IEC61850 Configuration from template

straton user guide – Rev. 5

sales@straton-plc.com

CP_R31_006_H

straton

Content

1. Overview

This document describes how to use the “IEC61850 Server incl. GOOSE From Templates” configurator and create configuration from the logic application.

2. Requirement and setup

straton Editor version 9.1 or higher is required.

The user of this tutorial should be familiar with straton Editor.

3. Configuring from template

3.1. Prerequisites

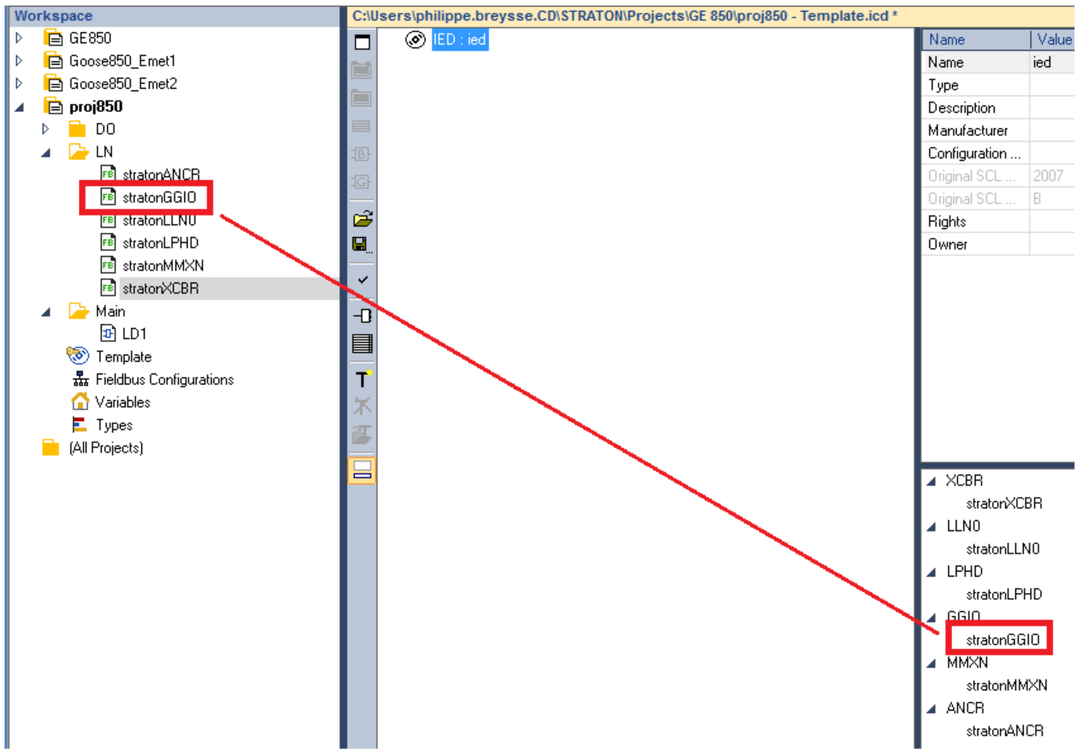
In order to use the IEC61850, the OEM should have provided:

"C" blocs or UDFBs (User Defined Function Blocs)

A template library for the IEC61850 Logical Nodes

Each Logical node in template file should have its corresponding "C" bloc or UDFB.

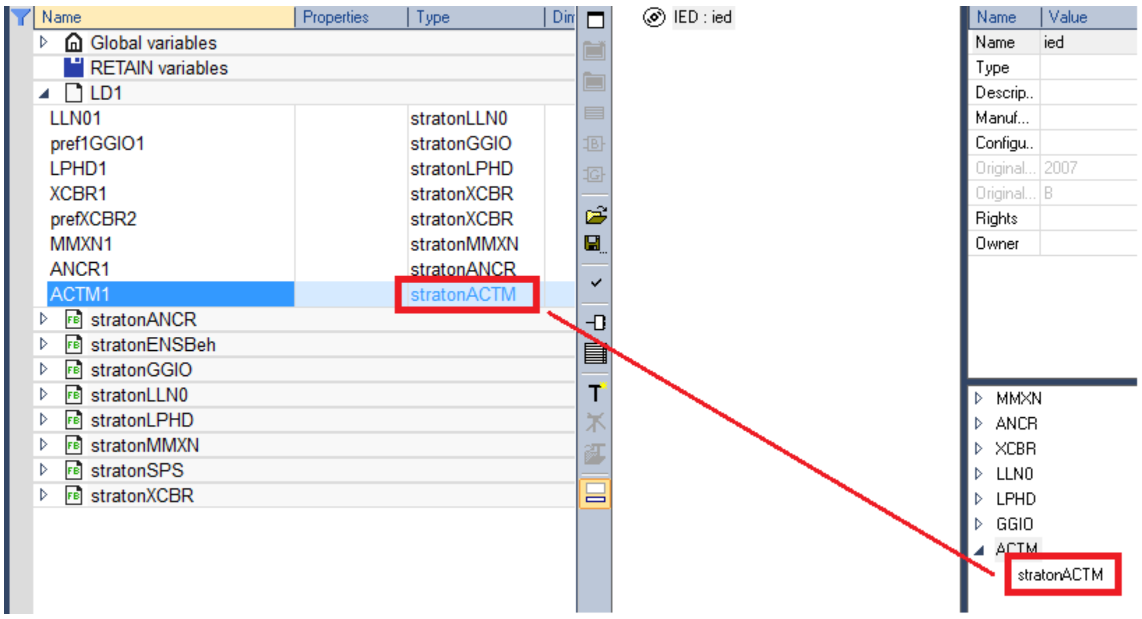
EXAMPLE WITH AN UDFB:


C:\Users\philippe.breysse.CD\STRATON\Projects\GE 850\proj850 - Template.icd *

Name	Value
Name	ied
Type	
Description	
Manufacturer	
Configuration ...	
Original SCL ...	2007
Original SCL ...	B
Rights	
Owner	

In SCL template file "Template" the user has declared the logical node "stratonGGIO". This corresponds to the UDFB "stratonGGIO".

EXAMPLE WITH A "C" FUNCTION BLOC:



Name	Properties	Type
Global variables		
RETAIN variables		
LD1		
LLN01		stratonLLN0
pref1GGIO1		stratonGGIO
LPHD1		stratonLPHD
XCBR1		stratonXCBR
prefXCBR2		stratonXCBR
MMXN1		stratonMMXN
ANCR1		stratonANCR
ACTM1		stratonACTM
stratonANCR		
stratonENSBeh		
stratonGGIO		
stratonLLN0		
stratonLPHD		
stratonMMXN		
stratonSPS		
stratonXCBR		

