ZELTWANGER
DICHTHEITSPRÜFUNG

Extended Documentation

Web API & SignalR

Additional Information

Target Audiences for the Manual

This manual is directed towards the following target audiences:

› Set-up technicians
› Maintenance personnel

This manual includes additional information for the following target audiences:

› Licensed electricians
› Personnel with expertise in pneumatic systems
› Manufacturer service personnel
› Personnel trained in packaging and transportation

Copyright

Table of Contents

1 Web API 6

1.1 Configure the Web API server 6
1.1.1 Active 6
1.1.2 Port 6
1.1.3 Ethernet connection 6
1.2 Available methods 7
1.2.1 Get the online state of the server 7
1.2.2 Get the current user 7
1.2.3 Get the available measuring programs 8
1.2.4 Start a measuring program 9
1.2.5 Start a dynamic measuring program 10
1.2.6 Stop the current measurement 11
1.2.7 Get the status of the measuring channel 11
1.2.8 Check the additional channel state 12
1.2.9 Check for channel error 13
1.2.10 Get live measuring values 14
1.2.11 Get custom live measuring values 15
1.2.12 Get the test results 16
1.2.13 Check if the measuring results are available 16
1.2.14 Get the measuring results 17
1.2.15 Get single measuring results 18
1.2.16 Get Program 19
1.2.17 Get a program parameter 20
1.2.18 Set a program parameter 21
1.2.19 Set multiple program parameters 22
1.2.20 Set external program ID 23
1.2.21 Set program name 23
1.2.22 Create new measuring program 24
1.2.23 Delete a program 24
1.2.24 Start the system verification 25
1.2.25 Reset the system verification 25
1.2.26 Get a specific value of the system verification 26
1.2.27 Get the device informations 27
1.2.28 Get the default program parameters 28
1.2.29 Acknowledge NOK Result 29

1.2.30 Check for needed NOK Acknowledgement 29
1.2.31 Get Charts 30
1.2.32 Get the last error 31
1.2.33 Get the last error of the measuring channel 31

2 SignalR 32

2.1 Configure the SignalR server 32
2.1.1 Active 32
2.1.2 Port 32
2.1.3 Ethernet Anschluss 32
2.2 Available methods 33
2.2.1 Get the online state of the server 33
2.2.2 Get the current user 33
2.2.3 Get the available measuring programs 34
2.2.4 Start a measuring program 35
2.2.5 Start a dynamic measuring program 36
2.2.6 Stop the current measurement 37
2.2.7 Get the status of the measuring channel 37
2.2.8 Check the additional channel state 38
2.2.9 Check for channel error 39
2.2.10 Get live measuring values 40
2.2.11 Get custom live measuring values 41
2.2.12 Get the test results 42
2.2.13 Check if the measuring results are available 42
2.2.14 Get the measuring results 43
2.2.15 Get single measuring results 44
2.2.16 Get Program 45
2.2.17 Get a program parameter 46
2.2.18 Set a program parameter 46
2.2.19 Set multiple program parameters 47
2.2.20 Set external program ID 48
2.2.21 Set program name 48
2.2.22 Create new measuring program 49
2.2.23 Delete a program 49
2.2.24 Start the system verification 50
2.2.25 Reset the system verification 50
2.2.26 Get a specific value of the system verification 51

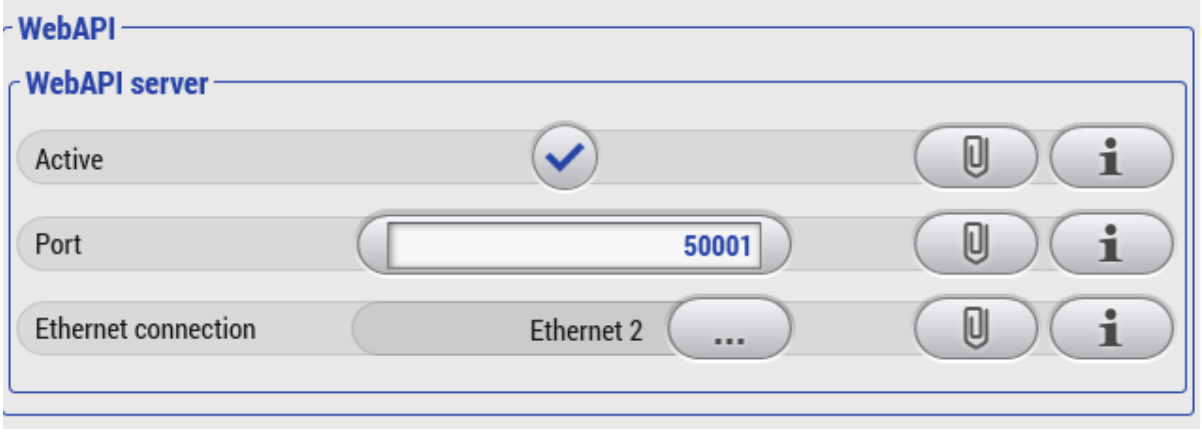
2.2.27 Get the device informations ... 52
2.2.28 Get the default program parameters ... 53
2.2.29 Acknowledge NOK Result ... 54
2.2.30 Check for needed NOK Acknowledgement ... 54
2.2.31 Get Charts ... 55
2.2.32 Get the last error ... 56
2.2.33 Get the last error of the measuring channel ... 56

3 Measuring program ... 57

3.1 Program header ... 57
3.2 Measuring parameters ... 57
3.2.1 Process parameters ... 57
3.2.2 Pressure change detection ... 60
3.2.3 Flow change detection ... 61
3.2.4 Limits ... 62
3.2.5 Offset ... 64
3.2.6 Temperature compensation ... 64
3.2.7 Process monitor ... 65
3.2.8 Ramps ... 67
3.2.9 Units ... 67
3.2.10 Testing configs ... 68
3.3 Enumerations ... 68
3.3.1 Channel mode ... 68
3.3.2 Measuring type ... 69
3.3.3 Venting mode ... 69
3.3.4 Temperature check mode ... 70
3.3.5 Flow ... 70
3.3.6 Pressure ... 70
3.3.7 Volume ... 70
3.3.8 Temperature ... 71
3.3.9 Time ... 71
3.3.10 Dimensionless Quantity ... 71

1 Web API

1.1 Configure the Web API server



1.1.1 Active

The Active parameter must be checked to enable the Web API interface.

1.1.2 Port

The Port parameter contains the port which will be used for the Web API connection and communication.

1.1.3 Ethernet connection

The Ethernet connection parameter defines the ethernet port which will be used for the Web API connection and communication.

1.2 Available methods

The Web API server offers various methods, e.g. to start a measurement or to get specific results from the last measurement.

All command calls follow the same structure which looks like this:
http://{IP address of the server}:{PORT}/api/zed/{METHOD}/*{PARAMETER}*
(For Example: http://127.0.0.1:50001/api/zed/start/1)
(The Value PARAMETER is not needed for this and some other methods.)

