



SIMFLEX™ IEC 61850

SCL Checker

User Manual

Jun 2019.



Index

1	Introduction.....	4
1.1	What is SimFlex™ SCL Checker Edition 2.....	4
1.2	Benefits.....	4
1.3	Applications.....	4
1.4	Key Features.....	5
1.5	What is the SimFlex™ SCL Checker used for?.....	6
1.5.1	SCL Header section test cases.....	6
1.5.2	SCL Substation section test cases.....	6
1.5.3	SCL Communication section test cases.....	6
1.5.4	SCL IED section test cases.....	8
1.5.5	SCL IED Services section test cases.....	9
1.5.6	SCL DataTypeTemplate section test cases.....	10
1.5.7	SCL Common IED and DataTypeTemplate section test cases.....	10
1.5.8	Data model (IEC 61850-7-3 and IEC 61850-7-4) test cases.....	11
1.5.9	How should I use the SimFlex™ SCL Checker?.....	12
1.6	System requirements.....	13
1.7	Basic skill requirements.....	13
1.7.1	IEC 61850 standard.....	13
1.7.2	Technical Issues (Tissues).....	13
1.7.3	UCA International Users Group testing.....	13
2	SCL Checker overview.....	14
2.1	Start the SimFlex™ SCL Checker.....	14
2.2	Details in the main screen.....	15
2.3	Configuring the SCL Checker settings.....	17
2.4	Detailed information of Settings dialog, General tab.....	18



- 2.5 Detailed information of Settings dialog, Device tab 19
- 2.6 Detailed information of Settings dialog, Conditionals tab 22
- 3 Use case: Starting and stopping SCL Checker..... 26
 - 3.1 Verify a connection with a DUT..... 26
 - 3.2 Start *SCL Checker* 26
 - 3.3 Configure the right network adapter in *SCL Checker* 26
 - 3.4 Select, Edit or Add a DUT to SCL Checker..... 26
 - 3.5 Load the data model from the DUT..... 28
 - 3.6 Load the SCL file (*.icd/*.cid/*.iid...) that belongs to the DUT 29
- 4 Use case: Select and run test case 31
 - 4.1 Opening and examining a log file 32
- 5 Use case: Configuring a test 35
 - 5.1 Configuration for sMdl2 and sMdl3 35
 - 5.2 Configuration for sMdl16 35
- 6 Use Case: Running a Test Session..... 39
 - 6.1 Opening a Test Session..... 39
 - 6.2 Running tests in a Test Session..... 40
 - 6.3 Overriding test cases results 40
 - 6.4 Saving and Closing a Test Session..... 41
- 7 Use Case: Request Support 43
 - 7.1 Using Support Feature..... 43
- 8 Advanced Topics..... 45
 - 8.1 How to use the data model..... 45
 - 8.1.1 Tree hierarchy 45
 - 8.1.2 How to read the values in the DUT data model (MMSData values) 45
 - 8.1.3 MMSData type and length information: 46



- 9 Frequently Asked Questions (FAQ) 47
 - 9.1 General 47
 - 9.1.1 How to install the SimFlex™ SCL Checker? 47
 - 9.1.2 How can I check if my device is available on the network? 47
 - 9.1.3 The SCL Checker does not start or complains about the license 47
- 10 List of possible error messages 49
 - 10.1 Conventions used 51
 - 10.2 Messages in all sMdl tests 52
 - 10.3 Messages available in all tests (sCnfx and sMdlx) 53
- 11 Additional Topics 55
 - 11.1 File name conventions 55
 - 11.2 Used abbreviations 55
 - 11.3 Referenced documents 55
 - 11.4 Referenced tools 55
- 12 Document Version History 56



1 Introduction

SimFlex™ is a software tool platform for testing and simulating Intelligent Electronic Devices (IED) that are based on the IEC 61850 standard.

This manual describes how to use the *SimFlex™ SCL Checker Edition 2* (“*SCL Checker*” in short). Note that there is also a *SimFlex™ SCL Checker Edition 1* available.

For the installation of the *SimFlex™ SCL Checker* please read the “Getting Started” manual.

1.1 What is SimFlex™ SCL Checker Edition 2

The *SCL Checker* is a tool that checks SCL file conformity to the IEC 61850 SCL schema and performs specific logical tests on the SCL file content. It clearly shows error location within SCL file. Process of verification and correction within file of more than 10 000 lines can be quite time consuming. The *SCL Checker* enables automatic verification of SCL files to Utilities, Manufacturers, System integrators and Conformance test laboratories. The *SCL Checker* comes with an extensive Test Suite that implements always up to date IEC 61850 conformance test cases that can be individually selected and executed. It can be used for:

- 1) IED capability description files (*.icd)
- 2) Configured IED description files (*.cid)
- 3) IED instantiated description files (*.iid)
- 4) System configuration description files(*.scd)
- 5) System specification description files (*.ssd)

Mainly, *SCL Checker* is designed to test *.icd, *.cid and *.iid files against the accompanying IED and the IEC 61850 standard.

SCL Checker is one of the tools from the *SimFlex™* family. The *SCL Checker* implements all sCnfx and sMdlx test cases defined in the document “*Conformance Test Procedures for Server Devices with IEC 61850-8-1 Edition 2 interface, Revision 2.0*” from the UCA® International Users Group.

1.2 Benefits

The main benefits of using the *SCL Checker* are:

- Fast and fully automated verification of SCL files including IEDs data and services
- Reduction of IEC 61850 design and configuration problems during product development, IEC 61850 system integration, commissioning and FAT
- IEC 61850 device certification on the fly (reduction of preparation cost)
- Easy to learn user interface
- Simple configuration

1.3 Applications

The *SCL Checker* has a wide range of applications:

- SCL verification during IEC 61850 device development. system design and integration. configuration and FAT



- IEC 61850 conformance and interoperability testing
- IED product development
- Preparation for UCA® International Users Group based IED certification

1.4 Key Features

The *SimFlex™ SCL Checker* is designed to be easy-to-use tool for the verification of IEC 61850 SCL files. Key features of the *SCL Checker* include:

- Executes the UCA International Users Group test procedures based on IEC 61850-10 Standard
- Verifies that the SCL files are well-formed and according to the schema defined in IEC 61850-6 Standard
- Compares SCL contents and exposed data and services in the IEDs
- Verifies the presence and order of data according the IEC 61850-7-3 and 7-4
- Always up to date to the latest UCA Iug test procedures
- Using official name space description (*.nsd) files keeps it bullet proof to the tool development mistakes
- Graphical view of the SCL file and the IED data model
- Logging of test progress and test results in human-readable text format
- Automatic and repetitive verification of SCL files
- Fast localization of SCL errors
- Remote testing of IEDs (Cloud ready)



1.5 What is the SimFlex™ SCL Checker used for?

The main goal of the *SCL Checker* is to make the conformance testing as automatic as possible and cover all tests specified by the UCA® International Users Group on the device and the SCL (*.icd/*.cid) file. Tests defined by the UCA® International Users Group are:

1.5.1 SCL Header section test cases

Test case	Test case description
sCnf1	Verify the SCL version = "2007", revision = "B"
sCnf2	Verify the XML encoding is UTF-8 or utf-8; <?xml version="1.0" encoding="UTF-8"?>
sCnf3	Verify that the ICD validates according to SCL schema: version 2007, revision B

1.5.2 SCL Substation section test cases

Test case	Test case description
sCnf10	Verify the ICD has at most one Substation or Line or Process exists at SCL level and the attribute "name" is "TEMPLATE". If voltagelevel and bay element are present, their name is "TEMPLATE" Condition: when substation section is present
sCnf11	Verify the ICD has none of the LNode bound to an IED different from "TEMPLATE" Condition: when substation section is present

1.5.3 SCL Communication section test cases

Test case	Test case description
sCnf20	Verify that the "Communication" element exists if <u>either</u> of (IED/Services/DynAssociation or IED/AccessPoint/ Services/DynAssociation is declared) <u>and</u> IED/AccessPoint/ Server is declared
sCnf21	If IED/Services/DynAssociation is declared, for each ConnectedAP/Address element: Verify that exactly one "P" element with attribute type="OSI-PSEL" with a valid value (non-empty, even number of characters, maximum 16 characters 0-9,A-F) Verify that exactly one "P" element with attribute type="OSI-SSEL" with a valid value (non-empty, even number of characters, maximum 16 characters 0-9,A-F)



Test case	Test case description
	<p>Verify that exactly one “P” element with attribute type=“OSI-TSEL” with a valid value (non-empty, even number of characters, maximum 8 characters 0-9,A-F)</p> <p>(Note that if xsi:type mechanism is used then schema validator can automatically verify the type)</p>
sCnf22	<p>Verify that for each accesspoint no more than one “P” element with attribute type=“OSI-AP-Title” and “OSI-AE-Qualifier and ”IP” and “IP-SUBNET”, “IP-GATEWAY”, OSI-NSAP, OSI-AP-Invoke, and OSI-AE-Invoke exists. For each of these that exist:</p> <p>Verify OSI-AP-Title value contains only decimal digits and non-repeating commas</p> <p>Verify OSI-AE-Qualifier value is decimal representation from 0-65535</p> <p>Verify IP and IP-SUBNET and IP-GATEWAY contain a “standard dotted-decimal” for Ipv4 (TISSUE #1208 forbids Ipv6 in Ed2)</p> <p>Verify OSI-AP-Invoke and OSI-AE-Invoke are between 0 and 65535.</p>
sCnf23	<p>For each GSE element, Address/P[type=MAC-Address] right digit of first octet is odd (1,3,5,7,9,B,D,F) (multicast).</p> <p>Condition: when GSE element is present</p>
sCnf24	<p>For each SMV element referencing a SampledValueControl whose attribute multicast=true or missing, verify Address/P[type=MAC-Address] right digit of first octet is odd (1,3,5,7,9,B,D,F) (multicast)</p> <p>For each SMV element referencing a SampledValueControl whose attribute multicast=false, verify Address/P[type=MAC-Address] right digit of first octet is even (0,2,4,6,8,A,C,E) (unicast)</p> <p>Condition: when SMV element is present</p>
sCnf25	Verify the ICD that each Subnetwork/ConnectedAP@iedName is “TEMPLATE”
sCnf26	Verify each Subnetwork/ConnectedAP@apName matches one of IED/AccessPoint@name
sCnf27	<p>Verify for each GSE element, the GSE@cbName points to a GSEControl within the AccessPoint pointed to by GSE//@apName and GSE@IdInst.</p> <p>Condition: when GSE element is present</p>
sCnf28	<p>Verify for each SMV element, the SMV@cbName points to a SampledValueControl within the AccessPoint pointed to by SMV//@apName and SMV@IdInst.</p> <p>Condition: when SMV element is present</p>

**1.5.4 SCL IED section test cases**

Test case	Test case description
sCnf40	Verify the ICD has exactly one IED element and that the attribute “name” of the element is “TEMPLATE”
sCnf41	<p>Verify all FCDA elements reference existing data and that doName and (optional) daName contain correct references. (ref 61850-6 §9.3.7 Table 22).</p> <ul style="list-style-type: none"> • Verify attributes IdInst, InClass, doName, and fc are declared. • Verify attribute InInst is declared if InClass is not “LLN0”. • Verify first component of doName references a DO@name and second component (if any) references a SDO@name within DO referenced by first component • Verify first component of daName (if present) references a DA@name and other component (if any) references a BDA@name within structure hierarchy of the DA referenced by first component • Verify that at most one component of doName/daName contains an index and that ix attribute is identical to this index (see 61850-6 Table 22). Valid example: <FCDA IdInst=“LD0” InClass=“MHA1” InInst=“1” fc=“MX” doName=“HA.phsAHar(0)” daName=“cVal.mag.f” ix=“0” />
sCnf42	Verify DOI/SDI/DAI structures match DataTypeTemplates (DOI@name is valid DO in LD/LN and DAI@name is a leaf within that DO and SDI@name form hierarchy between DOI and DAI)
sCnf43	<p>Verify that the ICD has none of the ExtRef references IEDs different from TEMPLATE or “@”</p> <p>Condition: when ExtRef IED element is present</p>
sCnf44	Verify that the ICD has no ClientLN elements exist within ReportControl and no IEDName elements within GSEControl and SampledValueControl
sCnf45	Verify all GSEControl/SampledValueControl/ReportControl have confRev>0 when datSet is not empty
sCnf46	Verify IED@originalSclVersion and IED@originalSclRevision attributes match corresponding attributes of SCL element (SCL@version and SCL@revision)
sCnf47	<p>Verify multiple identically named DOI/SDI/DAI elements at the same level differ by “ix” attribute (either different “ix” or “ix” attribute not present). See 61850-6 page 173.</p> <p>Condition: when DOI/SDI/DAI ix attribute is present</p>
sCnf48	<p>Verify multiple LLN0.SGCB do not appear in the same logical device hierarchy (defined by LLN0.GrRef which references the parent logical device)</p> <p>Condition: when multiple SGCB are present</p>



Test case	Test case description
sCnf49	Verify element “Log” exists only in LLNO Condition: when Log is present
sCnf50	Verify that the name length of IED, Logical Devices, Logical Nodes, data objects, data attributes, data sets and control blocks do not exceed the maximum length as specified in IEC 61850-7-2 clause 22.2 and SCSM
sCnf51	Verify that logical node LPHD is present in each root logical device (IEC 61850-7-1 clause 8.2.5)

1.5.5 SCL IED Services section test cases

Test case	Test case description
sCnf60	Verify that the attribute nameLength=“64” exists in the IED/Services element
sCnf61	Verify that the Services section must not contradict existing control block and data sets; <ul style="list-style-type: none"> • Nr of DataSet elements <= ConfDataSet.max (if provided). • Nr of ReportControl instances <= ConfReportControl.max (if provided) • Nr of GSEControl <= GOOSE.max (if provided) • Nr of SMVControl <= SMVsc.max (if provided) • Nr of LogControl <= ConfLogControl.max (if provided) • Nr of LGOS instances <= SupSubscription.maxGo (if provided) • Nr of LSVS instances <= SupSubscription.maxSv (if provided)
sCnf62	Verify the AccessPoint/Services element does not contain the attribute nameLength Condition: when AccessPoint Services element is present
sCnf63	Verify AccessPoint/Services element does not contain any of the elements ConfLNs, and ConfLdName Condition: when AccessPoint Services element is present
sCnf64	Verify that in case SupSubscription is claimed to be supported at least one instance of LGOS or LSVS must be in the ICD. Condition: when SupSubscription element is present
sCnf65	Verify that if serviceType=GOOSE is specified for ExtRef the ClientServices.goose=true. For serviceType=SMV the ClientServices.sv=true Condition: when serviceType=GOOSE or serviceType=SMV is present

**1.5.6 SCL DataTypeTemplate section test cases**

Test case	Test case description
sCnf70	Verify for each DAType/BDA or DOType/DA with attribute “bType”=Struct has attribute “type” whose value matches DAType@id; does not declare valKind (TISSUE #823); does not contain a <Val> element
sCnf71	Verify for each DAType/BDA or DOType/DA with attribute “bType”=Enum has attribute “type” whose value matches EnumType@id
sCnf72	Verify type names do not exceed 255 characters, contain no “whitespace” characters and contain only characters from Basic-Latin and Latin-1-Supplement
sCnf73	Verify that each DOType element contains at least one SDO or DA element
sCnf74	Verify for each DA with FC=“CO” (except “SBO”) that the associated DAType contains the element <ProtNs type=“8-MMS”>IEC 61850-8-1:2003</ProtNs> Verify for each DA name=“SBO” (FC=“CO”) contains the ProtNS element
sCnf75	Verify for each (instance of) DOType/DA[name=ctlModel] whose associated EnumType contains direct-with-normal-security has in the DOType a DA named “Oper”. If ctlModel has valKind=RO and valImport=missing/false then use the configured ctlModel value instead of EnumType. Similar for sbo-with-normal-security, Oper, Cancel and SBO Similar for direct-with-enhanced-security, Oper Similar for sbo-with-enhanced-security, Oper, Cancel and SBOw

