Instructions for using the CPS Emulator and the CPS Validation Adapter

Fraunhofer-Institut für productionsanlagen und konstruktionstechnik

Division manager: Prof. dr. holger kohl

Contact partner: Prof. Dr.-Ing. Thomas Knothe and Frank-Walter Jäkel

creator: tobias wolff and vincent happersberger

Date: 10.08.2021

Table of content

[1 Preface 3](#_Toc79326044)

[2 How to use the CPS Emulator 5](#_Toc79326045)

[2.1 Enter the IP address where the server should run (ENTER for localhost1) 5](#_Toc79326046)

[2.2 Select whether the "specificationConfig.xml"2 should be loaded. Press 'y' for yes or 'n' for no. 5](#_Toc79326047)

[2.2.1 Case - "Yes": The "specificationConfig.xml" is loaded, but must be adapted before! This loads the information models which are then available for selection. 7](#_Toc79326048)

[2.2.1.1 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). 7](#_Toc79326049)

[2.2.1.2 Case - "No": The "specificationConfig.xml" is not loaded. 9](#_Toc79326050)

[2.2.1.3 The emulation can be terminated by closing the window7. 11](#_Toc79326051)

[If the message "Please restart the server with edited information model" 11](#_Toc79326052)

[2.3 At successful start, "Emulation is running" is displayed every few seconds. 12](#_Toc79326053)

[3 Manuel of the CPS Validation Adapter 13](#_Toc79326054)

[3.1 Configuration. 13](#_Toc79326055)

[3.1.1 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. 13](#_Toc79326056)

[3.1.2 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 16 is used. 14](#_Toc79326057)

[3.2 Starting the CPS Validation Adapter. 15](#_Toc79326058)

[3.3 Configuration for starting the CPS Validation Adapter with the UaCPPServer. 15](#_Toc79326059)

[3.4 Evaluation and results 16](#_Toc79326060)

Table of Figures

[Figure 1: Windows Defender Safety Instructions 5](#_Toc79326146)

[Figure 2: Entering the IP address 5](#_Toc79326147)

[Figure 3: Selection regarding the use of "specificationConfig.xml", case "Yes" 6](#_Toc79326148)

[Figure 4: Path specification in the "specifiactionConfig.xml" 7](#_Toc79326149)

[Figure 5: Copy the path from Windows Explorer 7](#_Toc79326150)

[Figure 6: Listing of the information models under the specified path 7](#_Toc79326151)

[Figure 7: Created information model with timestamp 7](#_Toc79326152)

[Figure 8: Selection regarding use of "specificationConfig.xml", case "No" 8](#_Toc79326153)

[Figure 9: Path specification of on information model 8](#_Toc79326154)

[Figure 10: The emulation waits for the path input 8](#_Toc79326155)

[Figure 11: Specifying the path to the scenarios folder 10](#_Toc79326156)

[Figure 12: Output after creating a revised information model 10](#_Toc79326157)

[Figure 13: Output when reading an information model without changes 11](#_Toc79326158)

[Figure 14: Default settings for the demo 12](#_Toc79326159)

[Figure 15: Directory structure of the demo 13](#_Toc79326160)

[Figure 16: TestConfiguration.xml 13](#_Toc79326161)

[Figure 17: Folder structure of the Validation Adapter 14](#_Toc79326162)

[Figure 18: Visual output of the UaCPPServer 15](#_Toc79326163)

[Figure 19: Configuration of the TestConfiguration.xml for the UaCPPServer 15](#_Toc79326164)

[Figure 20: Results of the validation 15](#_Toc79326165)

[Figure 21: Incorrect node 16](#_Toc79326166)

[Table 1: Attributes of the AdapterConfig.properties 10](#_Toc78553873)

[Table 2: TestConfiguration.xml 11](#_Toc78553874)

# 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.

# How to use the CPS Emulator

The CPS(=cyper 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)

