



# WATS STANDARD JSON FORMAT MANUAL

DESCRIBES THE WSJF FORMAT AND HOW TO USE IT  
VIRINCO AS

**CONTENTS**

- General details ..... 2
- JSON ..... 2
  - Important ..... 3
  - Numeric and Date Formats ..... 3
- Report ..... 4
- UUT ..... 6
  - Miscinfo ..... 7
  - Sub units ..... 8
  - Additional Data ..... 9
- Steps ..... 12
  - Sequence steps ..... 14
  - Test steps ..... 16
- Chart ..... 22
  - Chart Series ..... 22
- Attachment ..... 24
- Message Pop Up ..... 24
- Call Executable ..... 25
- Additional Results ..... 25
- Comparison operators ..... 26
- Status Codes ..... 27

## GENERAL DETAILS

WATS format name	Wats Standard JSON Format (WSJF)
Version	1.0
File example name	WatsStandardJSONFormat.json
Last modified date	29.10.2019

The WATS Standard JSON Format is a standardised format for importing or exporting WATS reports. The WATS Standard JSON Format adheres to the WSJF JSON schema.

## JSON

WSJF is a JSON format. JSON defines objects with properties, which have values. Property values can be one of the following.

## STRING

Strings are quoted using ". They may have a format specified, for example date-time.

```
"property": "string value"
```

## NUMBER

Numbers are not quoted. The decimal separator is '.', so PI should be written as 3.14159. Do not use thousand separators. Only digits and decimal point is allowed.

```
"property": 3.14159
```

## INTEGER

Integers are not quoted. Do not use thousand separators.

```
"property": 100
```

## BOOLEAN

Boolean values are not quoted. They can be either "true" or "false".

```
"property": false
```

## ARRAY

Arrays contain multiple of another value type.

```
"property": ["value 1", "value 2", "value 3"]
```

---

## OBJECT

Objects have properties.

```
"property": {  
  "property1": "some value 1",  
  "property2": "some value 2",  
}
```

---

## NULL

Null is an empty value. The property then has no value.

```
"property": null
```

## IMPORTANT

Required properties must be defined. If a property has value type null, it can have no value, even if the property is required. If a property is not required, it can be omitted. For export, some properties are not included when their value is null.

## NUMERIC AND DATE FORMATS

Numeric: By default, '.' (period) is used as decimal separator, so PI should be written as 3.14159. **Do not use thousand separators. Only digits and decimal point is allowed.**

Date and time: Use the ISO 8601 format, e.g. 2019-09-12T14:26:16.977+02:00

## REPORT

WSJF must begin with an object. The root object describes the report.

Property Name	Description	Data Format	Importance
<b>type</b>	The type of report.	String (1) (T for a UUT report)	Required
<b>result</b>	The outcome of the test.	String (1) <a href="#">See status codes</a>	Required
<b>sn</b>	The serial number of the unit.	String (100)	Required
<b>pn</b>	The part number of the unit.	String (100)	Required
<b>rev</b>	The revision of the unit.	String (100)	Required
<b>start</b>	The start date and time in local time.	String (date-time)	Required
<b>startUTC</b>	The start date and time in UTC time.	String (date-time)	Required
<b>processCode</b>	The operation type code for a WATS process.	Integer (16)	Required
<b>processName</b>	The operation type name for a WATS process	String (100)	Optional
<b>productName</b>	The product name of the unit. This property is not used for incoming reports (read-only)	String (100)	Optional
<b>location</b>	The location where the test takes place.	String (100)	Optional
<b>purpose</b>	The purpose of the test machine.	String (100)	Optional
<b>stationName</b>	The name of the test station.	String (100)	Optional
<b>id</b>	A Globally Unique ID of the report. A report submitted with the same ID as another will overwrite the report. If missing it will be generated.	GUID	Optional
<b>origin</b>	An automated test equipment identifier.	String (100)	Optional
<b>miscInfos</b>	Searchable miscellaneous information about the report.	Array of MiscInfo objects	Optional
<b>subUnits</b>	Information about sub units of the unit.	Array of SubUnit objects	Optional
<b>additionalData</b>	Additional header data.	Array of AdditionalData objects	Optional
<b>root</b>	The root test step of the report.	Step object or null	Optional (Not included if null)
<b>uut</b>	The header data for a UUT report.	UUT object or null	Optional (Not included if null)

