# Distributed application

## straton user guide – Rev. 8

sales@straton-plc.com

CP_R31_006_H

straton

## Content

# 1. Overview

This document describes how to setup a communication through TCP/IP connection between Runtimes.

Download and install from https://straton-plc.com/telechargements/

# 2. Distributed applications – Binding

The T5 runtime version 2.10 or later provides an event driven protocol on TCP-IP for exchanging real time data among runtime systems. As the protocol is purely event-based, it ensures high performances and very low network traffic while running. This section explains how to use the T5 event-based protocol for binding variables of runtime systems in order to build a distributed application.

The *EnableEvents()* function can be used to enable or disable the production of events for binding.

### Exchange mechanism

The T5 protocol is based on a "publish-subscribe" model. Each runtime system may publish some variables on the network and consume variables from other runtime systems. Each produced variable is identified by a number. This identifier is used for linking the source and destination variables in the projects. The same published variable can be consumed in several destination projects.

### Exchanges

The variable's value is sent on the network only when it changes. For each produced variable, you can define a positive and negative hysteresis in order to adjust the network traffic according to the needs of your application. Each new value sent on the network is qualified with a date and time stamp. For each consumed variable in destination projects, you can have access to its date/time stamping and to its quality flag. You can also have global information about the status of the connection with each producer.

### Limitations

The maximum number of produced variables is limited. Refer to your OEM instructions for more information about the binding limitations imposed by the T5 runtime implementation.

Only boolean, numerical and TIME variables can be exchanged. STRING can be exchanged since version 9.4, only if the string's length is lower than 200 characters.

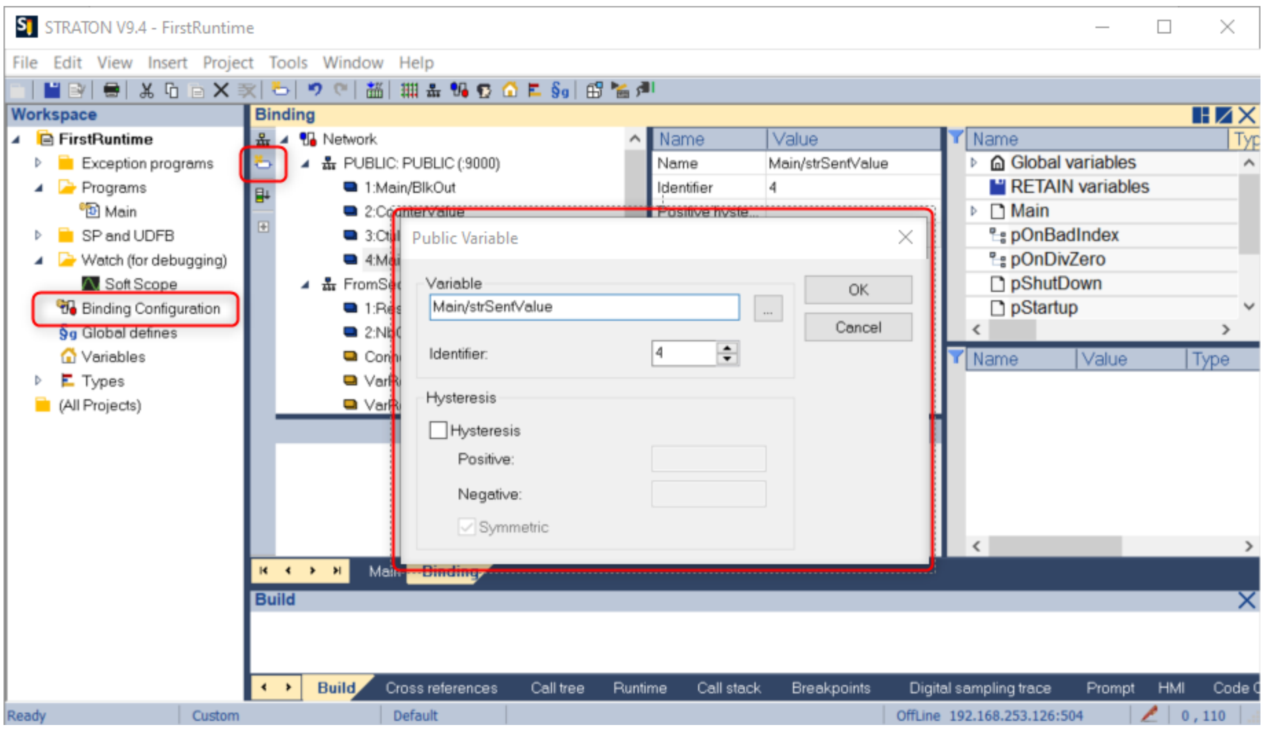
Function block instances cannot be exchanged on the network.