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

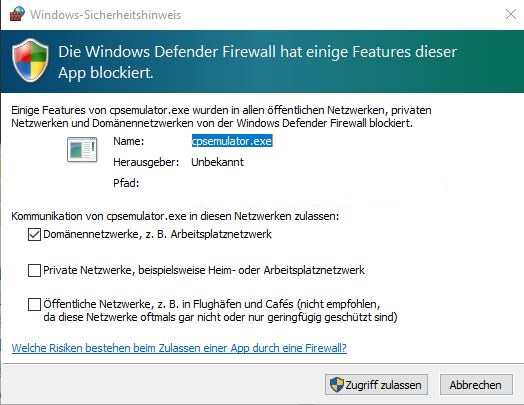


Figure 1: Windows Defender Safety Instructions

## Enter the IP address where the server should run (ENTER for localhost1)

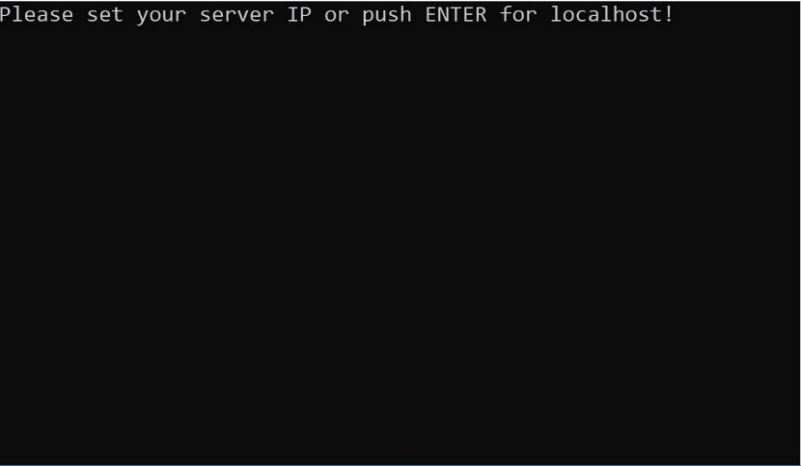


Figure 2: Entering the IP address

## Select whether the "specificationConfig.xml"2 should be loaded. Press 'y' for yes or 'n' for no.

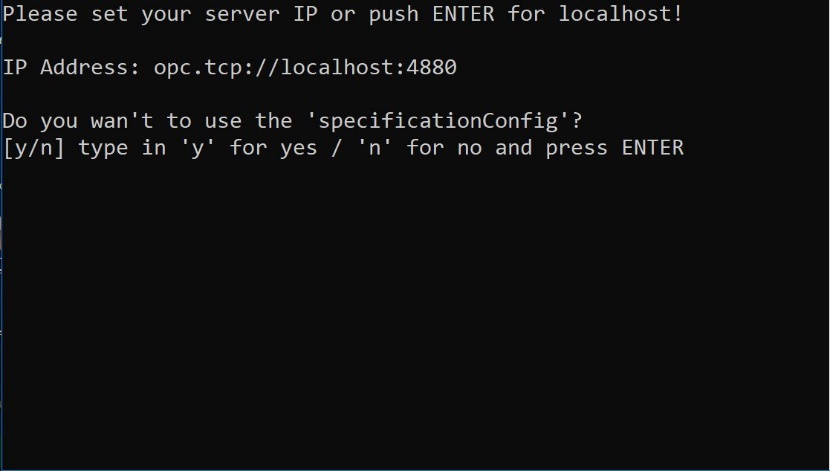


Figure 3: Selection regarding the use of "specificationConfig.xml", case "Yes"

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

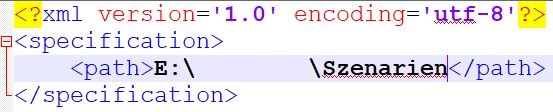


Figure 4: Path specification in the "specifiactionConfig.xml"

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>.



Figure 5: Copy the path from Windows Explorer

#### 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).

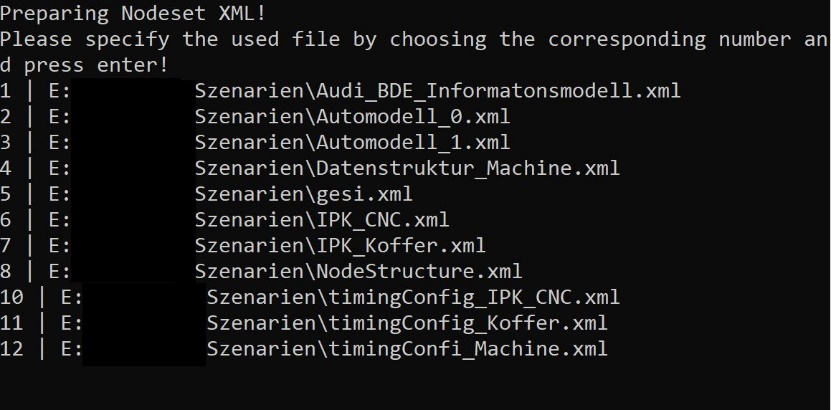


Figure 6: Listing of the information models under the specified path

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 name3. This file contains all changes, the originally read-in information model remains unchanged.

For example:



Figure 7: Created information model with timestamp

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

Select whether the internally stored IPC-Example4 shall be loaded or

ENTER to load an own information model.