1.5.7 SCL Common IED and DataTypeTemplate section test cases

Test case	Test case description
sCnf80	Verify that <Val> element values actually match a value in the corresponding EnumType, “ord” shall not be used, only EnumVal element values. Ref IEC 61850-6 Table 45.
sCnf81	Verify that <Val> elements values match IEC 61850-6 Table “Data type mapping” (if no table rows then Val element is not allowed at all)
sCnf82	Verify for each LD/LLN0.NamPlt.IdNs, a <Val> element exists with a valid namespace referring to Edition 2: IEC 61850-7-4:2007 or IEC 61850-7-4:2007A
sCnf83	Verify each ctlModel has an associated <Val> element
sCnf84	Verify CDC=ORG references use the ACSI format (with “.”, no “\$” and no functional constraint, TISSUE 1223) and that the reference does exist Condition: when a data object with CDC=ORG is present



Test case	Test case description
sCnf85	Verify for each LLNO, the existence and non-null value for Data Attribute NamPlt.configRev

1.5.8 Data model (IEC 61850-7-3 and IEC 61850-7-4) test cases

Test case	Test case description
sMdl1	Verify presence of mandatory data objects for each LN type and data attributes for each DO type. Passed when all objects/attributes are present
sMdl2	Verify presence of conditional presence true data objects for each LN type and data attributes for each DO type. Passed when all objects/attributes are present
sMdl3	Verify non-presence of conditional presence false data objects for each LN type and data attributes for each DO type. Passed when these objects/attributes are not present
sMdl4	Verify data model mapping according to applicable SCSM concerning name length and object expansion. Passed when mapping is according to applicable SCSM
sMdl5	Verify data model mapping according to applicable SCSM concerning organisation of functional components.
sMdl6	Verify data model mapping according to applicable SCSM concerning naming of control blocks and logs. Passed when mapping is according to applicable SCSM.
sMdl7	Verify type of all data objects for each LN type and all data attributes for each DO type. Passed when type of all objects/attributes do match with the IEC 61850-7-3, IEC 61850-7-4 and the applicable SCSM
sMdl8	Verify that the enum types and values from the SCL and in the device are in specified range. Passed when all enum types and values match the 2007A2.nsd.
sMdl9	Check if manufacturer specific data model extensions are implemented according to the extension rules in IEC 61850-7-1 clause 14.
sMdl10	Check if the order of the data attributes with the same functional constraint of the DO type match with IEC 61850-7-3. Passed when all attributes are in matching order
sMdl11	Moved to sCnf50
sMdl12	Check that the rules for multiple data object instantiation are kept (IEC 61850-7-1 clause 14.6, IEC 61850-7-4).
sMdl13	Moved to sCnf82
sMdl14	Check the correct use of name spaces for non-substation power utility applications like for example Hydro and DER. Condition: when non-substation name space is used



Test case	Test case description
sMdl15	Check if the SCL configuration file used to configure the DUT corresponds with the actual data object references, data types, data sets and pre-configured data values (settings) exposed by the DUT on the network.
sMdl16	Change one parameter/setting of each configurable data type and FC (FC can be DC, CF or SP) using the supplied configuration tool and check the updated online parameter/setting values correspond with the configured values in the SCL. Document the tested parameters in the test report. Condition when a parameter/setting is configurable
sMdl17	Check the “ldName” naming structure when supported. All online object references (including data sets, control block references and object references – CDC ORG) shall start with the “LDevice ldName” value instead of the “IED name” + “LDevice inst” Condition when Services ConfLdName is present
sMdl18	Verify that the indicated trigger option: <DA dchg, qchg, dupd > is conformant with the IEC 61850-7-3 standardized Trigger Option.

Not all tests are fully automatic; some of them needs manual intervention of the test engineer to reconfigure the DUT and reloading a changed SCL (*.icd/*.cid) file.

1.5.9 How should I use the SimFlex™ SCL Checker?

Typically, the user of the *SCL Checker* is a test engineer. The *SCL Checker* is used as a tool that will retrieve information from the DUT (Device Under Test) and its accompanying SCL file (typically an *.icd/*.cid file). Each test case can be executed separately or in any combination. The results are stored in a log file (per test case) and the test engineer can review the results of the tests by checking both the log file and, if applicable, the capture file with communication between DUT and the *SCL Checker*.

In order to fully deploy all capabilities of the *SCL Checker*, the tools need to be connected to an IED and its accompanying SCL file. The log files and capture files are stored in some storage facility, by default a folder on the local hard disk.



1.6 System requirements

The *SCL Checker* has the following software and hardware requirements:

- Windows 7 SP1, Windows 10
- 1GB of RAM or more
- 1 GHz processor or faster
- 250 MB of free hard disk space
- At least one 100 Mbit Network Interface Card

IMPORTANT: *SCL Checker* software has to be installed using an account with administrator rights.

IMPORTANT: The license for the *SimFlex™ SCL Checker* is not supported on virtual machines.

Before installing the *SCL Checker* software please make sure that you have installed Microsoft's .Net framework version 4.0 Full Profile (or better) on your computer. Without installing .Net Full Profile the *SimFlex™* installer and/or the program itself will not run properly.

The .Net Full Profile installation file can be downloaded from the Microsoft web site:

<http://www.microsoft.com/en-us/download/details.aspx?id=17718>

1.7 Basic skill requirements

The test engineer must have at least basic level of understanding of IEC 61850 standards. This is important because the tester will use it to test the device and accompanied SCL file, diagnose and analyze the result.

1.7.1 IEC 61850 standard

The IEC 61850 can be purchased on the IEC web site. See: <http://webstore.iec.ch>

1.7.2 Technical Issues (Tissues)

Information about technical issues is available at: <https://iec61850.tissue-db.com/default.mspix>

1.7.3 UCA International Users Group testing.

Information about the testing procedures could be retrieved by contacting the UCA® International Users Group.

Please be aware that to access certain documents on the UCA® web site, you might need to become a member of the UCA® International Users Group.

For more information see: <http://www.ucaiug.org>



2 SCL Checker overview

2.1 Start the SimFlex™ SCL Checker

Locate the *SCL Checker* icon on your desktop or in your Start menu and (double) click on it. The main screen of the program will open.

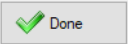


Figure 2-1 SCL Checker icon

Enter or change the name of test engineer that will use the program for testing

UserName

Figure 2-2 Log on procedure

Press  when finished. Main screen will appear.



2.2 Details in the main screen

The SCL Checker main screen is shown below:

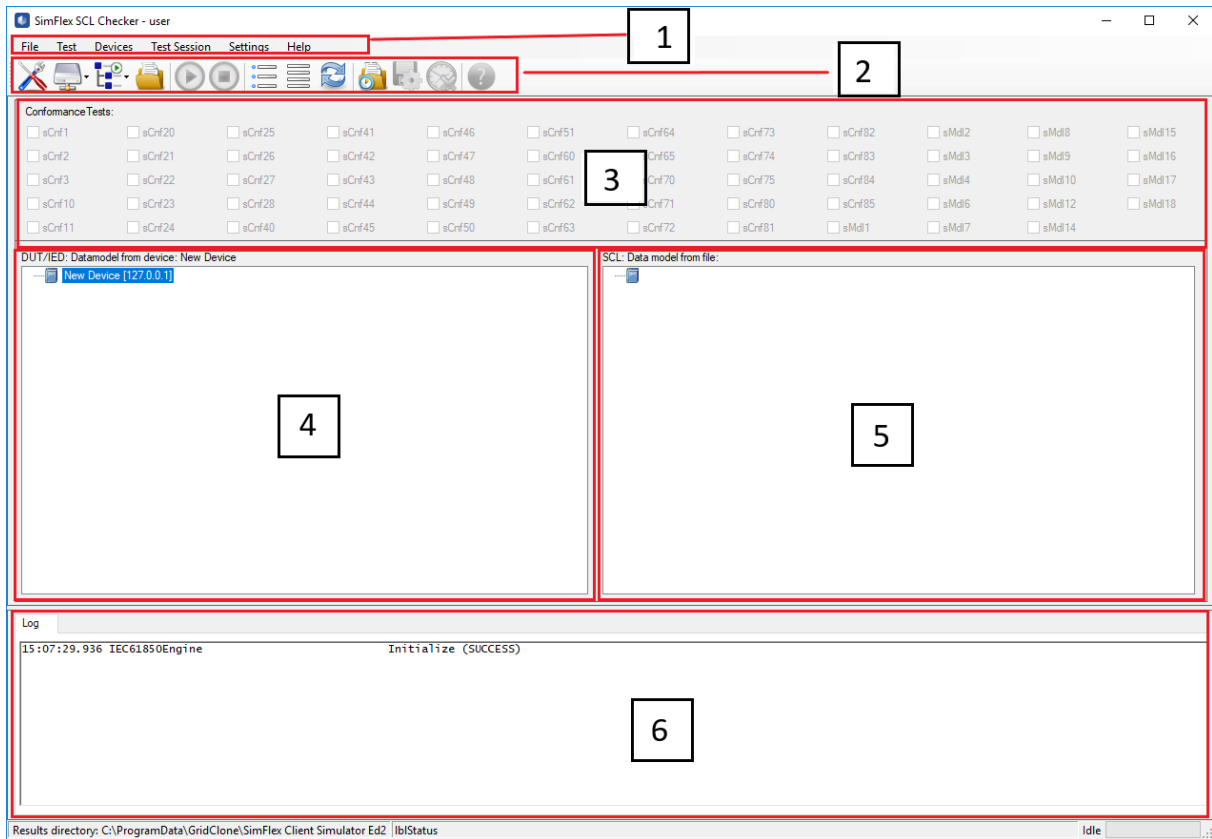















Figure 2-3 SimFlex™ SCL Checker Main screen

The main components of the application window:

- A Menubar (1) and Toolbar (2) with (from left to right)

-  The configuration icon
-  The device icon/submenu
-  The DUT model icon/submenu
-  The SCL (*.icd/*.cid/*.iid/*.scl...) model icon
-  The Run selected test icon.



-  The abort test icon.
-  The select all tests icon.
-  The deselect all tests icon.
-  The deselect and reset all tests icon.
-  The open session icon.
-  The save session icon.
-  The close session icon.
-  The Support icon

- The test case frame (3)

This frame shows all tests cases.

Available tests are enabled; not available tests are disabled.

A test becomes available when the data model for it (DUT and/or SCL) has been retrieved from the DUT or the applicable SCL file has been loaded.

- The DUT model frame (4)

Load the data model from the DUT by right-clicking on the frame and then select “Build data model” to build a data model with Readable information only. Select “Build data model and check RW status” for full Readable/Writable information.

You can also use the DUT model icon/submenu for this.

- The SCL (*.icd/*.cid/*.iid/*.scd...) model frame (5)

Load the data model from the SCL file by right-clicking on the frame and then select “Build data model”, select the proper SCL (*.icd/*.cid/*.iid/*.scd...) file in the upcoming file selector window.

- The Log and info frame (6)

This frame will keep all messages, warnings and other information and communication status messages.



2.3 Configuring the SCL Checker settings

In order to configure the *SCL Checker* settings:

- 1) In the menu bar click in *Settings -> General*

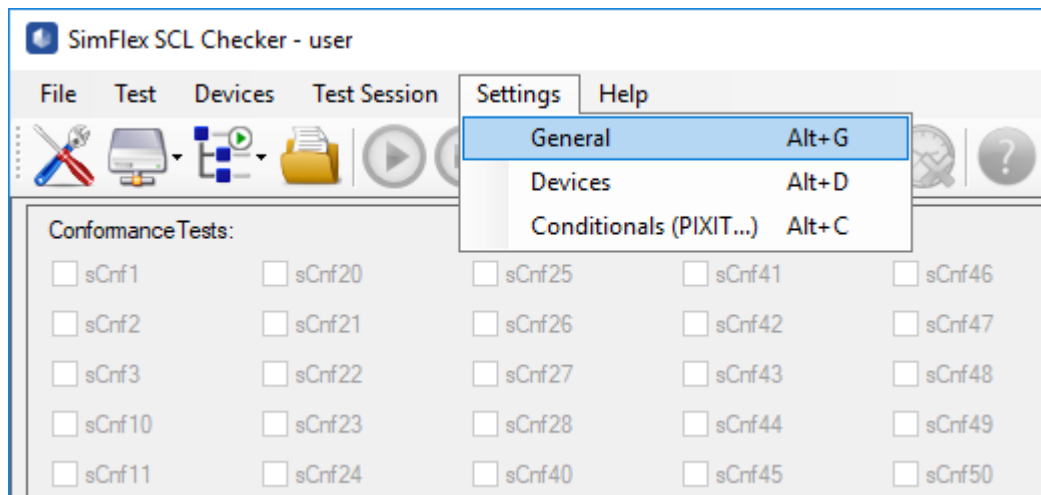


Figure 2-4 Settings from Menu Bar

- 2) Or, in the Toolbar, hover above that will say “Change settings” and click the icon.

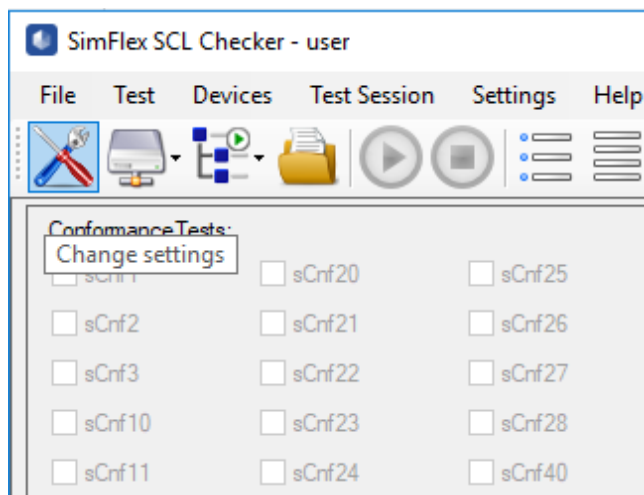


Figure 2-5 Settings from Tool Bar

New window will show up. The window consists of three tabs. [**General**] tab will be opened by default.



2.4 Detailed information of Settings dialog, General tab

In the Settings dialog, General tab, you can specify program settings.