## IMPLEMENTATION EXAMPLE

```
{
  "type": "T",
  "id": "bf5e5f36-8d25-4140-9ca9-dd1dea24154f",
  "pn": "WATS FAT",
  "sn": "1894.212031",
  "rev": "FAT",
  "productName": null,
  "processCode": 10,
  "processName": "SW Debug",
  "result": "F",
  "machineName": "VIC-OEF-TEST2",
  "location": "VM",
  "purpose": "apptest",
  "origin": null,
  "start": "2019-10-15T11:22:26.57+02:00",
  "startUTC": "2019-10-15T09:22:26.57Z",
  "root": null,
  "uut": null,
  "miscInfos": [],
  "subUnits": []
}
```

## UUT

The UUT object contains header data specific to a UUT report. It is recommended to fill out as much data as possible as this significantly improves the usability of the final WATS report.

Property Name	Description	Data Format	Importance
<b>user</b>	The name of the operator of the test system.	String (100)	Optional
<b>execTime</b>	The total duration of the test in seconds.	Number or null	Optional
<b>batchSN</b>	The serial number of the batch the unit belongs to.	String (100)	Optional
<b>testSocketIndex</b>	The index of the test socket used when parallel or batch execution.	Short (100)	Optional
<b>fixtureId</b>	The id of the fixture used to perform the tests.	String (100)	Optional
<b>errorCode</b>	The error code of any error that occurred during the test.	Integer (32)	Optional
<b>batchFailCount</b>	The number of tests that failed in this batch.	Integer (32)	Optional
<b>batchLoopIndex</b>	The number of failed tests in the batch.	Integer (32)	Optional
<b>stepIdCausedUUTFailure</b>	The id of the step that caused the report to fail.	Integer (32)	Optional
<b>errorMessage</b>	Message of the error that occurred during the test, if any.	String (500)	Optional
<b>comment</b>	A comment about the report.	String (5000)	Optional

IMPLEMENTATION EXAMPLE

```

"uut": {
  "execTime": 34.5362799,
  "testSocketIndex": -1,
  "batchSN": "BatchSerialNumber",
  "comment": "sgfsd",
  "errorCode": 0,
  "errorMessage": null,
  "fixtureId": "Your fixture ID",
  "user": "administrator",
  "batchFailCount": null,
  "batchLoopIndex": null,
  "stepIdCausedUUTFailure": 8
}
    
```

MISCINFO

Any further data concerning the UUT or the general test run must be entered as miscellaneous information. It can for example be used for software/firmware versions, or for whatever purpose suits the test. There can be any number of MiscInfo objects in a report.

Property Name	Description	Data Format	Importance
<b>description</b>	The name of the misc info	String (100)	Optional
<b>typedef</b>	The type defition of the misc info.	String (30)	Optional
<b>numeric</b>	The numeric value of the misc info	Integer (32)	Optional
<b>text</b>	The text value of the misc ifno	String (100)	Optional



### IMPLEMENTATION EXAMPLE

```
"miscInfos": [
  {
    "description": "Misc info 1",
    "typedef": null,
    "text": "Misc string 1",
    "numeric": 1
  },
  {
    "description": "Misc info 2",
    "typedef": null,
    "text": "Misc string 2",
    "numeric": 2
  }
]
```

### SUB UNITS

You can attach sub-units to the test report, which will be linked in the report header. This allow you track systems and their individual component, which gives traceability to the systems lifecycle.

Register a sub unit by adding a SubUnit object to the SubUnits array.