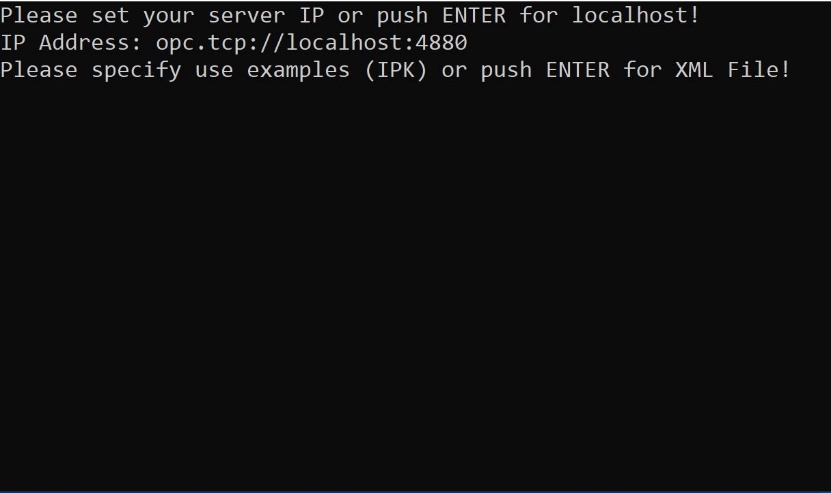


Figure 8: Selection regarding use of "specificationConfig.xml", case "No"

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).

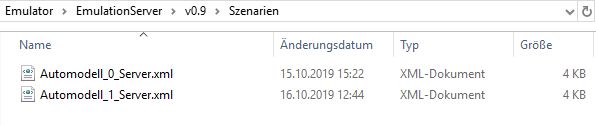


Figure 9: Path specification of on information model

(folder structure, where Automodell\_0.xml without NodeIds5 and Automodell\_1.xml with NodeIds)

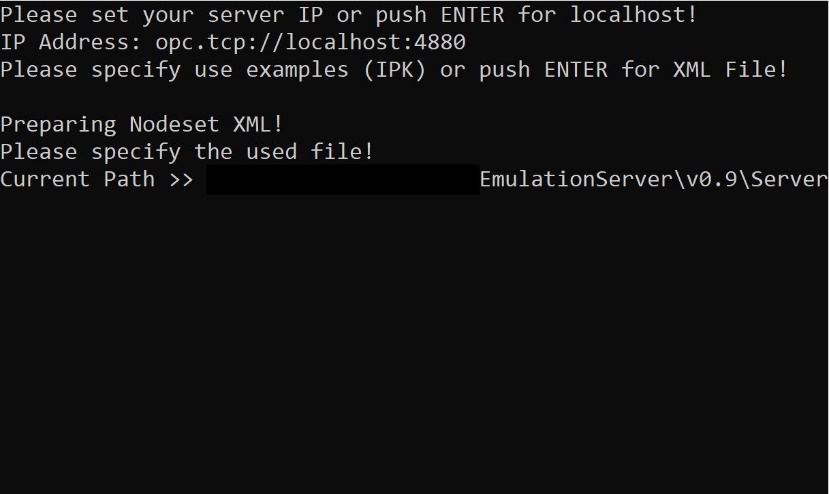


Figure 10: The emulation waits for the path input

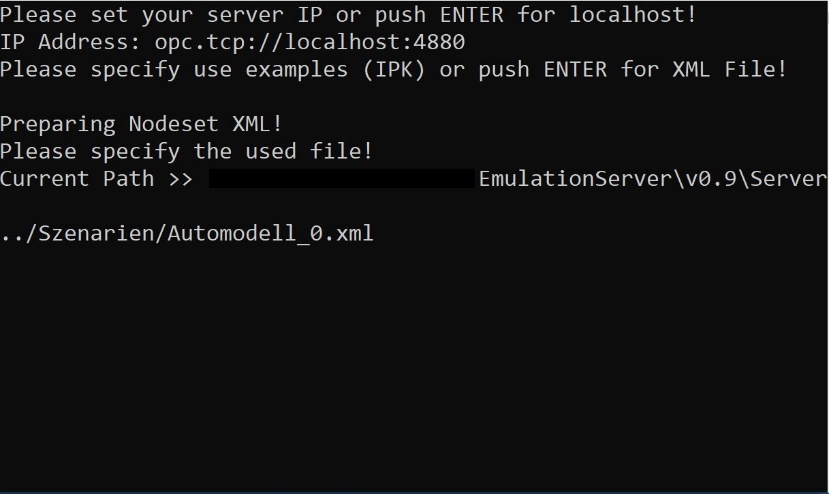


Figure 11: Specifying the path to the scenarios folder6

#### The emulation can be terminated by closing the window7.

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.