In addition, there is the possibility to request all available methods using the following call:
http://{IP address of the server}:{PORT}/api/zed/

1.2.1 Get the online state of the server

GetOnlineState	
Method name	getOnlineState
HTTP request method	Get
Return value	Boolean *(True = Server is online,* *False = Server is offline)*
Description	Returns if the Web API server is online or not.
Example call	http://127.0.0.1:50001/api/zed/getOnlineState/
Availability	Since software version 4.1.13.0

1.2.2 Get the current user

GetCurrentUser	
Method name	getCurrentUser
HTTP request method	Get
Return value	Text *(Name of the current user)*
Description	Returns the name of the user which is currently logged in on the leak testing device.
Example call	http://127.0.0.1:50001/api/zed/getCurrentUser/
Availability	Since software version 4.1.13.0

1.2.3 Get the available measuring programs

EnumeratePrograms	
Method name	enumeratePrograms
HTTP request method	Get
Return value	List of program objects.
Description	Returns all measuring programs which are available on the leak testing device.
Example call	http://127.0.0.1:50001/api/zed/enumeratePrograms/
Availability	Since software version 4.1.13.0

1.2.3.1 Program list example

This is an example of a result which could be returned from the EnumaratePrograms call.


```
{
  "Programs": [
    {
      "ExternalID": 2,
      "ChannelID": 1,
      "ProgramName": "Program 1",
      "Description": "",
      "ProgramType": "MeasuringProgram",
      "CreationTime": "2019-10-01T11:32:00",
      "LastChange": "2019-10-01T11:32:00"
    },
    {
      "ExternalID": 1,
      "ChannelID": 1,
      "ProgramName": "Selftest",
      "Description": "",
      "ProgramType": "MeasuringProgram",
      "CreationTime": "2019-09-19T09:11:18",
      "LastChange": "2019-09-19T09:11:18"
    }
  ]
}
```

1.2.4 Start a measuring program

Start	
Method name	start
HTTP request method	Post
Body content	Start information object.
Return value	Boolean *(True = Measurement successfully started,* *False = Could not start the measurement)*
Description	Starts a measuring with the program defined in the start information object.
Example call	http://127.0.0.1:50001/api/zed/start/
Availability	Since software version 4.1.13.0

1.2.4.1 Start information object example

This is an example for the start information object which is needed for the start call.

```
{
  "ChannelID": 1,
  "ExternalID": 1,
  "MeasuringMode": "LeakTest",
  "SerialNumber": ""
}
```

The SerialNumber field does only need to be set e.g. if the serialnumber is wanted in the result files of the leak testing device.

1.2.5 Start a dynamic measuring program

StartDynamicProgram	
Method name	startDynamicProgram
HTTP request method	Post
Body content	Measuring program object.
Return value	Boolean *(True = Measurement successfully started,* *False = Could not start the measurement)*
Description	Starts a measuring with the program defined in the measuring program object.
Example call	http://127.0.0.1:50001/api/zed/startDynamicProgram/
Availability	Since software version 4.1.13.0

1.2.5.1 Measuring program example

This is an example of a measuring program object of the StartDynamicProgram call:

```
{
  "ChannelID": 1,
  "MeasuringMode": "LeakTest",
  "MeasuringType": "PressureChangeGaugeLeakage",
  "ProgramName": "Program",
  "TestingParameters": [
    {
      "Name": "Phase.PreFilling",
      "Value": "3"
    },
    {
      "Name": "Phase.Filling",
      "Value": "3"
    }
  ]
}
```

The TeastingParameters list can be filled with every parameter that should differ from the default settings.

1.2.6 Stop the current measurement

Stop	
Method name	stop
HTTP request method	Post
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Boolean *(True = Measurement successfully stopped,* *False = Could not stop the measurement)*
Description	Stops the measurement running on the selected measuring channel.
Example call	http://127.0.0.1:50001/api/zed/stop/1
Availability	Since software version 4.1.13.0

1.2.7 Get the status of the measuring channel

GetChannelState	
Method name	getChannelState
HTTP request method	Get
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Text
Description	Returns the current state of the selected measuring channel.
Example call	http://127.0.0.1:50001/api/zed/getChannelState/1
Availability	Since software version 4.1.13.0

1.2.7.1 Return values

Value	Description
Initializing	The measuring channel is initializing.
WaitingForStart	The measuring channel waits for a start command.
Started	The measuring channel is currently running a measurement.
Paused	The measurement on this measuring channel got paused.
Stopped	The measurement on this measuring channel got stopped.
Finished	The measurement on this measuring channel is finished.

1.2.8 Check the additional channel state

CheckChannelAdditionalState	
Method name	checkChannelAdditionalState
HTTP request method	Post
Body content	Channel addition state information object.
Return value	Boolean *(True = The state is set,* *False = The state is not set)*
Description	Returns if the selected additional state is set on the measuring channel.
Example call	http://127.0.0.1:50001/api/zed/checkChannelAdditionalState/
Availability	Since software version 4.1.13.0

1.2.8.1 Channel additional state information object example

This is an example of a channel addition state information object:

```
{
  "ChannelID": 1,
  "ChannelAdditionalState": "MinLimit2NIO"
}
```

1.2.8.2 Available states of the ChannelAddidtionState field

Value	Description
MinLimit1NIO	The first lower NOK limit. (Fine leak)
MaxLimit1NIO	The first upper NOK limit. (Fine leak)
MinLimit2NIO	The second lower NOK limit. (Gross leak)
MaxLimit2NIO	The second upper NOK limit. (Gross leak)
SystemVerificationRequired	The system verification is required.
SystemVerificationActive	The system verification is active.

1.2.9 Check for channel error

CheckChannelError	
Method name	checkChannelError
HTTP request method	Post
Body content	Channel error information object.
Return value	Boolean *(True = The error is set,* *False = The error is not set)*
Description	Returns if the selected error is set on the measuring channel.
Example call	http://127.0.0.1:50001/api/zed/checkChannelError/
Availability	Since software version 4.1.13.0

1.2.9.1 Channel error information object example

This is an example of a channel error information object:

```
{
  "ChannelID": 1,
  "ChannelError": "NOK2"
}
```

1.2.9.2 Available states of the ChannelError field

Value	Description
ProgramNotFound	No program was found with the specified parameters.
SupplyAirNOK	The last measurement failed because of missing or to low air pressure.
NOK1	The test item is not tight. (Fine leak)
NOK2	The test item is not tight. (Gross leak)
ChannelIsNotAvailable	There is no measuring channel with the selected id.
SystemVerificationBlocked	The program is blocked because of a failed system verification.

1.2.10 Get live measuring values

GetMeasuringLiveValues	
Method name	getMeasuringLiveValues
HTTP request method	Get
Parameter	Positive integer *(ID of the measuring channel)*
Return value	List of live values.
Description	Returns the live values of the current measurement on the selected channel.
Example call	http://127.0.0.1:50001/api/zed/getMeasuringLiveValues/1
Availability	Since software version 4.1.13.0

1.2.10.1 Live values example

This is an example of a list of live values returned by the GetMeasuringLiveValues call:


```
{
  "CurrentPhase": "Measuring",
  "RemainingRunTime": 12,
  "Value1": -28.438550000006217,
  "Value2": 0.0001688551243637858
}
```

1.2.11 Get custom live measuring values

GetCustomMeasuringLiveValues	
Method name	getCustomMeasuringLiveValues
HTTP request method	Get
Parameter	Positive integer *(ID of the measuring channel)*
Return value	List of live values.
Description	Returns the live values of the current measurement on the selected channel. The selection of the values depends on the selected GUI fields on the device.
Example call	http://127.0.0.1:50001/api/zed/getMeasuringLiveValues/1
Availability	Since software version 4.3.5.0

1.2.11.1 Live values example

This is an example of a list of live values returned by the GetCustomMeasuringLiveValues call:


```
{
  "MeasuringLiveValues": [
    {
      "Name": "PreFillingAndFillingPressure",
      "Value": "480000"
    },
    {
      "Name": "TestingPressure",
      "Value": "479520"
    },
    {
      "Name": "SensorPressure",
      "Value": "479492.1525125"
    },
    {
      "Name": "PressureChange",
      "Value": "-27.8474874999956"
    },
    {
      "Name": "CorrectionValue",
      "Value": "0"
    },
    {
      "Name": "Result",
      "Value": "0.000168882422841186"
    },
    {
      "Name": "Quantity",
      "Value": "12"
    },
    {
      "Name": "QuantityOk",
      "Value": "2"
    }
  ]
}
```

1.2.12 Get the test results

GetTestResult	
Method name	getTestResult
HTTP request method	Get
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Text
Description	Returns the result of the last measurement on the selected channel.
Example call	http://127.0.0.1:50001/api/zed/getTestResult/1
Availability	Since software version 4.1.13.0

1.2.12.1 Return values

Value	Description
Undefined	The result is undefined.
Aborted	The measurement got aborted by a user.
OK	The measurement was ok.
NOK	The measurement was not ok.
NoResult	No result available for this measuring channel.
Error	An error has occurred.

