

# Instructions for using the CPS Emulator and the CPS Validation Adapters

Tobias Wolff

Vincent Happersberger

IPK Fraunhofer

Berlin

0. Preface: In the first step the CPS Emulator(Server) must be started and configured before the CPS Validation Adapter(Client) can be configured and started to connect to the server. Before starting the CPS Emulator, please check if the following instructions are correct: EmulationServer\v0.9\Server the two files "specificationConfig.xml" and "timingConfig.xml" are included.

# Table of Figures

Illustration1: Windows-Defender Safety Instructions .....	3
Illustration2: Entering the IP address.....	3
Illustration3: Selection regarding the use of "specificationConfig.xml", case "Yes .....	3
Illustration4: Path specification in the "specificationConfig.xml" .....	4
Illustration5: Copy the path from Windows Explorer .....	4
Illustration6: Listing of the information models under the specified path .....	4
Illustration7: Created information model with timestamp .....	4
Illustration8: Selection regarding use of "specificationConfig.xml", case "No" .....	5
Illustration9: Path specification of an information model .....	5
Illustration10: The emulation waits for the path input .....	5
Illustration11: Specifying the path to the scenarios folder <sup>6</sup> .....	6
Illustration12: Output after creating a revised information model .....	6
Illustration13: Output when reading an information model without changes.....	7
Illustration14: Default settings for the demo .....	7
Illustration15: Directory structure of the demo .....	8
Illustration16: Testconfiguration.xml.....	8
Illustration17: Folder structure of the Validation Adapter .....	9
Illustration18: Results of the validation.....	9
Illustration19: Incorrect node .....	10
Table 1: Attributes of the AdapterConfig.properties .....	8
Table 2: Testconfiguration.xml .....	9

## How to use the CPS Emulator

The CPS(=cyber physical system) emulator is a software to emulate an OPC UA server. It makes data of a read-in information model available to clients via a network.

Execute the CPSEmulator.exe. (to be found under the path: EmulationServer\v0.9\Server)

1. It launches a Windows security warning, where you can cancel and allow access, both will let the CPSEmulator.exe start with all features.

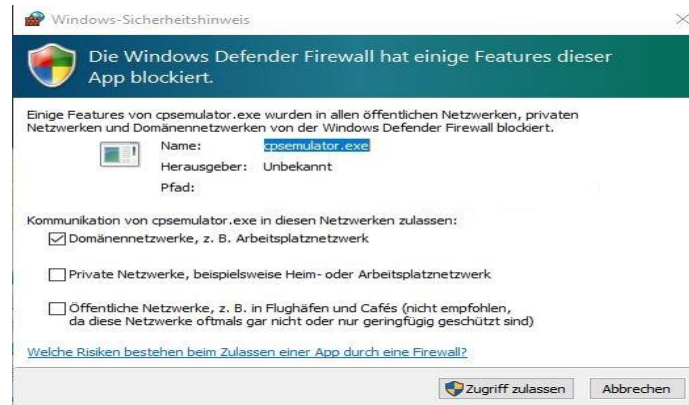


Illustration 1: Windows-Defender Safety Instructions

2. Enter the IP address where the server should run (ENTER for localhost<sup>1</sup>)

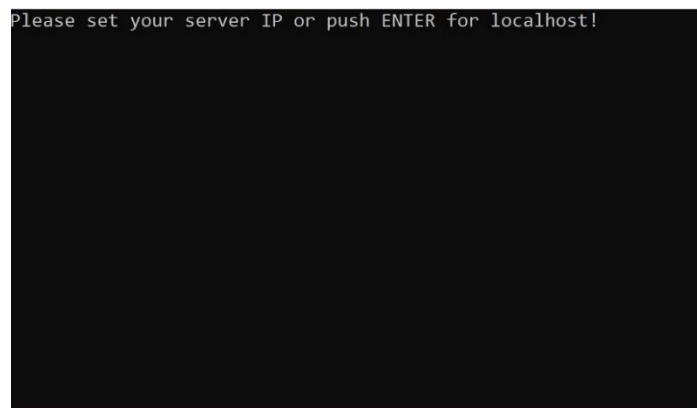


Illustration 2: Entering the IP address

3. Select whether the "specificationConfig.xml"<sup>2</sup> should be loaded. Press 'y' for yes or 'n' for no.



Illustration 3: Selection regarding the use of "specificationConfig.xml", case "Yes

<sup>1</sup> The standardized domain name of the virtual IP address, the server used on the running computer.

