scroll window with the **scroll bar⁷** if necessary)

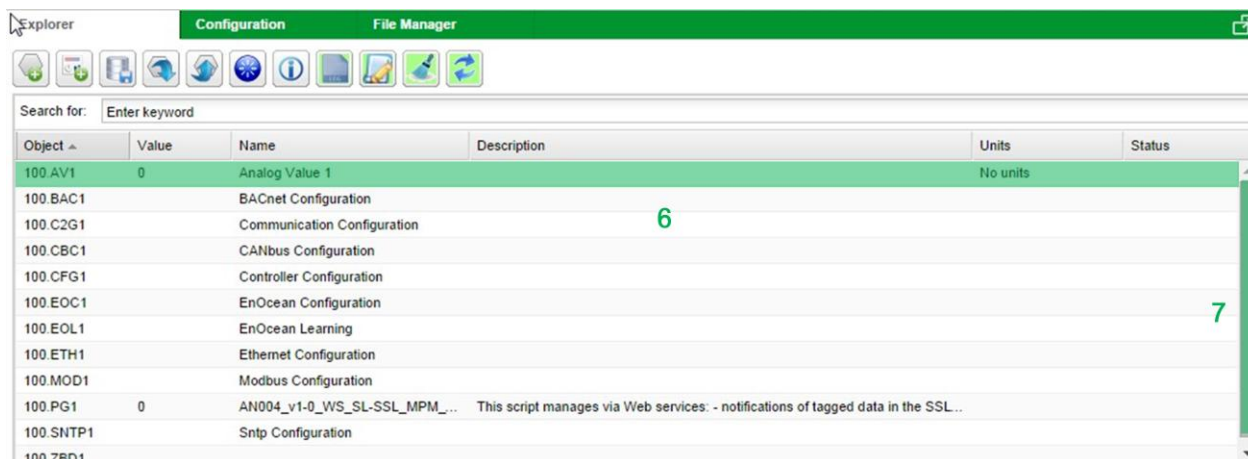


Figure 36: Create an Analog Value object in SSL-MPM, step 3

- In **Name⁸** field, set the name you choose for the object
example: **myNewSSLMPMObject** *(it is possible to use an object name with space character(s))*

The screenshot shows the 'Configuration' tab in the SSL-MPM interface. A table lists various objects, including 100.AV1, 100.BAC1, 100.C2G1, 100.CBC1, 100.CFG1, and 100.EOC1. Below the table, the 'Analog Value' configuration form is displayed. The 'Name' field is highlighted with a green box and labeled with a green '8'. The 'Node' field is set to 'N003F24', the 'Object BACnet Id' is 'AV1', the 'Value' is '0', and the 'Units' are set to 'No units'.

Figure 37: Create an Analog Value object in SSL-MPM, step 4

- If you want to use a existing object:
 - Left click on the **object⁹** to select the one you want (scroll window with the **scroll bar¹⁰** if necessary)

The screenshot shows the 'Configuration' tab in the SSL-MPM interface. A table lists various objects, including 100.AV1, 100.BAC1, 100.C2G1, 100.CBC1, 100.CFG1, and 100.EOC1. The first row (100.AV1) is highlighted with a green background. The 'Name' column is labeled with a green '9', and the 'Status' column is labeled with a green '10'.

Object	Value	Name	Description	Units	Status
100.AV1	0	myExistingSSLMPMObject		No units	
100.BAC1		BACnet Configuration			
100.C2G1		Communication Configuration			
100.CBC1		CANbus Configuration			
100.CFG1		Controller Configuration			
100.EOC1		EnOcean Configuration			

Figure 38: Select an Analog Value object in SSL-MPM

- In **Description¹¹** field, you have to write:
 - For new created object example in SL : “**NOTIFIED_BY_SL_VIA_WS**, myNewSLObject”
 note:
 - the first tag **NOTIFIED_BY_SL_VIA_WS** is case sensitive
 - the second tag, object name in SL (example: myNewSLObject) is case sensitive
 - if the **Description¹¹** field already contains **text** or you want to add one, the **text** has to be before and/or after the both tags with a comma as separation (both tags must not be separated and the order cannot be inverted):
example: “my text1, **NOTIFIED_BY_SL_VIA_WS**, myNewSLObject, my text2”
- Left click on **Save¹²**.

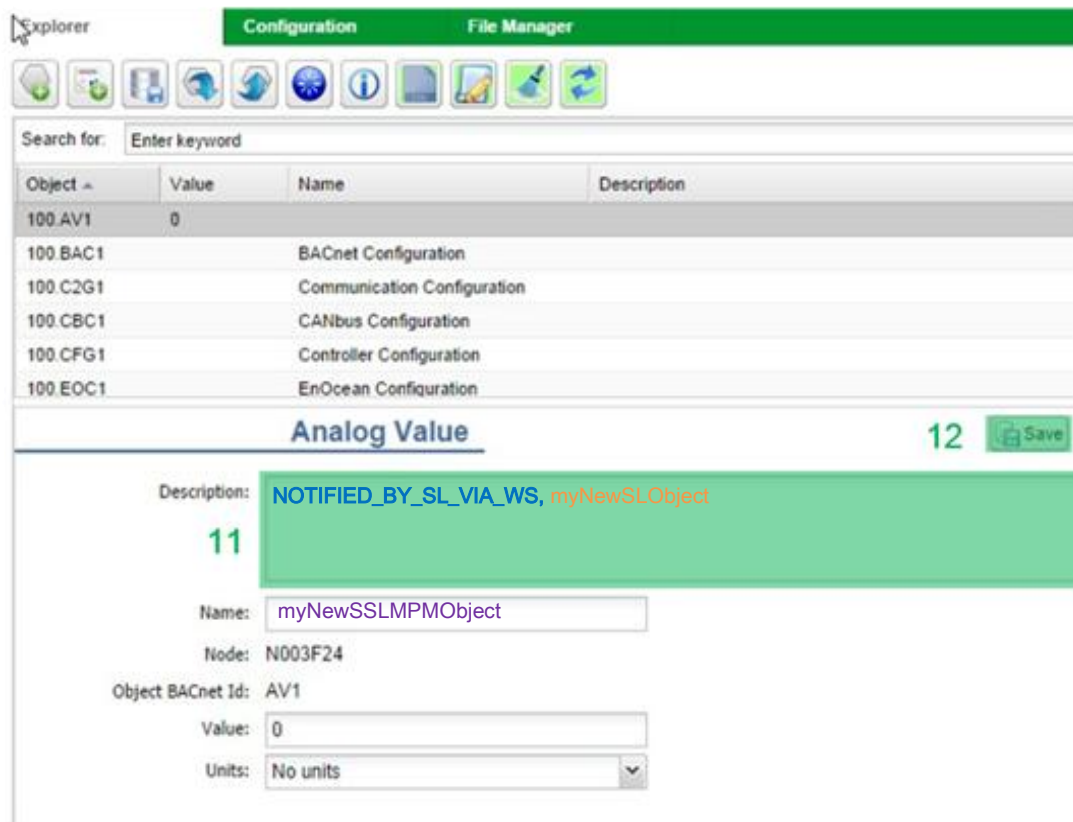


Figure 39: Tag the SSL-MPM object

In the SL configurator part:

- Left click on **Objects¹** Tab.
- Left click on object you selected before: **myNewSLObject²** for new object example (or **myExistingSLObject** for existing object example)
- In **Tags³** field, you have to write:
 - For new created or existing object in SL : “**NOTIFIED_TO_SSL_MPM_VIA_WS, 100.AV1**”
(100.AV1 corresponds to object name ID in SSL-MPM for object you selected)

note:

- the first tag **NOTIFIED_TO_SSL_MPM_VIA_WS** is case sensitive
- if the **Tags³** field already contains **TAG(s)** or you want to add some, the **TAG(s)** may be before and/or after the both tags with a comma as separation (both tags must not be separated and the order cannot be inverted)

example: “**TAG1, NOTIFIED_TO_SSL_MPM_VIA_WS, 100.AV1, TAG2**”

- Left click on **Save⁴**

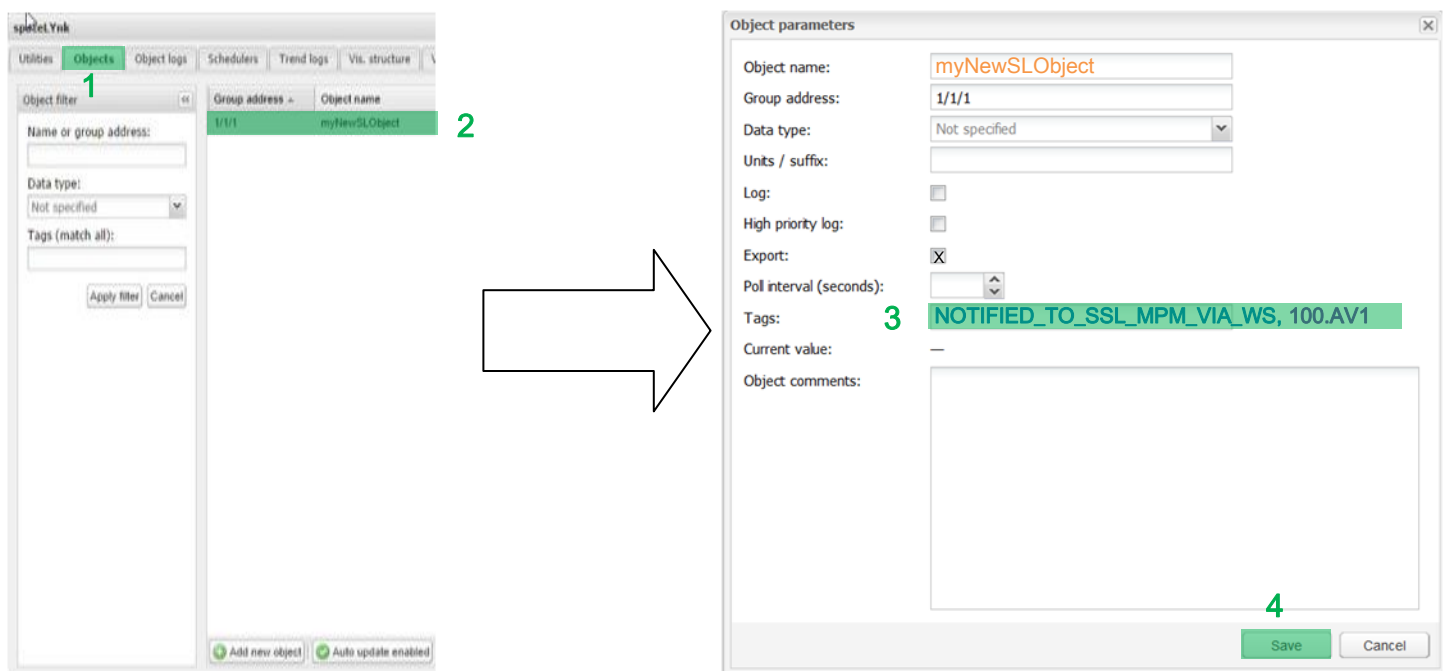


Figure 40:spaceLYnk object parameters

The object sharing of the spaceLYnk object with a SmartStruxure™ Lite MPM one is finished.

3.3.2 Verification of spaceLYnk => SmartStruxure™ Lite MPM objects synchronization

To validate the link between SSL-MPM and spaceLYnk, you have to change object value in spaceLYnk and to check the object value in SSL-MPM changes: if the object value in SSL-MPM changes and is same to spaceLYnk, the link is OK.

In the SL objects part:

- Left click on **Objects¹** Tab
- Select your object and left click on **Set value²**

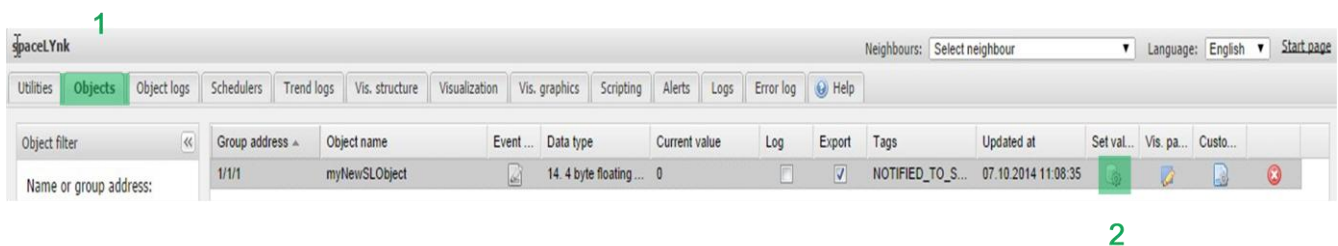


Figure 41: Select an object in spaceLYnk

- Change **New value¹** : set for example: **25**
- Left click on **Save²**

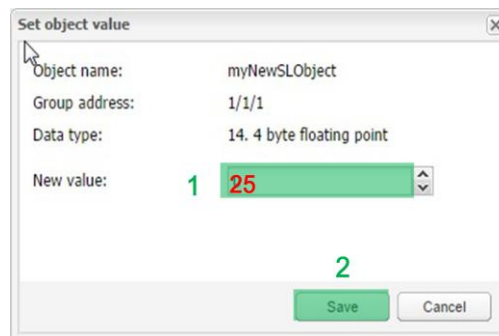
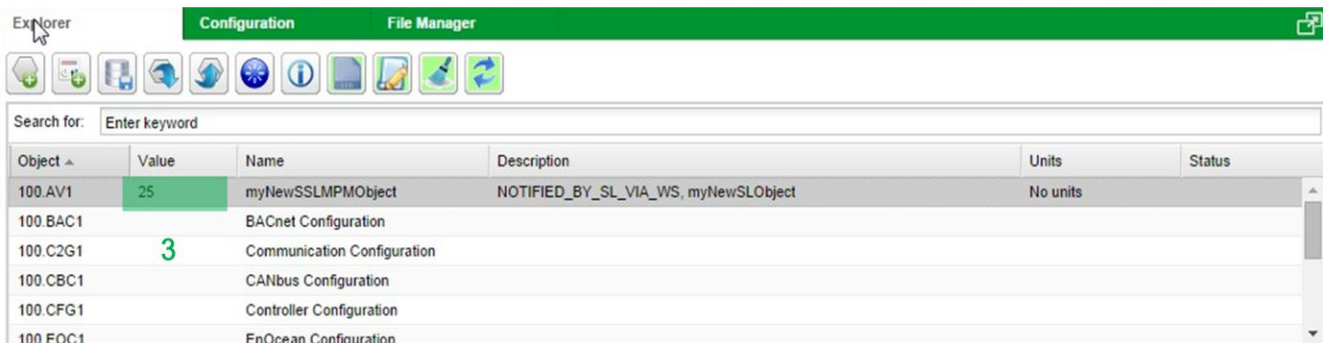