1.2.13 Check if the measuring results are available

MeasuringResultsAvailable	
Method name	measuringResultsAvailable
HTTP request method	Get
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Boolean *(True = The measuring result are available,* *False = The measuring results are not available)*
Description	Returns if the measuring results of the last measurement on the selected channel are available.
Example call	http://127.0.0.1:50001/api/zed/measuringResultsAvailable/1
Availability	Since software version 4.1.13.0

1.2.14 Get the measuring results

GetMeasuringResults	
Method name	getMeasuringResults
HTTP request method	Get
Parameter	Positive integer *(ID of the measuring channel)*
Return value	List of measuring result objects.
Description	Returns a list of measuring result objects of the selected channel. The return value is an empty string if the measuring results are not available. To ensure that the results are available use the call MeasuringResultsAvailable.
Example call	http://127.0.0.1:50001/api/zed/getMeasuringResults/1
Availability	Since software version 4.1.13.0

1.2.14.1 Measuring result object list example

This is an example of a list of measuring result objects returned from the GetMeasuringResults call:


```
{
  "MeasuringResults": [
    {
      "Name": "StartTime",
      "Value": "28-10-2019 08:53:50"
    },
    {
      "Name": "ProgramName",
      "Value": "Program 1"
    },
    {
      "Name": "Result",
      "Value": "OK"
    },
    {
      "Name": "ResultUnit",
      "Value": "Pa*m³/s"
    }
  ]
}
```


Which and how many measurement results are returned in the list depends on the setting of the leak test device. The list is based on the template that e.g. the .CSV results are based on.

1.2.14.2 Default layout

There also exists a default layout of the measuring result list which can be used.
This layout can be used by changing the method name from GetMeasuringResults to GetMeasuringResultsDefaultLayout.

The default layout looks like this:


```
{
  "MeasuringResults": [
    {
      "Name": "StartTime",
      "Value": "28-10-2019 08:53:50"
    },
    {
      "Name": "SerialNumber",
      "Value": ""
    },
    {
      "Name": "Result",
      "Value": "OK"
    },
    {
      "Name": "ResultValue",
      "Value": "0,000146745782278802"
    },
    {
      "Name": "ResultUnit",
      "Value": "Pa*m³/s"
    }
  ]
}
```


1.2.15 Get single measuring results

GetMeasuringResult	
Method name	getMeasuringResult
HTTP request method	Post
Body content	Measuring result information object.
Return value	Text *(The value of the requested measuring result)*
Description	Returns a specific measuring result of the last measurement on the selected channel.
Example call	http://127.0.0.1:50001/api/zed/getMeasuringResult/
Availability	Since software version 4.1.13.0

1.2.15.1 Measuring result information object example

This is an example of a measuring result information object for the GetMeasuringResult call:


```
{
  "ChannelID": 1,
  "ResultName": "StartTime"
}
```

1.2.16 Get Program

GetProgram	
Method name	getProgram
HTTP request method	Post
Body content	Program information object.
Return value	Program object
Description	Returns the program with the selected external id and channel.
Example call	http://127.0.0.1:50001/api/zed/getProgram/
Availability	Since software version 4.1.13.0

1.2.16.1 Program information object example

This is an example of the program information object used for the GetProram call:

```
{
  "ChannelID": 1,
  "ExternalID": 2
}
```

1.2.16.2 Program object example

This is an example of a program objects returned from the GetProgram call:

```
{
  "MeasuringType": "PressureChangeGaugeLeakage",
  "DimensionType": "Leakrate",
  "Header": {
    "ExternalID": 3,
    "ChannelID": 1,
    "ProgramName": "Programm 1",
    "Description": "",
    "ProgramType": "MeasuringProgram",
    "CreationTime": "2020-11-10T14:34:55",
    "LastChange": "2020-11-16T13:58:52"
  },
  "Parameters": [
    {
      "Name": "NOK.Active",
      "Value": "True"
    },
    {
      "Name": "NOK.AllowedRepetitions",
      "Value": "1"
    },
    {
      "Name": "DefaultUnits.Active",
      "Value": "True"
    },
    ...
  ]
}
```

1.2.17 Get a program parameter

GetProgramParameter	
Method name	getProgramParameter
HTTP request method	Post
Body content	Program parameter information object.
Return value	Text *(Value of the program parameter)*
Description	Returns the value of the selected parameter of the program with the selected external id.
Example call	http://127.0.0.1:50001/api/zed/getProgramParameter/
Availability	Since software version 4.1.13.0

1.2.17.1 Program parameter information object example

This is an example of the program parameter information object used for the GetProramParameter call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "ParameterName": "Phase.PreFilling",
  "Value": ""
}
```

1.2.18 Set a program parameter

SetProgramParameter	
Method name	setProgramParameter
HTTP request method	Post
Body content	Program parameter information object.
Return value	Boolean *(True = The parameter was set successfully,* *False = The parameter could not be set successfully)*
Description	Sets the selected parameter of the program with the selected external id to the given value.
Example call	http://127.0.0.1:50001/api/zed/setProgramParameter/
Availability	Since software version 4.1.13.0

1.2.18.1 Program parameter information object example

This is an example of the program parameter information object used for the SetProramParameter call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "ParameterName": "Phase.PreFilling",
  "Value": "3"
}
```

1.2.19 Set multiple program parameters

SetProgramParameters	
Method name	setProgramParameters
HTTP request method	Post
Body content	Program parameters information object.
Return value	Boolean *(True = The parameters were set successfully,* *False = The parameters could not be set successfully)*
Description	Sets the selected parameters of the program with the selected external id to the given values.
Example call	http://127.0.0.1:50001/api/zed/setProgramParameters/
Availability	Since software version 4.3.26.0

1.2.19.1 Program parameters information object example

This is an example of the program parameters information object used for the SetProramParameters call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "Parameters":
  [
    { "Name": "Pressure.Filling", "Value": "60000" },
    { "Name": "Pressure.PreFilling", "Value": "60000" }
  ]
}
```

1.2.20 Set external program ID

SetProgramExternalId	
Method name	setProgramExternalId
HTTP request method	Post
Body content	SetProgramExternalId information object.
Return value	Boolean *(True = The external Id was set successfully,* *False = The external Id could not be set successfully)*
Description	Sets the external Id of the program with the selected external id to the given value.
Example call	http://127.0.0.1:50001/api/zed/setProgramExternalId/
Availability	Since software version 4.3.45.0

1.2.20.1 SetProgramExternalId information object example

This is an example of the SetProgramExternalId information object used for the SetProgramExternalId call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "NewExternalID": 3
}
```

1.2.21 Set program name

SetProgramName	
Method name	setProgramName
HTTP request method	Post
Body content	SetProgramName information object.
Return value	Boolean *(True = The program name was set successfully,* *False = The program name could not be set successfully)*
Description	Sets the name of the program with the selected external id to the given value.
Example call	http://127.0.0.1:50001/api/zed/setProgramName/
Availability	Since software version 4.3.45.0

1.2.21.1 SetProgramName information object example

This is an example of the SetProgramName information object used for the SetProgramName call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "ProgramName": "Leaktest program"
}
```

1.2.22 Create new measuring program

CreateMeasuringProgram	
Method name	createMeasuringProgram
HTTP request method	Post
Body content	CreateMeasuringProgram information object.
Return value	Boolean *(True = The program was created successfully,* *False = The program could not be created successfully)*
Description	Creates a new measuring program.
Example call	http://127.0.0.1:50001/api/zed/createMeasuringProgram/
Availability	Since software version 4.3.45.0

1.2.22.1 CreateMeasuringProgram information object example

This is an example of the CreateMeasuringProgram information object used for the CreateMeasuringProgram call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "MeasuringType": "PressureChangeGaugeLeakage",
  "ProgramName": "Leaktest program"
}
```

1.2.23 Delete a program

DeleteProgram	
Method name	deleteProgram
HTTP request method	Post
Body content	DeleteProgram information object.
Return value	Boolean *(True = The program was deleted successfully,* *False = The program could not be deleted successfully)*
Description	Deletes an existing program.
Example call	http://127.0.0.1:50001/api/zed/deleteProgram/
Availability	Since software version 4.3.45.0