IED : ied

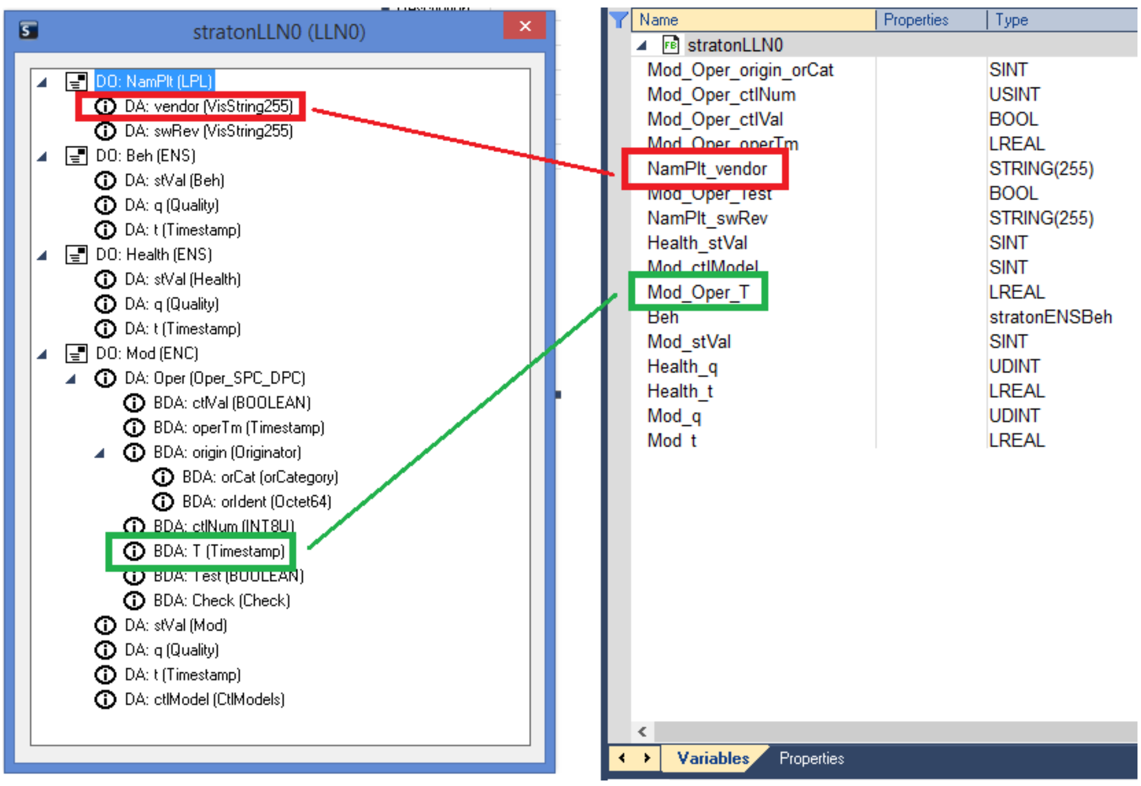
Name	Value
Name	ied
Type	
Descrip..	
Manuf...	
Configu..	
Original...	2007
Original...	B
Rights	
Owner	

MMXN
ANCR
XCBR
LLN0
LPHD
GGIO
ACTM
stratonACTM

In SCL template file "Template" the user has declared the logical node "stratonACTM". This corresponds to the "C" function bloc "stratonACTM".

Each Data objects and Data attributes should correspond in both ICD Template file and in UDFB/"C" function blocs.

EXAMPLE:



stratonLLN0 (LLN0)

- DO: NamPlt (LPL)
  - DA: vendor (VisString255)
  - DA: swRev (VisString255)
- DO: Beh (ENS)
  - DA: stVal (Beh)
  - DA: q (Quality)
  - DA: t (Timestamp)
- DO: Health (ENS)
  - DA: stVal (Health)
  - DA: q (Quality)
  - DA: t (Timestamp)
- DO: Mod (ENC)
  - DA: Oper (Oper_SPC_DPC)
    - BDA: ctlVal (BOOLEAN)
    - BDA: operTm (Timestamp)
    - BDA: origin (Originator)
      - BDA: orCat (orCategory)
      - BDA: orIdent (Octet64)
    - BDA: ctlNum (INT8U)
    - BDA: T (Timestamp)
    - BDA: Test (BOOLEAN)
    - BDA: Check (Check)
  - DA: stVal (Mod)
  - DA: q (Quality)
  - DA: t (Timestamp)
  - DA: ctlModel (CtlModels)

Name	Properties	Type
stratonLLN0		
Mod_Oper_origin_orCat		SINT
Mod_Oper_ctlNum		USINT
Mod_Oper_ctlVal		BOOL
Mod_Oper_operTm		LREAL
NamPlt_vendor		STRING(255)
Mod_Oper_Test		BOOL
NamPlt_swRev		STRING(255)
Health_stVal		SINT
Mod_ctlModel		SINT
Mod_Oper_T		LREAL
Beh		stratonENSBeh
Mod_stVal		SINT
Health_q		UDINT
Health_t		LREAL
Mod_q		UDINT
Mod_t		LREAL

On left side, the declaration of Logical Node "stratonLLN0" in ICD template file. On right side, the declaration of UDFB "stratonLLN0".

For each DA, there is one corresponding parameter in the UDFB.

For example, `NamPlt/vendor` corresponds to `NamPlt_vendor` in the UDFB.

Parameter in UDFB must have DO_DA syntax with a "_" as separator.

Another example, `Mod/Oper.T` corresponds to `Mod_Oper_T` parameter in the UDFB.

Notes:

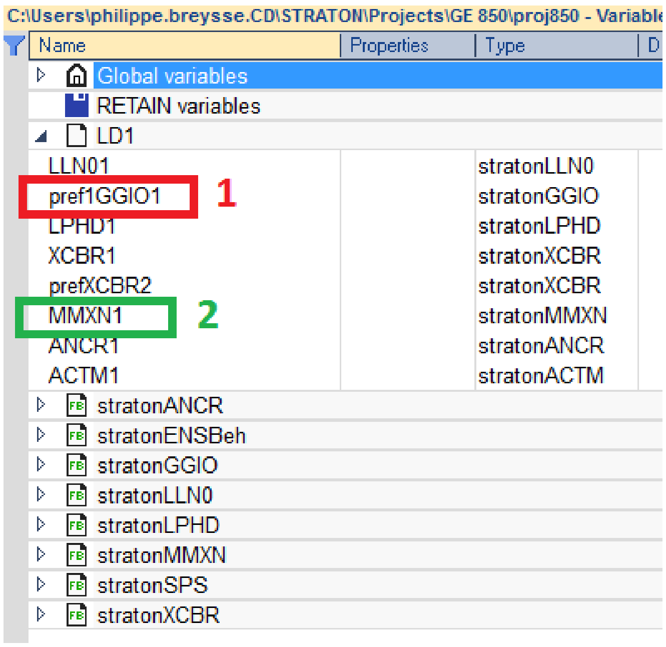
- Instead of declaring each parameter as a single variable, you can declare the UDFB parameter as a structure. In the example above, the DO "Beh" corresponds to the parameter "Beh" in the UDFB -> this Beh parameter is a structure of type stratonENSBeh (with stVal, q and t fields).
- It is not necessary to declare every variables, one can choose to declare only variables useful for the project application

3.2. Declaring instance of UDFB or instance of “C” function blocks

Before creating the configuration, the user must declare instances of “UDFB” and “C” function blocs. The instance name syntax must follow this syntax “*xxxxYYYYzz*” where:

```
xxxx: Correspond to the LN type Prefix (can have more than 4 letters)
YYYY: Correspond to a Logical Node Type (must have 4 letters)
zz : Is the LN instance number (can have more than 2 digits)
```

See example below:



C:\Users\philippe.breysse.CD\STRATON\Projects\GE 850\proj850 - Variable

Name	Properties	Type	D
Global variables			
RETAIN variables			
LD1			
LLN01		stratonLLN0	
pref1GGIO1		stratonGGIO	
LPHD1		stratonLPHD	
XCBR1		stratonXCBR	
prefXCBR2		stratonXCBR	
MMXN1		stratonMMXN	
ANCR1		stratonANCR	
ACTM1		stratonACTM	
stratonANCR			
stratonENSBeh			
stratonGGIO			
stratonLLN0			
stratonLPHD			
stratonMMXN			
stratonSPS			
stratonXCBR			

Example 1: in a main program “LD1”, we have declared instances “pref1GGIO1”

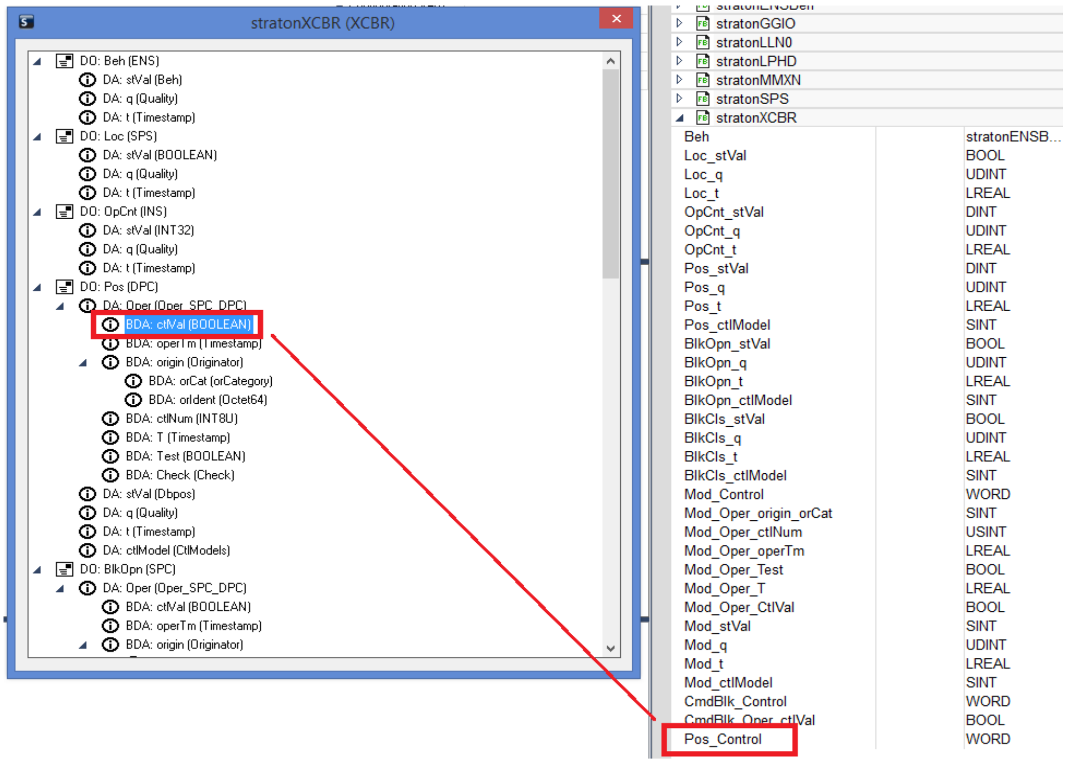
```
xxxx = “pref1”
YYYY = ”GGIO”
zz = ”1”
```

Example 2: in a main program “LD1”, we have declared instances “MMXN1”

```
xxxx = empty (optional)
YYYY =”MMXN”
zz =”1”
```

3.3. Control Models

For the management of the control models (Select, Cancel, Operate) an additional variable should be mapped and declare in the logic application. This variable must be called "Control". **Example**: In case of the position of a circuit breaker in the Logical Device "LD1", instance 1. LD1/XCBR1.Pos.Control



```
DO: Beh (ENS)
    DA: stVal (Beh)
    DA: q (Quality)
    DA: t (Timestamp)
DO: Loc (SPS)
    DA: stVal (BOOLEAN)
    DA: q (Quality)
    DA: t (Timestamp)
DO: OpCnt (INS)
    DA: stVal (INT32)
    DA: q (Quality)
    DA: t (Timestamp)
DO: Pos (DPC)
    DA: Oper (Oper_SPC_DPC)
        BDA: ctlVal (BOOLEAN)
        BDA: operTm (Timestamp)
        BDA: origin (Originator)
            BDA: orCat (orCategory)
            BDA: orIdent (Octet64)
        BDA: ctlNum (INT8U)
        BDA: T (Timestamp)
        BDA: Test (BOOLEAN)
        BDA: Check (Check)
    DA: stVal (Dbpos)
    DA: q (Quality)
    DA: t (Timestamp)
    DA: ctlModel (CtlModels)
DO: BlkOpn (SPC)
    DA: Oper (Oper_SPC_DPC)
        BDA: ctlVal (BOOLEAN)
        BDA: operTm (Timestamp)
        BDA: origin (Originator)
```

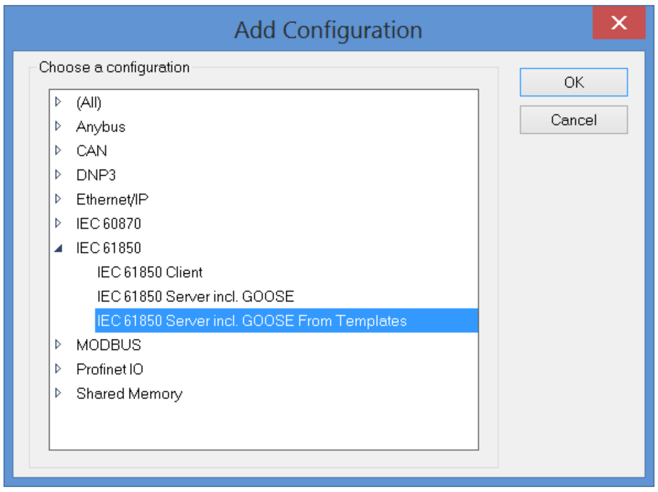
```
stratonGGIO
stratonLLN0
stratonLPHD
stratonMMXN
stratonSPS
stratonXCBR
```

Beh	stratonENSB...
Loc_stVal	BOOL
Loc_q	UDINT
Loc_t	LREAL
OpCnt_stVal	DINT
OpCnt_q	UDINT
OpCnt_t	LREAL
Pos_stVal	DINT
Pos_q	UDINT
Pos_t	LREAL
Pos_ctlModel	SINT
BlkOpn_stVal	BOOL
BlkOpn_q	UDINT
BlkOpn_t	LREAL
BlkOpn_ctlModel	SINT
BlkCls_stVal	BOOL
BlkCls_q	UDINT
BlkCls_t	LREAL
BlkCls_ctlModel	SINT
Mod_Control	WORD
Mod_Oper_origin_orCat	SINT
Mod_Oper_ctlNum	USINT
Mod_Oper_operTm	LREAL
Mod_Oper_Test	BOOL
Mod_Oper_T	LREAL
Mod_Oper_CtlVal	BOOL
Mod_stVal	SINT
Mod_q	UDINT
Mod_t	LREAL
Mod_ctlModel	SINT
CmdBlk_Control	WORD
CmdBlk_Oper_ctlVal	BOOL
Pos_Control	WORD

