# SEMA Cloud How To

For ADLINK internal use only!

For external use: Remove all references to “ILS” and “DeviceWise” and rework the layout! ☺

## Control Software Installation (DeviceWise Workbench)

The DeviceWise Workbench is a comfortable tool to remotely configure and control DeviceWise Gateway installations. The Workbench installation package (only for Windows at the moment) can be found at \\Fileserver2\Products\General\SEMA-Cloud (ILS)\Installation Packages\Workbench\ or in Alfresco. All path and filenames refer to the fileserver at Mannheim, the file structure on Alfresco should be similar.

Currently, the Workbench with the original ILS DeviceWise design is available. Later, an ADLINK branded version will be provided by ILS upon request.

The Workbench is available for Windows, Linux in various flavors and other operating systems. The operating system of the machine running the Workbench is independent of the operating systems of the clients to control. That means that a Workbench installed on a WindowsXP machine can connect to and control Gateway installations running under Linux (x86 and iMX6) and Windows. The only condition is that the client machine is reachable over the network from the Workbench and no firewall is blocking the traffic.

## Client software installation under Windows (x86) and Linux (x86 and iMX6)

The DeviceWise Gateway is the central software component that runs on the client, has access to all local interfaces (using plugins) and handles the connection to the cloud server. The Gateway is available for various flavors of Windows and Linux. For accessing the Gateway through a plugin, the SEMA software must be installed first.

1. Install “normal” SEMA package (tested with SEMA 2.5 R4). This enables the use of GUI and command line tool for verification if the BMC is accessible. Make sure that the libsema (and related) DLLs resp. SOs are in the search path.
2. Install the Gateway software on the client. The part of the filename printed in bold represents the software version, replace by newer version if available from ILS.

Under Linux (x86 and iMX6) the Gateway is always installed to /opt/dw, the start/stop script is /etc/software/init.d/devicewise.

Under Windows, C:\deviceWISE\ is assumed as installation directory for this document.

The DeviceWise Gateway software installation packages are stored at \\fileserver2\Products\General\SEMA-Cloud (ILS)\Installation Packages\Gateway

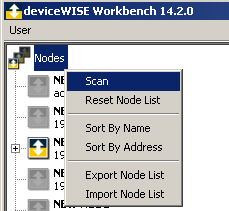
|  |  |
| --- | --- |
| **Operating system** | **Installation package file name** |
| Linux (x86) | DWGateway\_Install.Linux-X86-Generic.**14\_2\_0-011**.tar.gz |
| Linux (iMX6) | DWGateway\_Install.Linux-ARM-Raspbian.**14\_1\_0-032**.tar.gz |
| Windows | DWGateway\_Install.Windows\_NT.**14\_2\_0-011**.exe |

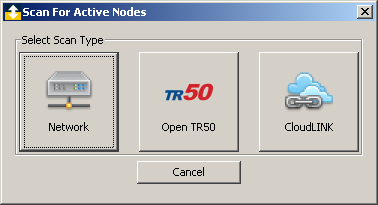
1. Install the SEMA Device Plugin. A compiled version is stored at \\fileserver2\Products\General\SEMA-Cloud (ILS)\Installation Packages\Plugin\$PLATFORM
   1. Linux (x86 and iMX6): Copy libdwadlink.so to /opt/dw/dwcore/plugins/ and make sure that the executable bit is set.
   2. Windows: Copy dwadlink.dll to C:\deviceWISE\Gateway\dwcore\plugins\

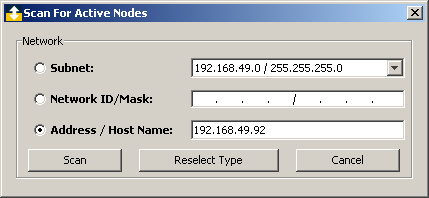
If you need to add some features or recompile the plugin, please refer to the chapter “Compiling the plugin” (TODO).

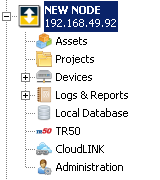
## Configuring an new client

This chapter assumes that a Workbench is available and the steps from chapter “Client software installation” are successfully completed. The screenshots show the Windows based variant but a Workbench running under $YOUR\_FAVORITE\_OS should work as well.