1.2.23.1 DeleteProgram information object example

This is an example of the DeleteProgram information object used for the DeleteProgram call:

```
{
  "ChannelID": 1,
  "ExternalID": 2
}
```

1.2.24 Start the system verification

StartSystemVerification	
Method name	startSystemVerification
HTTP request method	Post
Body content	System verification information object.
Return value	Boolean *(True = The system verification started successfully,* *False = The system verification could not be started)*
Description	Starts the system verification for the program with the selected external id.
Example call	http://127.0.0.1:50001/api/zed/startSystemVerification/
Availability	Since software version 4.1.13.0

1.2.24.1 System verification information object example

This is an example of the system verification information object used for the StartSystemVerification and ResetSystemVerification call:

```
{
  "ChannelID": 1,
  "ExternalID": 2
}
```

1.2.25 Reset the system verification

ResetSystemVerification	
Method name	resetSystemVerification
HTTP request method	Post
Body content	System verification information object.
Return value	Boolean *(True = The system verification got successfully reseted,* *False = The system verification could not be reseted)*
Description	Resets the system verification of the program with the selected external id and returns if it worked. This call is used if a program is blocked by a failed system verification.
Example call	http://127.0.0.1:50001/api/zed/resetSystemVerification/
Availability	Since software version 4.1.13.0

1.2.26 Get a specific value of the system verification

GetSystemVerificationValue	
Method name	getSystemVerificationValue
HTTP request method	Post
Body content	System verification value information object.
Return value	Text (The value of the selected system verification value)
Description	Returns the value of the selected system verification value for the last system verification of the program with the selected external id.
Example call	http://127.0.0.1:50001/api/zed/getSystemVerificationValue/
Availability	Since software version 4.1.13.0

1.2.26.1 System verification value information object example

This is an example of the system verification value information object used for the GetSystemVerificationValue call:

```
{
  "SystemVerificationValue": "DeviationOfTestleak",
  "ChannelID": 1,
  "ExternalID": 2
}
```

1.2.26.2 Available values of the SystemVerificationValue field

Value	Description
DifferenceValue	The difference between the measurements with and without a leak.
DeviationOfTestleak	The deviation from the set test leak.

1.2.27 Get the device informations

GetDeviceInformation	
Method name	getDeviceInformation
HTTP request method	Get
Return value	Device information object.
Description	Returns the most important information about the device and it's measuring channels.
Example call	http://127.0.0.1:50001/api/zed/getDeviceInformation/
Availability	Since software version 4.1.13.0

1.2.27.1 Device information object example

This is an example of the device information object returned from the GetDeviceInformation call:

```
{
  "SerialNumber": "1909001",
  "SoftwareVersion": "4.1.15.2",
  "DeviceType": "ZEDBase",
  "MeasuringChannels": [
    {
      "ChannelID": 1,
      "CalibrationDate": "",
      "ChannelType": "MF",
      "RangeMin": 0.0,
      "RangeMax": 100000.0
    }
  ]
}
```

1.2.28 Get the default program parameters

GetDefaultProgramParameters	
Method name	getDefaultProgramParameters
HTTP request method	Get
Parameter	Positive integer *(ID of the measuring channel)*
Return value	List of program parameter objects.
Description	Returns the default program parameters of the selected measuring channel.
Example call	http://127.0.0.1:50001/api/zed/getDefaultProgramParameters/1
Availability	Since software version 4.1.13.0

1.2.28.1 Program parameter list object example

This is an example of the program parameter list object returned from the GetDefaultProgramParameters call:


```
{
  "MeasuringTypeParameterList": [
    {
      "MeasuringType": "PressureChangeGauge",
      "ProgramParameters": [
        {
          "Name": "NOK.Active",
          "Value": "False"
        },
        {
          "Name": "NOK.AllowedRepetitions",
          "Value": "1"
        },
        {
          "Name": "DefaultUnits.Active",
          "Value": "True"
        },
        {
          "Name": "SafeVenting.PressureLimit",
          "Value": "10000"
        }
      ]
    }
  ]
}
```

1.2.29 Acknowledge NOK Result

NOKAcknowledgeChannel	
Method name	nokAcknowledgeChannel
HTTP request method	Post
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Boolean *(True = The NOK result got acknowledged successfully,* *False = The NOK result could not get acknowledged)*
Description	Acknowledges a NOK Result if needed.
Example call	http://127.0.0.1:50001/api/zed/nokAcknowledgeChannel/1
Availability	Since software version 4.3.13.0

1.2.30 Check for needed NOK Acknowledgement

CheckNOKAcknowledgeNeeded	
Method name	checkNokAcknowledgeChannel
HTTP request method	Get
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Boolean *(True = The NOK acknowledgement is needed,* *False = The NOK acknowledgement is not needed)*
Description	Checks if the NOK Acknowledgement is needed.
Example call	http://127.0.0.1:50001/api/zed/checkNokAcknowledgeNeeded/1
Availability	Since software version 4.3.29.0

1.2.31 Get Charts

GetCharts	
Method name	getCharts
HTTP request method	Get
Parameter	Positive integer *(ID of the measuring channel)*
Return value	List of chart objects.
Description	Returns the charts of the last measurement of the selected measuring channel.
Example call	http://127.0.0.1:50001/api/zed/getCharts/1
Availability	Since software version 4.3.74.0

1.2.31.1 Chart list object example

This is an example of the charts list object returned from the GetCharts call


```
{
  "Charts": [
    {
      "Name": "Normal chart",
      "ChartLines": [
        {
          "Name": "Pressure",
          "XAxisUnit": "s",
          "YAxisUnit": "bar",
          "ChartPoints": [
           {
              "X": "0.03735",
              "Y": "0.0"
           },
           {
              "X": "0.0485",
              "Y": "0.0"
           },
           {
              "X": "0.060724",
              "Y": "5.3384"
           }
          ]
        },
      ]
    }
  ]
}
```

1.2.32 Get the last error

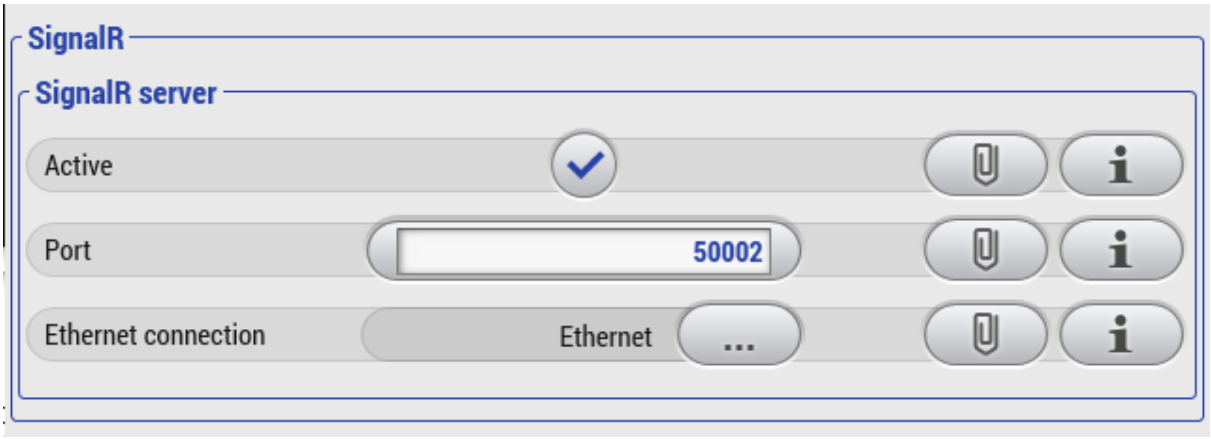
GetLastError	
Method name	getLastError
HTTP request method	Get
Return value	Text *(The last error message)*
Description	Returns the last error of the Web API server. If the text is empty, no error occurred since the start of the server.
Example call	http://127.0.0.1:50001/api/zed/getLastError/
Availability	Since software version 4.1.13.0

1.2.33 Get the last error of the measuring channel

GetLastChannelError	
Method name	getLastChannelError
HTTP request method	Get
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Text *(The last error message of the measuring channel)*
Description	Returns the last error of the Web API server that occurred on the selected measuring channel. If the text is empty, no error occurred on the measuring channel since the start of the server.
Example call	http://127.0.0.1:50001/api/zed/getLastChannelError/1
Availability	Since software version 4.1.13.0

2 SignalR

2.1 Configure the SignalR server



2.1.1 Active

The Active parameter must be checked to enable the SignalR interface.