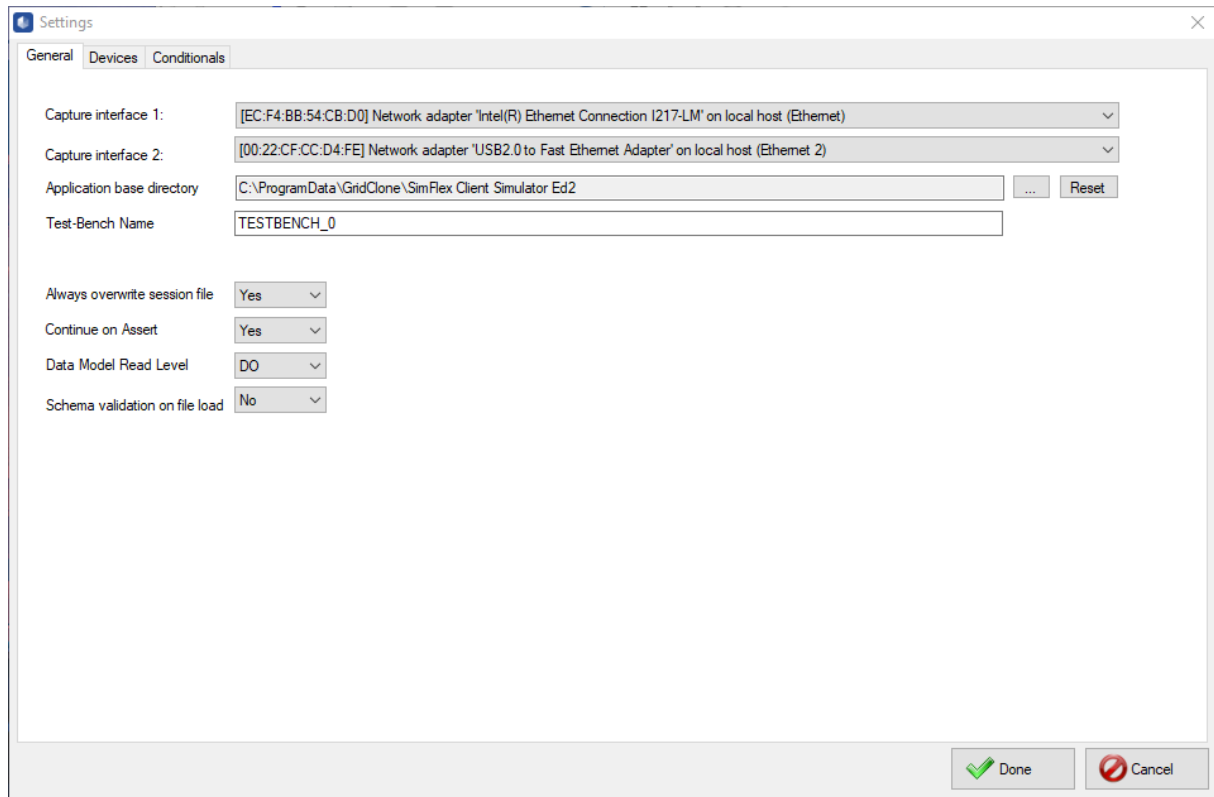


Figure 2-6: Settings > General

Capture interface 1

Select Primary Network Interface Card that will capture the network traffic during the execution of test cases and be used as main connection interface for communication with DUT

Capture interface 2

Select Secondary Network Interface Card (not used)

Application base directory

Select the directory that will be the base for the storage of all program related and created files. This includes amongst others the network capture files, the log files, the defined DUTs and all session related files.

Test-Bench name

Specify the name of the test bench. The test bench is the system on which the SCL Checker has been installed on.

Always overwrite session file

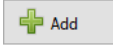
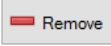
Select **Yes** when the session file should be overwritten. This setting is only applicable when running test cases during a session.



Continue on Assert	Select Yes if user need for testing script to continue until the end even though Failure is detected. It is helpful, when complete overview of test script results is needed.
Data Model Read Level	Selection between LN, FC, DO i.e. level of DUT data model readout will be stopped on selected level.
Schema validation on file load	If Yes, SCL file will be submitted to initial Schema validation and will not allow import of defected files. Selecting No, will allow more detailed tests and file diagnostics to be performed.

Note: This configuration item is shared with the *SimFlex™ Client Simulator*. Changing it here will also change it for the *Client Simulator*.

2.5 Detailed information of Settings dialog, Device tab

In the Settings dialog, Devices tab, you can define the Devices Under Test (DUT). On the left side of the dialog a list of defined DUTs is shown. Selecting a DUT shows the settings in the *Details* part of the dialog. You can add new devices by pressing the  button and remove devices by pressing the  button. In the *Details* part of the dialog you can specify the **Device name**, **Vendor Name** and the **IP address** of the DUT. You can also enter text to describe the version of the DUT (e.g. the HW version, the SW version and the firmware version of the DUT). This text field is free-format. In the **Comments** field you can enter any information you may find important for the DUT.

The following figure shows the settings that can be configured in the Devices tab.

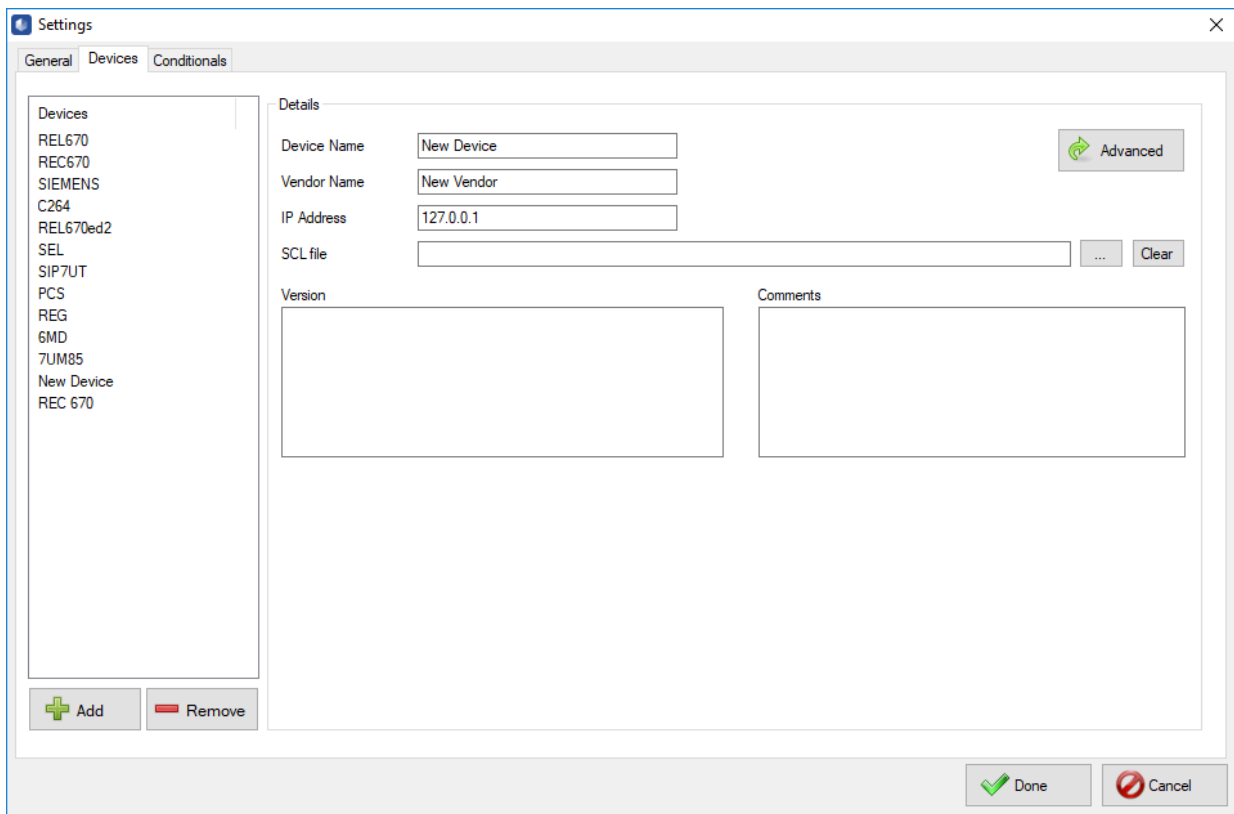
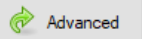
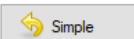


Figure 2-7: Settings > Devices

The  **Advanced** button will reveal fields that might be important during the association of the *SCL Checker* with the DUT. By default, the pre-configured values will do for most DUTs. Please refer to the PICS document of the manufacturer of the DUT for information regarding the settings. Changing these settings might cause the *SCL Checker* not being able to properly connect to a DUT.

By pressing button , **Simple**, Advanced setting will be hidden again.

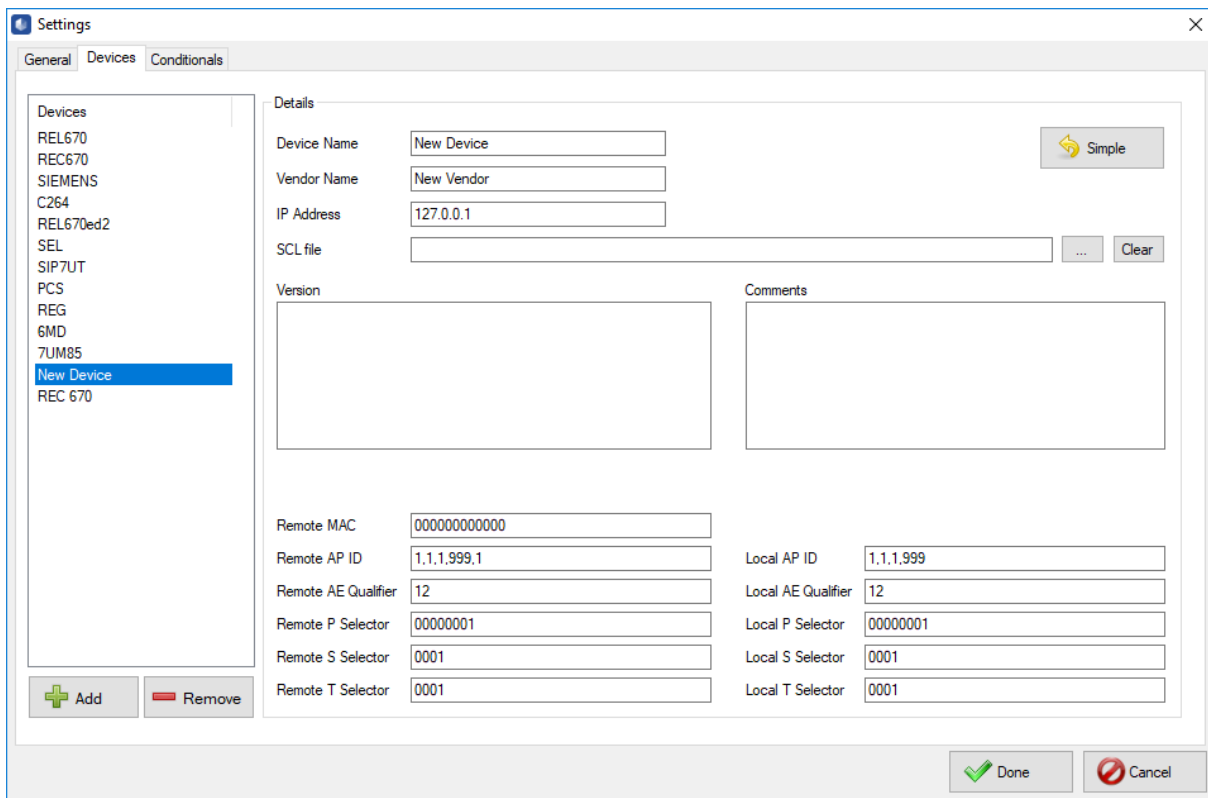


Figure 2-8: Settings > Devices with Advance view



2.6 Detailed information of Settings dialog, Conditionals tab

Conditional tab is showing presence conditions defined by the Standard (parts IEC 61850-7-3, 8-1...) with possibility to change values per each presence condition as specified in DUT PICS and PIXIT. Thus, it will condition execution i.e. final results of certain tests on DUT.

Group	Name	Value	Range	Description
-7-3:2003, -7-3:2009A 5	Mandatory	True	{True}	Attribute is mandatory.
-7-3:2003, -7-3:2009A 5	PICS_SUBST	False	{True, False}	Attribute is mandatory, if substitution is supported (for substitution, see IEC 61850-7-2).
-7-3:2003, -7-3:2009A 5	GC_1	True	{True}	At least one of the attributes shall be present for a given instance of DATA.
-7-3:2003 5	GC_2	True	{True}	All or none of the data attributes belonging to the same group (n) shall be present for a given instance...
-7-3:2009A 5	GC_2_n	True	{True}	All or none of the data attributes belonging to the same group (n) shall be present for a given instance...
-7-3:2003 5	GC_CON	True	{True}	A configuration data attribute shall only be present, if the (optional) specific data attributes to which thi...
-7-3:2009A 5	GC_CON_attr	True	{True}	A configuration data attribute shall only be present, if the (optional) specific data attributes (attr) to wh...
-7-3:2009A 5	GC_2_XOR_n	True	{True}	A configuration data attribute shall only be present, if the (optional) specific data attributes (attr) to wh...
-7-3 5	AC_LLN0_M	True	{True}	The attribute shall be present if the data NamPit belongs to LLN0; otherwise it may be optional.
-7-3:2003, -7-3:2009A 5	AC_LLN0_EX	True	{True}	The attribute shall be present only if the data NamPit belongs to LLN0 (applies to IdNs in CDC LPL o...
-7-3:2003, -7-3:2009A 5	AC_DLD_M	True	{True}	The attribute shall be present, if LN name space of this LN deviates from the LN name space referen...
-7-3:2003, -7-3:2009A 5	AC_DLN_M	True	{True}	The attribute shall be present, if LN name space of this LN deviates from the LN name spacereferen...
-7-3:2003, -7-3:2009A 5	AC_DLNDA_M	True	{True}	The attribute shall be present, if CDC name space of this data deviates from the CDC namespace ref...
-7-3:2003, -7-3:2009A 5	AC_SCAV	True	{True}	The presence of the configuration data attribute depends on the presence of i and f of the Analog Va...
-7-3:2003, -7-3:2009A 5	AC_ST	False	{True, False}	The attribute is mandatory, if the controllable status class supports status information.
-7-3:2003 5, -8-1:2003 E.12	AC_CO_M	False	{True, False}	-7-3: If the controllable status class supports control, this attribute is available and a mandatory attribu...
-7-3:2003, -7-3:2009A 5	AC_CO_O	False	{True, False}	If the controllable status class supports control, this attribute is available and an optional attribute.
-7-3:2009A 5	AC_CO_SBO	False	{True, False}	If the controllable status class supports control and if the control model support the values "sbo-with-n...
-7-3:2003, -7-3:2009A 5	AC_SG_M	False	{True, False}	The attribute is mandatory, if setting group is supported.
-7-3:2003, -7-3:2009A 5	AC_SG_O	False	{True, False}	The attribute is optional, if setting group is supported.
-7-3:2009A 5	AC_SG_C1	False	{True}	One of the attributes is mandatory, if this data shall be member of a setting group.
-7-3:2003, -7-3:2009A 5	AC_NSNG_M	False	{True, False}	The attribute is mandatory, if setting group is not supported.
-7-3:2003, -7-3:2009A 5	AC_NSNG_O	False	{True, False}	The attribute is optional, if setting group is not supported.
-7-3:2009A 5	AC_NSNG_C1	False	{True, False}	One of the attributes is mandatory, if this data shall be setting ousite a setting group.

Figure 2-9 Settings > Devices

Description of the columns:

- Group** Reference to the IEC 61850 part and chapter where this condition is specified. Fixed, for reference only.
- Name** Name of the condition (as described in IEC 61850 Standard). Fixed, for reference only.
- Value** True (the condition is applicable) or False (the condition is **not** applicable). Can be set to a value available in the Range column (next column to the right.)
- Range** Accepted values for the previous column. If only one value is available in this column then the Value column is set and fixed to this value. This column is informative. Fixed, for reference only.
- Description** A description of the condition as given in the IEC 61850. This column is informative. Fixed, for reference only.

Default setting are given in following table.



Group	Name	Value	Range	Description
-7-3:2009A 5	AC_CLC_O	False	{True, False}	The attribute shall be optional, when the calculation type (according to data ClcMth) for this LN is Peak fundamental or RMS fundamental. The attribute shall not be available, if ClcMth is TRUE RMS.
-8-1:2003 E.12	AC_CO_E_M	False	{True, False}	Attribute is mandatory, if control models with enhanced security are used
-7-3:2003 5, -8-1:2003 E.12	AC_CO_M	False	{True, False}	-7-3: If the controllable status class supports control, this attribute is available and a mandatory attribute. -8-2: Attribute is mandatory, if the controllable status class supports control.
-7-3:2003, -7-3:2009A 5	AC_CO_O	False	{True, False}	If the controllable status class supports control, this attribute is available and an optional attribute.
-7-3:2009A 5	AC_CO_SBO	False	{True, False}	If the controllable status class supports control and if the control model supports the values "sbo-with-normal-security" or "sbo-with-enhanced-security" or both, that attribute shall be mandatory.
-8-1:2003 E.12	AC_CO_SBO_N_M	False	{True, False}	Attribute is mandatory, if the control model SBO with normal security is used.
-8-1:2003 E.12	AC_CO_SBOW_E_M	False	{True, False}	Attribute is mandatory, if the control model SBO with enhanced security is used.
-8-1:2003 E.12	AC_CO_TA_E_M	False	{False}	Attribute is mandatory, if time activated control and enhanced security are used.
-7-3:2003, -7-3:2009A 5	AC_DLD_M	True	{True}	The attribute shall be present, if LN name space of this LN deviates from the LN name space referenced by IdNs of the logical device in which this LN is contained (applies to InNs in CDC LPL only).
-7-3:2003, -7-3:2009A 5	AC_DLN_M	True	{True}	The attribute shall be present, if LN name space of this LN deviates from the LN name space referenced by IdNs of the logical device in which this LN is contained (applies to InNs in CDC LPL only).
-7-3:2003, -7-3:2009A 5	AC_DLNDA_M	True	{True}	The attribute shall be present, if CDC name space of this data deviates from the CDC name space referenced by either the dataNs of the data, the InNs of the logical node in which the data is defined or IdNs of the logical device in which the data is contained (applies to cdcNs and cdcName in all CDCs only).
-7-3:2003, -7-3:2009A 5	AC_LNO_EX	True	{True}	The attribute shall be present only if the data NamPlt belongs to LLNO (applies to IdNs in CDC LPL only).
-7-3 5	AC_LNO_M	True	{True}	The attribute shall be present if the data NamPlt belongs to LLNO; otherwise it may be optional.
-7-3:2009A 5	AC_NS_G_C1	False	{True, False}	One of the attributes is mandatory, if this data shall be setting outside a setting group.
-7-3:2003, -7-3:2009A 5	AC_NS_G_M	False	{True, False}	The attribute is mandatory, if setting group is not supported.
-7-3:2003, -7-3:2009A 5	AC_NS_G_O	False	{True, False}	The attribute is optional, if setting group is not supported.
-7-3:2003, -7-3:2009A 5	AC_RMS_M	False	{True}	The attribute is mandatory when the harmonics reference type is rms.
-7-3:2003, -7-3:2009A 5	AC_SCAV	True	{True}	The presence of the configuration data attribute depends on the presence of i and f of the Analog Value of the data attribute to which this configuration attribute relates. For a given data object, that



Group	Name	Value	Range	Description
				attribute 1) shall be present, if both i and f are present, 2) shall be optional if only i is present and 3) is not required if only f is present NOTE If only is present in a device without floating point capabilities, the configuration parameter may be exchanged offline.
-7-3:2009A 5	AC_SG_C1	False	{True}	One of the attributes is mandatory, if this data shall be member of a setting group.
-7-3:2003, -7-3:2009A 5	AC_SG_M	False	{True, False}	The attribute is mandatory, if setting group is supported.
-7-3:2003, -7-3:2009A 5	AC_SG_O	False	{True, False}	The attribute is optional, if setting group is supported.
-7-3:2003, -7-3:2009A 5	AC_ST	False	{True, False}	The attribute is mandatory, if the controllable status class supports status information.
-7-3:2003, -7-3:2009A 5	GC_1	True	{True}	At least one of the attributes shall be present for a given instance of DATA.
-7-3:2003 5	GC_2	True	{True}	All or none of the data attributes belonging to the same group (n) shall be present for a given instance of DATA.
-7-3:2009A 5	GC_2_n	True	{True}	All or none of the data attributes belonging to the same group (n) shall be present for a given instance of DATA.
-7-3:2009A 5	GC_2_XOR_n	True	{True}	A configuration data attribute shall only be present, if the (optional) specific data attributes (attr) to which this configuration relates, is also present.
-7-3:2003 5	GC_CON	True	{True}	A configuration data attribute shall only be present, if the (optional) specific data attributes to which this configuration relates, is also present.
-7-3:2009A 5	GC_CON_attr	True	{True}	A configuration data attribute shall only be present, if the (optional) specific data attributes (attr) to which this configuration relates, is also present. (Note: similar to -7-3:2003 GC_CON).
-7-4:2009A 5	LN_ALG	False	{True}	Data shall be linked to PID algorithm.
-7-4:2009A 5	LN_COUNT	False	{True}	Multiple instances are only allowed in case of a compound data type.
-7-4:2009A 5	LN_FIXED	False	{True, False}	If a value is fixed this attribute acts as a setting
-7-4:2009A 5	LN_MODEL	False	{True, False}	Mandatory if modeling is needed.
-7-4:2009A 5	LN_MULTIPLE	False	{True, False}	Mandatory if more than one pair is needed.
-7-4:2009A 5	LN_OOST	False	{True, False}	Mandatory if RPSB is used for "Out of step tripping".
-7-4:2009A 5	LN_PSB	False	{True, False}	Mandatory if RPSB is used for "Power swing blocking".
-7-4:2009A 5	LN_REF	False	{True}	The DATA is mandatory if the referred DATA is available and is not allowed if the referred DATA is not available.
-7-4:2009A 5	LN_VISIBLE	False	{True, False}	Mandatory if the DATA is visible.
-7-3:2003, -7-3:2009A 5	Mandatory	True	{True}	Attribute is mandatory.



Group	Name	Value	Range	Description
-7-2:2007B	MFcond	True	{True, False}	Parameter condID: condition number (>0). Textual presence condition (non-machine processable) with reference condID to context specific text. If satisfied, the element is mandatory, otherwise forbidden.
-7-2:2007B	MFsubst	False	{True, False}	Element is mandatory if substitution is supported (for substitution, see IEC 61850-7-2), otherwise forbidden.
-7-2:2007B	MOcond	True	{True, False}	Parameter condID: condition number (>0). Textual presence condition (non-machine processable) with reference condID to context specific text. If satisfied, the element is mandatory, otherwise optional.
-7-2:2007B	MOoperTm	False	{True, False}	Element is mandatory if at least one controlled object on the IED supports time activation service; otherwise it is optional.
-7-2:2007B	OFcond	True	{True, False}	Parameter condID: condition number (>0). Textual presence condition (non-machine processable) with reference condID to context specific text. If satisfied, the element is optional, otherwise forbidden.
-7-3:2003, -7-3:2009A 5	PICS_SUBST	False	{True, False}	Attribute is mandatory, if substitution is supported (for substitution, see IEC 61850-7-2).



3 Use case: Starting and stopping SCL Checker

3.1 Verify a connection with a DUT

Make sure that both IED and the PC that is running the *SCL Checker* are connected to the same network and that they are in the same IP address range.

Example:

IED

IP address = 192.168.0.100, mask = 255.255.255.0

SimFlex SCL Checker

IP address = 192.168.0.90, mask 255.255.255.0

Choose the right interface that connects the client (PC used to run *SCL Checker*) and the server/device.

NOTE: Please check your Windows OS manual on how to change adapt the IP address and network mask.

3.2 Start *SCL Checker*

Open the *SCL Checker* as you would normally open any Windows application. You may find a link under the Start menu in the GridClone folder.

Another way to start the program is to double-click the *SimFlex™ IEC 61850 SCL Checker* icon on the desktop (available when this option has selected during the installation of the software).

3.3 Configure the right network adapter in *SCL Checker*

Please check chapter **2.4 Detailed information of Settings dialog, General tab** for more information on how to configure the network interface cards in the *SCL Checker*.

3.4 Select, Edit or Add a DUT to *SCL Checker*

Use the menu item *Devices* to *Edit*, *Add* or *Select* devices.

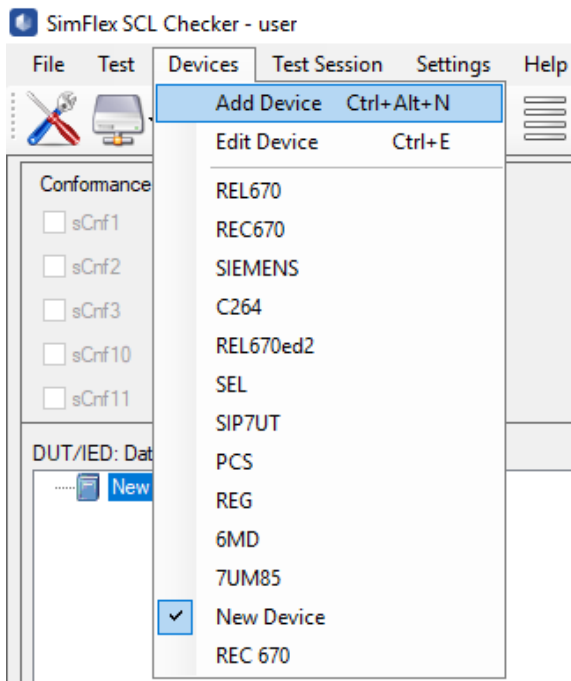


Figure 3-1 Edit, Add or Select device from Menu bar

You may also use the button **Devices**.

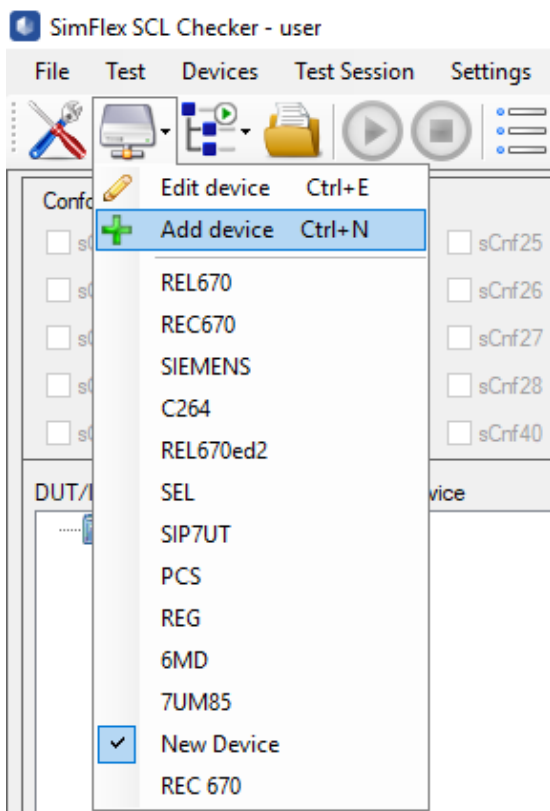


Figure 3-2 Edit, Add or Select device from Tool bar



3.5 Load the data model from the DUT

Once you have configured and selected device in *SCL Checker* you can connect to it and let the program build a data model. This data model will become available in the data model frame. During this process the log frame at the bottom will give information about the progress.

Use the [**Data model**] → **Build Datamodel** icon in the Toolbar to retrieve the data model from the DUT (there is no associated menu item).

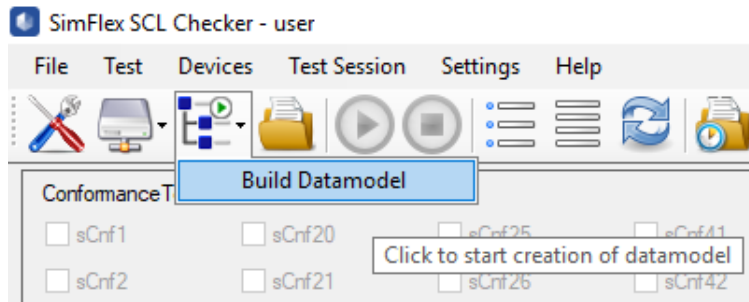


Figure 3-3: Build a data model

Or with right-click on the device name in the DUT frame and select [**Build datamodel**] in the popup window.

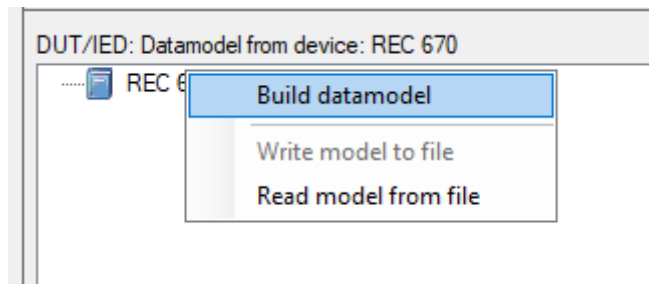


Figure 3-4: Build a data model

After building the model a model icon will appear in the data model frame, indicating that information about the data model are available.

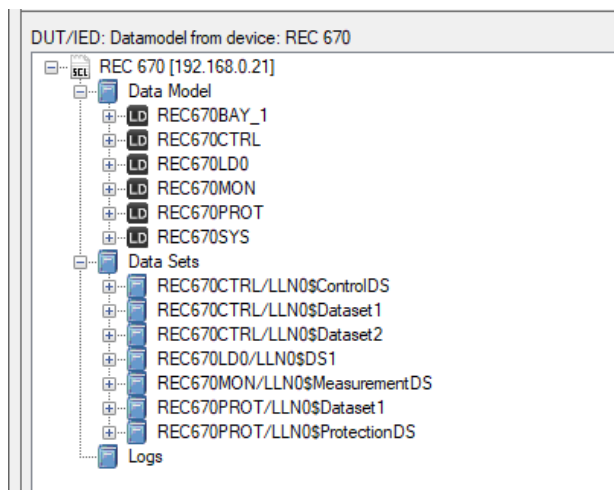




Figure 3-5: Data model retrieved from the DUT

Once loaded, user can right click on Datamodel icon and save model for offline tests (when DUT is not available) by selecting [**Build datamodel**] >**Write model to file**.

Reimport can be later done by selecting [**Build datamodel**] >**Read model from file**:

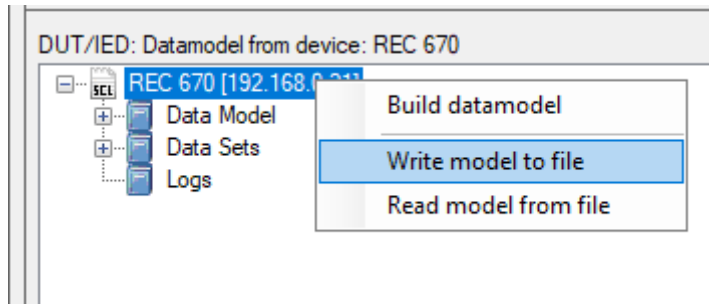


Figure 3-6: Write/Read Datamodel to/from File

3.6 Load the SCL file (*.icd/*.cid/*.iid...) that belongs to the DUT

Most of the validation by the *SCL Checker* is not done with the DUT but with the associated SCL file describing the DUT, its capabilities and configuration. This file must be loaded as well.

Right-click on the SCL model frame and select [**Build data model**] to start loading it.

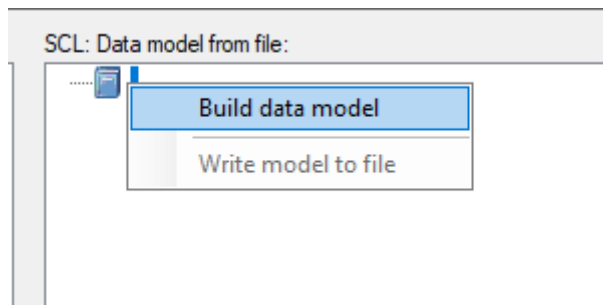


Figure 3-7: Build SCL Datamodel from SCL model frame

Or, use the [**Open configuration file**] icon from Toolbar:

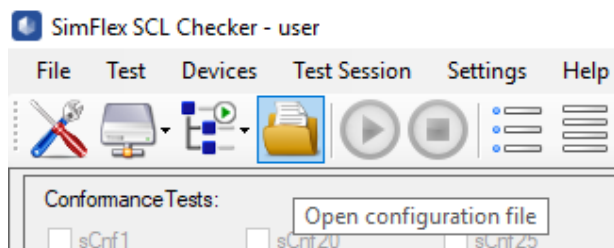


Figure 3-8: Build SCL Datamodel from Toolbar

A file selector window will open to enable you to select the correct SCL file (*.icd/*.cid).

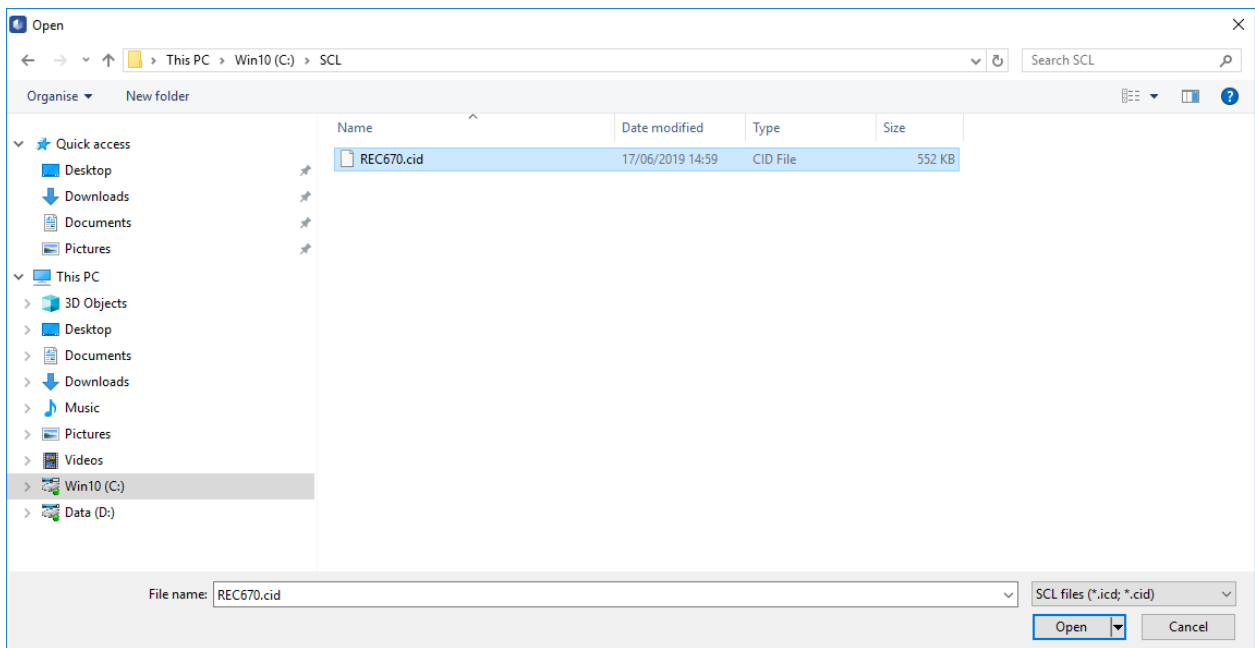


Figure 3-9 File selector window

Select the correct SCL file and click [**Open**].

When the file is loaded, a data model will be built from it and will be shown in the SCL model frame.

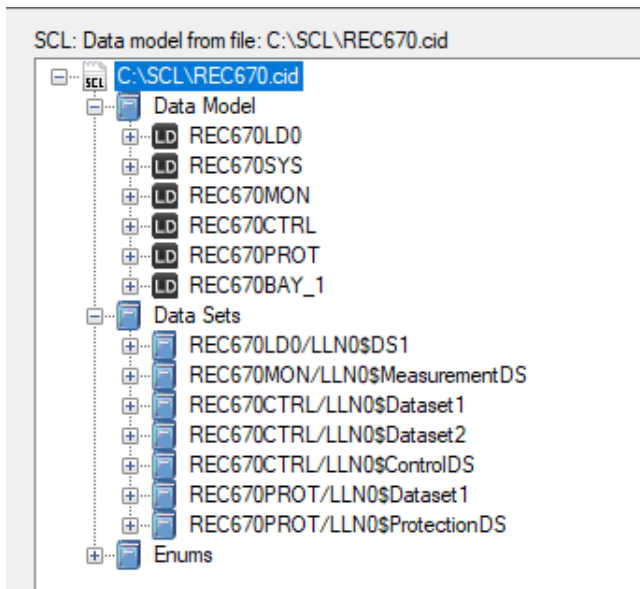


Figure 3-10 SCL data model

NOTE: In Open window, user can select all files filter (*.*) so other file formats such as *.scl, *.iid etc. could be loaded also. In this case additional window will appear for selecting IED from list of available IEDs.

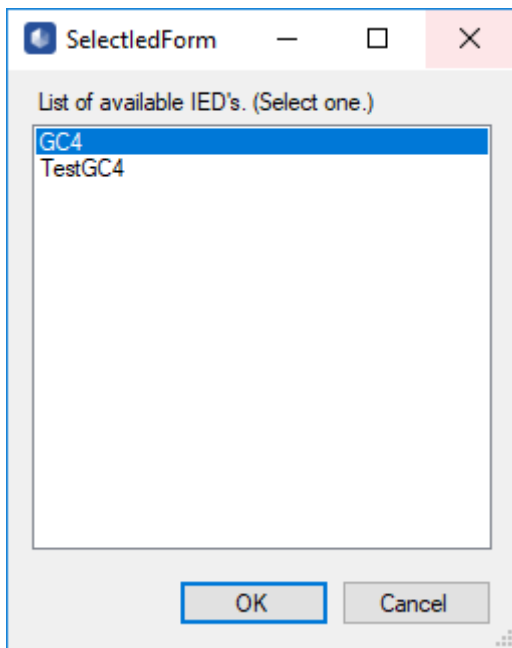


Figure 3-11 SCL data model – list of available IEDs in SCD file

4 Use case: Select and run test case

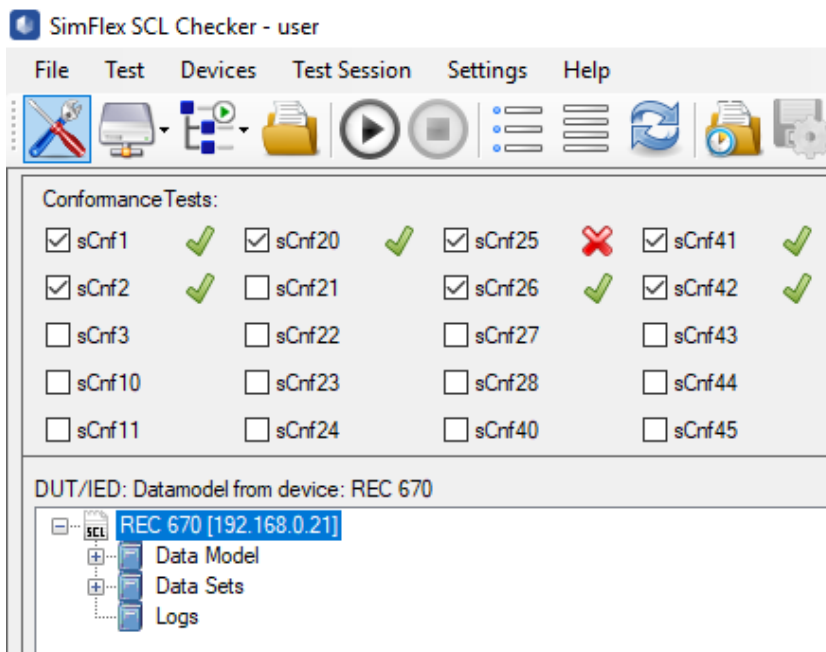


Figure 4-1: Selected test cases

When both DUT model and SCL model are loaded all test cases are available. By clicking on check box next to test case name, test will be selected for test run. In the example, tests sCnf1, sCnf2, sCnf20, sCnf25, sCnf26, sCnf41, sCnf42 are selected. Next, when [Run] button is pressed – test will be executed. The [Run] button is disabled during the test.



Figure 4-2: Start and stop test run

Test execution can be stopped in case that its duration is long enough for Tool operator to react (if necessary). Most of the test are finished in less than a second so stop operation will be not viable.

A lot of activities in the log frame at the bottom could be noticed.

After a normal run the [Run] button is enabled again and the test are marked as 'Passed' (green tag), 'Inconclusive' (blue question mark) or 'Failed' (red cross).

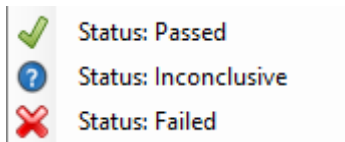


Figure 4-3: Possible test outcomes

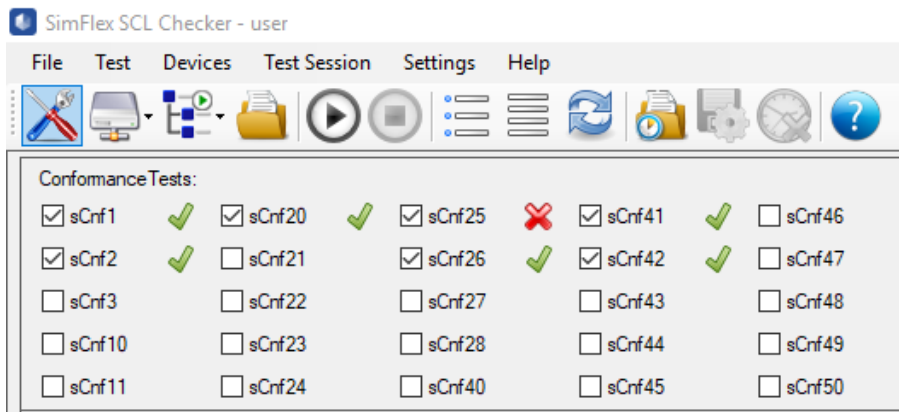


Figure 4-4: Test result (positive and negative)

Inspect the log for the exact reason(s) of failing.

4.1 Opening and examining a log file

A test that has already been executed has an associated log file. The log file will be opened automatically by double clicking on the test result icon (PASS, FAIL or INCONCLUSIVE).

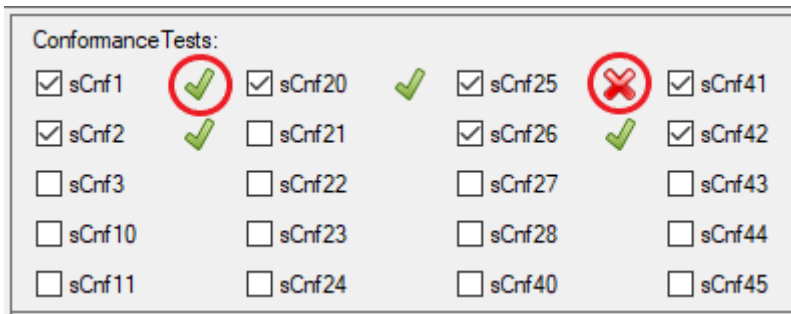


Figure 4-5: Open Log file by double-click on test result icon

Or, Right-click the test result icon to open a sub menu. In the sub-menu the user can open the test case log file and/or change/override the test case result.

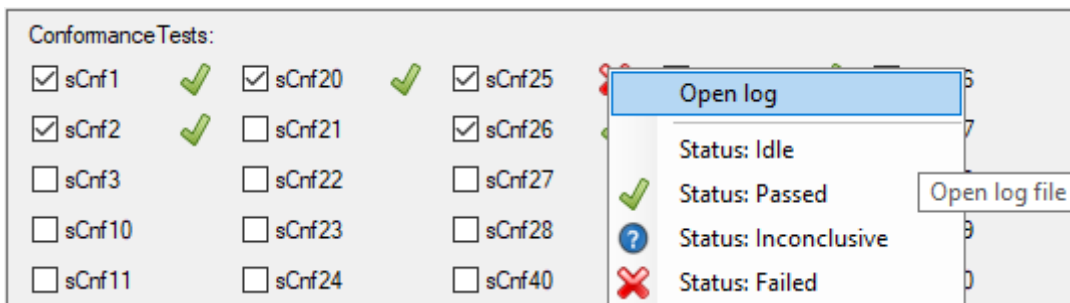


Figure 4-6: Open Log file by right-click on test result icon

The log file can also be opened with an external editor (e.g. Notepad). In this case browse to the location of the log file (e.g. with Windows Explorer) and open it with a text editor like Notepad.

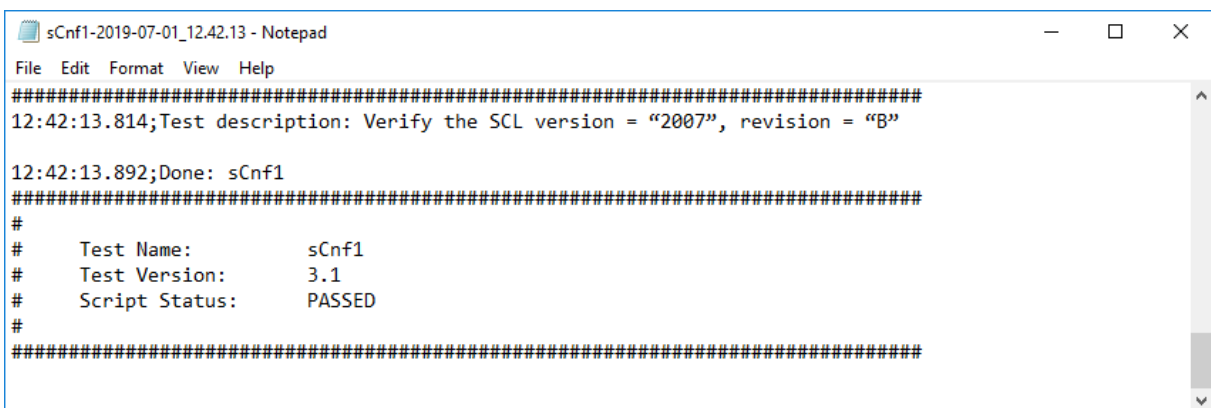


Figure 4-7: Details of result for sCnf1 test case – test PASSED



```
sCnf25-2019-07-01_12.42.13 - Notepad
File Edit Format View Help
#####
12:42:13.555;Test description: Verify each Subnetwork/ConnectedAP@iedName is "TEMPLATE"

12:42:13.634;ERROR: line ( 11, 20): Invalid iedName 'REC670'
12:42:13.638;Done: sCnf25
#####
#
#   Test Name:      sCnf25
#   Test Version:   3.1
#   Script Status:  FAILED
#
#####
```

Figure 4-8: Details of result for sCnf25 test case – test FAILED



5 Use case: Configuring a test

Some of the test, require either setting the conditionals prior to execution or User interaction and reconfiguration of test files or DUT during execution of tests. This relates only to some sMdl tests such as (sMdl2, sMdl3, sMdl16). Configuration for sMdl2 and sMdl3 relates to definition of supported presence conditions in **Settings>Conditionals** as mentioned in **2.6 Detailed information of Settings dialog, Conditionals tab**

5.1 Configuration for sMdl2 and sMdl3

sMdl2 and sMdl3 check numerous different conditions divided in two groups - conditions for attribute inclusion and data object conditionality in Logical node. These conditions are partly fixed and partly depending on the manufacturer. The manufacturer must supply this information through the PICS/PIXIT documents. This information must be included in the configuration's [**Conditionals**] tab.

Steps to be taken for configuring the PICS/PIXIT information:

1. Open the configuration form and select the [**Conditionals**] tab by selecting the configuration icon or selecting the menu **Configuration → Conditionals (PIXIT...)**.
2. For each line in the list, check the setting in the "Value" column.
The "Group" column is a reference to the IEC61850 document and chapter that is applicable.
The "Name" column is the name for the condition as used in the IEC61850 documents.
The "Value" column is the current value (and is the only column that can be edited).
The "Range" column sums up all acceptable values, if only one value is given then the "Value" is 'fixed' to this single value and cannot be changed.
The "Description" column is the description for the condition as given in the IEC61850.

The configuration data for sMdl2 and sMdl3 is stored in the "Session results directory".

5.2 Configuration for sMdl16

This test needs an additional SCL file with relevant changes in it. (See the UCA test description for more details).

The DUT must be configured during this test with the changed SCL file and the *SCL Checker* will ask the test engineer during test to either mark changes or to load the SCL file and to reconfigure the DUT.

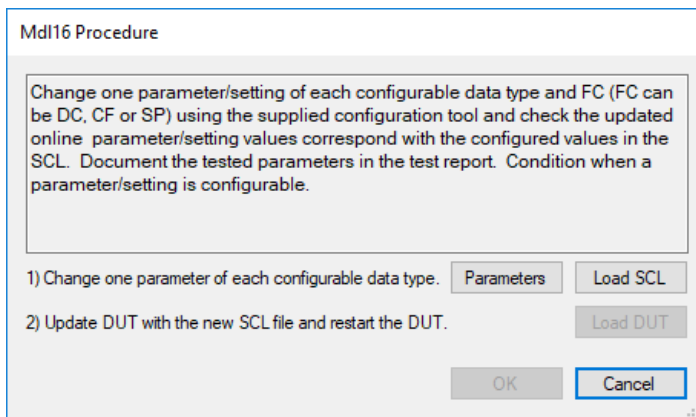


Figure 5-1: sMdl16 request for User to load SCL with changes and update DUT afterwards

To inspect which parameters are available for configuration, user can press **Parameters** button:

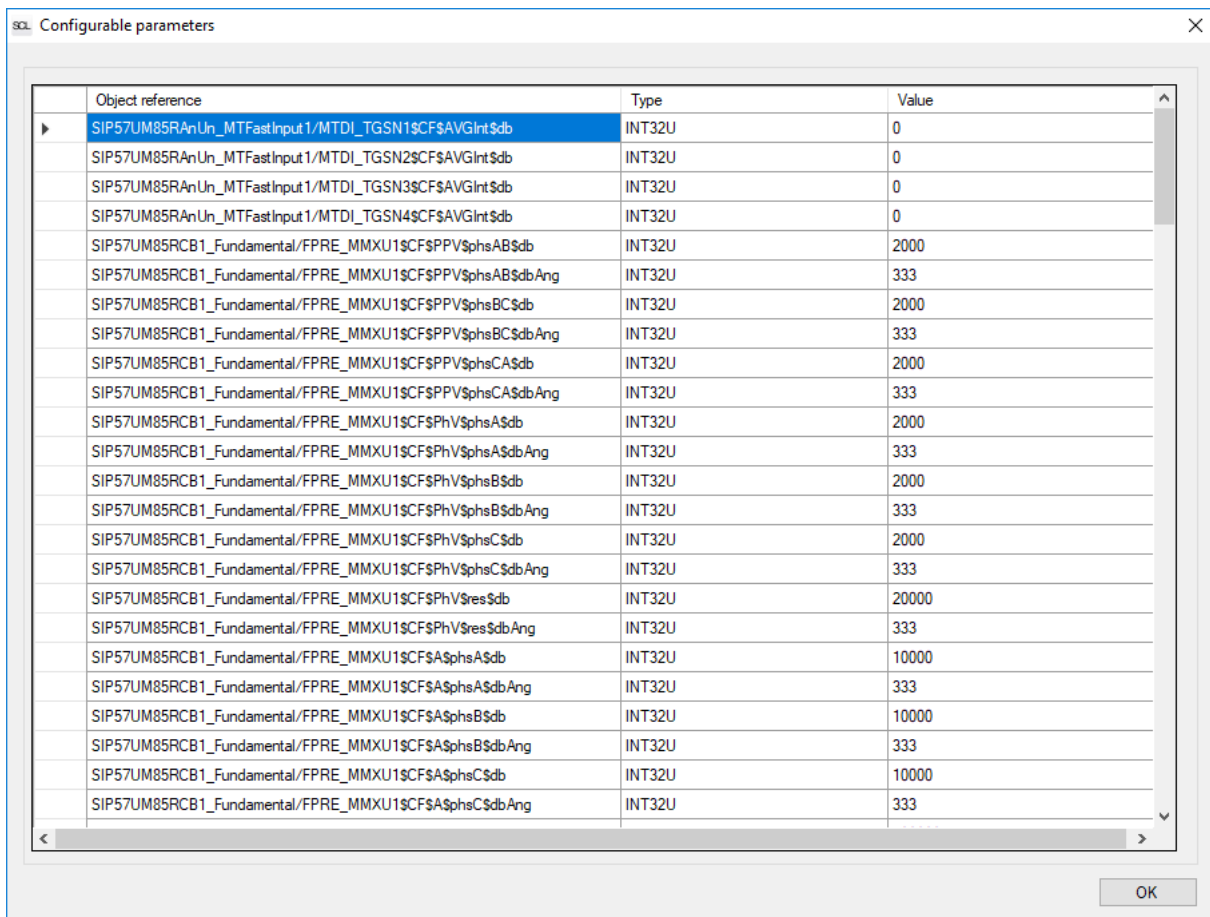


Figure 5-2: Configurable parameters

To load SCL with changes, user should press button **Load SCL**, and then button Load DUT will be activated for reloading DUT model (user should load SCL with changes to DUT first)

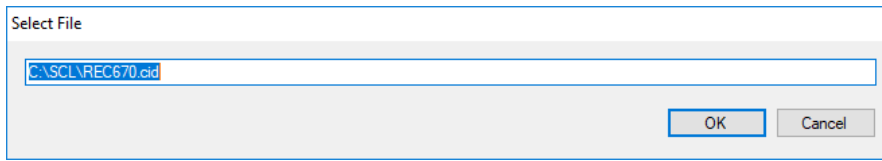


Figure 5-3: Load SCL file

NOTES:

Logging is divided in three parts:

- **SCL differences:** it is reporting Data Attributes with original values and than same Data Attributes with changed value (so for one change there will be two lines of report)
- **DUT differences:** it is reporting Data Attributes with original values and than same Data Attributes with changed value (so for one change there will be two lines of report)
- **Differences comparison:**
 ERROR: if there is no changes in SCL model and DUT model or
 ERROR: if changes from SCL model are not found in DUT model or
 WARNING if DUT model changes are not found in SCL model

Example 1 - DUT model is not following SCL model changes:

```
SCL (unchanged): C:\SCL\REC670.cid
.
SCL: differences
1: ObjectReference="REC670/QA1CSWI1$CF$Pos$ctlModel", TypeInfo="INT8", Value="4"
2: ObjectReference="REC670/QA1CSWI1$CF$Pos$sboTimeout", TypeInfo="INT32U", Value="+30000"
3: ObjectReference="REC670/QA1CSWI1$DC$NamPlt$vendor", TypeInfo="VisString255", Value=""ABB""
4: ObjectReference="REC670/QA1CSWI1$CF$Pos$ctlModel", TypeInfo="INT8", Value="0"
5: ObjectReference="REC670/QA1CSWI1$CF$Pos$sboTimeout", TypeInfo="INT32U", Value="+10000"
6: ObjectReference="REC670/QA1CSWI1$DC$NamPlt$vendor", TypeInfo="VisString255", Value=""Test Change""
.
DUT: differences
.
ERROR: Changes in DUT model: 0: not enough changes visible for this test (need at least 1 changes visible).
ERROR: Change in SCL model not found in DUT model: ObjectReference="REC670/QA1CSWI1$CF$Pos$ctlModel", TypeInfo="INT8", Value="4"
ERROR: Change in SCL model not found in DUT model: ObjectReference="REC670/QA1CSWI1$CF$Pos$sboTimeout", TypeInfo="INT32U", Value="+30000"
ERROR: Change in SCL model not found in DUT model: ObjectReference="REC670/QA1CSWI1$DC$NamPlt$vendor", TypeInfo="VisString255", Value=""ABB""
```

Example 2 - SCL model is not following DUT model changes:

```
SCL: differences
.
DUT: differences
1: ObjectReference="REC670/QA1CSWI1$CF$Pos$ctlModel", TypeInfo="INT8", Value="0"
2: ObjectReference="REC670/QA1CSWI1$CF$Pos$sboTimeout", TypeInfo="INT32U", Value="+10000"
3: ObjectReference="REC670/QA1CSWI1$DC$NamPlt$vendor", TypeInfo="VisString255", Value=""Test Change""
4: ObjectReference="REC670/QA1CSWI1$CF$Pos$ctlModel", TypeInfo="INT8", Value="4"
5: ObjectReference="REC670/QA1CSWI1$CF$Pos$sboTimeout", TypeInfo="INT32U", Value="+30000"
6: ObjectReference="REC670/QA1CSWI1$DC$NamPlt$vendor", TypeInfo="VisString255", Value=""ABB""
.
```



```
ERROR: Changes in SCL model: 0: not enough for this test (need at least 1 changes).  
WARNING: Change in DUT model not found in SCL model:  
ObjectReference="REC670/QA1CSWI1$CF$Pos$ctlModel", TypeInfo="INT8", Value="0"  
WARNING: Change in DUT model not found in SCL model:  
ObjectReference="REC670/QA1CSWI1$CF$Pos$sboTimeout", TypeInfo="INT32U", Value="+10000"  
WARNING: Change in DUT model not found in SCL model:  
ObjectReference="REC670/QA1CSWI1$DC$NamPlt$vendor", TypeInfo="VisString255", Value=""Test Change""
```

Please refer to the manual of the DUT how to update its configuration.



6 Use Case: Running a Test Session

This section explains what a test session is, how to load a test session into *SCL Checker*, how to run one or more tests in a session and how to evaluate the results of the test runs.

First of all, what is a test session?

A test session is a compilation of a test suite and session information like DUT, test lab, vendor and settings grouped together in one file. The extension of test session files is “*.tsf”. Internally it is XML file but it is not intended to be edited by the end user, instead the SimFlex Client Simulator should be used for any editing of a test session.

NOTE: SCL Checker will only open and save test sessions. The user will have to use the SimFlex Client Simulator to create and modify test sessions. Refer to the SimFlex Client Simulator User Manual in order to do so.

6.1 Opening a Test Session

In order to open a test session that was previously saved, the user has several options:

1. Click in the menu bar under: **Test Session** → **Open Session**, then select a test session in the *Open Test Session* dialog window.

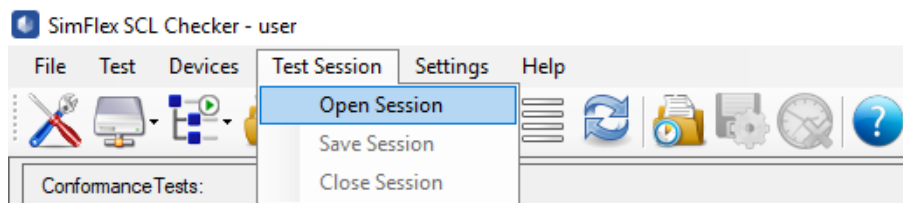


Figure 6-1: Open Session menu

2. Click in the tool bar button: **Open Test Session** and then select a test session in the *Open Test Session* dialog window.



Figure 6-2: Open Session button

An Open Test Session dialog window will be opened:

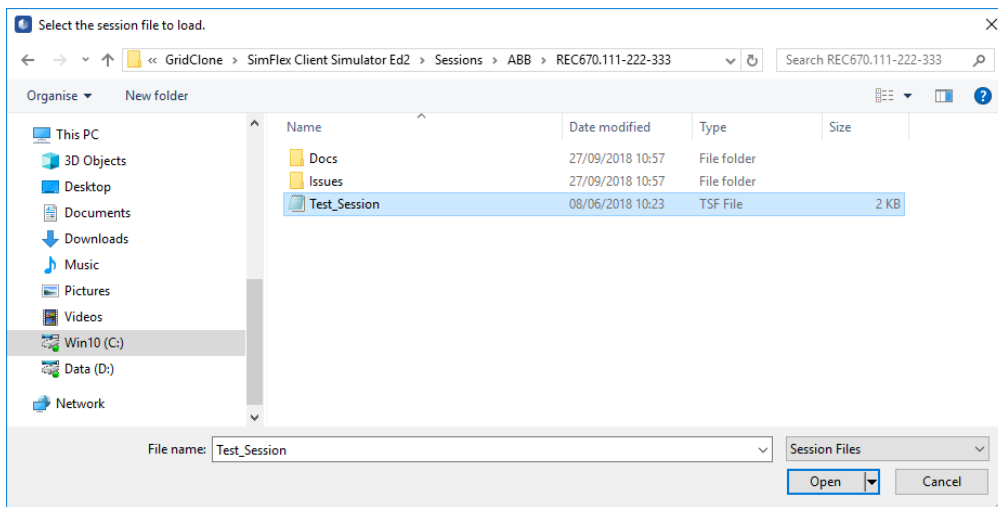


Figure 6-3: Open Session dialog

6.2 Running tests in a Test Session

While in session mode tests can be run as specified in chapter 4 Use case: **Select and run test case**.

6.3 Overriding test cases results

In some cases, the test engineer might want to override the test case result after manually examining the test results (log file, SCL file, DUT data model). To do so, test engineer can *right-click* the test result icon to open a sub menu. In the sub-menu the user can open the test case log file and/or change/override the test case result.

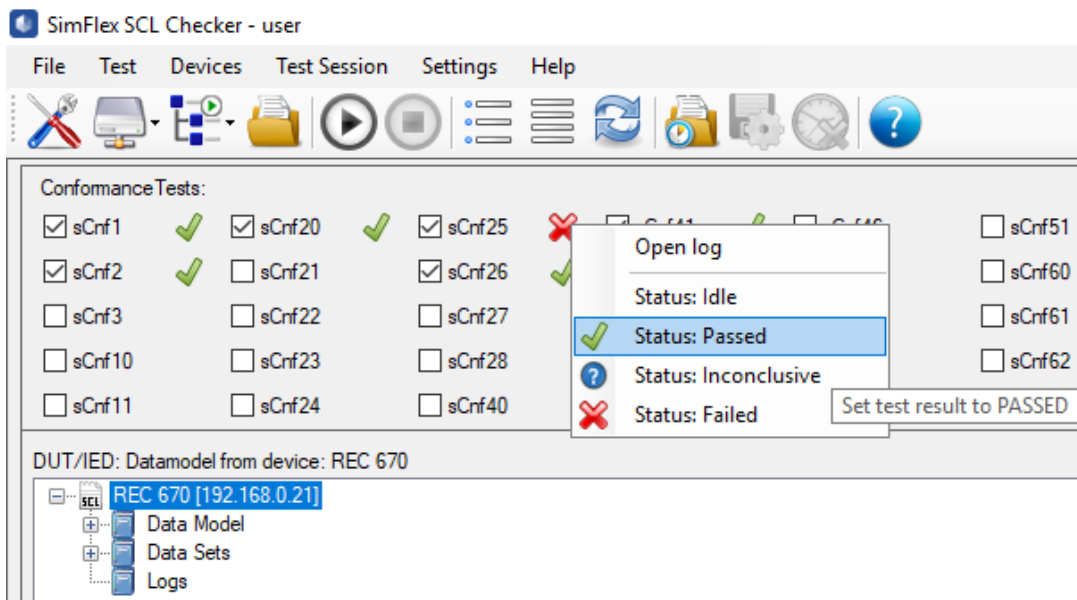


Figure 6-4: Overriding test results

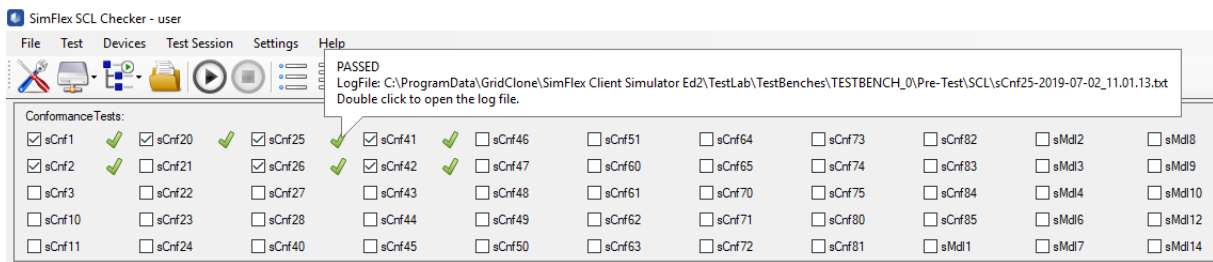


Figure 6-5: sCnf25 test overridden

When changing the test case result, the test session file will also be updated. This means that next time the test engineer opens the test session file, the result is preserved.

NOTE: The test log file also shows the result of the test case. As the log file is a simple text file, the test result is not changed after changing the test result in the SCL Checker.

6.4 Saving and Closing a Test Session

After modifying a session, it can be manually saved by three methods:

- 1) Click in the menu bar under: **Test Session** → **Save Session**.

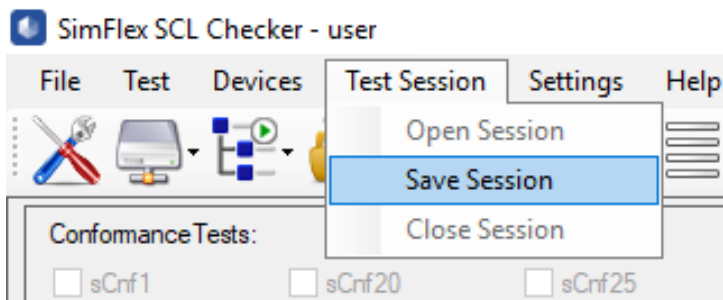


Figure 6-6: Save Session menu

- 2) Click in the tool bar button: **Save Test Session**.

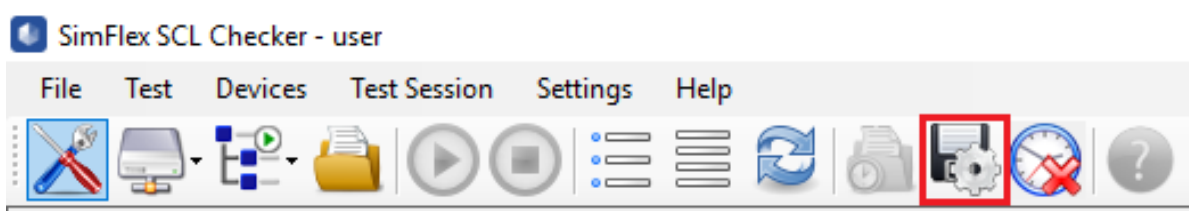


Figure 6-7: Save Session button

Additionally, test sessions will be automatically saved in the following cases: after a test is run, before the session is closed and before the application is closed.

The user can close a test session manually by:

- 1) Clicking in the menu bar under: **Test Session** → **Close Session**.

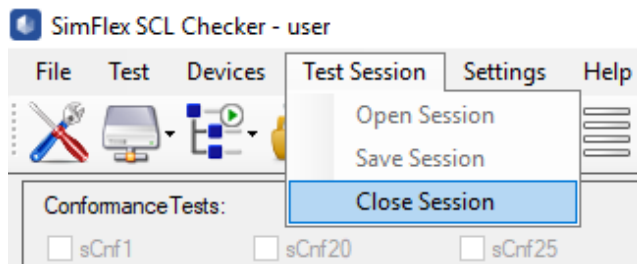


Figure 6-8: Close Session menu

2) Clicking in the tool bar button: **Close Test Session**.

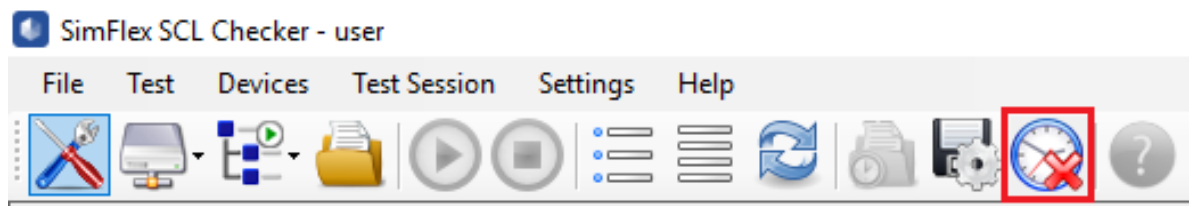


Figure 6-9: Close Session button



7 Use Case: Request Support

If you're having trouble using SimFlex SCL Checker, you can send email to GridClone support team. In order to understand problem and how it occurred, GridClone support team needs to know the logs and configurations of your SCL Checker application.

7.1 Using Support Feature

You can use Support feature from the application to send email for requesting support. This feature can be run by click the Support button in the icon bar. This button is available after DUT model or SCL model are loaded. While the test is running, this button will be disabled.

Follow this step to use support feature.

1. Click the Support button from the icon bar



Figure 7-1 Support Icon

2. User will be asked to confirm the action:

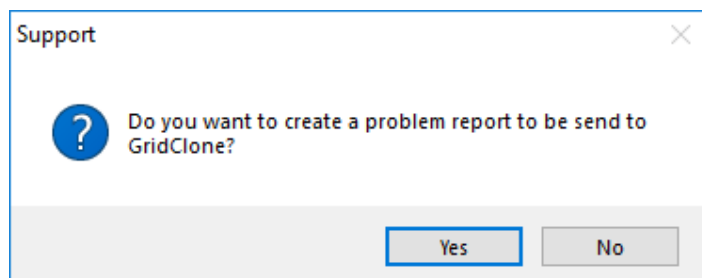


Figure 7-2 Support request confirmation

3. Next message box will give you information about the location of the log files have been saved.

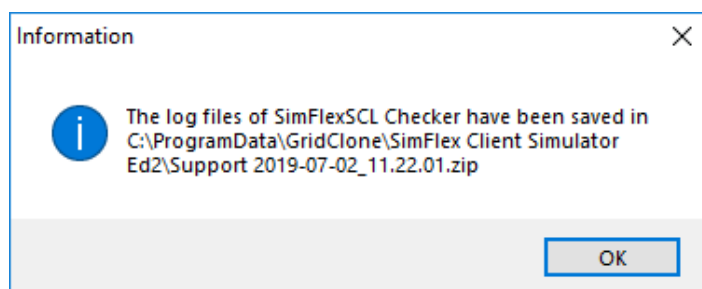


Figure 7-3 Message Box that shows the location of the Log files



- 4. A new dialog window will appear. Click Yes button if you want to send the email to GridClone support team.

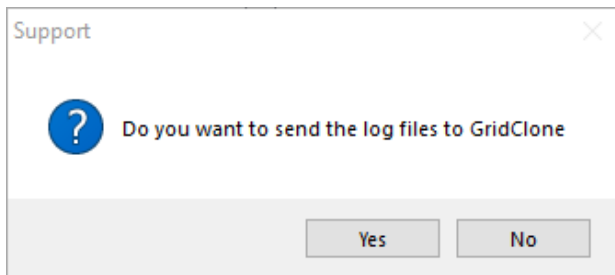


Figure 7-4 Confirmation of sending Email to Support team

- 5. Default email client will open. Finish your email and send it to GridClone support.

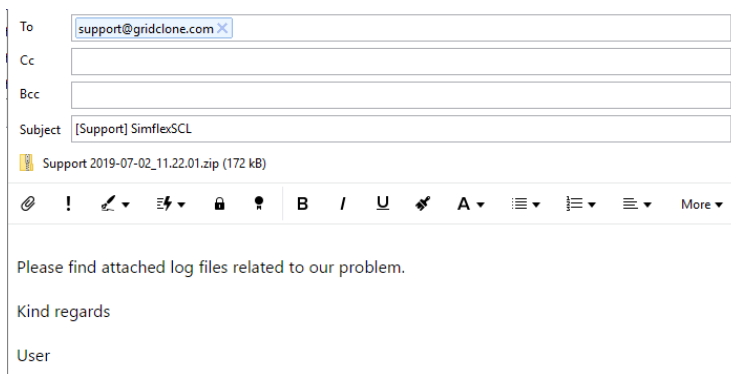


Figure 7-5 Composing Email to GridClone Support

This feature works if you have default mail application on your computer. However, if the email pop-up doesn't appear, you still can manually send email to GridClone including the log files.

The log files are saved in a zip file. The location of this zip file is written in the previous message box and you can also find it in log frame.

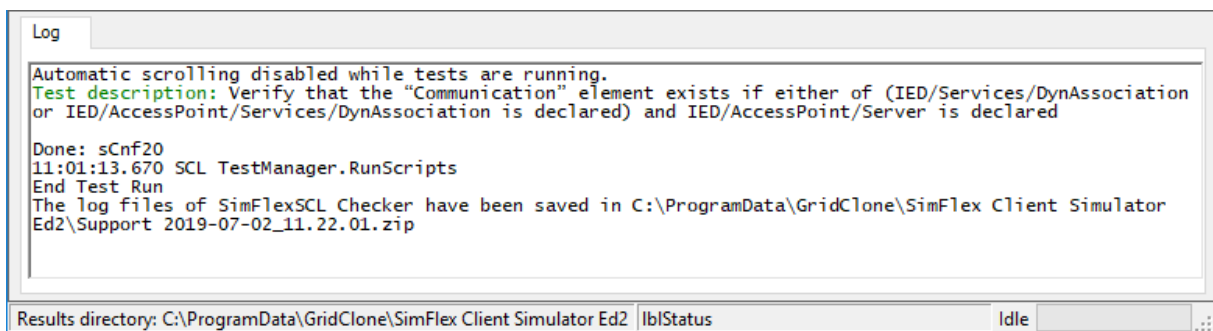


Figure 7-6 Log message with information of Support Log location



8 Advanced Topics

This section includes more information on test cases and data models.

8.1 How to use the data model

8.1.1 Tree hierarchy

Tree hierarchy is containing following nodes:

LD = Logical Device

LN = Logical Node

FC = Functional Constraint

DO = Data Object

DA = Data Attributes

8.1.2 How to read the values in the DUT data model (MMSData values)

All values from and to a DUT are communicated using the MMS protocol (ISO 9506), as described in IEC 61850-6. Values shown in the DUT data model are restricted to the format described within the MMS protocol.

The values displayed in the DUT data model do not only contain the MMSData type but also includes some length information, a flag indicating if the value is 'Readable', 'Writable' or both and the actual value retrieved from the DUT.

The normal format for an item with a value in the data model is:

Name, [MMSData type], Readable/Writable flag, value.

Example:

```
DUT_NAME [DUT IP address]

Data Model
  LD name
    FC ST
      DO Mod
        DA stVal, [INTEGER~8], R., value = 1
        DA q, [BITSTRING~-13], R., value = [0000000000000]
        DA t, [UTC_TIME], R., value = u24-10-2019_16:06:52.130
```



8.1.3 MMSData type and length information:

The length information with the MMSData type is according to IEC 61850-8-1, paragraph 8.1, table 11, mapping of ACSI data types.

Most MMSData types with a fixed length (such as UTC_TIME) have no length information, with the exception of BOOLEAN. INTEGER and UNSIGNED which have length information of one digit (8 for 8 bits, 16 for 16 bits etc.). String based types (such as VISIBLE_STRING) have length information that can be negative indicating a Maximum length (the string may be shorter than the length but only if the length information is negative.). And FLOATINGPOINT values have two length information values, one to indicate the total length (in bits) and the second to indicate the length of the exponent (in bits).

Readable/Writable:

At maximum two characters indicate the data attribute readable/writable information. The possible characters are:

<u>Character</u>	<u>Meaning</u>
-	The Readable/Writable state is unknown.
.	The Readable/Writable state is untested.
R	The data attribute is Readable
W	The data attribute is Writable
w	The data attribute should be writable but is not available at the moment for writing.

The most common combinations are:

<u>Combination</u>	<u>Meaning</u>
-	The data attribute Readable/Writable state is unknown.
.	The data attribute Readable/Writable state is untested.
R.	The data attribute is Readable but not tested for Writable.
R-	The data attribute is Readable but not Writable.
Rw	The data attribute is Readable and should be Writable but was not available for writing at the moment of testing.
RW	The data attribute is Readable and Writable.



9 Frequently Asked Questions (FAQ)

Some questions do reoccur and become frequently asked (that is, more than once). Check this section before you contact GridClone for support on your question.

9.1 General

9.1.1 How to install the SimFlex™ SCL Checker?

Read the *Getting started* document that comes with the program for a detailed description of the installation procedure. If you do not have **administrator rights** on your PC, please contact your IT department.

9.1.2 How can I check if my device is available on the network?

See that your DUT is turned on and connected to the same LAN as your *SCL Checker*. Find out the IP address of the DUT.

A simple test to check if the DUT is connected and within range of the *SCL Checker* is to open a Windows Command Window (cmd.exe) and typing the following instruction at the caret:

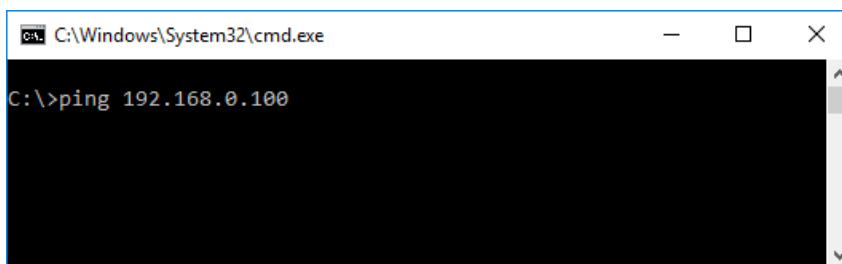


Figure 9-1: Ping command

The IP address behind the *Ping* command shall be the IP address of the DUT. In case the DUT is visible it will react with positive responds.

Next, be sure that you selected the appropriate **network interface** in the [**General**] tab in the Settings Dialog of the *SCL Checker*.

After that, be sure that you created a DUT in the [**Devices**] tab in the Settings Dialog of the *SCL Checker*. Check if the IP address is the same as the IP address of your DUT.

Try to retrieve the data model of the DUT.

9.1.3 The SCL Checker does not start or complains about the license

In the system tray check if the CodeMeter service is running (clicking on the small system-tray arrow).