1. **Identify the board:** Find out which IP address and MAC address are used by the client’s network interface that is connected to the local area network (Windows: “ifconfig”, Linux: “ipconfig”).
2. **Add board to the Workbench:**
   1. In the left area of the Workbench, right-click on “Nodes”, then left-click on “Scan”:
   2. Click on „Network“:

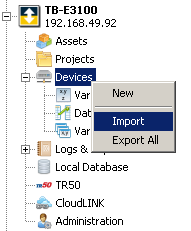


* 1. Enter the client’s IP address (e.g. 192.168.49.92) in the “Address / Host Name:” data field or select the correct Subnet (e.g. 192.168.49.0 / 255.255.255.0) if the Workbench and the client are located in the same local area network. 
  2. A new entry in the nodes list should appear:

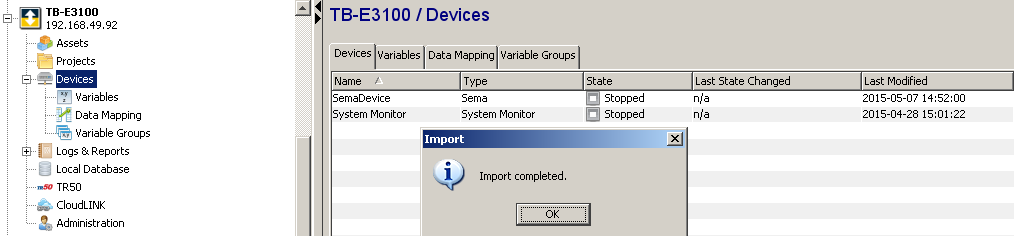


* 1. Go to “Administration”->”Node Administration”. Verify that the “MAC Address” and IP Address match the data from step 1 (in other words: Make sure that you’re looking at the right node). If so, enter a descriptive name and press the “Save Details” button.

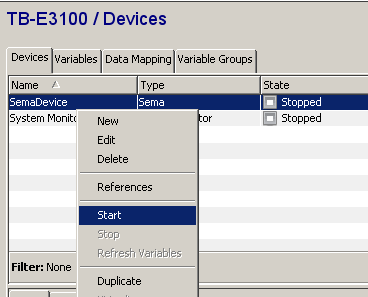
1. **Add a license:** Go to “Administration”->”Licenses”, press the “Import” button and select a valid license file. Currently, there are two types of license files:
   1. MAC independent: Can be installed to any node but the licenses have a short expiration period. The latest license file of this type was created on 2015-04-28 and has an expiration period of 60 days (\\Fileserver2\Products\General\SEMA-Cloud (ILS)\Installation Packages\License files\licenses-04282015 (60days MAC independent).txt). A new MAC independent license file can be obtained from ILS upon request.
   2. Fixed MAC: License files for the Q7-BT and the NuPro demo boards with a one year expiration period are available at \\Fileserver\Products\General\SEMA-Cloud (ILS)\Installation Packages\License files\
   3. Future scenario: If a SEMA cloud installation is deployed, the clients will get their licenses automatically from the server.
2. **Add SEMA support:** Some steps are required to enable BMC access for the gateway.
   1. Right-click on “Devices”, select “Import”.



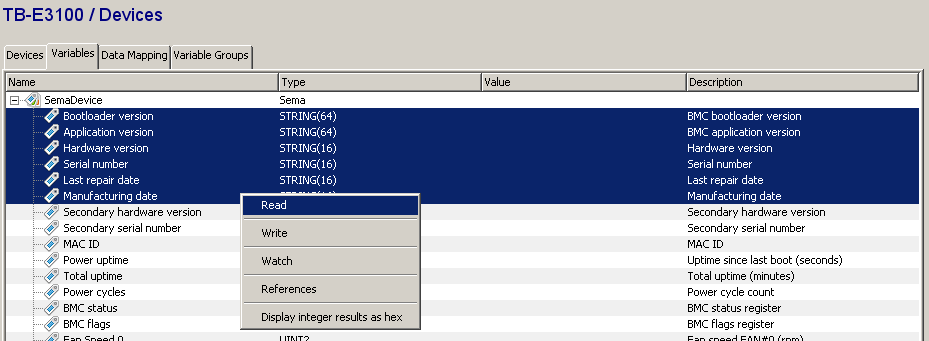
* 1. Chose file “\\Fileserver2\Products\General\SEMA-Cloud (ILS)\Installation Packages\Gateway\Device\_SemaDevice.dwx” and press “Import”. If the operation was successful, a message box “Import completed” should appear and the new device “SemaDevice” of type “Sema” is added to the devices list.