2.1.2 Port

The Port parameter contains the port which will be used for the SignalR connection and communication.

2.1.3 Ethernet Anschluss

The Ethernet connection parameter defines the ethernet port which will be used for the SignalR connection and communication.

2.2 Available methods

The Web API server offers various methods, e.g. to start a measurement or to get specific results from the last measurement.

All calls are available over the “ZED” hub.

2.2.1 Get the online state of the server

GetOnlineState	
Method name	GetOnlineState
Return value	Boolean *(True = Server is online,* *False = Server is offline)*
Description	Returns if the Web API server is online or not.
Availability	Since software version 4.1.13.0

2.2.2 Get the current user

GetCurrentUser	
Method name	GetCurrentUser
Return value	Text *(Name of the current user)*
Description	Returns the name of the user which is currently logged in on the leak testing device.
Availability	Since software version 4.1.13.0

2.2.3 Get the available measuring programs

EnumeratePrograms	
Method name	EnumeratePrograms
Return value	List of program objects.
Description	Returns all measuring programs which are available on the leak testing device.
Availability	Since software version 4.1.13.0

2.2.3.1 Program list example

This is an example of a result which could be returned from the EnumaratePrograms call:


```
{
  "Programs": [
    {
      "ExternalID": 2,
      "ChannelID": 1,
      "ProgramName": "Program 1",
      "Description": "",
      "ProgramType": "MeasuringProgram",
      "CreationTime": "2019-10-01T11:32:00",
      "LastChange": "2019-10-01T11:32:00"
    },
    {
      "ExternalID": 1,
      "ChannelID": 1,
      "ProgramName": "Selftest",
      "Description": "",
      "ProgramType": "MeasuringProgram",
      "CreationTime": "2019-09-19T09:11:18",
      "LastChange": "2019-09-19T09:11:18"
    }
  ]
}
```

2.2.4 Start a measuring program

Start	
Method name	Start
Parameter	Start information object.
Return value	Boolean *(True = Measurement successfully started,* *False = Could not start the measurement)*
Description	Starts a measuring with the program defined in the start information object.
Availability	Since software version 4.1.13.0

2.2.4.1 Start information object example

This is an example for the start information object which is needed for the start call.

```
{
  "ChannelID": 1,
  "ExternalID": 1,
  "MeasuringMode": "LeakTest",
  "SerialNumber": ""
}
```

The SerialNumber field does only need to be set e.g. if the serialnumber is wanted in the result files of the leak testing device.

2.2.4.2 Listen for the finish of the measurement

The hub offers an event which gets called as soon as the measurement is finished.
To get notified when the measurement is finished, all you must do is listen to the LeaktestFinished event.

2.2.5 Start a dynamic measuring program

StartDynamicProgram	
Method name	StartDynamicProgram
Parameter	Measuring program object.
Return value	Boolean *(True = Measurement successfully started,* *False = Could not start the measurement)*
Description	Starts a measuring with the program defined in the measuring program object.
Availability	Since software version 4.1.13.0

2.2.5.1 Measuring program example

This is an example of a measuring program object of the StartDynamicProgram call:

```
{
  "ChannelID": 1,
  "MeasuringMode": "LeakTest",
  "MeasuringType": "PressureChangeGaugeLeakage",
  "ProgramName": "Program",
  "TestingParameters": [
    {
      "Name": "Phase.PreFilling",
      "Value": "3"
    },
    {
      "Name": "Phase.Filling",
      "Value": "3"
    }
  ]
}
```

The TeastingParameters list can be filled with every parameter that should differ from the default settings.

2.2.5.2 Listen for the finish of the measurement

The hub offers an event which gets called as soon as the measurement is finished.
To get notified when the measurement is finished, all you must do is listen to the LeaktestFinished event.

2.2.6 Stop the current measurement

Stop	
Method name	Stop
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Boolean *(True = Measurement successfully stopped,* *False = Could not stop the measurement)*
Description	Stops the measurement running on the selected measuring channel.
Availability	Since software version 4.1.13.0

2.2.7 Get the status of the measuring channel

GetChannelState	
Method name	GetChannelState
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Text
Description	Returns the current state of the selected measuring channel.
Availability	Since software version 4.1.13.0

2.2.7.1 Return values

Value	Description
Initializing	The measuring channel is initializing.
WaitingForStart	The measuring channel waits for a start command.
Started	The measuring channel is currently running a measurement.
Paused	The measurement on this measuring channel got paused.
Stopped	The measurement on this measuring channel got stopped.
Finished	The measurement on this measuring channel is finished.

2.2.8 Check the additional channel state

CheckChannelAdditionalState	
Method name	CheckChannelAdditionalState
Parameter	Channel addition state information object.
Return value	Boolean *(True = The state is set,* *False = The state is not set)*
Description	Returns if the selected additional state is set on the measuring channel.
Availability	Since software version 4.1.13.0

2.2.8.1 Channel additional state information object example

This is an example of a channel addition state information object:

```
{
  "ChannelID": 1,
  "ChannelAdditionalState": "MinLimit2NIO"
}
```

2.2.8.2 Available states of the ChannelAddidtionState field

Value	Description
MinLimit1NIO	The first lower NOK limit. (Fine leak)
MaxLimit1NIO	The first upper NOK limit. (Fine leak)
MinLimit2NIO	The second lower NOK limit. (Gross leak)
MaxLimit2NIO	The second upper NOK limit. (Gross leak)
SystemVerificationRequired	The system verification is required.
SystemVerificationActive	The system verification is active.

2.2.9 Check for channel error

CheckChannelError	
Method name	CheckChannelError
Parameter	Channel error information object.
Return value	Boolean *(True = The error is set,* *False = The error is not set)*
Description	Returns if the selected error is set on the measuring channel.
Availability	Since software version 4.1.13.0

2.2.9.1 Channel error information object example

This is an example of a channel error information object:

```
{
  "ChannelID": 1,
  "ChannelError": "NOK2"
}
```

2.2.9.2 Available states of the ChannelError field

Value	Description
ProgramNotFound	No program was found with the specified parameters.
SupplyAirNOK	The last measurement failed because of missing or to low air pressure.
NOK1	The test item is not tight. (Fine leak)
NOK2	The test item is not tight. (Gross leak)
ChannelIsNotAvailable	There is no measuring channel with the selected id.
SystemVerificationBlocked	The program is blocked because of a failed system verification.