Figure 42: Change an object value in spaceLYnk

In SSL-MPM manager home page:

- Check if the **Value³** is the same than spaceLYnk one: if it is ok it means your configuration is ok



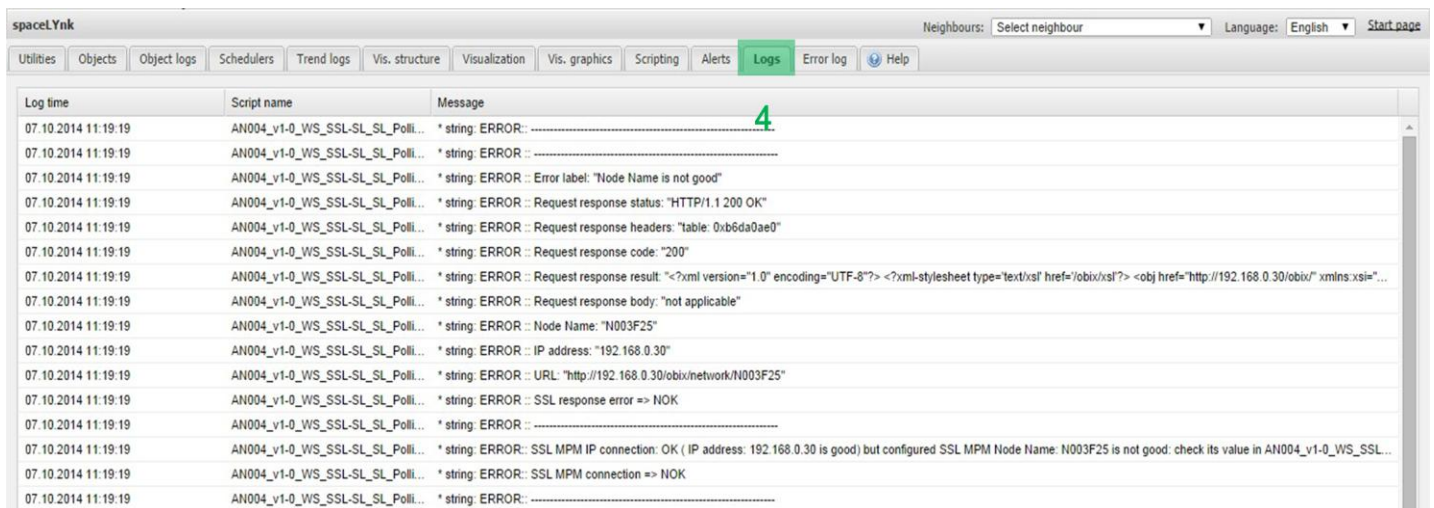
Object	Value	Name	Description	Units	Status
100.AV1	25	myNewSSLMPMObject	NOTIFIED_BY_SL_VIA_WS, myNewSLObject	No units	
100.BAC1	3	BACnet Configuration			
100.C2G1		Communication Configuration			
100.CBC1		CANbus Configuration			
100.CFG1		Controller Configuration			
100.EOC1		EnOcean Configuration			

Figure 43: Check the object value in SSL-MPM

Error cases :

If the SL object value is not replicated in SSL-MPM object: you may find help in SL logs

- Have a look in **Logs⁴** tab of SL and read the text
 - For instance, you may have written a bad name for object tag in SL: check the name at both side (SSL-MPM & SL), correct if necessary and retry.



Log time	Script name	Message
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR:
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR:
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: Error label: "Node Name is not good"
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: Request response status: "HTTP/1.1 200 OK"
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: Request response headers: "table: 0xb6da0ae0"
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: Request response code: "200"
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: Request response result: "<?xml version='1.0' encoding='UTF-8'> <?xmlstylesheet type='text/xsl' href='obix/xsl?'> <obj href='http://192.168.0.30/obix/' xmlns:xsl='..."
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: Request response body: "not applicable"
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: Node Name: "N003F25"
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: IP address: "192.168.0.30"
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: URL: "http://192.168.0.30/obix/network/N003F25"
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: SSL response error => NOK
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR:
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: SSL MPM IP connection: OK (IP address: 192.168.0.30 is good) but configured SSL MPM Node Name: N003F25 is not good: check its value in AN004_v1-0_WS_SSL...
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR: SSL MPM connection => NOK
07.10.2014 11:19:19	AN004_v1-0_WS_SSL-SL_SLPoli...	* string: ERROR:

Figure 44: Log example with Node Name error in spaceLYnk

If you see no error in the SL **Logs**⁴ tab and the object sharing is always not OK:

you can have a look in **Error log**⁵ tab:

- If you have logs in this tab for one of AN004_v1-0_WS_SSL-SL_xxxx scripts
 - we recommend you to remove all object sharing you did and to do it again
 - if after the new object sharing configuration, the object sharing is always not operational, please contact your country Schneider Electric support

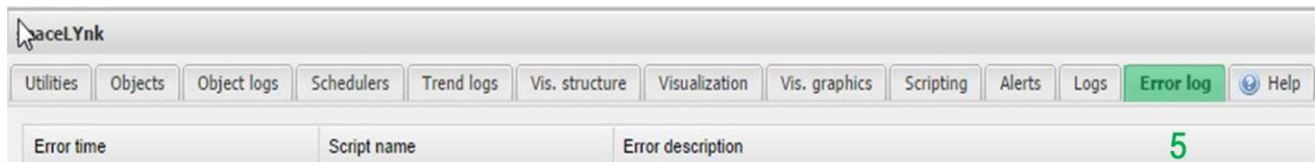


Figure 45: spaceLYnk Error log Tab

3.3.3 Share a SmartStruxure™ Lite MPM object with a spaceLYnk one

It is possible to share some SSL-MPM objects with full data type compliance using SL following types:
{"14. 4 byte floating point", "16. 14 byte ASCII string", "01. 1 bit (boolean)"}

Here is the fully compliant mapping table between SSL-MPM and SL data types:

SSL-MPM data types	SL data types
AI, AV	14. 4 byte floating point 16. 14 byte ASCII string
BI, BV	01. 1 bit (boolean)