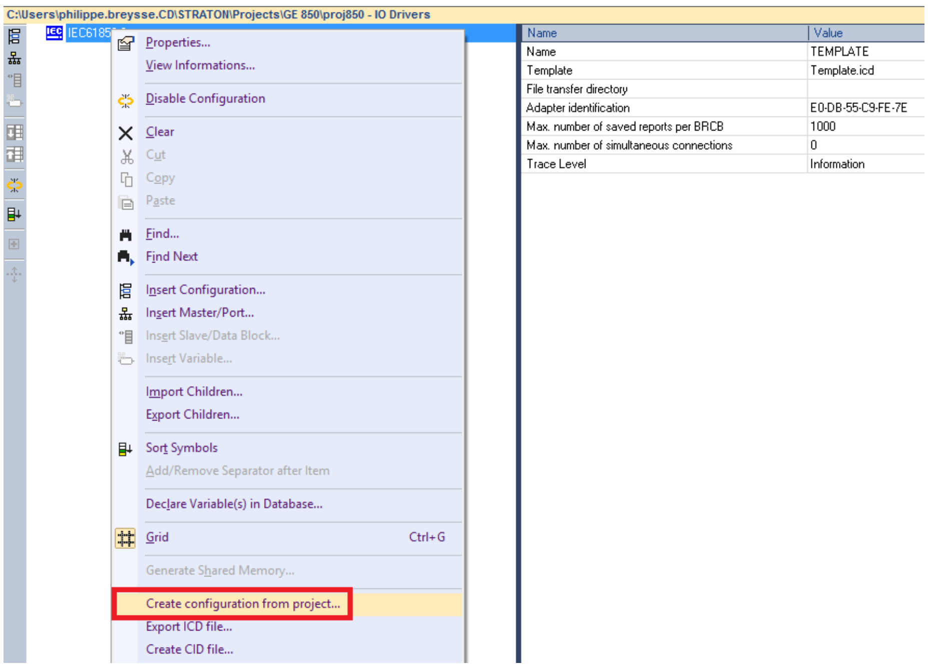
3.4. Create configuration from logic

Open fieldbus configuration

Add the IEC61850 Server configuration:

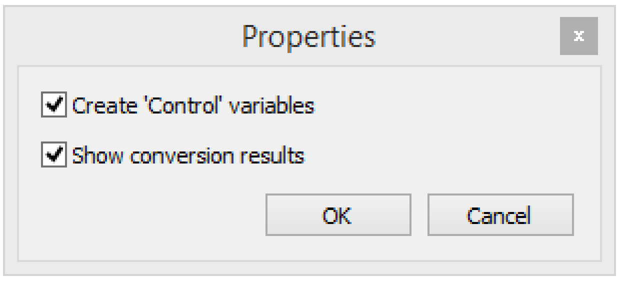


Right click on Server node and click on "Create configuration from project…"


C:\Users\philippe.breysse.CD\STRATON\Projects\GE 850\proj850 - IO Drivers

Name	Value
Name	TEMPLATE
Template	Template.icd
File transfer directory	
Adapter identification	E0-DB-55-C9-FE-7E
Max. number of saved reports per BRCB	1000
Max. number of simultaneous connections	0
Trace Level	Information

The "settings" dialog appears:

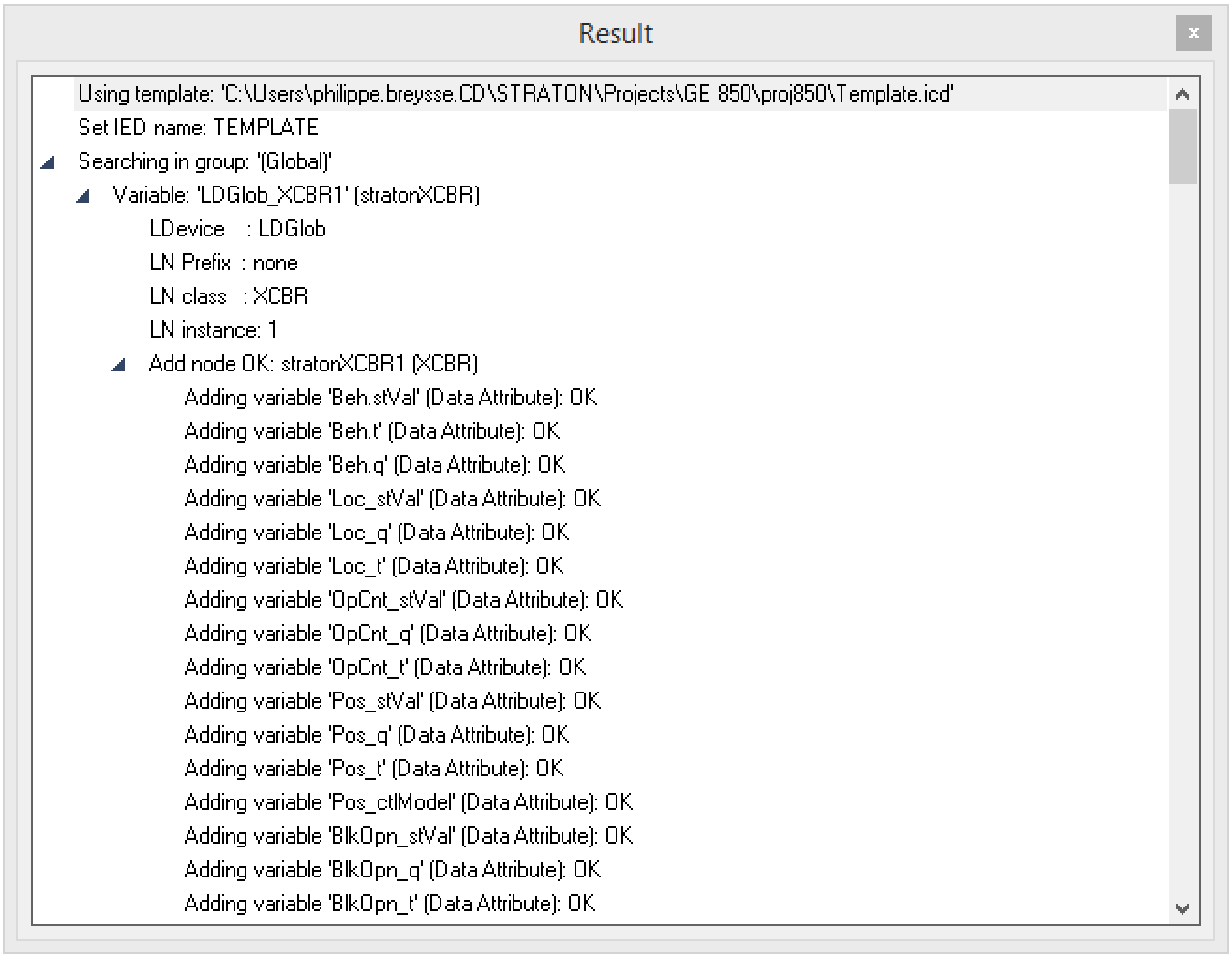


[✓] Create 'Control' variables

[✓] Show conversion results

Choose “Create Control variables” to create control variables (see the example about “Pos_Control”)