<sup>2</sup> Configuration file to simplify the configuration of the emulator, through shorter user inputs.

3.1. Case - "Yes": The "specificationConfig.xml" is loaded, but must be adapted before! This loads the information models which are then available for selection.

```
<?xml version='1.0' encoding='utf-8'?>
<specification>
  <path>E:\Szenarien</path>
</specification>
```

Illustration 4: Path specification in the "specificationConfig.xml"

In line 3 the path to the scenarios folder must be adjusted. This can easily be copied from the Windows-Explorer. Please insert between the brackets <path>...</path>.

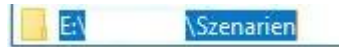


Illustration 5: Copy the path from Windows Explorer

3.1.1. Now all information models are listed under the specified path, whereby the selection is made by entering a number (to be found to the left of the path).

```
Preparing Nodeset XML!
Please specify the used file by choosing the corresponding number and
press enter!
1 | E:      Szenarien\Audi_BDE_Informatonsmodell.xml
2 | E:      Szenarien\Automodell_0.xml
3 | E:      Szenarien\Automodell_1.xml
4 | E:      Szenarien\Datenstruktur_Machine.xml
5 | E:      Szenarien\gesi.xml
6 | E:      Szenarien\IPK_CNC.xml
7 | E:      Szenarien\IPK_Koffer.xml
8 | E:      Szenarien\NodeStructure.xml
10 | E:     Szenarien\timingConfig_IPK_CNC.xml
11 | E:     Szenarien\timingConfig_Koffer.xml
12 | E:     Szenarien\timingConfi_Machine.xml
```

Illustration 6: Listing of the information models under the specified path

If the message "Please restart the server with edited information model" appears after starting and reading the information model, this means that the CPS Emulator must be restarted with the edited information model and the same configurations. The same applies to the CPS Validation Adapter. In this case the emulation creates a new file containing the file name of the imported information model and an addition of numbers after the file name<sup>3</sup>. This file contains all changes, the originally read-in information model remains unchanged.

For example:

 Automodell_011583.xml	28.10.2019 11:58	XML-Dokument	5 KB
---	------------------	--------------	------

Illustration 7: Created information model with timestamp

<sup>3</sup> The file name of the revised information model is displayed by the emulator.

3.2.1. Case - "No": The "specificationConfig.xml" is not loaded.

- 3.2.1.1. Select whether the internally stored IPC-Example<sup>4</sup> shall be loaded or ENTER to load an own information model.

```
Please set your server IP or push ENTER for localhost!
IP Address: opc.tcp://localhost:4880
Please specify use examples (IPK) or push ENTER for XML File!
```

Illustration 8: Selection regarding use of "specificationConfig.xml", case "No"

3.2.2. Specify the path of the information model (it is easiest to place the .xml file in the same folder as the CPSEmulator.exe, or in the folder "Scenarios", one folder higher in the structure).

Emulator > EmulationServer > v0.9 > Szenarien			
Name	Änderungsdatum	Typ	Größe
Automodel_0_Server.xml	15.10.2019 15:22	XML-Dokument	4 KB
Automodel_1_Server.xml	16.10.2019 12:44	XML-Dokument	4 KB

Illustration 9: Path specification of an information model

(folder structure, where Automodel\_0.xml without NodeIds<sup>5</sup> and Automodel\_1.xml with NodeIds)

```
Please set your server IP or push ENTER for localhost!
IP Address: opc.tcp://localhost:4880
Please specify use examples (IPK) or push ENTER for XML File!

Preparing Nodeset XML!
Please specify the used file!
Current Path >> EmulationServer\v0.9\Server
```

Illustration 10: The emulation waits for the path input

<sup>4</sup> An internally predefined information model.