Property name	Description	Data Format	Importance
<b>pn</b>	The partnumber of the sub unit.	String (100)	Required
<b>sn</b>	The serial number of the sub unit.	String (100)	Required
<b>rev</b>	The revision of the sub unit.	String (100)	Optional
<b>partType</b>	The type of sub unit.	String (50)	Optional

IMPLEMENTATION EXAMPLE

```
"subUnits": [
  {
    "partType": "Sub UUT 1",
    "pn": "Sub PN 1",
    "rev": "Sub rev 1",
    "sn": "Sub SN 1"
  },
  {
    "partType": "Sub UUT 2",
    "pn": "Sub PN 2",
    "rev": "Sub rev 2",
    "sn": "Sub SN 2"
  }
]
```

ADDITIONAL DATA

You can add additional header data to the report. Additional data is a free-form structure of properties and values. If you use the name "Station info", that additional data will be treated as additional station data.

Add an additional data by adding a AdditionalData object to the AdditionalData array.

Property Name	Description	Data Format	Importance
<b>name</b>	Name of additional data.	String	Required
<b>props</b>	Properties of additional data	Array of AdditionalDataProperty objects	Required

#### ADDITIONAL DATA PROPERTY

Stores information about an additional data property.

Property Name	Description	Data Format	Importance
<b>name</b>	Name of property.	String	Required
<b>type</b>	Value type of property.	Number, String, Bool, Obj, or Array	Required
<b>props</b>	Properties of additional data	Array of AdditionalDataProperty objects	Required
<b>flags</b>	Bit flags of property.	Number	Optional (not included if null)
<b>value</b>	Value string of property	String	Required for Number, String, and Bool types, else Optional (not included if null)
<b>comment</b>	Comment of property	String	Required for Number, String, and Bool types, else not included.
<b>props</b>	Array of sub-properties. Used for type Obj.	Array of AdditionalDataProperty objects	Required for Obj type, else not included.
<b>array</b>	Array information. Used for type Array.	AdditionalDataArray object	Required for Array type, else not included.

#### ADDITIONAL DATA ARRAY

A property can be an array. Information about the array is stored in an AdditionalDataArray object.

Property Name	Description	Data Format	Importance
<b>dimension</b>	Dimension of array.	Number	Required
<b>type</b>	Type of values in the array.	Number, String, Bool, Obj, or Array	Required
<b>indexes</b>	Indexes of the array.	Array of AdditionalDataArrayIndex objects	Required

ADDITIONAL DATA ARRAY INDEX

Each index in an array and its value is stored as an AdditionalDataArrayindex.

Property Name	Description	Data Format	Importance
<b>text</b>	The index written as text.	String	Required
<b>indexes</b>	Array of indexes ordered by dimension.	Array of Number	Required
<b>value</b>	The value of the index.	AdditionalDataProperty	Required

IMPLEMENTATION EXAMPLE

```

"name": "Additional UUT info",
"props": [
  {
    "name": "Port"
    "type": "Number"
    "value": 13
  },
  {
    "name": "Coordinates"
    "type": "Obj"
    "props": [
      {
        "name": "X"
        "type": "Number"
        "value": 4
      },
      {
        "name": "Y"
        "type": "Number"
        "props": 3.5
      }
    ]
  }
]

```

STEPS

The root property of the root object contains the actual test data. It is the root step of the step hierarchy and must therefore only contain a sequence call.

Where the content of the step is depends on the step type. For example, if the step is a numeric limit step, the content is in the numericMeas property.