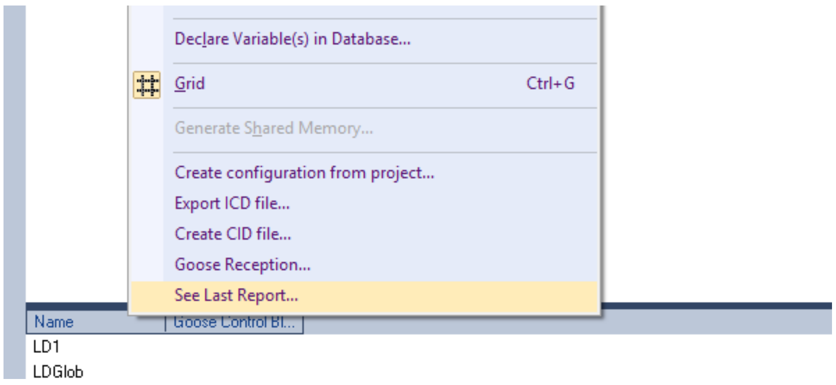
Choose “Show conversion results” to see results at the end of generation (each step of generation are described)



```
Using template: 'C:\Users\philippe.breysse.CD\STRATON\Projects\GE 850\proj850\Template.icd'
Set IED name: TEMPLATE
Searching in group: '(Global)'
  Variable: 'LDGlob_XCBR1' (stratonXCBR)
    LDevice    : LDGlob
    LN Prefix  : none
    LN class   : XCBR
    LN instance: 1
    Add node OK: stratonXCBR1 (XCBR)
      Adding variable 'Beh.stVal' (Data Attribute): OK
      Adding variable 'Beh.t' (Data Attribute): OK
      Adding variable 'Beh.q' (Data Attribute): OK
      Adding variable 'Loc_stVal' (Data Attribute): OK
      Adding variable 'Loc_q' (Data Attribute): OK
      Adding variable 'Loc_t' (Data Attribute): OK
      Adding variable 'OpCnt_stVal' (Data Attribute): OK
      Adding variable 'OpCnt_q' (Data Attribute): OK
      Adding variable 'OpCnt_t' (Data Attribute): OK
      Adding variable 'Pos_stVal' (Data Attribute): OK
      Adding variable 'Pos_q' (Data Attribute): OK
      Adding variable 'Pos_t' (Data Attribute): OK
      Adding variable 'Pos_ctlModel' (Data Attribute): OK
      Adding variable 'BlkOpn_stVal' (Data Attribute): OK
      Adding variable 'BlkOpn_q' (Data Attribute): OK
      Adding variable 'BlkOpn_t' (Data Attribute): OK
```

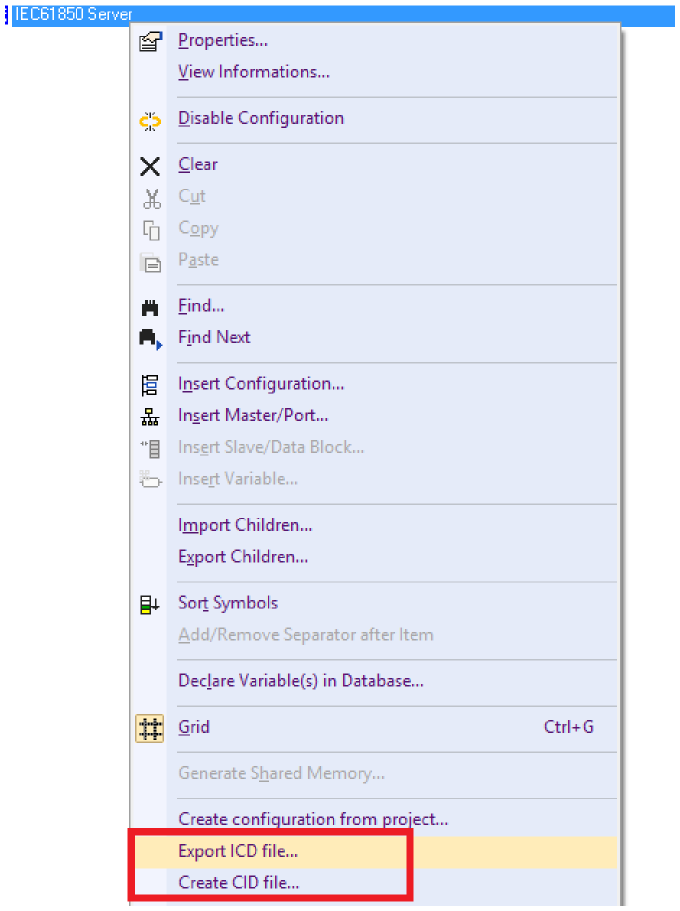
The conversion result is saved in project in file “__K5BusDDK850s_Result.txt”.

The user can see the last report later using the popup menu:



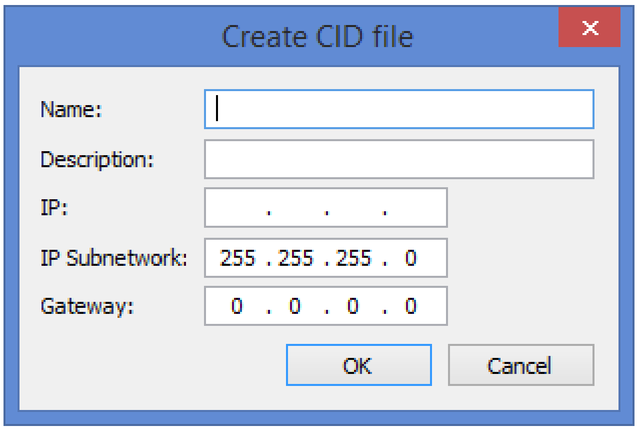
3.5. Export ICD

The proposed straton IDE driver has a command to export the current IEC61850 configuration to a specific folder. In this case the configuration is an ICD (IED Capability Description) file.



3.6. Create CID

A second command allows to create a CID (Configured IED Device information) file. A dialog allows to manually set some parameters that are required for the CID. Several instances of the CID can be created.