# 3. Publish variables

## 3.1 Binding configured per project

Open the binding configuration. In the "PUBLIC" level" the "Port" and "Address" parameters are not useful; these are determined automatically by the Runtime.

Add a Public Variable. If variables are already existing, the variable's Identifier is incremented automatically.



| Name | Value |
|---|---|
| Name | Main/strSentValue |
| Identifier | 4 |

Public Variable

Variable: Main/strSentValue

Identifier: 4

Hysteresis

☐ Hysteresis

Positive:

Negative:

☑ Symmetric

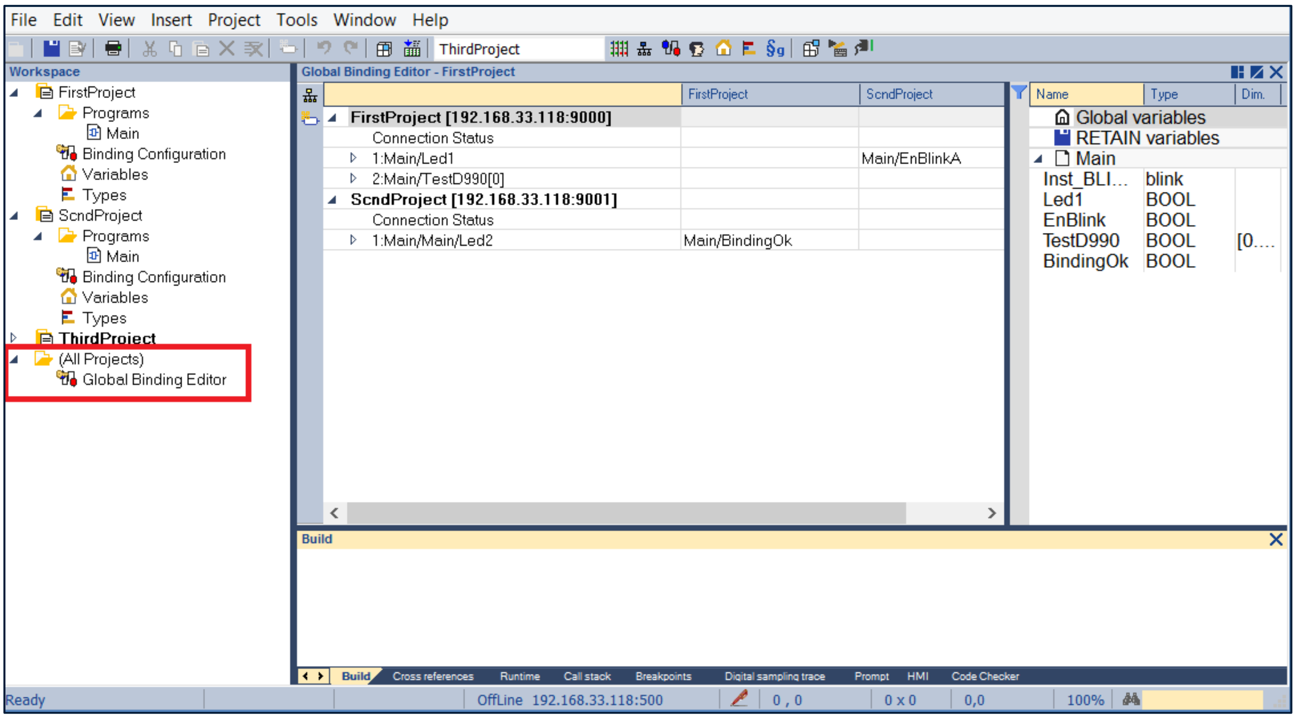
OK

Cancel

## 3.2 Global binding

Since straton 8.7 version it is also possible to directly access to the "Global Binding Editor".