Property Name	Description	Data Format	Importance
<b>group</b>	The step group	String (1) (S (Startup), M (Main), or C (Cleanup))	Required
<b>name</b>	The name of the test step.	String (100)	Required
<b>status</b>	The outcome of the test step.	String (1) <a href="#">See Status Codes</a>	Required
<b>id</b>	The id of the step. This property is not used for incoming reports (read-only).	Integer (32)	Required
<b>stepType</b>	The step type, a textual description of the step.	String. Defined in each step type below	Optional
<b>start</b>	The start date and time of the step execution.	String (date-time)	Optional
<b>totTime</b>	Duration of the step execution in seconds.	Number	Optional
<b>errorCode</b>	The error code of the error that occurred during the step execution	Integer (32)	Optional
<b>errorMessage</b>	The error message of the error that occurred during the step execution	String (200)	Optional
<b>causedSeqFailure</b>	A flag indicating if this step caused the sequence to fail.	Boolean	Optional
<b>causedUUTFailure</b>	A flag indicating if this step caused the report to fail.	Boolean	Optional
<b>reportText</b>	The step comment.	String (5000)	Optional
<b>interactiveExeNum</b>	The interactive exe number of the step.	Integer (32)	Optional
<b>tsGuid</b>	The step id from TestStand.	String (30)	Optional
<b>steps</b>	A list of sub steps for this step. Only for steps of type SequenceCall.	Array of Step objects	Optional (not included if null)
<b>numericMeas</b>	A list of numeric measurements belonging to this step.	Array of NumericMeasurement objects	Optional (not included if null)

<b>stringMeas</b>	A list of text measurements belonging to this step.	Array of StringMeasurement objects	Optional (not included if null)
<b>booleanMeas</b>	A list of boolean measurements belonging to this step.	Array of BooleanMeasurement objects	Optional (not included if null)
<b>additionalResults</b>	A list of additional results for the step.	Array of AdditionalData objects	Optional (not included if null)
<b>seqCall</b>	The information about the sequence call.	SequenceCall object	Optional (not included if null)
<b>callExe</b>	Information about the executable called by this step.	CallExe object	Optional (not included if null)
<b>messagePopup</b>	Information about the popup message belonging to this step.	MessagePopup object	Optional (not included if null)
<b>chart</b>	The chart belonging to this step.	Chart object	Optional (not included if null)
<b>attachment</b>	The attachment belonging to this step.	Attachment object	Optional (not included if null)
<b>loop</b>	Information about the loop the step is in.	LoopInfo object or null	Optional (not included if null)

### IMPLEMENTATION EXAMPLE

```

{id": 1,
"group": "M",
"stepType": "SequenceCall",
"interactiveExeNum": null,
"name": "MainSequence Callback",
"start": "2019-10-15T11:22:26.94+02:00",
"status": "F",
"errorCode": 0,
"errorMessage": null,
"tsGuid": "ID#:xnW0PT0ORE2ei7wF7uaxyB",
"totTime": 34.5362799,
"causedSeqFailure": true,
"causedUUTFailure": false,
"reportText": null

```

---

## STEP TYPES

Step type	Description	Dailed information
<b>SequenceCall</b>	Used to create a hierarchy structure	<a href="#">SequenceCall</a>
<b>NumericLimit</b>	Numeric limit test	<a href="#">NumericLimit</a>
<b>StringValue</b>	String value test	<a href="#">StringValue</a>
<b>PassFail</b>	Pass/Fail test	<a href="#">PassFail</a>
<b>Chart</b>	Graphs with data series	<a href="#">Chart</a>
<b>CallExe</b>	Logs data from a call .exe step	<a href="#">CallExe</a>
<b>Attachment</b>	Attachments such as README, images, etc.	<a href="#">Attachment</a>
<b>MessagePopup</b>	Logs data from a message popup step	<a href="#">MessagePopup</a>
<b>Action</b>	Used to log test action data and information	<a href="#">Action</a>

## SEQUENCE STEPS

Sequence steps are steps which contains a sequence call. Sequence calls are used to create hierarchy structured data. A sequence step can have the following attributes:

Property Name	Description	Data Format	Importance
<b>name</b>	The name of the sequence.	String (200)	Optional
<b>version</b>	The version of the sequence file	String (30)	Optional
<b>path</b>	The path to the sequence file	String (500)	Optional