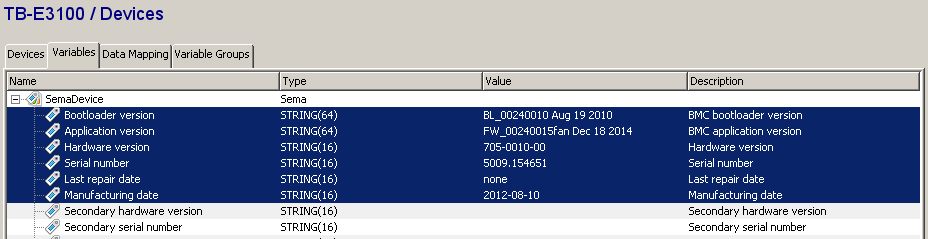


* 1. Right-click on “SemaDevice” and select “Start”. State should change from “Stopped” to “Started”.

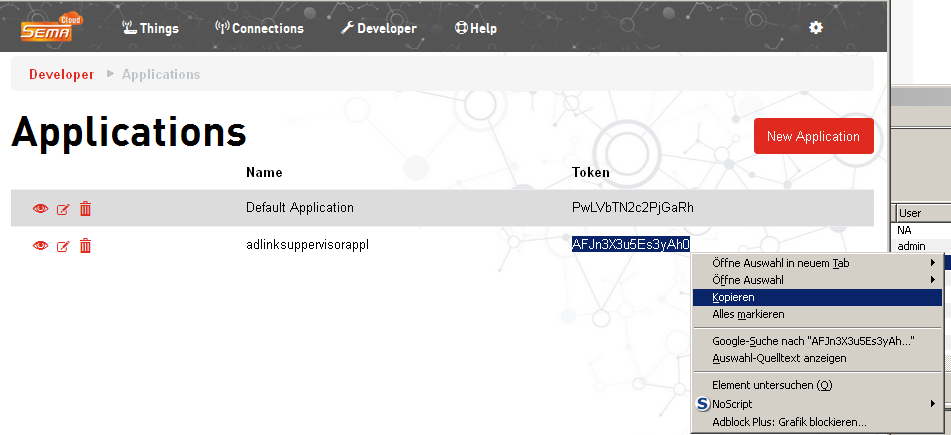


* 1. Verify operation: Change to the “Variables” tab, expand “SemaDevice”, select some variables, right-click and select “Read”.

The selected variables should be read from the SEMA library through the gateway plugin.



1. **Attach to an Organization:** Each node has to be attached to a specific organization (e.g. a customer’s account).
   1. Login to the management portal, switch organization if needed, click on “Developer” in the menu on top of the page then click “Applications” in the menu on the left. Locate the application called adlinksuppervisorappl (this stands for “ADLINK supervisor application” and contains important thing definitions) and copy the associated “Token” to the clipboard.



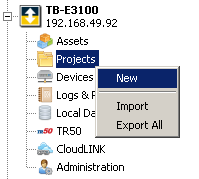
* 1. Go back to the Workbench, select the node and navigate to “TR50”->”TR50 Connection Management”. Enter the correct “TR50 Server Address”:
     1. Europe: api-de.devicewise.com
     2. USA: api.devicewise.com
     3. Asia: api-ap.devicewise.com
  2. Set “Connection Type” to “MQTT via HTTPS Proxy”, “Proxy Type” to “None”, “Thing Key” to “MAC Address”.
  3. Paste the “Application Token” from the management portal (step 5.a).
  4. Click “Save”, then “Start”. Wait a couple of seconds and then press “Refresh”.
  5. The Connection State and Status should change to “Started” resp. “Connected” and “Organization Key” and “Open Server” should contain something like “ADLINK” and “prod-de-open02”.
  6. Click on “Things” in the management portal and check if the thing appears as “Connected”.

## Creating and configuring projects and triggers

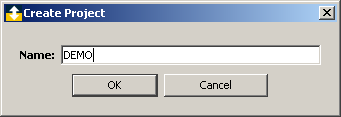
A trigger is the central element for publishing properties or attributes, calling functions taking actions and so on. Triggers can be grouped together in so called projects. Before we create any trigger, we must create a project first. Any node can contain an arbitrary number of projects, a project can contain any number of triggers. For practical applications it is strongly advised to maintain a well-structured layout of projects and triggers. This eases up the task of configuring a big number of identical nodes by simply transferring a complete project at once from one note to another.