SSL-MPM has less data types than SL has, you can share SSL-MPM data types such AI, AV in other SL specialized data types but you have to check the AI or AV object values you share in SL will not overflow the chosen SL data type as described in following mapping table:

SSL-MPM data types	Limited range for SSL-MPM object	SL data types
AI, AV	0 to 4	02. 2 bit (1 bit controlled)
AI, AV	0 to 16	03. 4 bit (3 bit controlled)
AI, AV	One ASCII character	04. 1 byte ASCII character
AI, AV	0 to 255	05. 1 byte unsigned integer
AI, AV	-128 to 127	06. 1 byte signed integer
AI, AV	0 to 65 535	07. 2 byte unsigned integer
AI, AV	-32 768 to 32 767	08. 2 byte signed integer
AI, AV	-671 088.64 to 670 760.96	09. 2 byte floating point
AI, AV	0 to 16777215	3 byte unsigned integer
AI, AV	<i>mapping is not possible</i>	10. 3 byte time / day
AI, AV	<i>mapping is not possible</i>	11. 3 byte date
AI, AV	0 to 4 294 967 295	12. 4 byte unsigned integer
AI, AV	-2 147 483 648 to 2 147 483 647	13. 4 byte signed integer
AI, AV	<i>mapping is not possible</i>	15. 4 byte access control

Note: in case you map a SSL-MPM AI or AV object in SL one with an out of range value, SL value will be set to its minimum or maximum. Ex: AV = 5005 => 06. 1 byte signed integer SL object = 127

In addition to tag principle, the created or existing object in SSL-MPM will have a second tag corresponding to the object name in SL:

For example:

- SL object name = **mySLObject**
- => SSL-MPM object
 - name = mySSLMPMObject
 - TAG: **mySLObject**

Here is the general process to implement a SSL-MPM to SL object sharing:

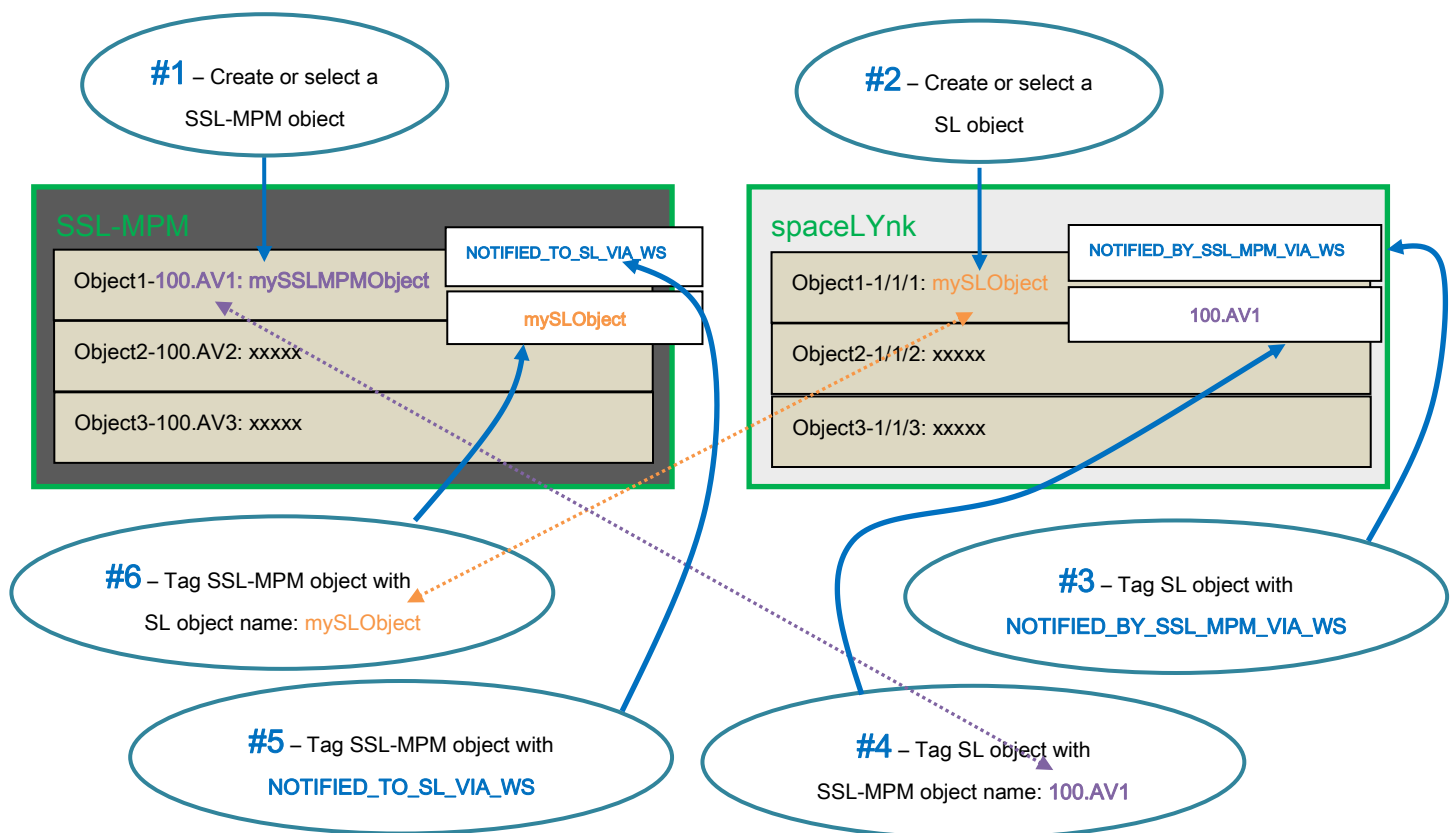


Figure 46: General process to implement a SSL-MPM to SL object sharing

Following parts will detail how to configure SSL-MPM and SL to set those different steps.

In SSL-MPM manager home page:

- If you want to create a new object:
 - Right click on **SmartStruxure™ Controller¹** and left click on **Add Objects²**

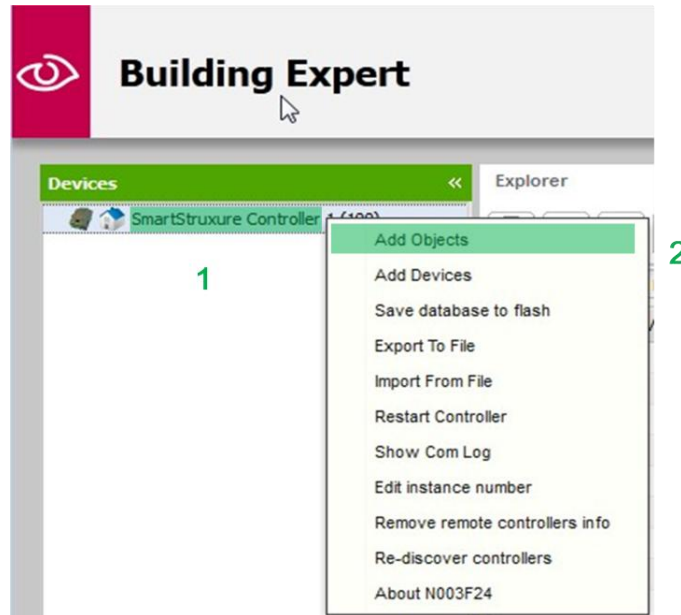


Figure 47: SSL-MPM manager home page

- Select **Filter³** corresponding to a type compliant with SL one
(Possible cases: **Analog Input** : "AI", **Analog Value** : "AV", **Binary Input** : "BI", **Binary Value** : "BV", see table before for more information)
Example: **Analog Value**