---

## IMPLEMENTATION EXAMPLE FOR SEQUENCE CALLS

```
{
  "id": 3,
  "group": "M",
  "stepType": "SequenceCall",
  "interactiveExeNum": null,
  "name": "NI steps (NI seq call)",
  "start": "2019-10-15T11:22:27.636+02:00",
  "status": "F",
  "errorCode": 0,
  "errorMessage": null,
  "tsGuid": "ID#:+u20X0V9vkGTSuFPK0137D",
  "totTime": 18.768614,
  "causedSeqFailure": true,
  "causedUUTFailure": false,
  "reportText": null,
  "steps": [
    {
      "id": 4,
      [...]
      "reportText": null
    },
    {
      "id": 5,
      [...]
      "booleanMeas": [
        {
          "name": null,
          "status": "P"
        }
      ]
    }
  ]
},
"seqCall": {
  "path": "Z:\\Test and Demo\\FAT-SAT\\TS2016\\SeqFiles\\TS2016 - FATv1.seq",
  "name": "NI steps",
  "version": null
}
}
```



**TEST STEPS**

A step containing a measurement is defined as a test step. There are three test step types:

- NumericLimitTest
- StringValueTest
- PassFailTest

A step with one of these types must contain the corresponding measurement property.

Test steps comes in two different forms:

1. Single steps (one measurement) (*recommended*). Do **NOT** use the name property.
2. Multiple steps (two or more measurements). The name property **MUST** be set and have a unique name within the step.

Test steps must have status attribute filled in (Passed/Failed). For single steps it is also possible to use Skipped, Done, Terminated and Error.

This table explains the proper keyword to use for single and multiple steps.

Type	Keyword (single)	Keyword (multiple)
Numeric Limit Test	StepType="ET_NLT"	StepType="ET_MNLT"
String Value Test	StepType="ET_SVT"	StepType="ET_MSVT"
Pass Fail Test	StepType="ET_PFT"	StepType="ET_MPFT"

## NUMERIC MEASUREMENT TEST

Numeric tests are used to measure numeric values. Numeric limit tests are defined with a value and a lower and/or higher limit. There is also a comparison operator to specify how to compare the measurement to the limit(s).

Property Name	Description	Data Format	Importance
<b>value</b>	The measured value.	Number	Required
<b>compOp</b>	Which comparison operation to use on the measures value.	String (10) <a href="#">See Comparison Operations</a>	Required
<b>status</b>	The status of the test.	String (1) <a href="#">See Status Codes</a>	Required
<b>name</b>	The name of the limit test. Only use for multiple measurements.	String (100)	Forbidden if single test in step. Required & Unique if multiple tests in same step.
<b>lowLimit</b>	The limit used to compare the measured value in greater than and equal/not equal comparisons.	Number	Optional / Required if comparison operation is GE, EQ, or NE
<b>highLimit</b>	The high limit used to compare measured value in less than comparisons.	Number	Optional / Required if Compare Operator is LE
<b>unit</b>	The unit of the measured value.	String (20)	Optional

## IMPLEMENTATION EXAMPLE

```
{
  "id": 72,
  "group": "M",
  "stepType": "ET_NLT",
  "interactiveExeNum": null,
  "name": "Numeric Limit Test",
  "start": "2019-10-15T11:22:46.604+02:00",
  "status": "P",
  "errorCode": 0,
  "errorMessage": null,
  "tsGuid": "ID#:F4CrB6yGQEChoY6s9p4QyB",
  "totTime": 0.0443013,
  "causedSeqFailure": null,
  "causedUUTFailure": false,
  "reportText": null,
  "numericMeas": [
    {
      "compOp": "GELE",
      "name": null,
      "status": "P",
      "unit": null,
      "value": 0.0,
      "highLimit": 10.0,
      "lowLimit": 0.0
    }
  ]
}
```

### STRING MEASUREMENT TEST

String tests are used to compare text. String tests are defined with a value and a limit. There is also a comparison operator to specify how to compare the text to the limit.