### Creating a project

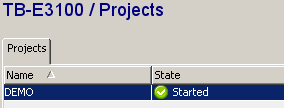
A fresh gateway installation comes without any projects. To create the first project, right-click on the “Projects” entry and select “New”:



Fill in a meaningful “Name” for the new project and press OK:



The new (empty) project appears in the projects list in a “stopped” state. Right-click on the project’s entry (anywhere in the line) and select “Start”. The state changes to “Started”:



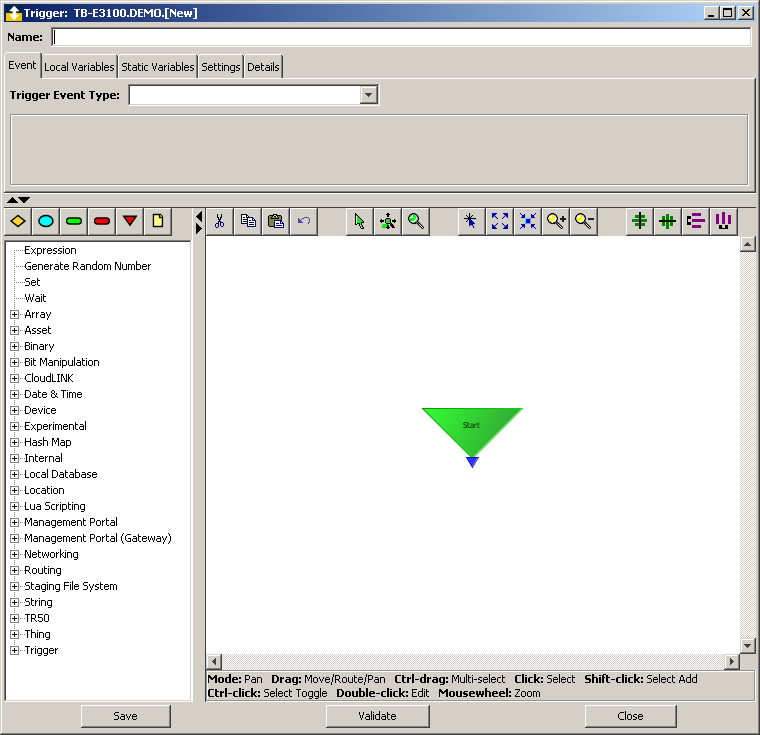
As long as we have not defined (and started) any triggers inside the project, it makes no difference if the state is started or stopped for now. But to avoid confusion later when we start a trigger inside a stopped project and nothing happens, we can start the project at this point.

Double-click on the project to open it. A new tab with the name of the project appears and a green round symbol indicates it’s state as started.



### Creating triggers

To add a new trigger to a project, navigate to the project view, right-click in the (empty) trigger list, select “New” and “use canvas Editor”. The canvas editor opens and the new trigger is ready to be configured.



### Attributes, properties and methods

An attribute is a key / value pair that is associated to a thing where no history is kept on server. A prominent application for an attribute is the serial number of the board on which the gateway is currently running. The serial number usually does not change during a board’s lifetime. Whenever the gateway software on the node starts, the serial number is read from BMC through the SEMA library and transmitted to the server. This ensures that the serial number displayed in the web portal matches the serial number of the board at all times. Since the serial number does not change, only the last transmitted value needs to be stored and any history is useless.

A property is a key / value pair that is associated to a thing with history to be kept on server. Each time a new key / value pair is published, it is stored in the thing’s database along with the timestamp of this publish event. The published values can be viewed in graphical form or as a table, downloaded as CSV and so on.

A method is TODO

#### Example: A trigger that publishes attributes whenever a node starts

Create a new trigger, enter a meaningful name and as “Trigger Event Type”, select “Internal”->”System”. Make sure the “Event Condition” reads “Node Start”.

From the action library on the left side, select “TR50”->”Send Raw Message” and click somewhere in the free space below the green “Start” triangle. A connection line from the small blue output arrow on the bottom of the start triangle to the blue input arrow on the top side of the “Send Raw Message” block should appear automagically. If not, connect these two symbols using your mouse.