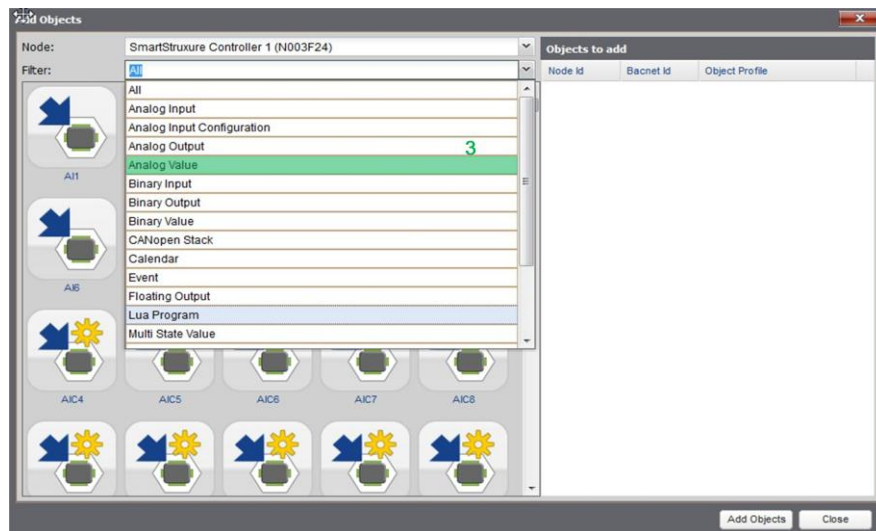


Figure 48: Create an Analog Value object in SSL-MPM, step 1

- Left click on an available **Object⁴**: example **AV2**
- Left click on **Add Objects⁵**

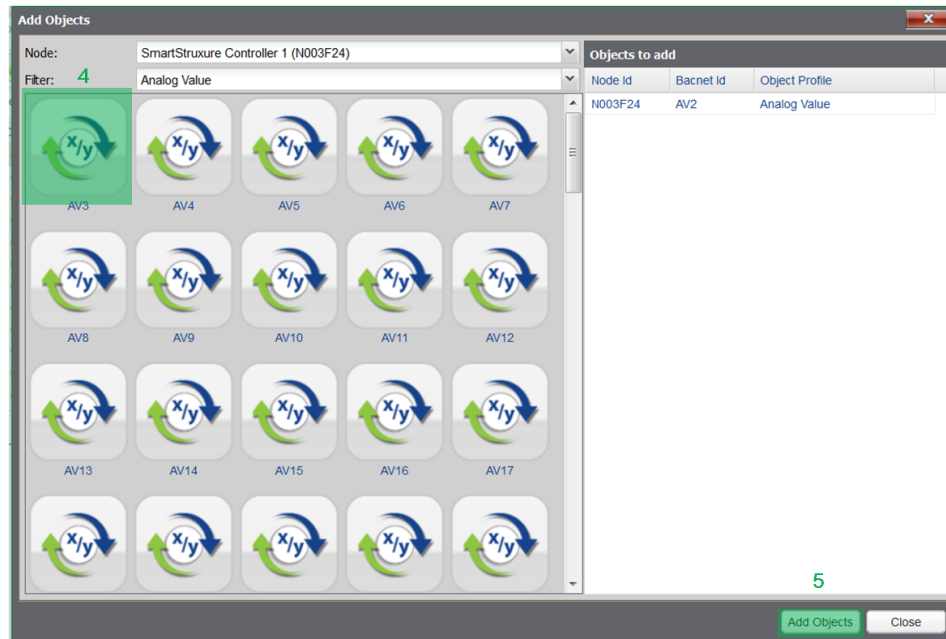


Figure 49: Create an Analog Value object in SSL-MPM, step 2

- Left click on the **Object⁶** to select the one you created (scroll window with the **scroll bar⁷** if necessary)

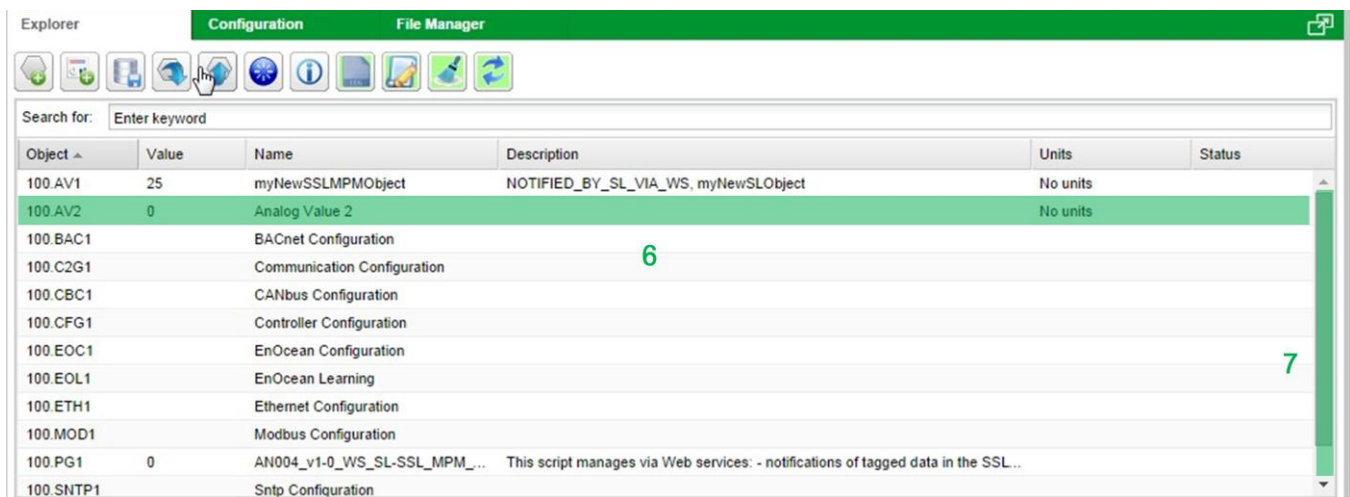


Figure 50: Create an Analog Value object in SSL-MPM, step 3

- In **Name⁸** field, set the name you choose for the object
example: **myNewSSLMPMObject1** *(it is possible to use an object name with space character(s))*
- Left click on **Save⁹**

The screenshot shows a software interface with a green header bar containing 'Explorer', 'Configuration', and 'File Manager'. Below the header is a toolbar with various icons. A search bar labeled 'Search for: Enter keyword' is present. A table lists objects with columns: Object, Value, Name, Description, Units, and Status.