Property Name	Description	Data Format	Importance
<b>value</b>	The text to compare.	String(100)	Required
<b>compOp</b>	Which comparison operation to use on the text.	String (10) <a href="#">See Comparison Operations</a>	Required
<b>status</b>	The status of the test.	String (1) <a href="#">See Status Codes</a>	Optional
<b>name</b>	The name of the measurement test. Only use for multiple measurements.	String (100)	Forbidden if single test in step. Required & Unique if multiple tests in same step.
<b>limit</b>	The text to compare against.	String (100)	Required / Optional if Comparison Operator is Logging

### IMPLEMENTATION EXAMPLE

```
{
  "id": 74,
  "group": "M",
  "stepType": "ET_SVT",
  "interactiveExeNum": null,
  "name": "String Value Test",
  "start": "2019-10-15T11:22:46.696+02:00",
  "status": "P",
  "errorCode": 0,
  "errorMessage": null,
  "tsGuid": "ID#:0iGa1tIVIEqgfYvRXkWwyD",
  "totTime": 0.0023617,
  "causedSeqFailure": null,
  "causedUUTFailure": false,
  "reportText": null,
  "stringMeas": [
    {
      "compOp": "IgnoreCase",
      "name": null,
      "status": "P",
      "value": "this is a string result",
      "limit": "this is a string result"
    }
  ]
}
```

### BOOLEAN MEASUREMENT TEST

Boolean or pass/fail tests are tests without measurements. They only have a status.

Property Name	Description	Data Format	Importance
<b>status</b>	The status of the test.	String (1) <a href="#">See Status Codes</a>	Required
<b>name</b>	The name of the measurement test. Only use for multiple measurements.	String (100)	Forbidden if single test in step. Required & Unique if multiple tests in same step.

### IMPLEMENTATION EXAMPLE

```

{
  "id": 76,
  "group": "M",
  "stepType": "ET_PFT",
  "interactiveExeNum": null,
  "name": "Pass/Fail Test",
  "start": "2019-10-15T11:22:47.754+02:00",
  "status": "P",
  "errorCode": 0,
  "errorMessage": null,
  "tsGuid": "ID#:WV00LdKgZEii64JAako6vC",
  "totTime": 0.0035034,
  "causedSeqFailure": null,
  "causedUUTFailure": false,
  "reportText": null,
  "booleanMeas": [
    {
      "name": null,
      "status": "P"
    }
  ]
}

```

---

## ACTION STEPS

Used to log test action data and information. These steps are always single steps, never multiple.

---

## IMPLEMENTATION EXAMPLE

```
{
  "id": 71,
  "group": "M",
  "stepType": "Action",
  "interactiveExeNum": null,
  "name": "WATS test steps",
  "start": "2019-10-15T11:22:46.563+02:00",
  "status": "D",
  "errorCode": 0,
  "errorMessage": null,
  "tsGuid": "ID#:Kw7PDR4uKEuNqK6qbHI9CA",
  "totTime": 0.0009073,
  "causedSeqFailure": null,
  "causedUUTFailure": false,
  "reportText": null
}
```

## CHART

Charts can be added to empty steps, or in addition to tests.

Property Name	Description	Data Format	Importance
<b>chartType</b>	The type of chart.	String (30) (Line, LineLogXY, LineLogX, LineLogY)	Optional
<b>label</b>	The name of the chart	String (100)	Optional
<b>xLabel</b>	The name of the X axis	String (50)	Optional
<b>xUnit</b>	The unit of the X axis	String (20)	Optional
<b>yLabel</b>	The name of the Y axis	String (50)	Optional
<b>yUnit</b>	The unit of the Y axis	String (20)	Optional

## CHART SERIES

A Series element (or XY-plot) contains a semicolon separated list of x-values and y-values. This version only supports the XYG datatype. Each chart can have up to 10 series.

- YData and XData can't have more than 10000 entries each.
- The number of elements in Y and X must match.

Property Name	Description	Data Format	Importance
<b>name</b>	Name of plot.	String (100)	Required
<b>xdata</b>	All data for x-axis.	String (Semicolon-separated list)	Required
<b>ydata</b>	All data for y-axis.	String (Semicolon-separated list)	Required
<b>dataType</b>	Attribute (only XYG supported)	String (10)	Optional.