2.2.10 Get live measuring values

GetMeasuringLiveValues	
Method name	GetMeasuringLiveValues
Parameter	Positive integer *(ID of the measuring channel)*
Return value	List of live values.
Description	Returns the live values of the current measurement on the selected channel.
Availability	Since software version 4.1.13.0

2.2.10.1 Live values example

This is an example of a list of live values returned by the GetMeasuringLiveValues call:

```
{
  "CurrentPhase": "Measuring",
  "RemainingRunTime": 12,
  "Value1": -28.438550000006217,
  "Value2": 0.0001688551243637858
}
```

2.2.11 Get custom live measuring values

GetCustomMeasuringLiveValues	
Method name	getCustomMeasuringLiveValues
Parameter	Positive integer *(ID of the measuring channel)*
Return value	List of live values.
Description	Returns the live values of the current measurement on the selected channel. The selection of the values depends on the selected GUI fields on the device.
Availability	Since software version 4.3.5.0

2.2.11.1 Live values example

This is an example of a list of live values returned by the GetCustomMeasuringLiveValues call:


```
{
  "MeasuringLiveValues": [
    {
      "Name": "PreFillingAndFillingPressure",
      "Value": "480000"
    },
    {
      "Name": "TestingPressure",
      "Value": "479520"
    },
    {
      "Name": "SensorPressure",
      "Value": "479492.1525125"
    },
    {
      "Name": "PressureChange",
      "Value": "-27.8474874999956"
    },
    {
      "Name": "CorrectionValue",
      "Value": "0"
    },
    {
      "Name": "Result",
      "Value": "0.000168882422841186"
    },
    {
      "Name": "Quantity",
      "Value": "12"
    },
    {
      "Name": "QuantityOk",
      "Value": "2"
    }
  ]
}
```

2.2.12 Get the test results

GetTestResult	
Method name	GetTestResult
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Text
Description	Returns the result of the last measurement on the selected channel.
Availability	Since software version 4.1.13.0

2.2.12.1 Return values

Value	Description
Undefined	The result is undefined.
Aborted	The measurement got aborted by a user.
OK	The measurement was ok.
NOK	The measurement was not ok.
NoResult	No result available for this measuring channel.
Error	An error has occurred.

2.2.13 Check if the measuring results are available

MeasuringResultsAvailable	
Method name	MeasuringResultsAvailable
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Boolean *(True = The measuring result are available,* *False = The measuring results are not available)*
Description	Returns if the measuring results of the last measurement on the selected channel are available.
Availability	Since software version 4.1.13.0

2.2.14 Get the measuring results

GetMeasuringResults	
Method name	GetMeasuringResults
Parameter	Positive integer *(ID of the measuring channel)*
Return value	List of measuring result objects.
Description	Returns a list of measuring result objects of the selected channel. The return value is an empty string if the measuring results are not available. To ensure that the results are available use the call MeasuringResultsAvailable.
Availability	Since software version 4.1.13.0

2.2.14.1 Measuring result object list example

This is an example of a list of measuring result objects returned from the GetMeasuringResults call:

```
{
  "MeasuringResults": [
    {
      "Name": "StartTime",
      "Value": "28-10-2019 08:53:50"
    },
    {
      "Name": "ProgramName",
      "Value": "Program 1"
    },
    {
      "Name": "Result",
      "Value": "OK"
    },
    {
      "Name": "ResultUnit",
      "Value": "Pa*m³/s"
    }
  ]
}
```

Which and how many measurement results are returned in the list depends on the setting of the leak test device. The list is based on the template that e.g. the .CSV results are based on.

2.2.14.2 Default layout

There also exists a default layout of the measuring result list which can be used.
This layout can be used by changing the method name from GetMeasuringResults to GetMeasuringResultsDefaultLayout:


```
{
  "MeasuringResults": [
    {
      "Name": "StartTime",
      "Value": "28-10-2019 08:53:50"
    },
    {
      "Name": "SerialNumber",
      "Value": ""
    },
    {
      "Name": "Result",
      "Value": "OK"
    },
    {
      "Name": "ResultValue",
      "Value": "0,000146745782278802"
    },
    {
      "Name": "ResultUnit",
      "Value": "Pa*m³/s"
    }
  ]
}
```


2.2.15 Get single measuring results

GetMeasuringResult	
Method name	GetMeasuringResult
Parameter	Measuring result information object.
Return value	Text *(The value of the requested measuring result)*
Description	Returns a specific measuring result of the last measurement on the selected channel.
Availability	Since software version 4.1.13.0

2.2.15.1 Measuring result information object example

This is an example of a measuring result information object for the GetMeasuringResult call:


```
{
  "ChannelID": 1,
  "ResultName": "StartTime"
}
```

2.2.16 Get Program

GetProgram	
Method name	getProgram
HTTP request method	Post
Body content	Program information object.
Return value	Program object
Description	Returns the program with the selected external id and channel.
Example call	http://127.0.0.1:50001/api/zed/getProgram/
Availability	Since software version 4.1.13.0

2.2.16.1 Program information object example

This is an example of the program information object used for the GetProram call:

```
{
  "ChannelID": 1,
  "ExternalID": 2
}
```

2.2.16.2 Program object example

This is an example of a program objects returned from the GetProgram call:

```
{
  "MeasuringType": "PressureChangeGaugeLeakage",
  "DimensionType": "Leakrate",
  "Header": {
    "ExternalID": 3,
    "ChannelID": 1,
    "ProgramName": "Programm 1",
    "Description": "",
    "ProgramType": "MeasuringProgram",
    "CreationTime": "2020-11-10T14:34:55",
    "LastChange": "2020-11-16T13:58:52"
  },
  "Parameters": [
    {
      "Name": "NOK.Active",
      "Value": "True"
    },
    {
      "Name": "NOK.AllowedRepetitions",
      "Value": "1"
    },
    {
      "Name": "DefaultUnits.Active",
      "Value": "True"
    },
    ...
  ]
}
```

2.2.17 Get a program parameter

GetProgramParameter	
Method name	GetProgramParameter
Parameter	Program parameter information object.
Return value	Text *(Value of the program parameter)*
Description	Returns the value of the selected parameter of the program with the selected external id.
Availability	Since software version 4.1.13.0

2.2.17.1 Program parameter information object example

This is an example of the program parameter information object used for the GetProramParameter call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "ParameterName": "Phase.PreFilling",
  "Value": ""
}
```

2.2.18 Set a program parameter

SetProgramParameter	
Method name	GetProgramParameter
Parameter	Program parameter information object.
Return value	Boolean *(True = The parameter was set successfully,* *False = The parameter could not be set successfully)*
Description	Sets the selected parameter of the program with the selected external id to the given value.
Availability	Since software version 4.1.13.0

2.2.18.1 Program parameter information object example

This is an example of the program parameter information object used for the SetProramParameter call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "ParameterName": "Phase.PreFilling",
  "Value": "3"
}
```

2.2.19 Set multiple program parameters

SetProgramParameters	
Method name	setProgramParameters
Parameter	Program parameters information object.
Return value	Boolean *(True = The parameters were set successfully,* *False = The parameters could not be set successfully)*
Description	Sets the selected parameters of the program with the selected external id to the given values.
Availability	Since software version 4.3.26.0

2.2.19.1 Program parameters information object example

This is an example of the program parameters information object used for the SetProramParameters call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "Parameters":
  [
    { "Name": "Pressure.Filling", "Value": "60000" },
    { "Name": "Pressure.PreFilling", "Value": "60000" }
  ]
}
```

2.2.20 Set external program ID

SetProgramExternalId	
Method name	SetProgramExternalId
Parameter	SetProgramExternalId information object.
Return value	Boolean *(True = The external Id was set successfully,* *False = The external Id could not be set successfully)*
Description	Sets the external Id of the program with the selected external id to the given value.
Availability	Since software version 4.3.45.0

2.2.20.1 SetProgramExternalId information object example

This is an example of the SetProgramExternalId information object used for the SetProgramExternalId call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "NewExternalID": 3
}
```

2.2.21 Set program name

SetProgramName	
Method name	SetProgramName
Parameter	SetProgramName information object.
Return value	Boolean *(True = The program name was set successfully,* *False = The program name could not be set successfully)*
Description	Sets the name of the program with the selected external id to the given value.
Availability	Since software version 4.3.45.0

2.2.21.1 SetProgramName information object example

This is an example of the SetProgramName information object used for the SetProgramName call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "ProgramName": "Leaktest program"
}
```

2.2.22 Create new measuring program

CreateMeasuringProgram	
Method name	CreateMeasuringProgram
Parameter	CreateMeasuringProgram information object.
Return value	Boolean *(True = The program was created successfully,* *False = The program could not be created successfully)*
Description	Creates a new measuring program.
Availability	Since software version 4.3.45.0

2.2.22.1 CreateMeasuringProgram information object example

This is an example of the CreateMeasuringProgram information object used for the CreateMeasuringProgram call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "MeasuringType": "PressureChangeGaugeLeakage",
  "ProgramName": "Leaktest program"
}
```