Add a project in the binding editor ( ) and add some variables ( ). Then it is possible to link this variable directly to another one from another project.



- FirstProject
  - Programs
    - Main
  - Binding Configuration
  - Variables
  - Types
- ScndProject
  - Programs
    - Main
  - Binding Configuration
  - Variables
  - Types
- ThirdProject
- (All Projects)
  - Global Binding Editor

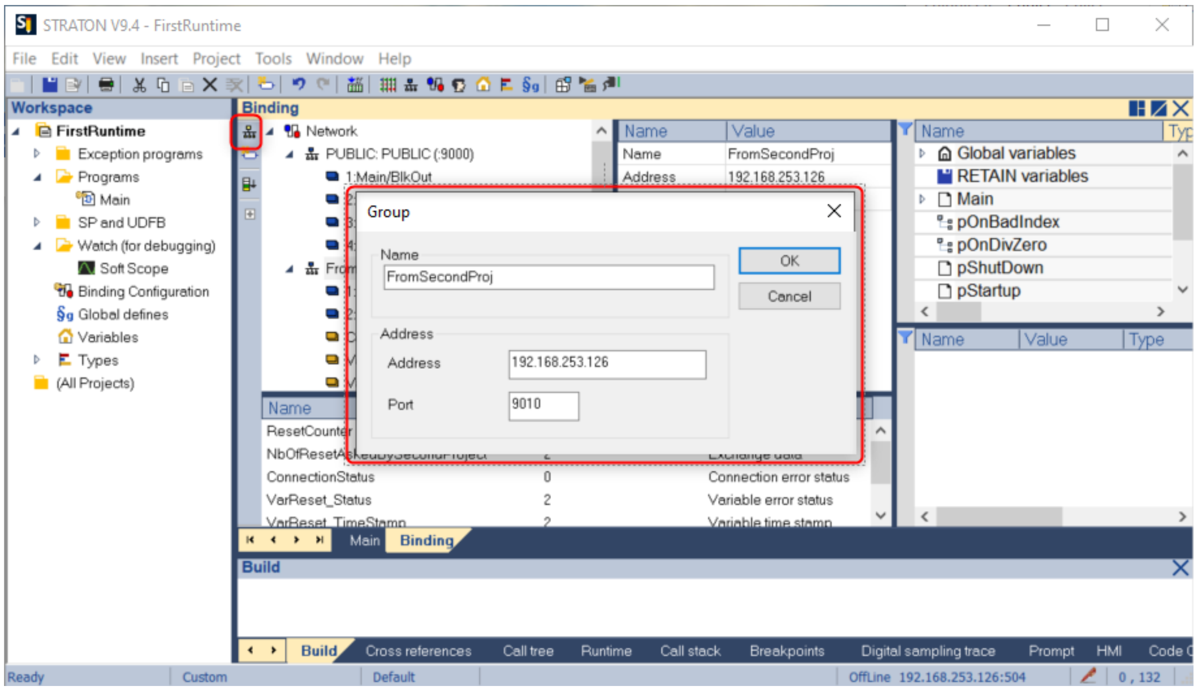
| | FirstProject | ScndProject |
|---|---|---|
| FirstProject [192.168.33.118:9000] | | |
| Connection Status | | |
| 1:Main/Led1 | | Main/EnBlinkA |
| 2:Main/TestD990[0] | | |
| ScndProject [192.168.33.118:9001] | | |
| Connection Status | | |
| 1:Main/Main/Led2 | Main/BindingOk | |

| Name | Type | Dim. |
|---|---|---|
| Global variables | | |
| RETAIN variables | | |
| Main | | |
| Inst_BLI… | blink | |
| Led1 | BOOL | |
| EnBlink | BOOL | |
| TestD990 | BOOL | [0.… |
| BindingOk | BOOL | |

# 4. Subscribe to variables

Except fi the Global Binding window is used, if variables must be received from an external Runtime, an additional network level must be added.

Click on “Insert Master/Port” and configure it with the Address and binding Port of the external runtime.