Double-click on the “Send Raw Message” block and enter the following as “JSON Message:”

{

"SetSerNo":

{

"command":"thing.attr.set",

"params":

{

"key": "serno",

"value": "$(serno)"

}

},

"SetHWRev":

{

"command":"thing.attr.set",

"params":

{

"key": "hwrev",

"value": "$(hwrev)"

}

},

"SetMACID":

{

"command":"thing.attr.set",

"params":

{

"key": "macid",

"value": "$(macid)"

}

}

}

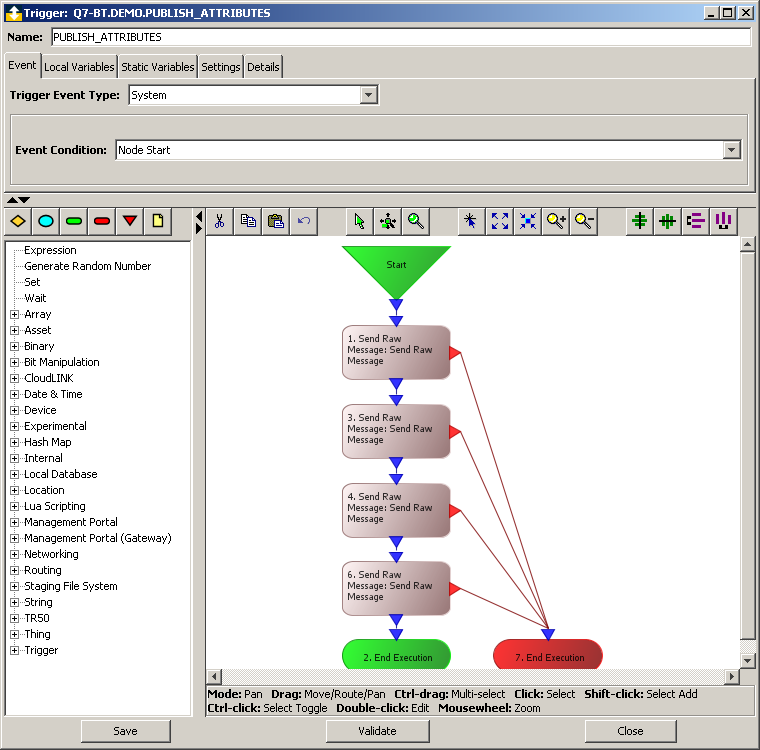
This JSON message consists of three similar blocks, each one transmits a separate attribute and has the following components:

* Message name (e.g. "SetSerNo" ): Can be any name, chose a meaningful and unique name for each block.
* Command ("command":"thing.attr.set"): Command for setting a thing’s attribute.
* Parameters: The thing.attr.set command requires two parameters, a “key” and a “value”. The key must match an attribute key in the thing definition (or tick “Auto def attributes” in the thing definition). And the value is the value to be set.

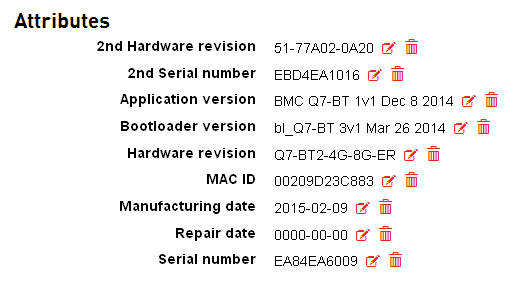
Please note the notation “$(serno)” for the “value” parameter. As soon as you add this string, the editor recognizes this as a variable name and adds an appropriate entry in the “Input” tab below. Select the line for the variable “serno”, open the drop-down view, click the plus symbol next to “SemaDevice” and select “Serial number”. The variable type (here: STRING(16) will be set automatically as this is the data type associated to “Serial number” by the SEMA plugin. Repeat these steps for the other two variables accordingly then close the configuration dialog and save the trigger.

A JSON message can handle up to three commands. If you need more, add more “Send Raw Message” blocks and configure them to your application specific needs.

The example trigger (Trigger\_DEMO\_PUBLISH\_ATTRIBUTES.dwx) uses four “Raw Messages” to send all nine (3+2+2+2) attributes whenever a node starts.