2.2.23 Delete a program

DeleteProgram	
Method name	DeleteProgram
Parameter	DeleteProgram information object.
Return value	Boolean *(True = The program was deleted successfully,* *False = The program could not be deleted successfully)*
Description	Deletes an existing program.
Availability	Since software version 4.3.45.0

2.2.23.1 DeleteProgram information object example

This is an example of the DeleteProgram information object used for the DeleteProgram call:

```
{
  "ChannelID": 1,
  "ExternalID": 2
}
```

2.2.24 Start the system verification

StartSystemVerification	
Method name	StartSystemVerification
Parameter	System verification information object.
Return value	Boolean *(True = The system verification started successfully,* *False = The system verification could not be started)*
Description	Starts the system verification for the program with the selected external id.
Availability	Since software version 4.1.13.0

2.2.24.1 System verification information object example

This is an example of the system verification information object used for the StartSystemVerification and ResetSystemVerification call:

```
{
  "ChannelID": 1,
  "ExternalID": 2,
  "ParameterName": "Phase.PreFilling",
  "Value": "3"
}
```

2.2.25 Reset the system verification

ResetSystemVerification	
Method name	ResetSystemVerification
Parameter	System verification information object.
Return value	Boolean *(True = The system verification got successfully reseted,* *False = The system verification could not be reseted)*
Description	Resets the system verification of the program with the selected external id and returns if it worked. This call is used if a program is blocked by a failed system verification.
Availability	Since software version 4.1.13.0

2.2.26 Get a specific value of the system verification

GetSystemVerificationValue	
Method name	GetSystemVerificationValue
Parameter	System verification value information object.
Return value	Text (The value of the selected system verification value)
Description	Returns the value of the selected system verification value for the last system verification of the program with the selected external id.
Availability	Since software version 4.1.13.0

2.2.26.1 System verification value information object example

This is an example of the system verification value information object used for the GetSystemVerificationValue call:

```
{
  "SystemVerificationValue": "DeviationOfTestleak",
  "ChannelID": 1,
  "ExternalID": 2
}
```

2.2.26.2 Available values of the SystemVerificationValue field

Value	Description
DifferenceValue	The difference between the measurements with and without a leak.
DeviationOfTestleak	The deviation from the set test leak.

2.2.27 Get the device informations

GetDeviceInformation	
Method name	GetDeviceInformation
Return value	Device information object.
Description	Returns the most important information about the device and it's measuring channels.
Availability	Since software version 4.1.13.0

2.2.27.1 Device information object example

This is an example of the device information object returned from the GetDeviceInformation call:


```
{
  "SerialNumber": "1909001",
  "SoftwareVersion": "4.1.15.2",
  "DeviceType": "ZEDBase",
  "MeasuringChannels": [
    {
      "ChannelID": 1,
      "CalibrationDate": "",
      "ChannelType": "MF",
      "RangeMin": 0.0,
      "RangeMax": 100000.0
    }
  ]
}
```

2.2.28 Get the default program parameters

GetDefaultProgramParameters	
Method name	GetDefaultProgramParameters
Parameter	Positive integer *(ID of the measuring channel)*
Return value	List of program parameter objects.
Description	Returns the default program parameters of the selected measuring channel.
Availability	Since software version 4.1.13.0

2.2.28.1 Program parameter list object example

This is an example of the program parameter list object returned from the GetDefaultProgramParameters call:


```
{
  "MeasuringTypeParameterList": [
    {
      "MeasuringType": "PressureChangeGauge",
      "ProgramParameters": [
        {
          "Name": "NOK.Active",
          "Value": "False"
        },
        {
          "Name": "NOK.AllowedRepetitions",
          "Value": "1"
        },
        {
          "Name": "DefaultUnits.Active",
          "Value": "True"
        },
        {
          "Name": "SafeVenting.PressureLimit",
          "Value": "10000"
        }
      ]
    }
  ]
}
```

2.2.29 Acknowledge NOK Result

NOKAcknowledgeChannel	
Method name	NOKAcknowledgeChannel
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Boolean *(True = The NOK result got acknowledged successfully,* *False = The NOK result could not get acknowledged)*
Description	Acknowledges a NOK Result if needed.
Availability	Since software version 4.3.13.0

2.2.30 Check for needed NOK Acknowledgement

CheckNOKAcknowledgeNeeded	
Method name	CheckNokAcknowledgeChannel
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Boolean *(True = The NOK acknowledgement is needed,* *False = The NOK acknowledgement is not needed)*
Description	Checks if the NOK Acknowledgement is needed.
Availability	Since software version 4.3.29.0

2.2.31 Get Charts

GetCharts	
Method name	GetCharts
Parameter	Positive integer *(ID of the measuring channel)*
Return value	List of chart objects.
Description	Returns the charts of the last measurement of the selected measuring channel.
Availability	Since software version 4.3.74.0

2.2.31.1 Chart list object example

This is an example of the charts list object returned from the GetCharts call


```
{
  "Charts": [
    {
      "Name": "Normal chart",
      "ChartLines": [
        {
          "Name": "Pressure",
          "XAxisUnit": "s",
          "YAxisUnit": "bar",
          "ChartPoints": [
            {
              "X": "0.03735",
              "Y": "0.0"
            },
            {
              "X": "0.0485",
              "Y": "0.0"
            },
            {
              "X": "0.060724",
              "Y": "5.3384"
            }
          ]
        },
      ]
    }
  ]
}
```

2.2.32 Get the last error

GetLastError	
Method name	GetLastError
Return value	Text *(The last error message)*
Description	Returns the last error of the Web API server. If the text is empty, no error occurred since the start of the server.
Availability	Since software version 4.1.13.0

2.2.33 Get the last error of the measuring channel

GetLastChannelError	
Method name	GetLastChannelError
Parameter	Positive integer *(ID of the measuring channel)*
Return value	Text *(The last error message of the measuring channel)*
Description	Returns the last error of the Web API server that occurred on the selected measuring channel. If the text is empty, no error occurred on the measuring channel since the start of the server.
Availability	Since software version 4.1.13.0

3 Measuring program

3.1 Program header

Name	Description	Type	Unit
Channel	Number of the channel that will be used by this program.	Positive integer	-
ChannelMode	The channel mode which should be used for the measuring channel.	Enumeration	Channel mode
MeasuringType	The measuring type used for the measuring program.	Enumeration	Measuring type
ProgramName	The name of the measuring program. The name will be used e.g. for the user interface and the measuring results.	Text	-

3.2 Measuring parameters

3.2.1 Process parameters

3.2.1.1 Pressures (Relative pressure)

Name	Description	Type	Unit
Pressure.PreFilling	The pressure that is used for pre-filling the test setup during pre-filling time.	Floating point number	Pascal
Pressure.Filling	The pressure that is used for filling the test setup during filling time.	Floating point number	Pascal
Pressure.Measuring	The pressure that is used for filling the test setup during measuring time.	Floating point number	Pascal
Pressure.LeakDetection	Defined permanent pressure for locating leaks with leak detection spray.	Floating point number	Pascal

3.2.1.2 Pressures (Absolute pressure)

Name	Description	Type	Unit
PressureAbs.PreFilling	The pressure that is used for pre-filling the test setup during pre-filling time.	Positive floating point number	Pascal
PressureAbs.Filling	The pressure that is used for filling the test setup during filling time.	Positive floating point number	Pascal
PressureAbs.Measuring	The pressure that is used for filling the test setup during measuring time.	Positive floating point number	Pascal
PressureAbs.LeakDetection	Defined permanent pressure for locating leaks with leak detection spray.	Positive floating point number	Pascal

3.2.1.3 Flows

Name	Description	Type	Unit
Flow.PreFilling	The flow that is used for pre-filling the test setup during pre-filling time.	Floating point number	Pa*m³/s
Flow.Filling	The flow that is used for filling the test setup during filling time.	Floating point number	Pa*m³/s
Flow.LeakDetection	Defined permanent flow for locating leaks with leak detection spray.	Floating point number	Pa*m³/s