- FirstRuntime
  - Exception programs
  - Programs
    - Main
  - SP and UDFB
  - Watch (for debugging)
    - Soft Scope
  - Binding Configuration
  - Global defines
  - Variables
  - Types
- (All Projects)

- Network
  - PUBLIC: PUBLIC (9000)
    - 1:Main/BlkOut

| Name | Value |
|---|---|
| Name | FromSecondProj |
| Address | 192.168.253.126 |

Group

Name: FromSecondProj

Address

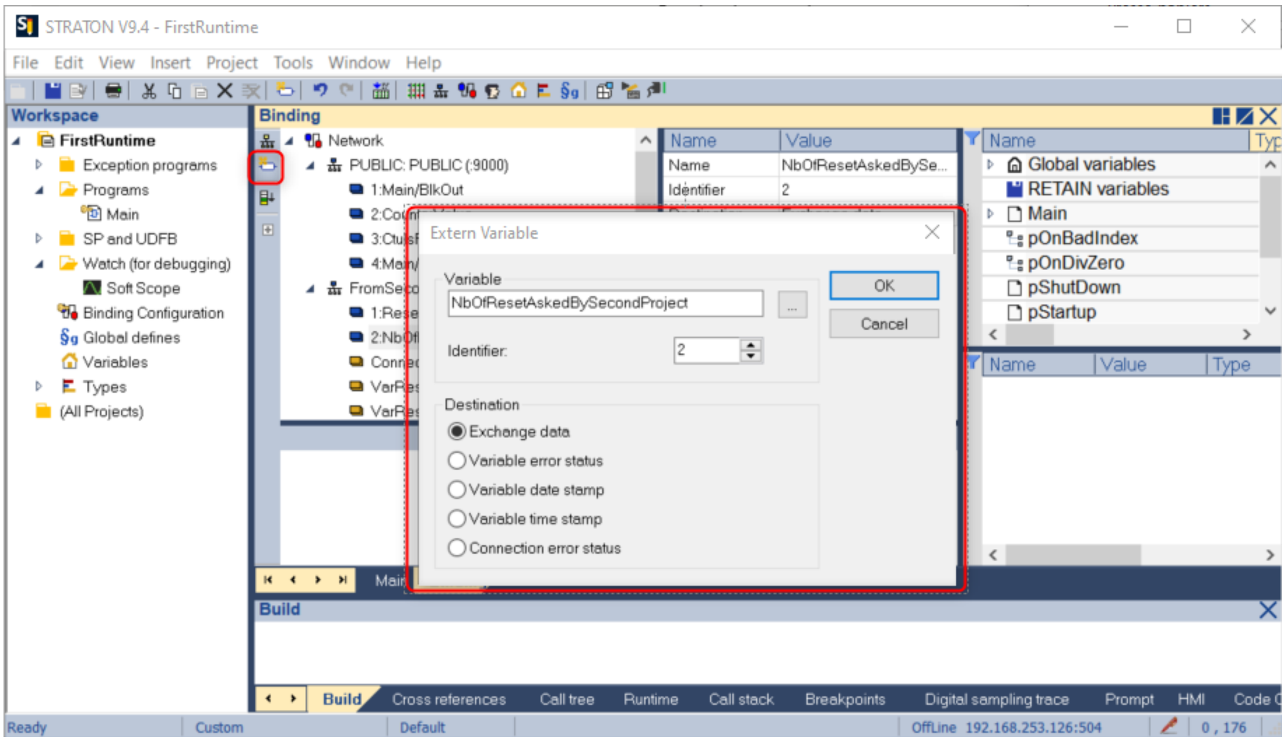
Address: 192.168.253.126

Port: 9010

| Name | | |
|---|---|---|
| ResetCounter | | |
| NbOfResetAskedBySecondProject | 2 | Exchange data |
| ConnectionStatus | 0 | Connection error status |
| VarReset_Status | 2 | Variable error status |
| VarReset_TimeStamp | 2 | Variable time stamp |

| Name | Typ |
|---|---|
| Global variables | |
| RETAIN variables | |
| Main | |
| pOnBadIndex | |
| pOnDivZero | |
| pShutDown | |
| pStartup | |