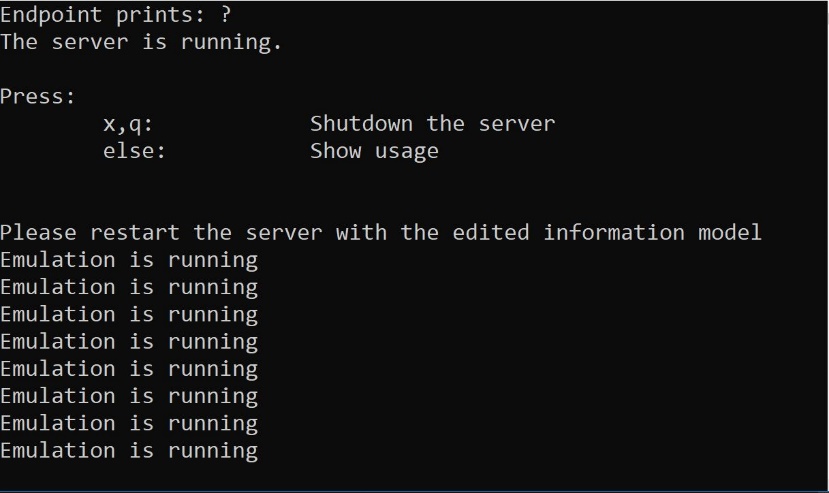


Figure 12: Output after creating a revised information model

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

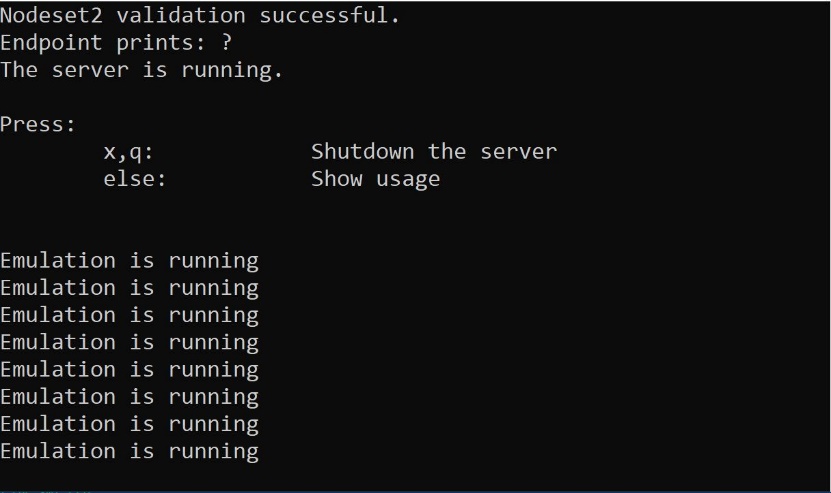


Figure 13: Output when reading an information model without changes

# 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".

## Configuration.

### 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.

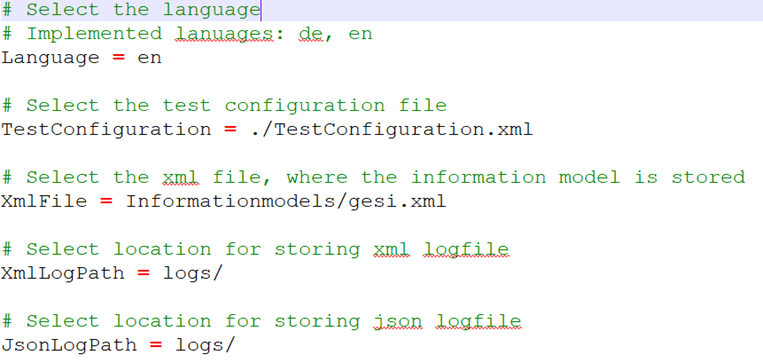


Figure 14: Default settings for the demo

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.

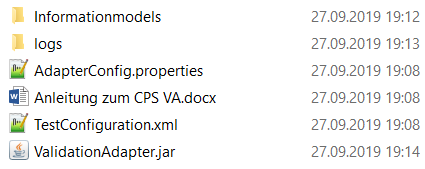


Figure 15: Directory structure of the demo

For the demo, the configuration as shown in Illustration 15 is used. Illustration 6 shows the standard directory structure of the demo. The relative paths depend on the location of the ValidationAdapter.jar.