Create CID file

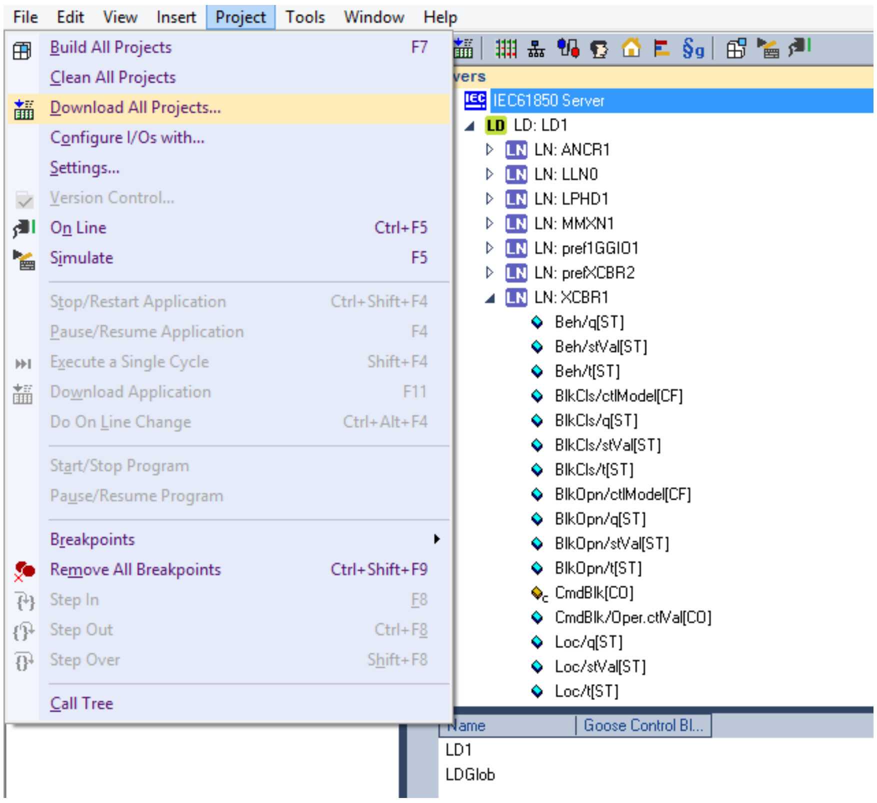
Name:	
Description:	
IP:	. . .
IP Subnetwork:	255 . 255 . 255 . 0
Gateway:	0 . 0 . 0 . 0

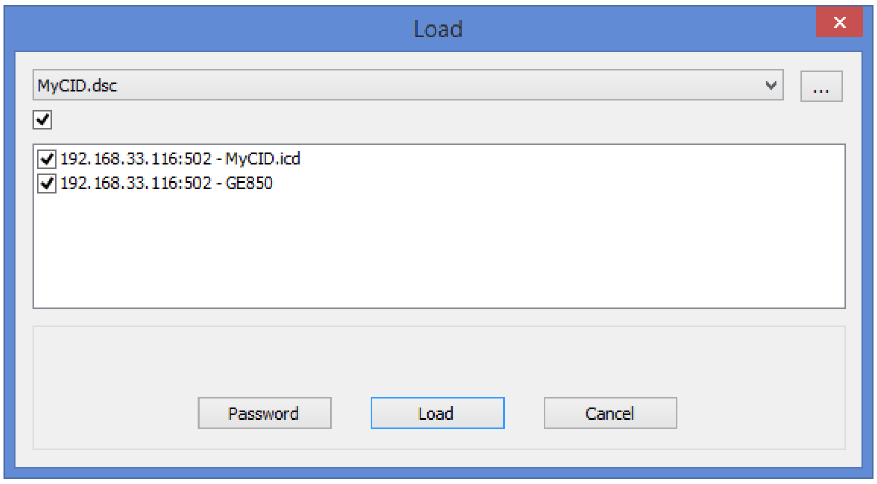
OK Cancel

This allows the deployment of the same application on different IEDs.

The CID file can be modified to add Data sets and Report control blocks.

When user creates a new CID file, the workbench automatically creates a new script for downloading the file. The user will have to use the Download scipts to download his CID file on the target:





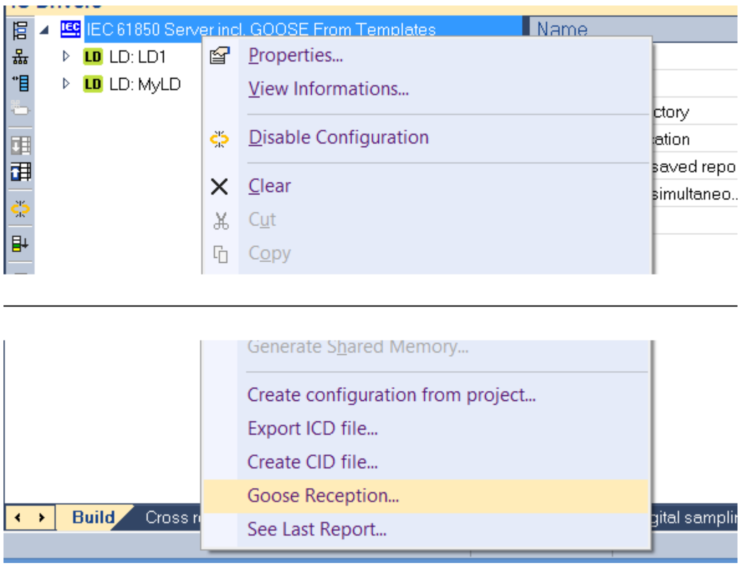
When using this script the CID file will be downloaded on the target and then the application will start.

Note:

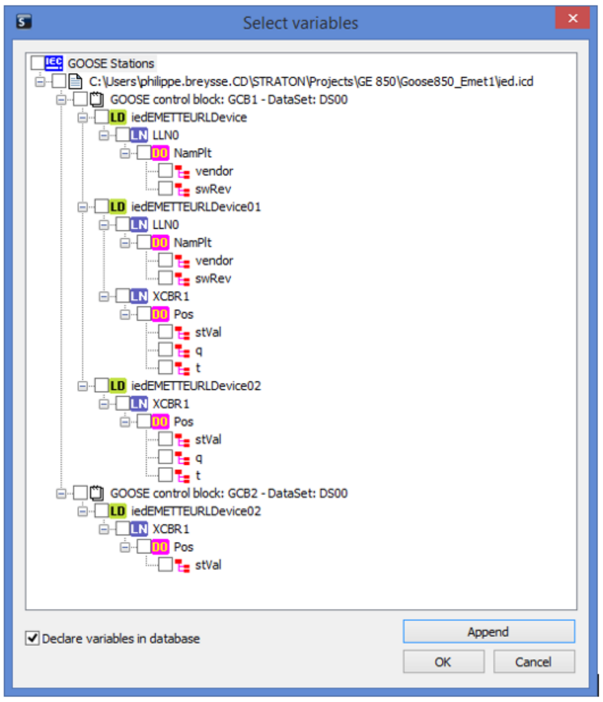
- More information about the Download scripts can be found in the Annex A of this document.

3.7. Using Goose reception

When the user needs to create a list of variables for Goose reception, he will have to use the "Goose Reception…" popup menu from the Fieldbus Configurations:



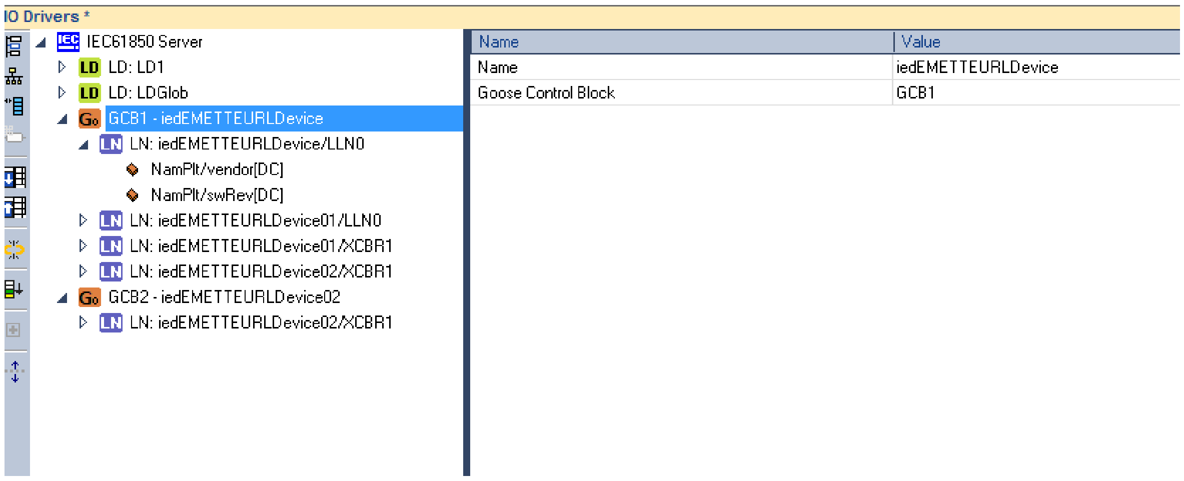
Click on “Append” button, select the corresponding *.icd file and select the goose data:



```
GOOSE Stations
  C:\Users\philippe.breysse.CD\STRATON\Projects\GE 850\Goose850_Emet1\ied.icd
    GOOSE control block: GCB1 - DataSet: DS00
      LD iedEMETTEURLDevice
        LN LLN0
          DO NamPlt
            vendor
            swRev
      LD iedEMETTEURLDevice01
        LN LLN0
          DO NamPlt
            vendor
            swRev
        LN XCBR1
          DO Pos
            stVal
            q
            t
      LD iedEMETTEURLDevice02
        LN XCBR1
          DO Pos
            stVal
            q
            t
    GOOSE control block: GCB2 - DataSet: DS00
      LD iedEMETTEURLDevice02
        LN XCBR1
          DO Pos
            stVal
```

[✓] Declare variables in database

The wizard will automatically create “Goose reception” node and the associated variables:



```
IEC61850 Server
  LD: LD1
  LD: LDGlob
  GCB1 - iedEMETTEURLDevice
    LN: iedEMETTEURLDevice/LLN0
      NamPlt/vendor[DC]
      NamPlt/swRev[DC]
    LN: iedEMETTEURLDevice01/LLN0
    LN: iedEMETTEURLDevice01/XCBR1
    LN: iedEMETTEURLDevice02/XCBR1
  GCB2 - iedEMETTEURLDevice02
    LN: iedEMETTEURLDevice02/XCBR1
```

Name	Value
Name	iedEMETTEURLDevice
Goose Control Block	GCB1

Important:

After configuring the GOOSE reception, the CID must be updated using the “Create CID file” menu. The name of an existing CID can be filled, straton will automatically merge these CID so if some Data Sets or RCBs are already created, this will not be lost.

4. Frequently Asked Questions

The following error message occurs in the runtime's output:

"Could not start goose for adapter xxx"

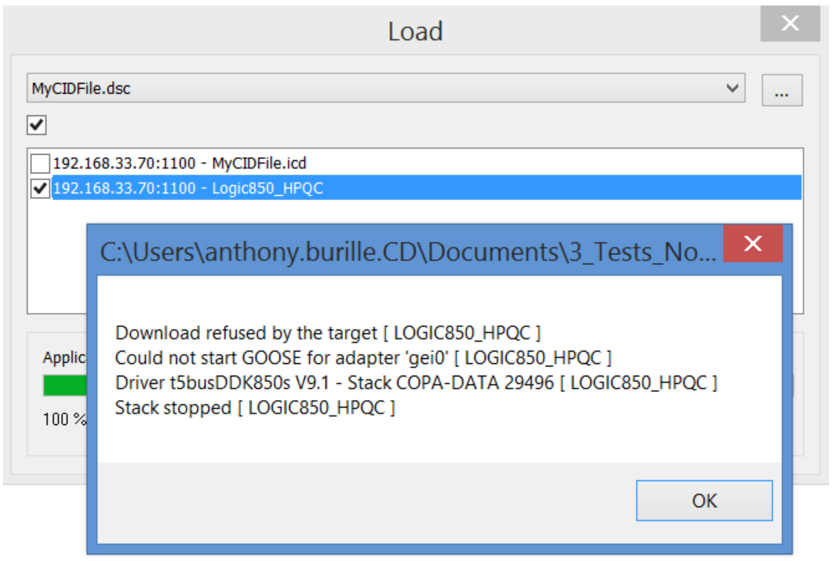
This happens because no GOOSE is configured in the Fieldbus Configurations.

"Could not start GOOSE because adapter identification xxx is not valid!"

The "Adapter identification" is not correct. For Windows runtimes, the adapter identification corresponds to the MAC address of the PC's Ethernet card. For other runtimes like Linux, QNX, VxWorks etc. this correspond to the name of the Ethernet card (can be eth0, eth1, gei0, fxp0 ...)

"Download refused by the target" error message occurs while my configuration is the good one:

For example:


Load
MyCIDFile.dsc
...
192.168.33.70:1100 - MyCIDFile.icd
192.168.33.70:1100 - Logic850_HPQC
C:\Users\anthony.burille.CD\Documents\3_Tests_No...
Applic
100 %
OK

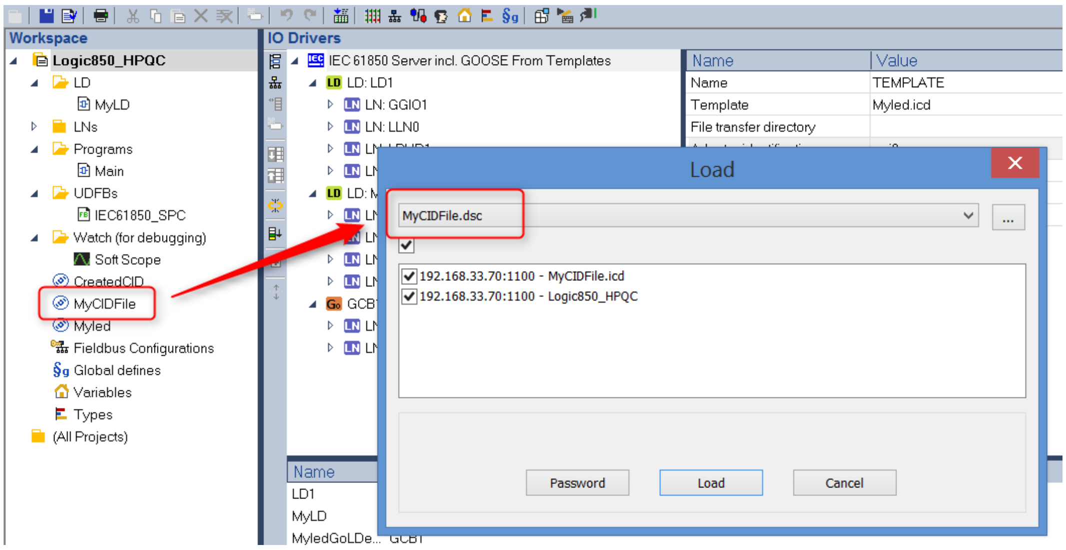
```
Download refused by the target [ LOGIC850_HPQC ]
Could not start GOOSE for adapter 'gei0' [ LOGIC850_HPQC ]
Driver t5busDDK850s V9.1 - Stack COPA-DATA 29496 [ LOGIC850_HPQC ]
Stack stopped [ LOGIC850_HPQC ]
```

This message can occur if the file is not downloaded to the right place on the target or is not accessible. Most of the time, files are stored in the "Custom" folder. On the target this folder must exist (case sensitive) and must be accessible in read/write.