<sup>5</sup> The representation of a node on an OPC UA server.

```

Please set your server IP or push ENTER for localhost!
IP Address: opc.tcp://localhost:4880
Please specify use examples (IPK) or push ENTER for XML File!

Preparing Nodeset XML!
Please specify the used file!
Current Path >> EmulationServer\v0.9\Server
../Szenarien/Automodell_0.xml

```

Illustration 11: Specifying the path to the scenarios folder<sup>6</sup>

- 3.2.3. The emulation can be terminated by closing the window<sup>7</sup>.
- 3.2.4. If the message "Please restart the server with edited information model" appears after starting and reading the information model, this means that the CPS Emulator must be restarted with the edited information model and the same configurations. The same applies to the CPS Validation Adapter.

In this case the emulation creates a new file containing the file name of the imported information model and an addition of numbers after the file name (will be output). This file contains all changes, the originally read-in information model remains unchanged.

```

Endpoint prints: ?
The server is running.

Press:
      x,q:      Shutdown the server
      else:      Show usage

Please restart the server with the edited information model
Emulation is running
Emulation is running
Emulation is running
Emulation is running
Emulation is running
Emulation is running
Emulation is running
Emulation is running

```

Illustration 12: Output after creating a revised information model

<sup>6</sup> „../“ stands for the unique backward running in the folder structure and can be concatenated.

<sup>7</sup>The emulation can also be stopped with the key combination "ctrl+c".

3.3. At successful start, "Emulation is running" is displayed every few seconds.

```
Nodeset2 validation successful.
Endpoint prints: ?
The server is running.

Press:
      x,q:      Shutdown the server
      else:     Show usage

Emulation is running
Emulation is running
Emulation is running
Emulation is running
Emulation is running
Emulation is running
Emulation is running
Emulation is running
```

*Illustration 13: Output when reading an information model without changes*

## Manuel of the CPS Validation Adapter

The cyper physical System Validation Adapter (CPS VA) is a software to validate communication interfaces, here especially a server, in OPC UA against a specification. For this purpose the CPS VA is configured with the help of two configuration files "AdapterConfig.properties" and "TestConfiguration.xml".

### 2. Configuration.

- 2.1 **AdapterConfig.properties:** This configuration file allows you to configure the CPS VA in terms of language, specification selection and location of files written by the CPS VA.

```
# Select the language
# Implemented languages: de, en
Language = en

# Select the test configuration file
TestConfiguration = ./TestConfiguration.xml

# Select the xml file, where the information model is stored
XmlFile = Informationmodels/gesi.xml

# Select location for storing xml logfile
XmlLogPath = logs/

# Select location for storing json logfile
JsonLogPath = logs/
```

*Illustration 14: Default settings for the demo*

Table 1: Attributes of the AdapterConfig.properties

Language	Default setting: en Demoversion: en Choice between German(de) and English(en)
TestConfiguration	This allows you to specify the location of the configuration for the test. For the demo version the same directory is selected in which the AdapterConfig.properties is located.
XmlFile	Which specification is to be checked and in which directory it is located
XmlLogPath	In which directory the LogFile.json should be saved
JsonLogPath	In which directory the LogFile+Zeistempel.xml should be saved

The location always depends on the directory in which the CPS VA application is stored. However, absolute paths can also be used.







 Informationmodels	27.09.2019 19:12
 logs	27.09.2019 19:13
 AdapterConfig.properties	27.09.2019 19:08
 Anleitung zum CPS VA.docx	27.09.2019 19:08
 TestConfiguration.xml	27.09.2019 19:08
 ValidationAdapter.jar	27.09.2019 19:14

Illustration 15: Directory structure of the demo

For the demo, the configuration as shown in Illustration 5 is used. Illustration 6 shows the standard directory structure of the demo. The relative paths depend on the location of the ValidationAdapter.jar.