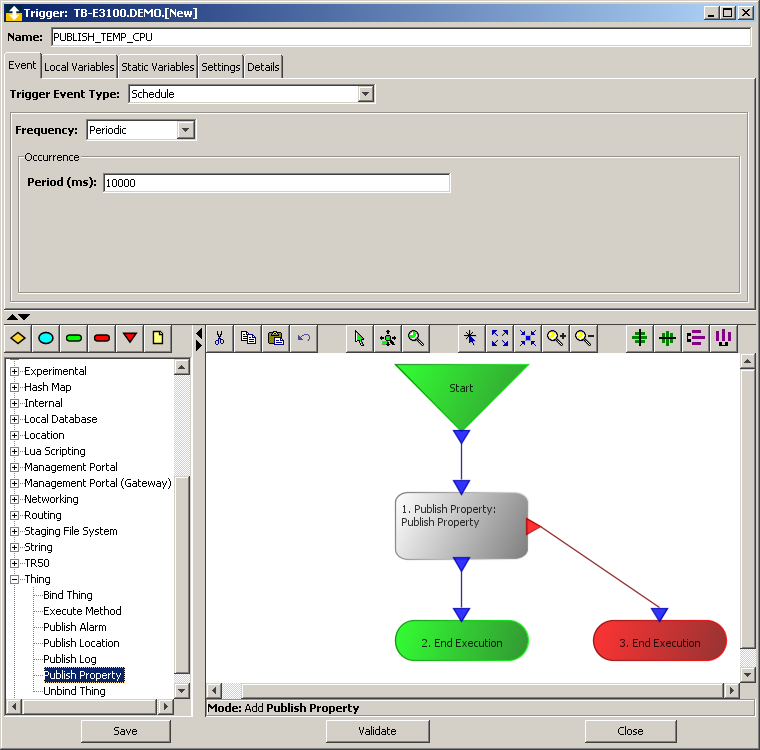


Here is the Demo-Q7-BT’s attributes view after successfully executing the trigger. Note that the order of attributes follows the alphabetical attribute name ordering and not the order in which the raw messages are sent.



#### Example: Add a trigger that publishes the CPU temperature every ten seconds

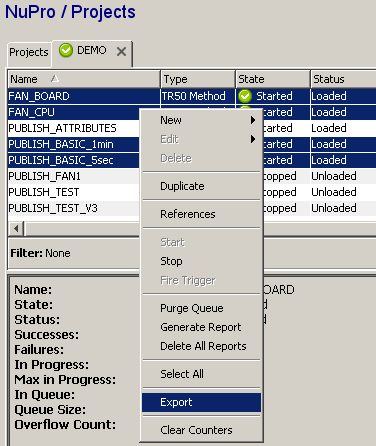
Create a new trigger, enter a meaningful name and as “Trigger Event Type”, select “Schedule”. Make sure that “Frequency” reads “Periodic” and “Period (ms)” is set to “10000”. Select “Thing” -> “Publish Property” from the menu on the left, place an action symbol on the canvas, add green (Success) and red (Failure) “End Execution” symbols and connect everything.



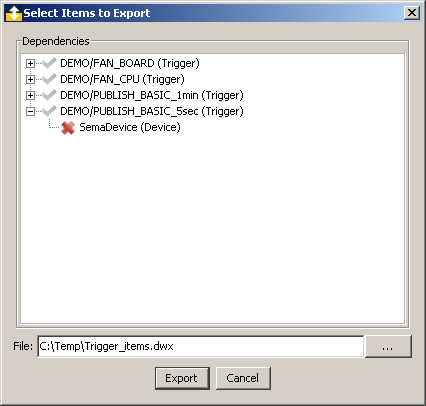
Double-click on the “Publish Property” box and click in cell “Value” of “Property”. Navigate to “Thing Definitions” -> “Adlink Supervisor Processor” -> “properties” and click on “CPU Temperature (C) (tempcpu)”. The string “tempcpu” is filled in as this is the actual property name assigned in the Things Definition. Repeat the process for “Value” of “Value” and select “SemaDevice” -> “CPU temperature”. Note that the data types are automatically filled in as defined in the plugin. Close the configuration dialog and save the trigger.

### Exporting and importing triggers

Copying a trigger, a set of triggers from a project, a whole project ore several projects from one node to another one is easy. Open the Source node in the workbench, select the entities to export and chose “Export” from the context menu.



The “Select Items to Export” dialog opens. Check the dependencies in the upper part of the dialog. You can exclude items you do not want to exclude on any level. In the example below, the “SemaDevice (Device)” entry is deselected because only the triggers shall be exported and not the Device the Triggers depend on. It is assumed that the SemaDevice already exists on the target node(s) and importing the file with SemaDevice definition will cause a conflict. Another possibility is to export Triggers and Device here and deselect the Device when importing the file later. All roads lead to Rome.



Navigate to the target node, right-click on “Projects” (as Triggers are subsets of projects), select “Import”, choose file, check items to import and hit “Import”. When everything worked as expected, you should see this message:

