7.2 Common data types ......................................................................................................... 18
7.2.1 Introduction ............................................................................................................. 18
7.2.2 Simple data types ................................................................................................... 18
7.2.3 Structured data types ............................................................................................ 19
7.3 Data model for TestDescription .................................................................................... 20
7.3.1 Introduction ........................................................................................................... 20
7.3.2 TestDescription element ......................................................................................... 20
7.4 Data model for TestInput ............................................................................................. 23
7.4.1 Introduction ........................................................................................................... 23
7.4.2 TestInput element .................................................................................................. 23
7.5 Data model for ScriptExecution .................................................................................... 23
7.5.1 Introduction ........................................................................................................... 23
7.5.2 ScriptExecution element .......................................................................................... 23
7.6 Data model for RequiredTestOutput ............................................................................. 24
7.6.1 Introduction ........................................................................................................... 24
7.6.2 RequiredTestOutput element .................................................................................. 24

Annex A (informative): Examples using test description template ........................................ 26
A.1 Example#1: VNF Traffic Forwarding Test Description ..................................................... 26

Annex B (informative): Bibliography .................................................................................... 28

Annex C (informative): Change history ................................................................................. 29

History ...................................................................................................................................... 30
Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for ETSI members and non-members, and can be found in ETSI SR 000 314: "Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, PLUGTESTS™, UMTS™ and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. 3GPP™ and LTE™ are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. oneM2M™ logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. GSM® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "shall", "shall not", "should", "should not", "may", "need not", "will", "will not", "can" and "cannot" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"must" and "must not" are NOT allowed in ETSI deliverables except when used in direct citation.
1 Scope

The present document is based on the requirement of establishing a test case description convention between operators and providers in DevOps automated testing. It proposes a test case description template, to be used for standardizing the input and output information exchanged for test execution and result analysis.

The standardized test case description template in the present document aims at determining a standardized machine-readable format for (but not restricted to):

1) Test case description information
2) Test case input information
3) Test case script designation information
4) Test case output information

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.


[2] ETSI GS NFV-SOL 001: "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; NFV descriptors based on TOSCA specification”.

[3] ETSI GS NFV-SOL 004: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; VNF Package and PNFD Archive specification”.


[7] ETSI GS NFV-SOL 014: "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; YAML data model specification for descriptor-based virtualised resource management”.


NOTE: Available at http://www.yaml.org/spec/1.2/spec.html.


NOTE: Available at https://json-schema.org/.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

Not applicable.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purpose of the present document, the following terms apply:

configuration: element of test case to describe information like network topology and the specific structure of the test environment

pre-test conditions: element of test case to describe the dependent conditions before the test sequence is performed in which the SUT/Test Function should be ready for executing the test and in which the test related parameters/measurements are defined

reference function: reference implementation of NFV functional components in the test environment

NOTE: The NFV functional components which are neither SUT nor Test Function in the test environment are Reference Function, the actual entity of Reference Function depends on the specific SUT.

test environment: environment which provides all the functional elements needed for the testing execution on SUT, consists of test functions, reference functions

test function: entity that will be controlled (for example, by test controller) for test execution and monitored to obtain measurements for test results in the test environment

test PNF: physical instrument as test function in the test environment

test sequence: element of test case which contains a series of test steps listed in sequence to describe the operation of each step like controlling or checking the SUT/Test Function

test system: specialized tool (system) built for the purpose of testing that has the abilities including test case management and execution, control and communication with SUT during testing, observation and measurement of test result

test verdict: element of test case that is used to describe how to record the test result according to the checking step's result in test sequence. For functionality or API testing, test verdict describes how the result is deemed as passed or failed; for benchmark testing, test verdict describes which values should be recorded

test VNF: virtual instrument as test function in the test environment

3.2 Symbols

Void.
3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

- **API** Application Programming Interface
- **HTML** Hyper Text Markup Language
- **JSON** JavaScript Object Notation
- **MANO** Management and Orchestration
- **NFV** Network Function Virtualization
- **NFVI** Network Function Virtualization Infrastructure
- **NFVO** NFV Orchestrator
- **NS** Network Service
- **SUT** System Under Test
- **URL** Uniform Resource Locator
- **VIM** Virtual Infrastructure Manager
- **VM** Virtual Machine
- **VNF** Virtual Network Function
- **VNFM** VNF Manager
- **vFW** virtual Firewall
- **XML** Extensible Markup Language
- **YAML** YAML Ain't Markup Language

4 Test Case Description use-cases (informative)

4.1 General

The following use cases describe the steps involved in NFV automatic testing, the relevant information may be referred in the Test Case Description file. The use cases capture the generic processes as well as the actions required to be performed by actors playing different roles in order to identify the requirements for the standard Test Case Description format. All the use cases presented in this clause are informative.

For the purpose of the use cases, the roles identified in table 4.1-1 have been identified.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case Description Provider</td>
<td>Provide the Test Case Description file according to corresponding automated test.</td>
</tr>
<tr>
<td>Test Case Description Consumer</td>
<td>System that parses the Test Case Description file and use the information obtained from the test description file to automatically execute the test. (For example, it could be a test framework.)</td>
</tr>
</tbody>
</table>

4.2 Test Case Description composition

Test cases are designed for SUT (in the present document are functional components of the NFV architecture), and automated tests are implemented. The standardized Test Case Description defined in the present document provides key information to describe the test, information about deployment automation, information about automated test execution, and information about test result collection. The Test Case Description composition encompasses steps to describe that the content in Test Case Description will be specified by which role.

<table>
<thead>
<tr>
<th>#</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test Case Description Provider</td>
</tr>
</tbody>
</table>
Table 4.2-2: Pre-conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Pre-conditions</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test case has been designed; automated test has been implemented</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2-3: Post-conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Post-conditions</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A Test Case Description file for corresponding test case</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2-4: Base Flow

<table>
<thead>
<tr>
<th>#</th>
<th>Role</th>
<th>Action/Description</th>
</tr>
</thead>
</table>
| 1  | Test Case Description provider            | According to the automated test, with the standard format, specify the information including:  
• key information to describe the test;  
• information about deployment automation;  
• information about automated test execution;  
• information about test result collection. |

4.3 Test case selection

By parsing the key description information of the test case provided in the test description files, the Test Case Description Consumer may have the capability to select test cases according to the actual test task.

Table 4.3-1: Roles

<table>
<thead>
<tr>
<th>#</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test Case Description Consumer</td>
</tr>
</tbody>
</table>

Table 4.3-2: Pre-conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Pre-conditions</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test Case Description composition is done</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3-3: Post-conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Post-conditions</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The test cases to be executed have been selected</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3-4: Base Flow

<table>
<thead>
<tr>
<th>#</th>
<th>Role</th>
<th>Action/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test Case Description Consumer</td>
<td>Obtain Test Case Description files.</td>
</tr>
<tr>
<td>2</td>
<td>Test Case Description Consumer</td>
<td>Parse the key description information of the test case.</td>
</tr>
<tr>
<td>3</td>
<td>Test Case Description Consumer</td>
<td>Select the test cases to be executed.</td>
</tr>
</tbody>
</table>

4.4 Automated test deployment

Automated test is closely related to the execution environment. SUT, Test Functions and Reference Functions may need to be deployed automatically before the test execution. Considering the functional components of the NFV architecture as different type of software, there are kinds of deployment automation solutions.
Typical examples of those solutions are:

- Setting up virtual machines by delivering VM images
- Managing VNF deployments via OpenStack Heat or according to ETSI GS NFV-SOL 001 [2], ETSI GS NFV-SOL 004 [3], ETSI GS NFV-SOL 006 [4] and ETSI GS NFV-SOL 007 [5]
- Managing container deployments via Kubernetes
- Installing application software on top of an operating system

When SUT is a VNF under test, the relevant test functions are VNFs too, the deployment will be that VNFs are instantiated through NFV MANO. In this case, Test Case Description file may need to specify the resource file information required for deployment operation: VNF package, and other required parameters depending on demands.

NOTE: Other situations for non-VNF SUTs are FFS.

Table 4.4-1: Roles

<table>
<thead>
<tr>
<th>#</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test Case Description Consumer</td>
</tr>
</tbody>
</table>

Table 4.4-2: Pre-conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Pre-conditions</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The test cases have been selected to be run</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4-3: Post-conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Post-conditions</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The SUT/Test Function/Reference Function involved in the automated test have been deployed</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4-4: Base Flow

<table>
<thead>
<tr>
<th>#</th>
<th>Role</th>
<th>Action/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test Case Description Consumer</td>
<td>Obtain Test Case Description files.</td>
</tr>
<tr>
<td>2</td>
<td>Test Case Description Consumer</td>
<td>Map the related parameters to the corresponding values in actual environment.</td>
</tr>
<tr>
<td>3</td>
<td>Test Case Description Consumer</td>
<td>With the parameter values, perform the deployment operation via execute automation scripts.</td>
</tr>
</tbody>
</table>

4.5 Automated test execution

If the role that implements the automated test and the role that executes the automated test are not hosted by the same organization, a method is needed between these 2 roles to provide the information about how to execute automated testing.

The test case description file may contain such information, so that the Test Case Description consumer can use the information to execute the test via parsing the machine-readable test case description file.

During this process, the Test Case Description consumer may need to know:

- By which type of operation the test is performed (e.g. by running a command, by executing a test script or by call an API).
- The details about the exact operation.
- On which system the operation is done.
• The relevant files for execution (e.g. the test script file).
• The parameters related to actual environment (mapping these parameters to the corresponding values in actual environment).
• The parameters specified for values related to test steps (may have default values, but also capable to specify the customized values when execute the test).

### Table 4.5-1: Roles

<table>
<thead>
<tr>
<th>#</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test Case Description Consumer</td>
</tr>
</tbody>
</table>

### Table 4.5-2: Pre-conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Pre-conditions</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The SUT/Test Function/Reference Function involved in the automated test have been deployed</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4.5-3: Post-conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Post-conditions</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The automated test execution has finished</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4.5-4: Base Flow

<table>
<thead>
<tr>
<th>#</th>
<th>Role</th>
<th>Action/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test Case Description Consumer</td>
<td>Obtain the test case description files.</td>
</tr>
<tr>
<td>2</td>
<td>Test Case Description Consumer</td>
<td>Parse and confirm the exact operation to execute the test, fetch or find the related files.</td>
</tr>
<tr>
<td>3</td>
<td>Test Case Description Consumer</td>
<td>Map the related parameters to the corresponding values in the actual environment.</td>
</tr>
<tr>
<td>4</td>
<td>Test Case Description Consumer</td>
<td>Specify the values related to test steps as parameters of test execution. (Optional)</td>
</tr>
<tr>
<td>5</td>
<td>Test Case Description Consumer</td>
<td>Execute the automated test.</td>
</tr>
</tbody>
</table>

### 4.6 Test result collection

The result of automated test execution may be presented differently depending on the implementation. For example, it may be printed directly in the output, or saved as files with format like JSON, XML, HTML, text. Test results presented in different forms increase complexity in the collection of test results.

The test case description file may provide information to describe how the test results are collected, such as by saving the printed information directly, by fetching information through the API, or by obtaining files with specified path/location. Then the Test Case Description Consumer can use the information above to collect test results.

### Table 4.6-1: Roles

<table>
<thead>
<tr>
<th>#</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test Case Description Consumer</td>
</tr>
</tbody>
</table>
Table 4.6-2: Pre-conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Pre-conditions</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test case description file contains information that can be parsed by Test Case Description Consumer about when to collect results (collection start/stop) and how to collect results</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>The automated test execution is ready to begin</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.6-3: Post-conditions

<table>
<thead>
<tr>
<th>#</th>
<th>Post-conditions</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test result has been collected</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.6-4: Base Flow

<table>
<thead>
<tr>
<th>#</th>
<th>Role</th>
<th>Action/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test Case Description Consumer</td>
<td>Parse the test description file and confirm how to collect the test result.</td>
</tr>
<tr>
<td>2</td>
<td>Test Case Description Consumer</td>
<td>Collect test results based on the information parsed from the file.</td>
</tr>
</tbody>
</table>

5 Requirements for Test case description template

5.1 Generic Requirements

Table 5.1-1 specifies generic requirements applicable to the standardised test case description file.

Table 5.1-1: Generic requirements for test case description file

<table>
<thead>
<tr>
<th>Numbering</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST_CASE_DESC.GEN.001</td>
<td>The test case description file shall contain the key information to describe the test, information about deployment automation, information about automated test execution, information about test result collection.</td>
</tr>
</tbody>
</table>

5.2 Requirements for test case description information

Table 5.2-1 specifies requirements applicable to the test case description information.

Table 5.2-1: Requirements for test case description information

<table>
<thead>
<tr>
<th>Numbering</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST_CASE_DESC.TD.001</td>
<td>The test description shall support a way to identify uniquely the test.</td>
</tr>
<tr>
<td>TEST_CASE_DESC.TD.002</td>
<td>The test description shall include the type of testing for test case selection.</td>
</tr>
<tr>
<td>TEST_CASE_DESC.TD.003</td>
<td>The test description shall include the information about SUT for test case selection.</td>
</tr>
<tr>
<td>TEST_CASE_DESC.TD.004</td>
<td>The test description shall include machine readable key information of test case for test case selection.</td>
</tr>
<tr>
<td>TEST_CASE_DESC.TD.005</td>
<td>The test description shall include human readable key information of test case for understanding.</td>
</tr>
<tr>
<td>TEST_CASE_DESC.TD.006</td>
<td>The test description shall include the information to describe the pre-conditions and post-conditions of test case.</td>
</tr>
<tr>
<td>TEST_CASE_DESC.TD.007</td>
<td>The test description shall include the information to describe main steps of test case.</td>
</tr>
</tbody>
</table>
5.3 Requirements for test case inputs information

Table 5.3-1 specifies requirements applicable to the test case inputs information.

<table>
<thead>
<tr>
<th>Numbering</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST_CASE_DESC.Inputs.001</td>
<td>The inputs shall provide the required information for the execution of test case.</td>
</tr>
<tr>
<td>TEST_CASE_DESC.Inputs.002</td>
<td>The inputs shall provide human readable description of the information for the test case execution, including functional description, default values, etc.</td>
</tr>
</tbody>
</table>

5.4 Requirements for test case script execution information

Table 5.4-1 specifies requirements applicable to the test case script execution information.

<table>
<thead>
<tr>
<th>Numbering</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST_CASE_DESC.SE.001</td>
<td>The test case description shall identify the location and the scripting language of the script to be executed.</td>
</tr>
<tr>
<td>TEST_CASE_DESC.SE.002</td>
<td>The test case description shall define the environment requirements of script execution.</td>
</tr>
</tbody>
</table>

5.5 Requirements for test case outputs information

Table 5.5-1 specifies requirements applicable to the test case outputs information.

<table>
<thead>
<tr>
<th>Numbering</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST_CASE_DESC.Outputs.001</td>
<td>If required, the output shall include the result of functional tests.</td>
</tr>
<tr>
<td>TEST_CASE_DESC.Outputs.002</td>
<td>If required, the output shall include one or more measurements of test result.</td>
</tr>
<tr>
<td>TEST_CASE_DESC.Outputs.003</td>
<td>The output shall be available in a machine readable format.</td>
</tr>
<tr>
<td>TEST_CASE_DESC.Outputs.004</td>
<td>The output shall be available in a human readable format.</td>
</tr>
</tbody>
</table>

6 Test case description information model

6.1 Introduction

The clauses below define the information elements related to the Test Case Descriptor, which is also the Test case description template.

6.2 TestCaseDescriptor information element

6.2.1 Introduction

A TestCaseDescriptor is a test case description template which describes a test case in terms of automated test including inputs, execution, outputs.

6.2.2 Attributes

The attributes of the TestCaseDescriptor information element shall follow the indications provided in table 6.2.2-1.
### Table 6.2.2-1: Attributes of the TestCaseDescriptor information element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Qualifier</th>
<th>Cardinality</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>M</td>
<td>1</td>
<td>Identifier</td>
<td>Specifies this template uniquely.</td>
</tr>
<tr>
<td>testDescription</td>
<td>M</td>
<td>1</td>
<td>TestDescription</td>
<td>Specifies a test description plus the type of test, type of evaluation, SUT to be tested, provider of test case. See note. See clause 6.3.</td>
</tr>
<tr>
<td>input</td>
<td>M</td>
<td>0..N</td>
<td>TestInput</td>
<td>Specifies the input information of test case.</td>
</tr>
<tr>
<td>scriptExecution</td>
<td>M</td>
<td>1</td>
<td>ScriptExecution</td>
<td>Specifies the script designation information of test case. See clause 6.5.</td>
</tr>
<tr>
<td>output</td>
<td>M</td>
<td>0..N</td>
<td>RequiredTestOutput</td>
<td>Specifies the output information of test case.</td>
</tr>
</tbody>
</table>

**NOTE:** The concept of a test description is coincident with the test description defined in other TST specifications like ETSI GS NFV-TST 010 [1].

### 6.3 TestDescription information element

#### 6.3.1 Introduction

A TestDescription information element describes the identifier of test case, the objective of test case, the type of testing, the information about SUT, the configuration of test case, the provider of test case, the pre-conditions, the applicability, the post-conditions and the test sequence of test case, etc.

#### 6.3.2 Attributes

The attributes of the TestDescription information element shall follow the indications provided in table 6.3.2-1.
### Table 6.3.2-1: Attributes of the TestDescription information element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Qualifier</th>
<th>Cardinality</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>testId</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Identifier of this test case.</td>
</tr>
<tr>
<td>testObjective</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Provides human-readable information on what is intended to be tested in the test case.</td>
</tr>
<tr>
<td>reference</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Indicates the specifications or requirements documents where the tested requirements are expressed.</td>
</tr>
<tr>
<td>testType</td>
<td>M</td>
<td>1</td>
<td>Enum</td>
<td>Specifies the type of testing that the test case belongs to. Allowed values: Compliance, Functionality, Benchmarking, Interoperability</td>
</tr>
<tr>
<td>testEvaluation</td>
<td>M</td>
<td>1</td>
<td>Enum</td>
<td>Specifies how the test case evaluate the test result. Allowed values: Verification, Measurement. See note 1.</td>
</tr>
<tr>
<td>sutInfo</td>
<td>M</td>
<td>1</td>
<td>SutInfo</td>
<td>Information that identifies the SUT targeted by the test case, and properties related to this SUT. See clause 6.3.3.</td>
</tr>
<tr>
<td>provider</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>The provider of this test case. This attribute may include name of the organization (like a company name) and specific person or department that defined the test case. See note 2.</td>
</tr>
<tr>
<td>preConditions</td>
<td>M</td>
<td>0..1</td>
<td>PreConditions</td>
<td>Defines the conditions that in which states the SUT should be before undergoing the actual test execution. This element may include identifiers (testId) of other test cases which need to be executed to reach the pre-conditions. See clause 6.3.4.</td>
</tr>
<tr>
<td>configId</td>
<td>M</td>
<td>1</td>
<td>Identifier</td>
<td>The name or id to identify the Configuration for this test case. See note 3.</td>
</tr>
<tr>
<td>applicability</td>
<td>M</td>
<td>0..1</td>
<td>Applicability</td>
<td>Specifies the features which are required to be supported for this test case execution. See clause 6.3.5.</td>
</tr>
<tr>
<td>postConditions</td>
<td>M</td>
<td>0..1</td>
<td>String</td>
<td>Defines the conditions that in which states the SUT should be after the test execution.</td>
</tr>
<tr>
<td>testSequence</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Provides human-readable brief description for the steps of test case.</td>
</tr>
</tbody>
</table>

**NOTE 1:** The 'testEvaluation' describes in which way this test case will evaluate the test result. The value can be one of two types: verification and measurement, 'verification' means at the end of the execution there will be a result as pass/fail, and 'measurement' means there will be a result value for recording or for further measure.

**NOTE 2:** A test case can be provided by vendor, Service Provider or third-party validator.

**NOTE 3:** In an ideal state, typical configurations used in specific type of testing for specific SUT are defined in relevant specifications, and name or id are assigned to those Configurations. These names or ids can be directly specified in 'configId'.

### 6.3.3 SutInfo information element

#### 6.3.3.1 Introduction

The SutInfo information element includes attributes related to what SUT would be tested, the type of SUT, which version would be tested, and by which vendor the SUT is provided.
6.3.3.2 Attributes

The attributes of the SutInfo information element shall follow the indications provided in table 6.3.3.2-1.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Qualifier</th>
<th>Cardinality</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sutType</td>
<td>M</td>
<td>1</td>
<td>Enum</td>
<td>Specifies the type of functional component of NFV architecture to which the SUT belongs. Allowed values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• VNF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• NS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• NFVO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• VNFFM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• NFVO+VNFFM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• NFVI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• VIM</td>
</tr>
<tr>
<td>product</td>
<td>M</td>
<td>0..1</td>
<td>String</td>
<td>Indicates what specific product to be tested in human-readable format. See note.</td>
</tr>
<tr>
<td>vendor</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Indicates by which vendor the SUT is provided</td>
</tr>
</tbody>
</table>

NOTE: The value is used to identify what exactly would be tested. For example, when the SUT is a vFW VNF, the 'product' could be 'FW'.

6.3.4 PreConditions information element

6.3.4.1 Introduction

The PreConditions information element includes attributes related to the testId values of dependent test cases, the explanation of each pre-condition and if any, the automated script to make sure pre-conditions.

6.3.4.2 Attributes

The attributes of the PreConditions information element shall follow the indications provided in table 6.3.4.2-1.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Qualifier</th>
<th>Cardinality</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dependency</td>
<td>M</td>
<td>0..N</td>
<td>String</td>
<td>The testId value of other test cases which need be executed to reach the pre-conditions.</td>
</tr>
<tr>
<td>condition</td>
<td>M</td>
<td>0..N</td>
<td>String</td>
<td>Provides human-readable information on the condition that SUT should reach before test execution.</td>
</tr>
<tr>
<td>actionScript</td>
<td>M</td>
<td>0..N</td>
<td>String</td>
<td>Specifies the automated script to be run to ensure pre-conditions are achieved. See note.</td>
</tr>
</tbody>
</table>

NOTE: The actionScript attribute shall not be present if the operations to make sure pre-conditions is implemented as part of the test script itself.

6.3.5 Applicability information element

6.3.5.1 Introduction

The Applicability information element includes attributes related to the functions which are required for this specific test case.

6.3.5.2 Attributes

The attributes of the Applicability information element shall follow the indications provided in table 6.3.5.2-1.
Table 6.3.5.2-1: Attributes of the PreConditions information element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Qualifier</th>
<th>Cardinality</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>applicableFeature</td>
<td>M</td>
<td>0..N</td>
<td>String</td>
<td>Lists the required features to be supported in order to execute this TD.</td>
</tr>
</tbody>
</table>

6.4 TestInput information element

6.4.1 Introduction

The TestInput information element is used to give the input parameters and their values for enabling script execution in the template, for example, testing traffic throughput and duration. The given attributes may differ from use cases.

6.4.2 Attributes

The attributes of the TestInput information element shall follow the indications provided in table 6.4.2-1.

Table 6.4.2-1: Attributes of the TestInput information element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Qualifier</th>
<th>Cardinality</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Specifies the name of the attribute required for testing execution.</td>
</tr>
<tr>
<td>description</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Provides human readable function description of the attribute.</td>
</tr>
<tr>
<td>type</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Specifies the input attribute data type, it can be string, integer, etc.</td>
</tr>
<tr>
<td>value</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Specifies the value for this attribute.</td>
</tr>
</tbody>
</table>

6.5 ScriptExecution information element

6.5.1 Introduction

The ScriptExecution information element is to give the information that is needed to enable the execution of the script.

6.5.2 Attributes

The attributes of the ScriptExecution information element shall follow the indications provided in table 6.5.2-1.

Table 6.5.2-1: Attributes of the ScriptExecution information element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Qualifier</th>
<th>Cardinality</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Specifies the fetching location (URL) of the script.</td>
</tr>
<tr>
<td>type</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Specifies the scripting language type of the script, for example, python.</td>
</tr>
<tr>
<td>environmentVariable</td>
<td>M</td>
<td>0..N</td>
<td>keyValuePair</td>
<td>Specifies the parameters that are related to the environment requirements of script execution, e.g. the environment version of script execution.</td>
</tr>
</tbody>
</table>
6.6 RequiredTestOutput information element

6.6.1 Introduction

The RequiredTestOutput information element is to describe the information required to be returned after test execution. The information can be the results of performance metrics, which can refer to requirements number within ETSI GS NFV-TST 009 [6], or vendor-specific information as agreed with the operator.

6.6.2 Attributes

The attributes of the RequiredTestOutput information element shall follow the indications provided in table 6.6.2-1.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Qualifier</th>
<th>Cardinality</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>The name of the attribute presenting the test results (for example, requirement number in ETSI GS NFV-TST 009 [6]).</td>
</tr>
<tr>
<td>description</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>Provides human readable function description of the attribute.</td>
</tr>
<tr>
<td>type</td>
<td>M</td>
<td>1</td>
<td>String</td>
<td>The data type of the attribute, it can be string, integer, etc.</td>
</tr>
</tbody>
</table>

7 Test case description Data Model

7.1 Overview

The following clauses define the data model based on information model definition in clause 6, and give the YAML-based data models, referring the definition in ETSI GS NFV-SOL 014 [7].

7.2 Common data types

7.2.1 Introduction

Clause 7.2 specifies the common data types that are used for declaring the parameters and grammar elements in Test Case Description Template.

7.2.2 Simple data types

The simple data types that can be used in Test Case Description Template are defined in table 7.2.2-1.

<table>
<thead>
<tr>
<th>Type name</th>
<th>Description</th>
<th>Example(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>A string as defined in YAML v1.2 [8].</td>
<td>&quot;a string&quot;</td>
</tr>
<tr>
<td>Number</td>
<td>A number as defined in IETF RFC 8259 [9] referred in JSON Schema [10].</td>
<td>&quot;23&quot;, &quot;-1.023E3&quot;</td>
</tr>
<tr>
<td>Boolean</td>
<td>A data type that can take the following values: true, false. The type is defined in JSON Schema and referred in YAML v1.2 [8].</td>
<td>&quot;true&quot;, &quot;false&quot;</td>
</tr>
</tbody>
</table>

NOTE: Enum and Identifier in information model are represented as string, with constrained values.

The data model definition for the parameters that are simple data types shall comply with the following YAML syntax.
Table 7.2.2-2: Data model for a parameter

<table>
<thead>
<tr>
<th>Field</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameter_name</td>
<td>yes</td>
<td>The name of the parameter.</td>
</tr>
<tr>
<td>type</td>
<td>yes</td>
<td>The type of the parameter.</td>
</tr>
<tr>
<td>description</td>
<td>yes</td>
<td>A human readable description for the parameter.</td>
</tr>
<tr>
<td>default</td>
<td>no</td>
<td>A default value of the parameter.</td>
</tr>
<tr>
<td>enum</td>
<td>no</td>
<td>A set of enumerated values for a parameter to restrict the possible values.</td>
</tr>
</tbody>
</table>

< parameter name >:
  type: <the type of the parameter>
  description: <the description of the parameter>
  default: <the default value of the parameter>
  enum:
    - <enumerated value 1>
    - <enumerated value 2>

7.2.3 Structured data types

The structured data types are used for complex data types, individual structured data type is represented in the present document using "\" recursive as inline definition. There are two structured data types defined in JSON schema: object and array.

Table 7.2.3-1: Data model for {parameter name}

<table>
<thead>
<tr>
<th>Parameter Name and Attributes</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{parameter name}</td>
<td>{object, array}</td>
<td>Type of the parameter.</td>
</tr>
<tr>
<td>{description}</td>
<td>-</td>
<td>Description of the parameter.</td>
</tr>
<tr>
<td>{attribute}</td>
<td>{attribute type}</td>
<td>Type of {attribute}.</td>
</tr>
<tr>
<td>&gt;{sub attribute}</td>
<td>{sub attribute type in the attribute}</td>
<td>Type of {sub attribute}.</td>
</tr>
</tbody>
</table>

The syntax of object for parameter definition is represented with the following definition:

{parameter name}:
  type: object
  required:
    - {1st mandatory attribute}
    - {2nd mandatory attribute}
  properties:
    {1st attribute}:
      type: e.g. object
      properties:
        {sub attribute}
    {2nd attribute}:
      ...

The syntax of array for parameter definition is represented with the following definition:

{parameter name}:
  description: <the description of the parameter>
  type: array
  minItems: { lower bound of cardinality}
  maxItems: { upper bound of cardinality}
  items:
    - type: e.g. object
      properties:
        {sub attribute}
7.3 Data model for TestDescription

7.3.1 Introduction

This clause specifies the data model for TestDescription information element which specifies the attributes needed of a specific test case to be executed.

7.3.2 TestDescription element

The parameters used when defining TestDescription shall follow the indicators provided in table 7.3.2-1.

Table 7.3.2-1: Data model for TestDescriptor

<table>
<thead>
<tr>
<th>Parameter Name and Attributes</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TestDescription</td>
<td>Object</td>
<td>Specifies a test description plus the type of test, type of evaluation, SUT to be tested, provider of test case.</td>
</tr>
<tr>
<td>&gt;testId</td>
<td>String</td>
<td>Identifier of the test case.</td>
</tr>
<tr>
<td>&gt;testObjective</td>
<td>String</td>
<td>Provides human-readable information on what is intended to be tested in the test case.</td>
</tr>
<tr>
<td>&gt;reference</td>
<td>String</td>
<td>Indicates the specifications or requirements documents where the tested requirements are expressed.</td>
</tr>
<tr>
<td>&gt;testType</td>
<td>String</td>
<td>Specifies the type of testing that the test case belongs to. Allowed values: Compliance, Functionality, Benchmarking, Interoperability</td>
</tr>
<tr>
<td>&gt;testEvaluation</td>
<td>String</td>
<td>Specifies how the test case evaluate the test result. Allowed values: Verification, Measurement</td>
</tr>
<tr>
<td>&gt;sutInfo</td>
<td>Object</td>
<td>Information that identifies the SUT targeted by the test case, and properties related to this SUT.</td>
</tr>
<tr>
<td>&gt;&gt;sutType</td>
<td>String</td>
<td>Specifies the type of functional component of NFV architecture to which the SUT belongs. Allowed values: VNF, NS, NFVO, VNF, NFVO+VNF, NFVI, VIM</td>
</tr>
<tr>
<td>&gt;&gt;product</td>
<td>String</td>
<td>Indicates what specific product to be tested in human-readable format. The value is used to identify what exactly is to be tested. For example, when the SUT is a vFW VNF, the 'product' could be 'FW'.</td>
</tr>
<tr>
<td>&gt;&gt;vendor</td>
<td>String</td>
<td>Indicates by which vendor the SUT is provided.</td>
</tr>
<tr>
<td>&gt;provider</td>
<td>String</td>
<td>The provider of this test case. This attribute may include name of the organization (like a company name) and specific person or department that defined the test case.</td>
</tr>
<tr>
<td>&gt;preConditions</td>
<td>Object</td>
<td>Defines the conditions that in which states the SUT should be before undergoing the actual test execution. This attribute may include testId values of other test cases which need to be executed to reach the pre-conditions.</td>
</tr>
<tr>
<td>&gt;&gt;dependency</td>
<td>Array of String</td>
<td>The testId value of other test cases which need be executed to reach the pre-conditions.</td>
</tr>
<tr>
<td>&gt;&gt;condition</td>
<td>Array of String</td>
<td>Provides human-readable information on the condition that SUT should reach before test execution.</td>
</tr>
<tr>
<td>&gt;&gt;actionScript</td>
<td>Array of String</td>
<td>Specifies the automated script to be run to ensure pre-conditions are achieved.</td>
</tr>
<tr>
<td>&gt;configId</td>
<td>String</td>
<td>The name or id to identify the configurations for this test case.</td>
</tr>
<tr>
<td>&gt;applicability</td>
<td>Object</td>
<td>Specifies the features which are required to be supported for this test case execution.</td>
</tr>
<tr>
<td>Parameter Name and Attributes</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>&gt;&gt; applicableFeature</td>
<td>Array of String</td>
<td>Lists the required features to be supported in order to execute this TestDescription.</td>
</tr>
<tr>
<td>&gt;postConditions</td>
<td>String</td>
<td>Provides human-readable information on the conditions in which states the SUT should be after the test execution.</td>
</tr>
<tr>
<td>&gt;testSequence</td>
<td>String</td>
<td>Provides human-readable brief description for the steps of the test case.</td>
</tr>
</tbody>
</table>

The syntax of the **TestDescription** shall comply with the following definition:

```json
TestDescription:
  description: >
  Specifies a test description, plus including the type of test, type of evaluation, SUT to be tested, provider of test case and so on.
  type: object
  required:
    - testId
    - testObjective
    - reference
    - testType
    - testEvaluation
    - sutInfo
    - provider
    - configId
    - testSequence
  properties:
    testId:
      description: >
      Identifier of this test case.
      type: string
    testObjective:
      description: >
      Provides human-readable information on what is intended to be tested in the test case.
      type: string
    reference:
      description: >
      Indicates the specifications or requirements documents where the tested requirements are expressed.
      type: string
    testType:
      description: >
      Specifies the type of testing that the test case belongs to.
      type: string
      enum:
        - Compliance
        - Functionality
        - Benchmarking
        - Interoperability
    testEvaluation:
      description: >
      Specifies how the test case evaluate the test result.
      type: string
      enum:
        - Verification
        - Measurement
    sutInfo:
      description: >
      Information that identifies the SUT targeted by the test case what SUT would be tested, and other properties related to the SUT.
      type: object
      required:
        - sutType
        - vendor
      properties:
        sutType:
          description: >
          Specifies the type of functional component of NFV architecture to which the SUT belongs.
          type: string
          enum:
            - VNF
            - NS
            - NFVO
            - VNFM
            - NFVO+VNFM
            - NFVI
```
- VIM
  product:
    description: >
    Indicates what specific product would be tested. The value is used to identify what
    exactly would be tested. For example, when the SUT is a vFW VNF, the 'product' could be 'FW'
    type: string
  vendor:
    description: >
    Indicates by which vendor the SUT is provided.
    type: string
  provider:
    description: >
    The provider of this test case. This attribute may include name of the organization
    (like a company name) and specific person or department that defined the test case.
    type: string
  preConditions:
    description: >
    Defines the conditions that in which states the SUT should be before undergoing the actual
    test execution. This attribute may include testId values of other test cases which need to be
    executed to reach the pre-conditions.
    type: object
    properties:
      dependency:
        description: >
        The testId value of other test cases which need to be executed to reach the pre-
        conditions.
        type: array
        minItems: 0
        items:
          type: string
      condition:
        description: >
        Describes the condition that SUT should reach before test execution.
        type: array
        minItems: 0
        items:
          type: string
      actionScript:
        description: >
        Specifies the automated script to be run to ensure pre-conditions are achieved.
        type: array
        minItems: 0
        items:
          type: string
  configId:
    description: >
    The name or id to identify the configurations for this test case.
    type: string
  applicability:
    description: >
    Specifies the features which are required to be supported for this test case execution.
    type: object
    properties:
      applicableFeature:
        description: >
        Lists the required features to be supported in order to execute this TestDescription.
        type: array
        minItems: 0
        items:
          type: string
  postConditions:
    description: >
    Defines the conditions that in which states the SUT should be after the test execution.
    type: string
  testSequence:
    description: >
    Provides human-readable brief description for the steps of the test case.
    type: string
7.4 Data model for TestInput

7.4.1 Introduction

This clause specifies the data model for TestInput information element which specifies the parameters needed of a specific test case to be executed.

7.4.2 TestInput element

The parameters used when defining TestInput shall follow the indicators provided in table 7.4.2-1.

<table>
<thead>
<tr>
<th>Parameter Name and Attributes</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TestInput</td>
<td>Array of Object</td>
<td>Specifies the input parameters of test case.</td>
</tr>
<tr>
<td>&gt; name</td>
<td>String</td>
<td>Specifies the name of the parameter required for testing execution.</td>
</tr>
<tr>
<td>&gt; description</td>
<td>String</td>
<td>Provides human readable function description of the parameter.</td>
</tr>
<tr>
<td>&gt; type</td>
<td>String</td>
<td>Specifies the input parameter data type, it can be string, integer, etc.</td>
</tr>
<tr>
<td>&gt; value</td>
<td>String</td>
<td>Specifies the value for this parameter.</td>
</tr>
</tbody>
</table>

The syntax of the TestInput shall comply with the following definition:

```json
TestInput:
  description: >
    Specifies the input information of test case.
  type: array
  minItems: 0 # lower bound of cardinality
  items:
    type: object
    required:
      - name
      - description
      - type
      - value
    properties:
      name:
        description: >
          Specifies the name of the parameter required for testing execution.
        type: string
      description:
        description: >
          Provides human readable function description of the parameter.
        type: string
      type:
        description: >
          Specifies the input parameter data type, it can be string, integer, etc.
        type: string
      value:
        description: >
          Specifies the value for this parameter.
        type: string
```

7.5 Data model for ScriptExecution

7.5.1 Introduction

This clause specifies the data model for ScriptExecution information element which specifies what is needed to enable the execution of the script.

7.5.2 ScriptExecution element

The parameters used when defining ScriptExecution shall follow the indicators provided in table 7.5.2-1.
Table 7.5.2-1: Data model for ScriptExecution

<table>
<thead>
<tr>
<th>Parameter Name and Attributes</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScriptExecution</td>
<td>Object</td>
<td>Specifies the script designation information of test case.</td>
</tr>
<tr>
<td>&gt; address</td>
<td>String</td>
<td>Specifies the fetching location (URL) of the script.</td>
</tr>
<tr>
<td>&gt; type</td>
<td>String</td>
<td>Specifies the scripting language type of the script, for example, python, shell script.</td>
</tr>
<tr>
<td>&gt; environmentVariable</td>
<td>Array of Object</td>
<td>List of environment variable key-value pairs that are related to the environment requirements of script execution.</td>
</tr>
<tr>
<td>&gt;&gt; key</td>
<td>String</td>
<td>Specify the name of the environment variable parameter.</td>
</tr>
<tr>
<td>&gt;&gt; value</td>
<td>String</td>
<td>Specify the value of the environment variable parameter.</td>
</tr>
</tbody>
</table>

The syntax of the ScriptExecution shall comply with the following definition:

```
ScriptExecution:
  description: >
  type: object
  required:
    - address
    - type
  properties:
    address:
      description: >
      type: string
    type:
      description: >
      type: string
  environmentVariable:
    description: >
    type: array
    minItems: 0 # lower bound of cardinality
    items:
      type: object
      properties:
        key:
          description: >
          type: string
        value:
          description: >
          type: string
```

7.6 Data model for RequiredTestOutput

7.6.1 Introduction

This clause specifies the data model for RequiredTestOutput information element which specifies the information required to be returned after test execution.

7.6.2 RequiredTestOutput element

The parameters used when defining RequiredTestOutput shall follow the indicators provided in table 7.6.2-1.
Table 7.6.2-1: Data model for RequiredTestOutput

<table>
<thead>
<tr>
<th>Parameter Name and Attributes</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequiredTestOutput</td>
<td>Array of Object</td>
<td>Specifies the output parameter of test case.</td>
</tr>
<tr>
<td>&gt; name</td>
<td>String</td>
<td>The name of the parameter presenting the test results (for example, requirement number in ETSI GS NFV-TST 009 [6]).</td>
</tr>
<tr>
<td>&gt; description</td>
<td>String</td>
<td>Provides human readable description of the parameter.</td>
</tr>
<tr>
<td>&gt; type</td>
<td>String</td>
<td>The data type of the parameter, it can be string, integer, etc.</td>
</tr>
</tbody>
</table>

The syntax of the `RequiredTestOutput` shall comply with the following definition:

```json
RequiredTestOutput:
  description: >
    Specifies the output parameter of test case.
  type: array
  minItems: 0 # lower bound of cardinality
  items:
    type: object
    required:
      - name
      - description
      - type
    properties:
      name:
        description: >
          The name of the parameter presenting the test results (for example, requirement number in ETSI GS NFV-TST 009 [6]).
        type: string
      description:
        description: >
          Provides human readable description of the parameter.
        type: string
      type:
        description: >
          The data type of the parameter, it can be string, integer, etc.
        type: string
```
Annex A (informative):
Examples using test description template

A.1 Example#1: VNF Traffic Forwarding Test Description

This test case is for testing VNF traffic forwarding functionality, two test VNFs will count the ingress and egress traffic frames and bits with the VNF under test to have the measurement.

The following gives an example test description of test case:

```json
schema_version: 1.0
name: traffic-forward-test
description: An example of automation test for VNF traffic forwarding functionality.

Test_example:
properties:
  id: b2CC0ce7-2222-4fC7-95ed-4330d70a2277
  TestDescription:
    testId: 45eb297c-3688-4902-9c37-88e4fdd5f1e38
    testObjective: Measure the packet handling capacity of bidirectional VNF traffic forwarding function.
    reference: clause 6.4.3.3.2 - ETSI GS NFV-SOL 002 v3.6.1
    testType: Benchmarking
    testEvaluation: Measurement
  sutInfo:
    sutType: VNF
    product: FW
    vendor: abcTelecom
    provider: xyzIntegration
  preConditions:
    dependency: d95c9680-3c67-41ff-ffdc-1ba25ab8990
    condition: The SUT VNF and Test VNF have been instantiated successfully.
    actionScript: file:///${EXECUTION_HOME}/script/topology.py
    configId: Config_prod_TOPOLOGY
  testSequence:
    Step 01. Test VNF A configures the IP and port of the SUT VNF and target Test VNF B.
    Step 02. Set the duration of Test VNF A.
    Step 03. Count the frames and bits sent and received by Test VNF A.
    Step 04. Count the frames and bits sent and received by Test VNF B.

TestInput:
- name: format
description: Output formats, supported formats such as table, csv, json, yaml.
type: string
  value: yaml
- name: labserver-ip
description: The IP address of license server.
type: string
  value: 10.10.10.1
- name: username
description: The username used for create session.
type: string
  value: Admin
- name: stcv1-mgmt-ip
description: The management IP address of Test VNF traffic generator west.
type: string
  value: 10.10.10.2
- name: stcv1-test-ip
description: The IP address of Test VNF traffic generator west test port.
type: string
  value: 10.10.10.3
- name: stcv2-mgmt-ip
description: The management IP address of Test VNF traffic generator east.
type: string
  value: 10.10.10.4
- name: stcv2-test-ip
description: The IP address of Test VNF traffic generator east test port.
type: string
  value: 10.10.10.5
- name: sut-left-ip
description: The IP address of SUT left port.
type: string
  value: 10.10.10.6
- name: sut-right-ip
```
description: The IP address of SUT right port.
type: string
value: 10.10.10.7

- name: duration-time
description: The traffic forwarding duration in seconds.
type: integer
value: 100

ScriptExecution:
address: file:///${EXECUTION_HOME}/script/traffic-forward-test.py
type: python

RequiredTestOutput:
- name: Stcv1 Tx frame
description: Transmitted frames from Stcv1.
type: string
- name: Stcv1 Tx bit
description: Transmitted bits from Stcv1.
type: string
- name: Stcv1 Rx frame
description: Received frames of Stcv1.
type: string
- name: Stcv1 Rx bit
description: Received bits of Stcv1.
type: string
- name: Stcv2 Tx frame
description: Transmitted frames from Stcv2.
type: string
- name: Stcv2 Tx bit
description: Transmitted bits from Stcv2.
type: string
- name: Stcv2 Rx frame
description: Received frames of Stcv2.
type: string
- name: Stcv2 Rx bit
description: Received bits of Stcv2.
type: string
Annex B (informative):
Bibliography

ETSI GS NFV-TST 001: "Network Functions Virtualisation (NFV); Pre-deployment Testing; Report on Validation of NFV Environments and Services".

ETSI GR NFV-TST 006 (V1.1.1): "Network Functions Virtualisation (NFV); Testing; Report on CICD and Devops".

ETSI GR NFV-TST 007 (V2.6.1): "Network Functions Virtualisation (NFV) Release 2; Testing; Guidelines on Interoperability Testing for MANO".

ETSI GR NFV-TST 011 (V1.1.1): "Network Functions Virtualisation (NFV); Testing; Test Domain and Description Language Recommendations".
Annex C (informative):
Change history

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Information about changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul 28th, 2020</td>
<td>V0.0.1</td>
<td>Start drafting the skeleton of TST013</td>
</tr>
</tbody>
</table>
| Mar 05th, 2021 | V0.0.2  | Updated the skeleton and added already approved contents:
1) NFVTST(20)000063r2: TST013-proposed contribution on scope in clause1
2) NFVTST(20)000062r1: TST013-proposed contribution on references in clause 2
3) NFVTST(20)000064r8: TST013-proposed contribution on test case description convention in clause3
4) NFVTST(20)000061r3: TST013–proposed contribution on automated test execution in clause 4
5) NFVTST(20)000090r7: TST013-proposed contribution on TestDescription element
6) NFVTST(21)000039r2: TST013-proposed contribution on modification of the skeleton |
| Oct 26th, 2021 | V0.0.3  | 1) NFVTST(21)000030r4: TST013-proposed contribution on clause 4
2) NFVTST(21)163000r1: TST013-proposed contribution on use cases clause 4.5
3) NFVTST(21)163004r2: TST013-proposed contribution on use cases clause 4.6
4) NFVTST(21)000031r2: TST013-proposed contribution on clause 5.1-5.2 requirements
5) NFVTST(21)000067r2: TST013-proposed contribution on clause 5.3 requirements
6) NFVTST(21)000081r2: TST013-proposed contribution on clause 5.4-5.5 requirements
7) NFVTST(21)000086: TST013-Update References of Adding SOL001_006 |
| Apr 15th, 2022 | V0.0.4  | 1) NFVTST(21)000089r2: TST013-Correct the Cardinality of Inputs in clause 6.2.2
2) NFVTST(21)000088r3: TST013- Contribution on clause 6.4 inputs information element
3) NFVTST(21)000016r2: TST013- Proposed contribution on clause 6.5 scriptExecution information element
4) NFVTST(21)000113r5: TST013- Proposed contribution on clause 6.6 outputs information element
5) NFVTST(22)000013r2: TST013- Contribution on clause 7.1~7.2.3 common data types |
| Jul 08th, 2022  | V0.0.5  | 1) NFVTST(22)000018r5_TST_013_Contribution_onClause_7_3_data_model_for_TestCaseInfo
2) NFVTST(22)000029r2_TST013-Clean_up_editor_note_in_clause_6.3.3.2
3) NFVTST(22)000030r2_TST013-Update chapter 6 to align with latest discussion
4) NFVTST(22)000036r1_TST013- Contribution on clause 7.4 Data model for Inputs
5) NFVTST(22)000042r1_TST013- Contribution on clause 7.5 Data model for ScriptExecution
6) NFVTST(22)000043r1_TST013- Contribution on clause 7.6 Data model for testCase Output
7) NFVTST(22)000044r2_TST013- Contribution on Annex_A
8) NFVTST(22)000049r1_TST013-Clean up the Editor's notes all over the document |
# History

<table>
<thead>
<tr>
<th>Document history</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.1.1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>