See the Annex A of this document for more details.

Annex A – Download scripts for the files and application

After creating a new CID file, its corresponding Download script is automatically created.

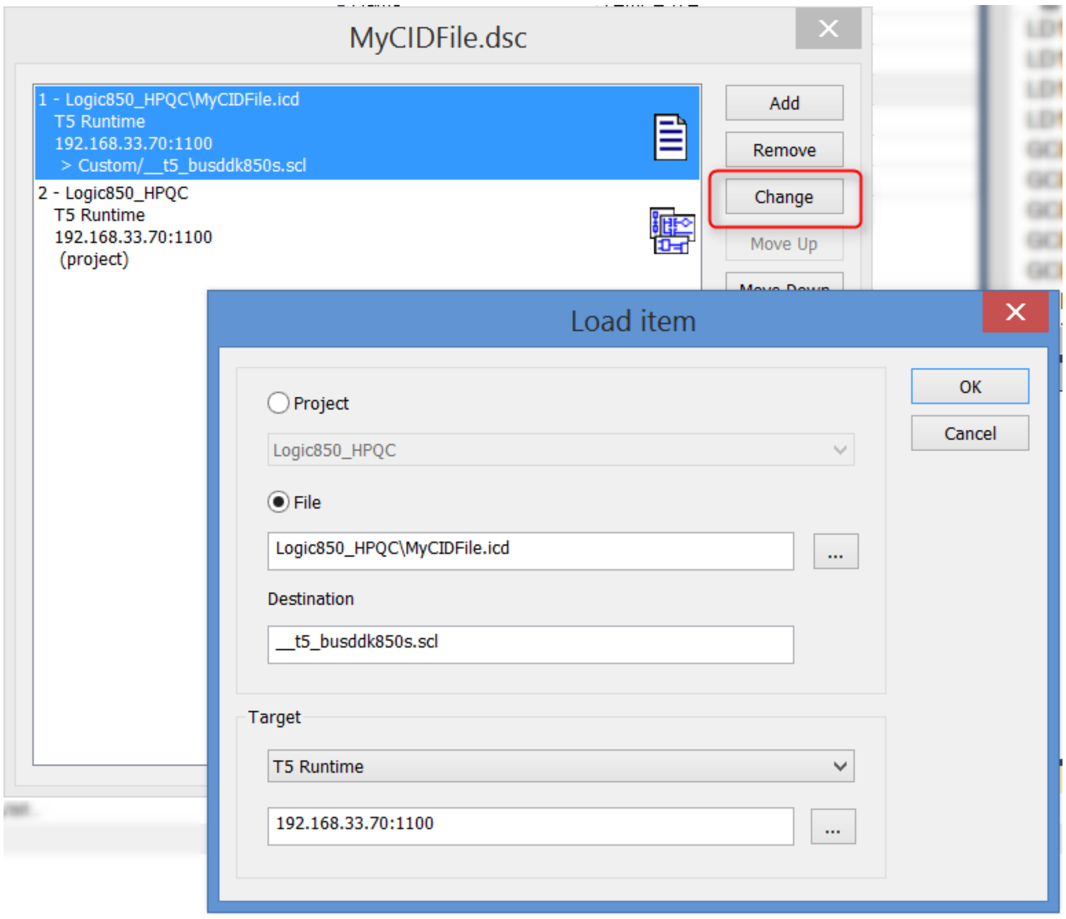


Name	Value
Name	TEMPLATE
Template	Myled.icd
File transfer directory	

MyCIDFile.dsc

192.168.33.70:1100 - MyCIDFile.icd
192.168.33.70:1100 - Logic850_HPQC

Existing scripts can be edited using the [...] button, then the “Edit” button.



1 - Logic850_HPQC\MyCIDFile.icd
T5 Runtime
192.168.33.70:1100
> Custom/__t5_busddk850s.scl

2 - Logic850_HPQC
T5 Runtime
192.168.33.70:1100
(project)

Project: Logic850_HPQC

[✓ File]: Logic850_HPQC\MyCIDFile.icd

Destination: __t5_busddk850s.scl

Target: T5 Runtime

192.168.33.70:1100

The script allows to download the necessary files (SCL files, application ...) to the target.

Files are downloaded in a precise order, from the top to the bottom. The download procedure can be modified, adding or removing files.

The application must be in the last position because when starting it will use the previously downloaded files.

The following fields can be filled:

Project:	if there are many projects in the same project list
File:	the file to be downloaded from your PC
Destination:	destination of the file on the target (*)

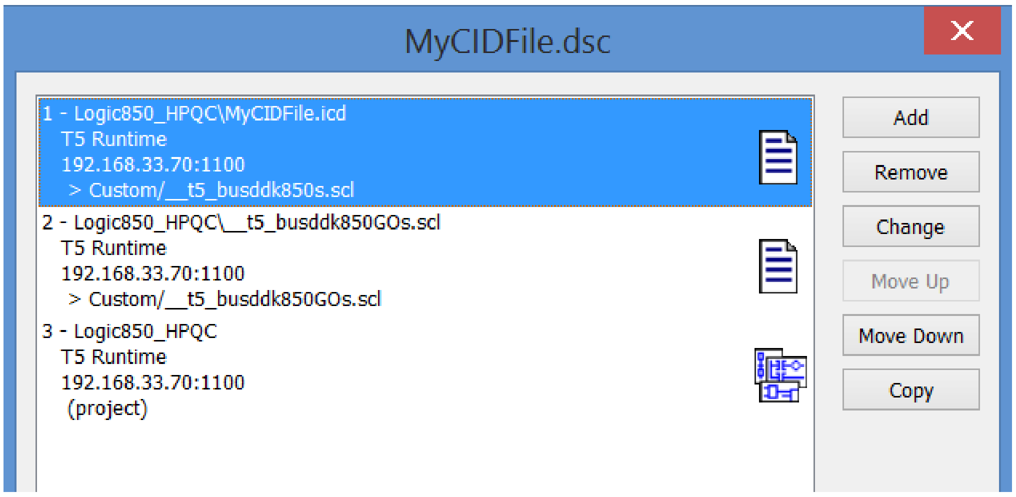
Target: Runtime type and IP address + port of the target

(*) The destination path correspond to the folder where the runtime is launched.

For example, for runtimes running under other OS than Windows (Linux, QNX, VxWorks...), it is important to take in account the "`/path850=xxx/`" option.

If the runtime is launched using "`/path850=Custom/`", it means that when using the IEC61850 driver it will look at the different SCL files in the Custom folder. This folder must exist on the target, moreover the script must be modified in order to take it into account.

For example, for an application downloaded on a Linux runtime at 192.168.33.70:1100 launched with the "/path850=Custom/" option and using GOOSE data, the script is:



MyCIDFile.dsc

```
1 - Logic850_HPQC\MyCIDFile.icd
    T5 Runtime
    192.168.33.70:1100
     > Custom/__t5_busddk850s.scl
2 - Logic850_HPQC\__t5_busddk850GOs.scl
    T5 Runtime
    192.168.33.70:1100
     > Custom/__t5_busddk850GOs.scl
3 - Logic850_HPQC
    T5 Runtime
    192.168.33.70:1100
     (project)
```