Object	Value	Name	Description	Units	Status
100.AV1	25	myNewSSLMPMObject	NOTIFIED_BY_SL_VIA_WS, myNewSObject	No units	
100.AV2	0			No units	
100.BAC1		BACnet Configuration			
100.C2G1		Communication Configuration			
100.CBC1		CANbus Configuration			
100.CFG1		Controller Configuration			

Below the table is a section titled 'Analog Value' with buttons for 'Save', 'Import', and 'Export'. The 'Description' field contains the number '9'. The 'Name' field contains 'myNewSSLMPMObject1' with a green highlight and a green '8' next to it. The 'Node' field contains 'N003F24'. The 'Object BACnet Id' field contains 'AV2'. The 'Value' field contains '0'. The 'Units' field is a dropdown menu set to 'No units'.

Figure 51: Create an Analog Value object in SSL-MPM, step 4

In the SL configurator part:

- Left click on **Objects¹** Tab
- If you want to create a new object:
 - Left click on **Add new object²**, Set the **Object name³**:
example: **myNewSLObject1** (*it is possible to use an object name with space character(s)*)
 - Set the **Group address⁴**, Select the **Data type⁵**, Set the **Export⁶** flag
- If you want to use a existing object:
 - Left click on the object you want the SSL-MPM to notify:
example: **myExistingSLObject1** (*it is possible to use an object name with space character(s)*)
 - Set the **Export⁶** flag
- In **Tags⁷** field, you have to write:

- For new created or existing object in SL : “**NOTIFIED_BY_SSL_MPM_VIA_WS, 100.AV2**”
(100.AV2 corresponds to object name ID in SSL-MPM for object you selected)

note: - the first tag **NOTIFIED_BY_SSL_MPM_VIA_WS** is case sensitive

- if the **Tags⁷** field already contains **TAG(s)** or you want to add some, the **TAG(s)** may be before and/or after the both tags with a comma as separation (both tags must not be separated and the order cannot be inverted):

example: “**TAG1, NOTIFIED_BY_SSL_MPM_VIA_WS, 100.AV2, TAG2**”

- Left click on **Save⁸**

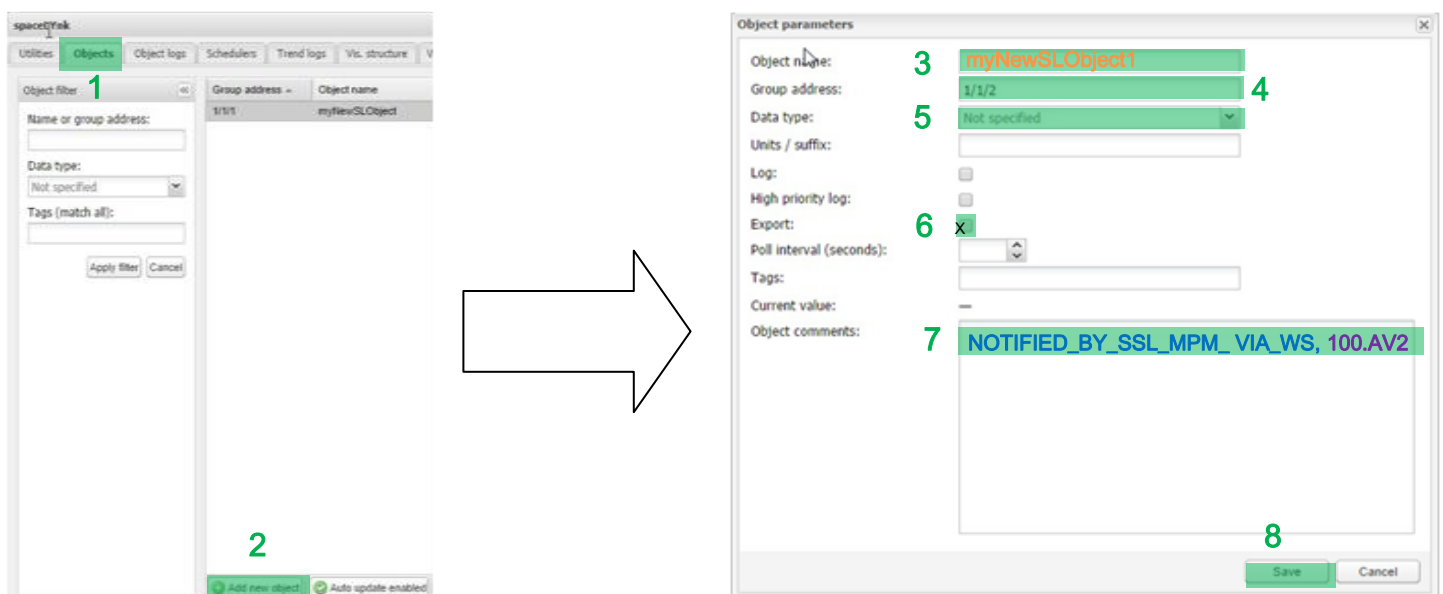


Figure 52: spaceLYnk object Tab

In SSL-MPM manager home page:

- Left click on the object to select the one you want to notify to SL:
example: `myNewSSLMPMObject1` (it is possible to use an object name with space character(s))
- In **Description**¹⁰ field, you have to write:
 - For new created object example in SL : `"NOTIFIED_TO_SL_VIA_WS, myNewSLObject1"`note:
 - the first tag `NOTIFIED_TO_SL_VIA_WS` is case sensitive
 - the second tag, object name in SL (example: `myNewSLObject1`) is case sensitive
 - if the **Description**¹⁰ field already contains `text` or you want to add one, the `text` has to be before and/or after the both tags with a comma as separation (both tags must not be separated and the order cannot be inverted):
example: `"myText1, NOTIFIED_TO_SL_VIA_WS, myNewSLObject1, myText2"`
- Left click on **Save**¹¹

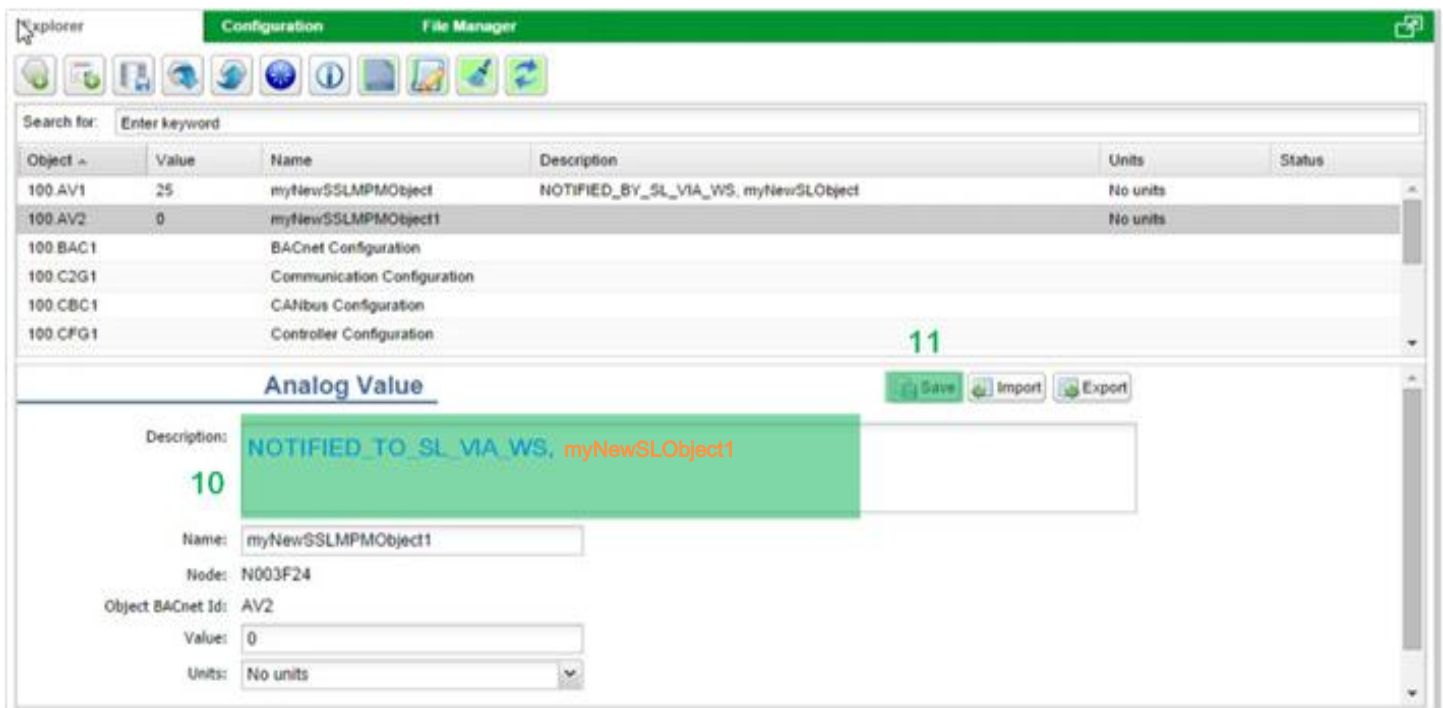


Figure 53: Tag the SSL-MPM object

The object sharing of the SmartStruxure™ Lite MPM object with a spaceLYnk is finished.

3.3.4 Verification of SmartStruxure™ Lite MPM => spaceLYnk objects synchronization

To validate the link between spaceLYnk and SSL-MPM, you have to change object value in SSL-MPM and to check the object value in SL changes: if the object value in SL changes and is same to SSL-MPM, the link is OK.

In SSL-MPM manager home page:

- Left click on the **Object¹** to select your object
- Change **Value²**: set for example: **35**
- Left click on **Save³**

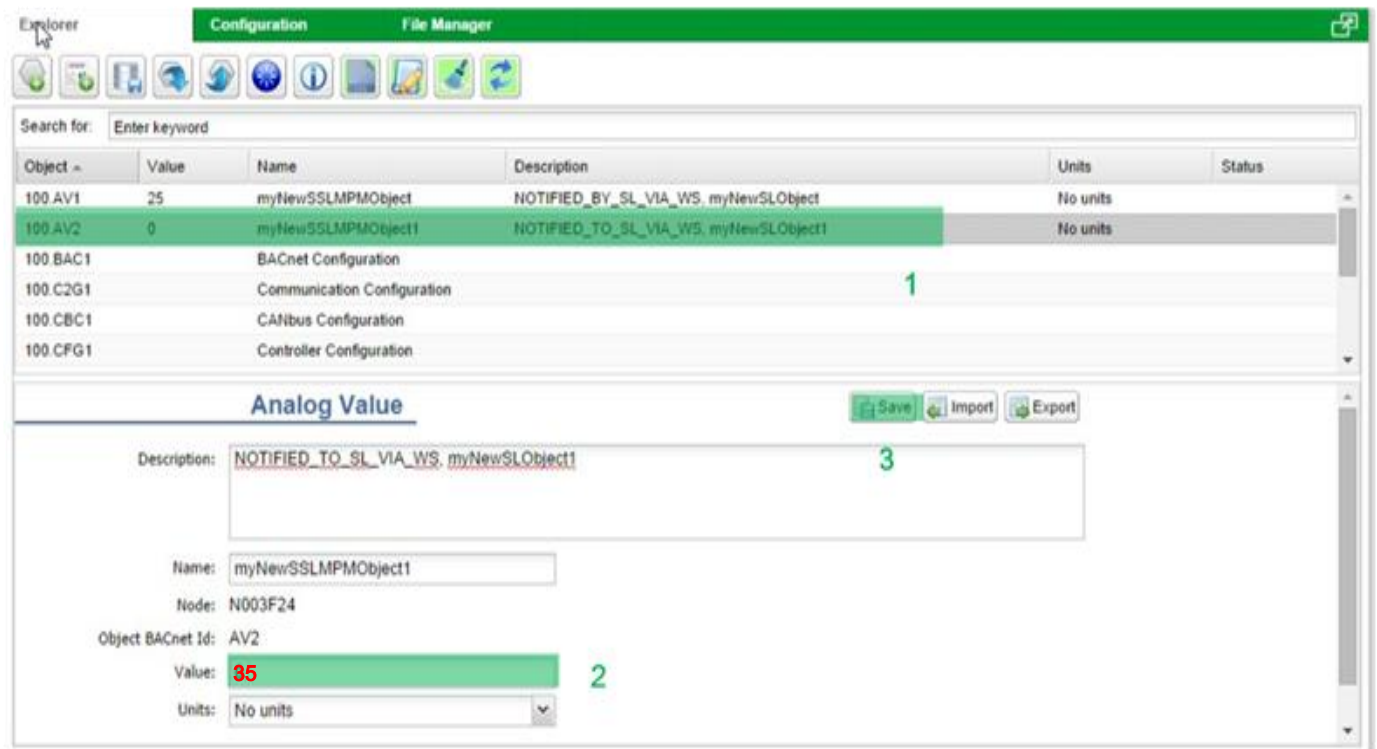


Figure 54: Select and change value object in SSL-MPM

In the SL objects part:

- Check if the **Current value¹** has changed to same one you set at SSL-MPM side

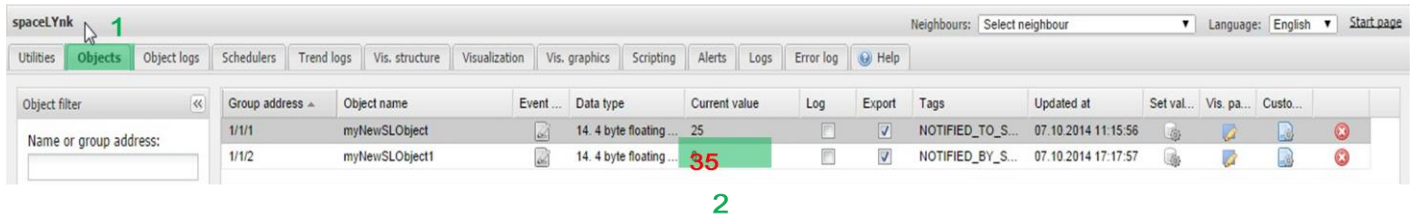


Figure 55: Check object value in spacelynk

Error cases :

If the SSL-MPM object value is not replicated in SL object: you may find help in SSL-MPM logs.