## IMPLEMENTATION EXAMPLE

```
"chart": {  
  "chartType": "Line",  
  "label": "Clabel",  
  "xLabel": "Xlabel",  
  "xUnit": "Xunit",  
  "yLabel": "Ylabel",  
  "yUnit": "Yunit",  
  "series": [  
    {  
      "dataType": "XYG",  
      "name": "PlotName 0",  
      "xdata": "0;1;2;3"  
      "ydata": "0.505966492022607;0.64645051547995;0.639659061179134;0.911752170787661"  
    }  
  ]  
}
```



## ATTACHMENT

Attachments can be included in a step by adding an Attachment object to a step. WSJF uses Base64 encoded contents to attach a file.

Property Name	Description	Data Format	Importance
<b>name</b>	The name for the attachment.	String (100)	Required
<b>contentType</b>	The Mime-type of the attachment.	String (100)	Required.
<b>data</b>	A base64 encoded string of the contents of the file.	String	Optional.

## IMPLEMENTATION EXAMPLE

```
"attachment": {
  "name": "stepdata.xml",
  "contentType": "text/xml",
  "data": "PFRoaXMgaXMgdGhlfhNTCBzdHJpbmcgXD4="
}
```

## MESSAGE POP UP

Message popup is used when a pop-up message is displayed.

Property Name	Description	Data Format	Importance
<b>button</b>	The code for which button was pressed.	Integer	Optional
<b>response</b>	The message from the pop up.	String (100)	Optional

## IMPLEMENTATION EXAMPLE

```
"messagePopup": {
  "response": null,
  "button": 2
}
```

## CALL EXECUTABLE

Step which contains a reference to an executable.

Property Name	Description	Data Format	Importance
<b>exitCode</b>	The exit code from the executable.	Number	Required

## IMPLEMENTATION EXAMPLE

```
"callExe": {
  "exitCode": 0
}
```

## ADDITIONAL RESULTS

A step can have additional results, which is a free-form structure of properties and values.

Property Name	Description	Data Format	Importance
<b>name</b>	Name of additional data.	String	Required
<b>props</b>	Properties of additional data	Array of AdditionalDataProperty objects	Required

## COMPARISON OPERATORS

The table below describes the different Comparison operators for test types.

CompOperator	Description	Step type
EQ	Equal	Numeric, String
NE	Not Equal	Numeric, String
GT	Greater than	Numeric, String
LT	Less than	Numeric, String
GE	Greater or equal	Numeric, String
LE	Less or equal	Numeric, String
GTLT	Greater than low limit, less than high limit	Numeric
GELE	Greater or equal than low limit, less or equal than high limit	Numeric
GELT	Greater or equal than low limit, less than high limit	Numeric
GTLE	Greater than low limit, less or equal than high limit	Numeric
LTGT	Less than low limit, greater than high limit	Numeric
LEGE	Less or equal than low limit, greater or equal than high limit	Numeric
LEGT	Less or equal than low limit, greater than high limit	Numeric
LTGE	Less than low limit, greater or equal than high limit	Numeric
LOG	Logging values (no comparison)	Numeric, String
CASESENSIT	Case sensitive	String
IGNORECASE	Ignore case / !CASESENSIT	String

**STATUS CODES**

Below is a list of supported status codes for a step.

Status	Code	UUT description	Step / measurement description	Additional information
Passed	P	UUT passed	Step passed	
Failed	F	UUT failed	Step failed	
Done	D	Not used.	Step completed.	Can be used for single steps, should not be used for multiple steps
Skipped	S	Not used.	Step execution skipped.	Can be used for single steps, with caution for multiple steps.
Error	E	An error occurred during test execution	An error occurred during step execution.	Can be used for single steps, should not be used for multiple steps.
Terminated	T	Operator terminated test execution	Operator terminated step execution.	Can be used for single steps, should not be used for multiple steps.

The UUT report itself can be Passed, Failed, Error or Terminated.

A single step can have all statuses.

A single measurement (in a multiple step) can only be Passed or Failed or Skipped.