- 2.2 **TestConfiguration.xml:** With the TestConfiguration.xml the test can be configured. OPC UA uses the Internet Protocol, which is why a connection can only be established using an IP address. For the demo the configuration as shown in illustration 7 is used.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<testconfiguration>
  <testsystem>GESI</testsystem>
  <guid>f470aeaa-04dd-47d8-9580-d1ff7c54cc7b</guid>
  <ip>localhost</ip>
  <port>4880</port>
  <security>none</security>
  <nodestructure>yes</nodestructure>
  <variable>false</variable>
  <framework>milo</framework>
</testconfiguration>
```

Illustration 16: Testconfiguration.xml



Table 2: Testconfiguration.xml

Testsystem	The name of the tested system. This is used as the file name for the server information model written by the CPS VA.
guid	Global unique identifier, used for unique test identification
IP	Parameters for establishing a connection in the Internet Protocol
Port	Parameters for establishing a connection in the Internet Protocol
Security	Encryption of the connection
Nodestructure	Should the node structure be tested?
Variable	Should variables be tested?
Framework	Selection of the framework for the validation process Currently Milo of the Eclipse Foundation and Java Legacy of the OPC Foundation are implemented

### 3. Starting the CPS Validation Adapter.







 Informationmodels	27.09.2019 19:12
 logs	27.09.2019 19:13
 AdapterConfig.properties	27.09.2019 19:08
 Anleitung zum CPS VA.docx	27.09.2019 19:08
 TestConfiguration.xml	27.09.2019 19:08
 ValidationAdapter.jar	27.09.2019 19:14

Illustration 17: Folder structure of the Validation Adapter

After the configuration for the next test run has been made via the AdapterConfig.properties and TestConfiguration.xml The validation is performed by executing ValidationAdapter.jar.

### 4. Evaluation and results

A validation process is completed with the creation of three files. The "LogFile.json" is a file that contains information about the validation process, the "LogFile + Timestamp.xml" contains information about the results of the validation and a file that represents the structure of the data in the server in an OPC UA specific format.





 GESI.xml
 LogFile.json
 LogFile20190927_1921.xml
 pom.xml

Illustration 18: Results of the validation

Illustration 8 shows a total of four files, which can be found in the "logs" folder. "GESI.xml" is the information model as implemented in the server. The XML corresponds to the standard of an information model as specified by the OPC Foundation. In "LogFile.json" information about the process itself can be found. "LogFile20190927\_1921.xml" is an example for an evaluation of a server. This file also corresponds to the standard of the OPC Foundation but error codes have been added to the faulty attributes in order to make any differences between specification and implementation clear.

```

<UAObject BrowseName="2:IPK" NodeId="ns=2;s=10019 Node is not implemented in Server" ParentNodeId:
  <DisplayName>IPK</DisplayName>
  <References>
    <Reference IsForward="false" ReferenceType="HasComponent">ns=2;s=5005</Reference>
    <Reference ReferenceType="HasTypeDefinition">i=58</Reference>
    <Reference ReferenceType="HasComponent">ns=2;s=22427</Reference>
    <Reference ReferenceType="HasComponent">ns=2;s=22424</Reference>
    <Reference ReferenceType="HasComponent">ns=2;s=19179</Reference>
    <Reference ReferenceType="HasComponent">ns=2;s=19020</Reference>
    <Reference ReferenceType="HasComponent">ns=2;s=999</Reference>
  </References>
</UAObject>

```

*Illustration 19: Incorrect node*

Illustration 9 shows a section of the "LogFile20190927\_1921.xml", with an error. In this case the specification specifies this node with the corresponding parameters, but the CPS VA cannot find these nodes with the corresponding NodeId in the server, so the result is "Node is not implemented in Server" appended to the parameter.