To view SSL-MPM logs, go in SSL-MPM manager home page:

- Left click on **100.PGx¹** corresponding to script named “*AN004_v1-0_WS_SL-SSL_MPM_script*”

(scroll window with the **scroll bar⁷** if necessary)

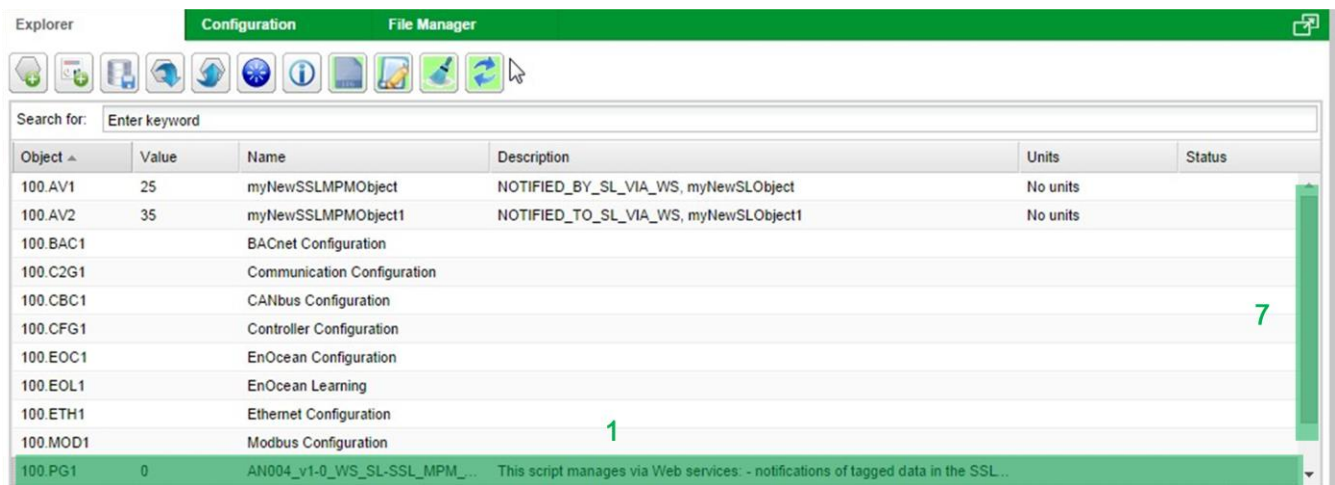


Figure 56: SSL-MPM home page with 100.PG1 script

- Select **Script²** Tab in Lua Program frame

- Left click on **Output³**



Figure 57: View 100.PG1 script logs

- Read the text and check if there is log with “ERROR ::”
 - For instance, you may have written a bad name for object tag in SSL-MPM: check the name at both side (SSL-MPM & SL), correct if necessary and retry

```

Building Expert - Google Chrome
192.168.0.30/external.html
OR::
2013-10-30 17:07:48 ERROR :: Notification for myNewSLObjct2 to spaceLYnk => NOK
2013-10-30 17:07:49 ERROR ::
2013-10-30 17:07:49 ERROR :: spaceLYnk response error => NOK
2013-10-30 17:07:49 ERROR :: URL: "http://remote:remote@192.168.0.50/cgi-bin/scada-remote/request.cgi?m=json&r=grp&fn=write&alias=myNewSLObjct2"
2013-10-30 17:07:49 ERROR :: IP address: "192.168.0.50"
2013-10-30 17:07:49 ERROR :: Remote access login: "remote"
2013-10-30 17:07:49 ERROR :: Remote access password: "remote"
2013-10-30 17:07:49 ERROR :: Request response result: "nil"
2013-10-30 17:07:49 ERROR :: Request response code: "200"
2013-10-30 17:07:49 ERROR :: Request response headers: "table: 22326518"
2013-10-30 17:07:49 ERROR :: Request response status: "HTTP/1.1 200 OK"
2013-10-30 17:07:49 ERROR :: Error label: "Failure to decode json message"
2013-10-30 17:07:49 ERROR ::
2013-10-30 17:07:49 ERROR ::
2013-10-30 17:07:52 ::
2013-10-30 17:07:52 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-30 17:07:52 ::
2013-10-30 17:07:56 ::
2013-10-30 17:07:56 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-30 17:07:56 ::
2013-10-30 17:08:00 ::
2013-10-30 17:08:00 :: spaceLYnk connection => OK ( IP address: 192.168.0.50 )
2013-10-30 17:08:00 ::

```

Figure 58: Log example with error in SSL-MPM

4 Conclusion

The presented Web service solution is a one-to-one and bi-directional solution enabling to easily share data between a spaceLYnk and SmartStruxure™ Lite Multi-Purpose Managers.

Once you installed at both controllers framework scripts, you may share as many objects you want.

Solution limitations:

- the solution is designed to share data between only one SSL-MPM and one SL
- the solution has been validated with hardware architecture presented in 2.2, and with 10 objects sharing too; so we cannot prevent you from some possible limitations if you use a quite bigger number of objects sharing with other tasks in spaceLYnk and/or SmartStruxure™ Lite products
- the solution has a transfer time (~1s) to replicate object data change from SSL-MPM to SL direction which may be a limitation, for instance to switch on/off a light with a push-button at SSL-MPM side toward a KNX actuator at SL side
- the SSL-MPM and SL are permanently connected by an Ethernet/IP connection but this connection may be broken, for instance if the Ethernet wire is disconnected; the solution software script in SSL-MPM will detect the failure through a polling function with a timer of 3 seconds.

As whole SSL-MPM LUA scripts are executed in a serial mode, a network failure between SL and SSL-MPM will delay all other SSL-MPM scripts executions during 3 seconds each 4 seconds. This may lead to others feature dysfunctions in other SSL-MPM LUA scripts if those features require cyclic real-time executions.

5 Appendix

5.1 Glossary

The following table describes the acronyms and defines the specific terms used in this document.

Abbreviation	Description
SSL	SmartStruxure™ Lite
SL	spaceLYnk
LAN	Local Area Network
MPM	Multi-Purpose Manager
JSON	JavaScript Object Notation is an open standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs.
HTTP	Hyper Text Transfer Protocol is an application protocol for distributed, collaborative, hypermedia information system.
IP	Internet Protocol
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
WS	Web Service
WSDL	Web Services Description Language
XML	Extensible Markup Language

Schneider Electric Industries SAS

Head Office

35, rue Joseph Monier

92506 Rueil-Malmaison Cedex

FRANCE

www.schneider-electric.com