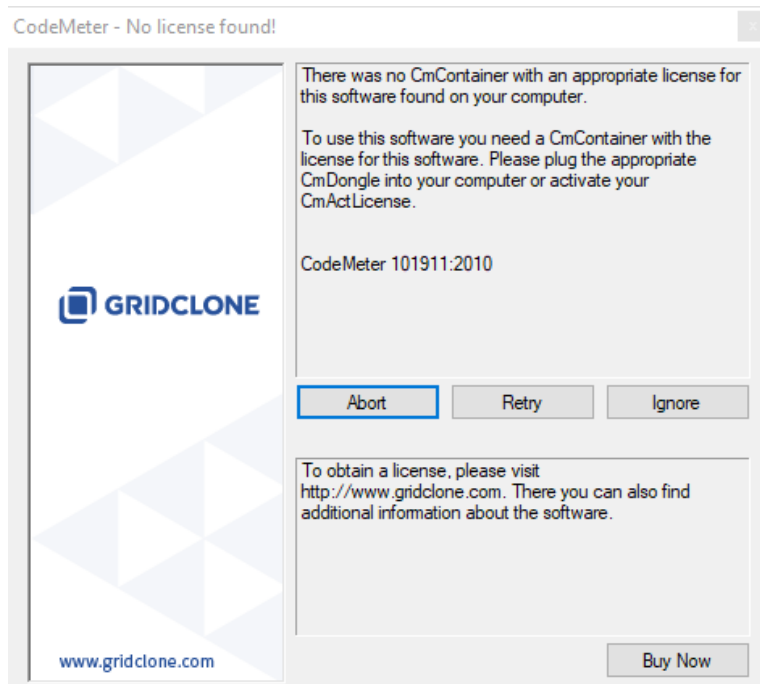


Figure 9-2: No license found

If the CodeMeter is not visible, it could be that it is not running, or that the service **is** running, but the CodeMeter program is not visible.

- a) In the first case: re-install the SimFlex™ IEC 61850 SCL Checker and make sure the WIBU software installs correctly.
- b) In the second case: press Windows Start button and locate the CodeMeter installation. Then start the “CodeMeter Control Center” manually.

Once the CodeMeter Control Center is started, open it (from the system tray) and check that there is a license shown in the left part of the dialog. Make sure you have inserted the USB license key in one of the available USB ports!

In case you are running a trial version of the SimFlex™ IEC 61850 SCL Checker:

If there is no license shown, select **File→Import License** from the menu and locate the “*.lic” file in the *SCL Checker* folder (In the folder where you installed the program, e.g. C:\Program Files\GridClone\...). Note that the CodeMeter Control Center might have a filter like “*.WibuCmRaU”. Change it to “*.*” or “*.lic” to make the license file visible.

Once the license file has been selected and opened it should be visible in the CodeMeter Control Center.

To be sure the license is activated: stop and start the CodeMeter Control Center service by pressing **Process→Stop CodeMeter Service** and then **Process→Start CodeMeter Service**.

Finally re-start the *SCL Checker*. If the problem persists please contact GridClone for support.



10 List of possible error messages

The *SCL Checker* has different kinds of messages incorporated in logging engine:

- 1) Test Case Error messages - strictly bind to the exact test case
- 2) General error messages – general purpose error notifications
- 3) BUG messages. (Please report these messages to GridClone support team).
- 4) Warnings – user notifications that test script identified possible deviations from expected behavior etc.
- 5) Information –notification on important behavior or characteristic that could help user in diagnostics

SimFlex SCL Checker has two levels of verbosity to report messages, “Normal mode” and “Verbose mode”. You can switch between the two by (de)selecting the menu **Test → Verbose** option.

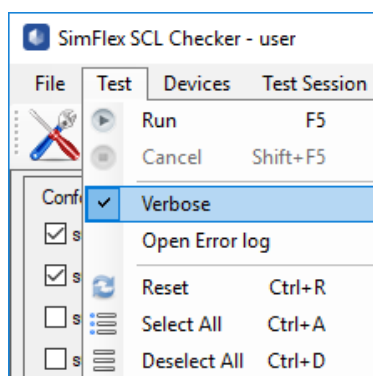


Figure 10-1: Selection Verbose or normal mode of reporting

In “Normal mode” (**Test → Verbose** unselected), only messages that will cause a test to fail or to be inconclusive is reported.



```
Log
Automatic scrolling disabled while tests are running.
Test description: Verify for each DAType/BDA or D0Type/DA with attribute "bType"=Struct has attribute "type"
whose value matches DAType@id; does not declare valKind (TISSUE #823);does not contain a <Val> element
ERROR: line (7146,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (7147,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (7148,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (7149,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (7150,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (7151,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6227,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6244,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6261,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6278,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6295,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6312,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6329,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6351,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6368,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6389,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6405,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6421,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6442,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
ERROR: line (6458,8): D0Type/DA with attribute "bType"=Struct declare valKind 'RO'
Done: sCnf70
11:59:53.764 SCL TestManager.RunScripts
End Test Run

Results directory: C:\ProgramData\GridClone\SimFlex Client Simulator Ed2 |lblStatus Idle
```

Figure 10-2: Normal mode of reporting

In “Verbose mode” (Test → Verbose selected) any decision made in a test is reported. This enables a test engineer to precisely follow all the actions by the test and to verify not only why a test is failing but also that a test is passing for the right reasons.



```

Log
Checking DAType/BDA, line (7000,8): DAType id = 'SB0wOperBOOLEAN', BDA name='origin'
Checking DAType/BDA, line (7008,8): DAType id = 'CancelBOOLEAN', BDA name='origin'
Checking DAType/BDA, line (7026,8): DAType id = 'RangeConfig', BDA name='hhLim'
Checking DAType/BDA, line (7027,8): DAType id = 'RangeConfig', BDA name='hLim'
Checking DAType/BDA, line (7028,8): DAType id = 'RangeConfig', BDA name='lLim'
Checking DAType/BDA, line (7029,8): DAType id = 'RangeConfig', BDA name='lLim'
Checking DAType/BDA, line (7030,8): DAType id = 'RangeConfig', BDA name='min'
Checking DAType/BDA, line (7031,8): DAType id = 'RangeConfig', BDA name='max'
Checking DAType/BDA, line (7035,8): DAType id = 'Vector', BDA name='mag'
Checking DAType/BDA, line (7036,8): DAType id = 'Vector', BDA name='ang'
Checking DAType/BDA, line (7045,8): DAType id = 'OperINT32', BDA name='origin'
Checking DAType/BDA, line (7053,8): DAType id = 'CancelINT32', BDA name='origin'
Checking DAType/BDA, line (7064,8): DAType id = 'OperTcmd', BDA name='origin'
Checking DAType/BDA, line (7072,8): DAType id = 'CancelTcmd', BDA name='origin'
Checking DAType/BDA, line (7079,8): DAType id = 'OperINT8', BDA name='origin'
Checking DAType/BDA, line (7087,8): DAType id = 'CancelINT8', BDA name='origin'
Checking DAType/BDA, line (7095,8): DAType id = 'CancelINT33', BDA name='origin'
Checking DAType/BDA, line (7102,8): DAType id = 'ABBIED600_Oper_ENC_c153659ef063668161e744abf3a32eeb', BDA
name='origin'
Checking DAType/BDA, line (7111,8): DAType id = 'ABBIED600_Oper_SPC_DPC_e0492d43942d06a49689daef6a52cfb4', BDA
name='origin'
Checking DAType/BDA, line (7120,8): DAType id = 'ABBIED600_Oper_INC_d6fbd7a30aeb616d03869bdd84db4310', BDA
name='origin'
Checking DAType/BDA, line (7135,8): DAType id = 'ABBIED600_Oper_ENC_ec2ecc9f5b6eebd11460553fe18e893d', BDA
name='origin'
Checking DAType/BDA, line (7146,8): DAType id = 'ABBIED600_RangeConfig_4bc7c85248954acee2d42c37b1d57247', BDA
name='hhLim'
ERROR: line (7146,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
Checking DAType/BDA, line (7147,8): DAType id = 'ABBIED600_RangeConfig_4bc7c85248954acee2d42c37b1d57247', BDA
name='hLim'
ERROR: line (7147,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
Checking DAType/BDA, line (7148,8): DAType id = 'ABBIED600_RangeConfig_4bc7c85248954acee2d42c37b1d57247', BDA
name='lLim'
ERROR: line (7148,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
Checking DAType/BDA, line (7149,8): DAType id = 'ABBIED600_RangeConfig_4bc7c85248954acee2d42c37b1d57247', BDA
name='lLim'
ERROR: line (7149,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
Checking DAType/BDA, line (7150,8): DAType id = 'ABBIED600_RangeConfig_4bc7c85248954acee2d42c37b1d57247', BDA
name='min'
ERROR: line (7150,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
Checking DAType/BDA, line (7151,8): DAType id = 'ABBIED600_RangeConfig_4bc7c85248954acee2d42c37b1d57247', BDA
name='max'
ERROR: line (7151,8): DAType/BDA with attribute "bType"=Struct declare valKind 'RO'
Checking DAType/BDA, line (7154,8): DAType id = 'ABBIED600_Vector_685b17234872474063899753003f1ab4', BDA
name='man'
Results directory: C:\ProgramData\GridClone\SimFlex Client Simulator Ed2 |lblStatus Idle

```

Figure 10-3: Verbose mode of reporting

WARNING: switching on to the **Test** → **Verbose** will generate huge amounts of logging data and can extend the time to finalize a test.

To find more information about an error message please look up the messages for a particular test and if no information is found, try the general section. These sections give details on the “Normal mode” messages. The extra “Verbose mode” messages follow a similar pattern and should be clear from the context and content alone. If not, please get in contact with GridClone.

Bug messages are clearly identified with the first (or last) word “BUG: ...”. In normal circumstances this kind of message will not show up but in case the program or parts of its data gets corrupted and the internal reflections and checking done by the program comes up with an unexpected, unexplainable or impossible result this message is given. Any details on how the message was triggered, what SCL files and DUT are used will be helpful to solve the problem.

10.1 Conventions used

Any message that will cause a test to fail has the word “ERROR:” at the beginning of the line.

E.g: “ERROR: line (1846, 11): DA MyLogicalDevice/LLN0\$C0\$Mod\$Oper: Missing ‘Check’”

Description of each message attribute is given below:



<Line number>	<p>A position in a SCL file given as (line, column) combination.</p> <p>E.g. “line (1846, 11):”</p> <p><Line number> value is normally followed by an <Object Reference> value. A <line number> is given when available and left out (including the colon) when not available. It refers to the line and column in the SCL file where a data object or value is specified. When data objects between the DUT model and SCL file are matched then any message based on data objects from the DUT model may include <line number> information on the matched data object in the SCL file. (Matching data objects between DUT and SCL (*icd) file is done on Data Object Reference level only.)</p>
<bType>	<p>Part of Basic IEC61850 Type information. <bType> is one of Int8, Int16, Enum, etc.</p> <p>NOTE: Type information in IEC61850 is defined as a (bType, type) tuple where the type part can be empty.</p>
<Object class>	<p>Data object classification, e.g. “LD”, “LN”, “FC”, “DO” etc.</p>
<Object name>	<p>Data object name, e.g. “LLN0”, “Mod”, “Oper”, “Check” etc.</p>
<Object Reference>	<p>A value of an object reference:</p> <p>E.g. “MyLogicalDevice/LLN0\$C0\$Mod\$Oper\$Check”</p> <p>An <Object Reference> is usually preceded with a <Line number> value.</p>
<Result>	<p>Any test result, one out of “PASSED”, “INCONCLUSIVE” or “FAILED”</p>
<Test>	<p>Test name, e.g. “sCnf1”, “sCnf2”, ... “sMdl1”, “sMdl2”,...</p>
<type>	<p>Part of Basic IEC61850 Type information. ‘type’ is a further specification on the bType part, most commonly given with bType is Struct, Array or Enum</p> <p>NOTE: Type information in IEC61850 is defined as a (bType, type) tuple where the type part can be empty.</p>

10.2 Messages in all sMdl tests

Message:	<i>“<Object class>: <Object Reference>, please check manually”</i>
Example:	<i>“DO: MyLogicalDevice/LLN0\$EX\$MyDO, please check manually”</i>
Meaning:	Information about the object reference (of given object class) is not found in the IEC61850 reference data. The object reference might be an extension or enhancement.



Correction: Check that the object reference is an extension or enhancement (see PICS/PIXIT information). If not, check that the object reference is not part of the IEC61850. Depending on the test, the test engineer has to validate this object reference and adjust the outcome of the test accordingly.

10.3 Messages available in all tests (sCnfx and sMdlx)

Message: ERROR: <Line number>: <Object class> <Object reference>: Missing <Object name>.”

Example: “ERROR: line (1846, 11): DA MyLogicalDevice/LLN0\$CO\$Mod\$Oper: Missing ‘Check’..”

Meaning: The data object ‘Check’ is missing in data object “MyLogicalDevice/LLN0\$CO\$Mod\$Oper” and as such, the test is FAILING.

Message: “ERROR: parent <Line number>: <Object class> <Object reference>: child <line number>: <Object name>: Not allowed.”

Example: “ERROR: line (1846, 11): DA MyLogicalDevice/LLN0\$CO\$Mod\$Oper: child (1890, 11): ‘Check’: Not allowed.”

Meaning: The data object ‘Check’ is included in data object “MyLogicalDevice/LLN0\$CO\$Mod\$Oper” while not allowed and as such, the test is FAILING

Message: “Test description: <some text>”

Example: “Test description: Verify that the indicated trigger option: <DA dchg, qchg, dupd > ...”

Meaning: This is what test sMdl18 is specified to do (according to the UCA testing procedure).

Message: “Done: <test>”.

Example: “sMdl5: Done: sMdl5”

Meaning: Test sMdl5 has finished; this is the last log message from within the test, still preceded with the <test> name.

Message: SCL file 'C C:\FileName.format ' cannot be read: Data at the root level is invalid. Line 1, position 1.:
Finished Reading SCL file.”

Example: “Reading SCL file: C:\FileName.format



*SCL file 'C C:\FileName.format ' cannot be read: Data at the root level is invalid.
Line 1, position 1.:*

Finished Reading SCL file."

Meaning: You did not load SCL file and information from it is needed for the test you selected to run.

Correction: Load the SCL file belonging to the DUT that you are testing to correct this.

Message: "No valid DUT data from model available (test inconclusive)"

Example: "No valid DUT data from model available (test inconclusive)"

Meaning: You did not build a data model for the DUT but information from it is needed for the test you selected to run.

Correction: Connect to a DUT and build a data model.

Message: "ERROR: <Some text>: Unforeseen problem."

Example: "ERROR: divide by zero error: Unforeseen problem."

Meaning: Due to some corrupt data, programming error or other reason not covered in testing the program has met a failing point from which it cannot recover. Due to that the test cannot continue and must fail. This message might imply a bug in the program but could as well be a bug or non-compliance condition in the DUT. Please report this message to GridClone. Any details on how the message was triggered, what SCL files and DUT are used is immensely helpful by solving the problem.



11 Additional Topics

In this section you will find information on additional topics.

11.1 File name conventions

Any *.txt file is a test result file and has the same content as the log frame.

Any *.log file is an error log file for errors related to the program and not to the test.

sMdl1-2019-01-07_09.28.43-CS.txt is the test result of test case *sMdl1* generated by the *SCL Checker* on the 7th of January 2019 at 09:28 local time.

2019-01-07_09.28.43-SimFlexError.log is an error log for the same run. This file is located in the “Application base directory”.

11.2 Used abbreviations

DUT Device Under Test.

IEC The International Electrotechnical Commission, managing international standards. See: <http://www.iec.ch/>

IEC 61850 The IEC 61850:2007 standard, version A and B

IED Intelligent Electronic Device

PICS Protocol Implementation Conformance Statement

PIXIT Protocol Implementation Extra Information for Testing

UCA The UCA® International Users Group. See: <http://www.ucaiug.org>

11.3 Referenced documents

IEC 61850 IEC 61850:2007 version A and B

MMS ISO 9506:2003 (as used by IEC 61850-8)

Getting started.pdf

11.4 Referenced tools

WinPcap For capturing network traffic, see: <http://www.winpcap.org/>

WireShark For analysing network traffic, see: <http://www.wireshark.org/>



12 Document Version History

The table below describes the version history of this document.

Version	Date	By	Subject
1.0	January 1, 2013	EM	Initial version
1.1	January 22, 2013	EM	Extending on several subjects.
1.2	January 26, 2013	EM	General rework on entire document
1.3	June 20, 2013	EM	General rework for Edition 2.
1.4	June 28, 2019	CV	General update