### 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 16 is used.

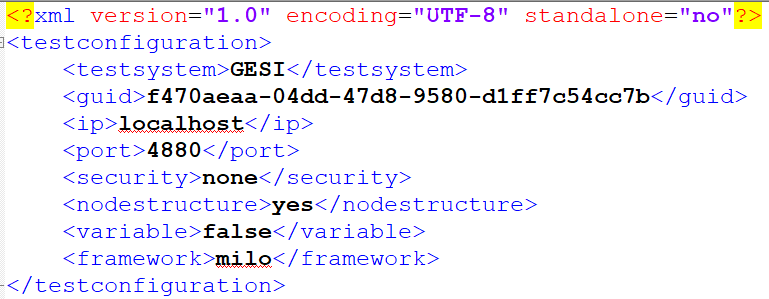


Figure 16: TestConfiguration.xml

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 |

## Starting the CPS Validation Adapter.

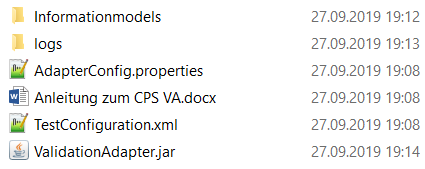


Figure 17: Folder structure of the Validation Adapter

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.

## Configuration for starting the CPS Validation Adapter with the UaCPPServer.

The following pictures will illustrate how to configure the CPS Validation Adapter.

Link to the UaCPPServer (you need to register):

<https://www.unified-automation.com/de/downloads/opc-ua-servers.html>

After starting the UaCPPServer, the default uses the user profile name and a port. You can change the connection type via the UaAdminDialog shown in the instructions.

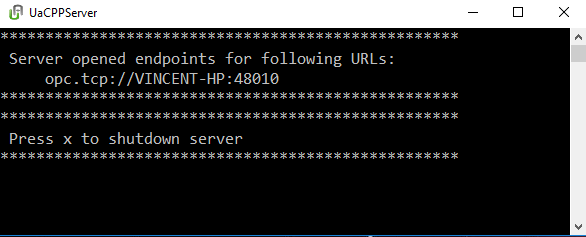


Figure 18: Visual output of the UaCPPServer

The UaCPPServer directly runs after starting the application and shows the IP-Adress and port to connect to.

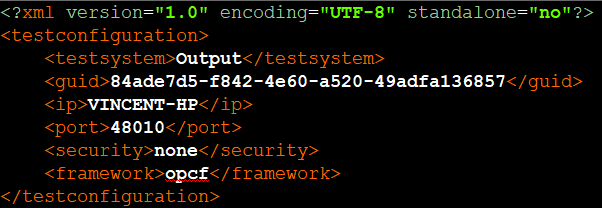


Figure 19: Configuration of the TestConfiguration.xml for the UaCPPServer

After starting the UaCPPServer configure the TestConfiguration.xml by opening it with any text editor. Change the ip as shown in Figure19 and the port right after the two dots.

## 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.

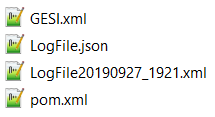


Figure 20: Results of the validation

Illustration 20 shows a total of four files, which can be found in the "logs" folder. 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.

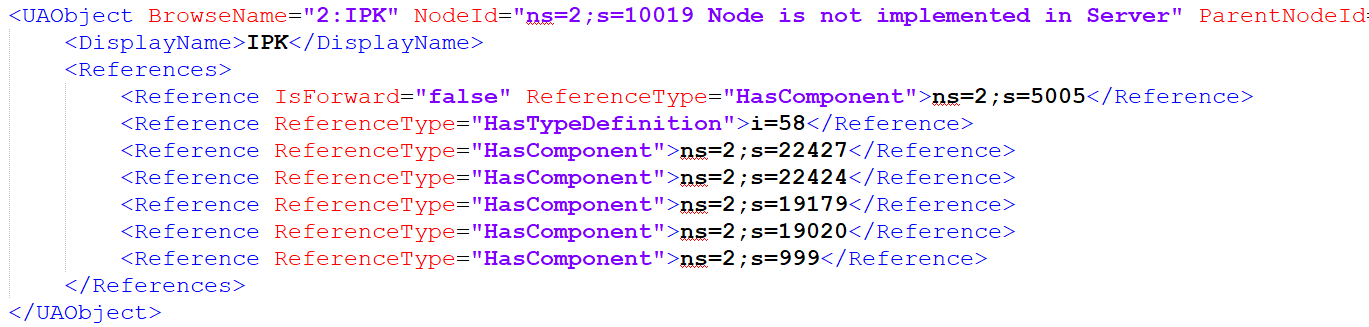


Figure 21: Incorrect node

Illustration 21 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.