Variables which must be received from the external application must be declared in the project's dictionary. These must have exactly the same identifier as what is set in the external application.



| Name | Value |
|---|---|
| Name | NbOfResetAskedBySe... |
| Identifier | 2 |

Extern Variable

Variable: NbOfResetAskedBySecondProject

Identifier: 2

Destination
- [✓ Exchange data]
- Variable error status
- Variable date stamp
- Variable time stamp
- Connection error status

OK Cancel

Several choices are available when declaring an External Variable:

| | |
|---|---|
| Exchange data: | data from the external application |
| Variable error status: | status variable, linked to a variable and its identifier. |

- o 0 = the variable exists on the external application and has been received at least one time
- o 1 = the variable has never been received or it does not exist on the external application.

| | |
|---|---|
| Variable date stamp: | variable's date stamp as a DINT* |
| Variable timestamp: | variable's time stamp as DINT* |
| Connection error status: | general connection status |

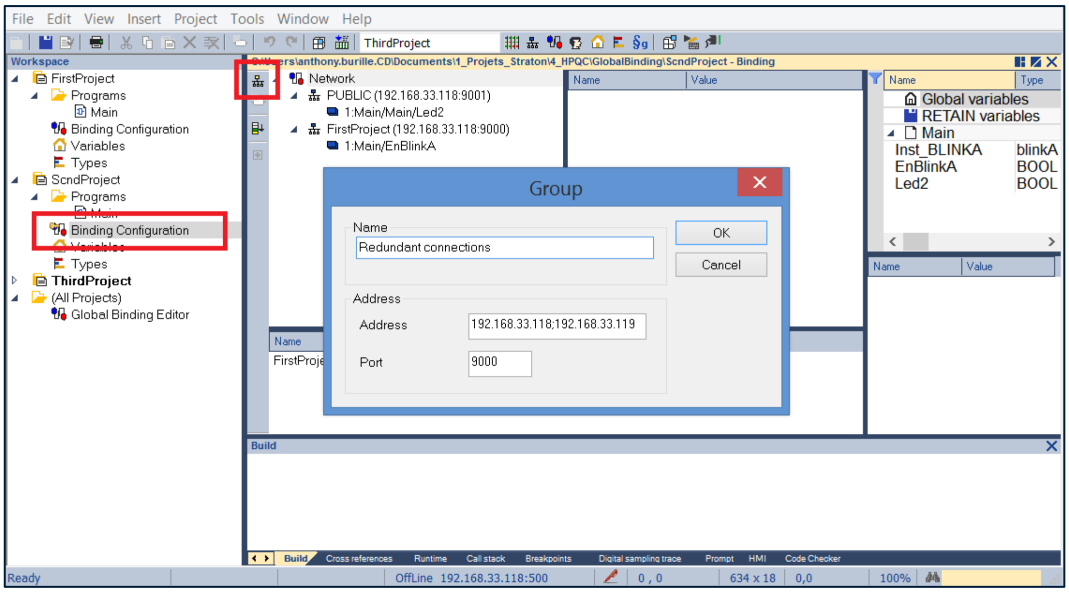
- o 0 = the connection is successful
- o 1 = it tries to connect
- o 2 = it failed to connect (can be a wrong port)
- o 4 = the external application/runtime is stopped (can be a wrong IP address)

*Use blocks as DTDAY, DTHOUR, DTMIN… in order to extract information from DINT time/date-stamps

# 5. Redundant connections

## 5.1. Dual connections

The support of redundant Ethernet connection is ease, simply enter both IP addresses separate by a ";".



Group

Name: Redundant connections

Address

Address: 192.168.33.118;192.168.33.119

Port: 9000

## 5.2. Multiple connections

Redundant network connection to redundant PLCs are supported up to 6 IP addresses.

In this case, IP addresses must be separated by "|" characters. E.g.:

```
192.168.0.20|192.168.0.21|192.168.1.20|192.168.1.21
```

The status of the connection set a bit at TRUE for each failed connection.

# 6. Diagnostics

For any consumers it is possible to get the status of the connection with status variable.

The producer cannot detect which consumers are connected; to do so the only way is to create a cross link to retrieve the status of the link for instance. Status of the connection can be published and read by the consumer.