3.2.1.4 Phases / Times

Name	Description	Type	Unit
Phase.PreRun	Waiting time after the start of the program.	Positive floating point number	Seconds
Phase.RefFilling.Active	Activates the prefilling of the reference volume.	Boolean	-
Phase.RefFilling	Time to fill the reference volume.	Positive floating point number	Seconds
Phase.PreFilling	Time for the pre-filling of the test setup. The pre-filling is used to pre-stretch the part, to stabilize mechanically and to minimize temperature effects (phase is not required).	Positive floating point number	Seconds
Phase.Filling	Time for filling the test setup.	Positive floating point number	Seconds
Phase.PreBalancing	Time for stabilizing the test setup with opened bypass-valve.	Positive floating point number	Seconds
Phase.Balancing	Time for stabilizing the test setup.	Positive floating point number	Seconds
Phase.Measuring	Time for measuring the pressure change or the flow.	Positive floating point number	Seconds
Phase.Bypass.Balancing	Time for stabilizing the test setup with opened bypass-valve.	Positive floating point number	Seconds
Phase.Overflowing	Time for the Overflowing.	Positive floating point number	Seconds
Phase.VolumeOverflow.Balancing	Time for stabilizing after overflow into the expansion volume.	Positive floating point number	Seconds
Phase.VolumeOverflow.Measuring	Time for measuring the pressure change after overflow into the expansion volume.	Positive floating point number	Seconds

Phase.Venting	Time for venting the test setup.	Positive floating point number	Seconds
Phase.FollowUp	Waiting time before the end of the program.	Positive floating point number	Seconds
Phase. LeakDetectionVenting	Time for venting the test setup after a leak location.	Positive floating point number	Seconds
Phase.Volumecheck. Auto	Activates the automatic measuring phase fractions.	Boolean	-
Phase.Volumecheck. Balancing	The percentage of the time stabilizing the test setup.	Positive floating point number	Percent
Phase.Volumecheck. Overflowing	The percentage of the time for the Overflowing.	Positive floating point number	Percent

3.2.1.5 Volumes

Name	Description	Type	Unit
Volume.Part	Volume of the entire test setup (equivalent to the volume of the part and the volume of the leak testing device including hose system and adaption).	Positive floating point number	m^3
Volume.VolumeCheck	Volume for overflowing, to get the volume of the test setup.	Positive floating point number	m^3
Volumefactor.Part	Consider the proportion of the volume of the part to the volume of the reference.	Floating point number	Factor
Volume.Reference. MeasuringCircuit	Volume of the reference volume, used for the measurement.	Positive floating point number	m^3

3.2.1.6 Reference volumes

Name	Description	Type	Unit
MassflowLeakage. VrefVentingActive	Activation of the venting of the reference volume after the measurement.	Boolean	-

3.2.2 Pressure change detection

Name	Description	Type	Unit
PressureChangeDetection. InhibitTime	Timeout between first detection and actual result evaluation (noise compensation).	Floating point number	Milliseconds
PressureChangeDetection. UpperLimitsEnvelope	Minimum pressure required for detection.	Floating point number	Pascal
PressureChangeDetection. LowerLimitsEnvelope	Maximum pressure required for detection.	Floating point number	Pascal
PressureChangeDetectionAbs. UpperLimitsEnvelope	Minimum pressure required for detection.	Positive floating point number	Pascal
PressureChangeDetectionAbs. LowerLimitsEnvelope	Maximum pressure required for detection.	Positive floating point number	Pascal

3.2.3 Flow change detection

Name	Description	Type	Unit
FlowChangeDetection. InhibitTime	Timeout between first detection and actual result evaluation (noise compensation).	Floating point number	Pa*m³/s
FlowChangeDetection. UpperFlowLimit	Minimum flow required for detection.	Floating point number	Pa*m³/s
FlowChangeDetection. LowerFlowLimit	Maximum flow required for detection.	Floating point number	Pa*m³/s

3.2.3.1 PID control

Name	Description	Type	Unit
PID.Active	Activation of the PID control. The PID control is used to optimize the pressure control.	Boolean	-
PID.Offset	Offset for the PID control.	Floating point number	Pascal
PID.KP	Proportional element. This effects proportional to the input signal.	Floating point number	-
PID.KI	Integral element. This effects the value by temporal integration of the control deviation.	Floating point number	Amount per second
PID.KD	D element (differentiator). This must be used in combination with the integral element or the proportional element and affects the rate of change.	Floating point number	Seconds
PID.Correction.Max	Maximum total correction.	Floating point number	Pascal
PID.I.Max	Maximum integral element.	Floating point number	Pascal
PID.SampleRate	Sampling rate that is used for the calculation of the values.	Floating point number	Milliseconds

3.2.3.2 Additional parameters

Name	Description	Type	Unit
TestLeakActive	Activation of the test leak. If the test leak is enabled, it is connected automatically for this test program.	Boolean	-
VentingMode	Determines which venting valve is used.	Enumeration	Venting mode
SafeVenting.Active	If safe venting is enabled, the safe venting phase will not stop until the specified pressure limit is reached.	Boolean	-
SafeVenting.PressureLimit	Specifies the pressure limit at which the venting phase is ended.	Floating point number	Pascal
FillingBypassActive	Opens the filling bypass valve during the pre-fill phase.	Boolean	-
PermanentVenting.Active	Vent the test setup after ending the test permanent.	Boolean	-
SwitchingValves.SwitchValve1	Switch the switching valve 1 during the measurement.	Boolean	-
SwitchingValves.SwitchValve2	Switch the switching valve 2 during the measurement.	Boolean	-

3.2.4 Limits

Name	Description	Type	Unit
Leakrate.UpperLimit1	First upper limit (fine leak). Corresponds to the specified leak rate.	Floating point number	Pa*m³/s
Leakrate.UpperLimit2	Second upper limit (gross leak). Parts, where reworking is possible, can still be reworked within this limit.	Floating point number	Pa*m³/s
Leakrate.LowerLimit1	First lower limit (fine leak). For tests with positive pressure a slightly negative leakage should always be allowed because of temperature effects.	Floating point number	Pa*m³/s
Leakrate.LowerLimit2	Second lower limit (gross leak).	Floating point number	Pa*m³/s
PressureChange.UpperLimit1	First upper limit (fine leak). Corresponds to the specified leak.	Floating point number	Pascal
PressureChange.UpperLimit2	Second upper limit (gross leak). Parts, where reworking is possible, can still be reworked within this limit.	Floating point number	Pascal
PressureChange.LowerLimit1	First lower limit (fine leak). For tests with positive pressure a slightly negative pressure change	Floating point number	Pascal

	should always be allowed because of temperature effects.		
PressureChange.LowerLimit2	Second lower limit (gross leak).	Floating point number	Pascal
Volume.UpperLimit1	First upper limit for the volume.	Floating point number	m^3
Volume.UpperLimit2	Second upper limit for the volume.	Floating point number	m^3
Volume.LowerLimit1	First lower limit for the volume.	Floating point number	m^3
Volume.LowerLimit2	Second lower limit for the volume.	Floating point number	m^3
DirectFlow.UpperLimit1	First upper limit (fine leak). Corresponds to the specified leak rate.	Floating point number	$Pa*m^3/s$
DirectFlow.UpperLimit2	Second upper limit (gross leak). Parts, where reworking is possible, can still be reworked within this limit.	Floating point number	$Pa*m^3/s$
DirectFlow.LowerLimit1	First lower limit (fine leak). For tests with positive pressure a slightly negative leakage should always be allowed because of temperature effects.	Floating point number	$Pa*m^3/s$
DirectFlow.LowerLimit2	Second lower limit (gross leak).	Floating point number	$Pa*m^3/s$
Pressure.UpperLimit1	First upper limit (fine leak). Corresponds to the specified leak.	Floating point number	Pascal
Pressure.UpperLimit2	Second upper limit (gross leak). Parts, where reworking is possible, can still be reworked within this limit.	Floating point number	Pascal
Pressure.LowerLimit1	First lower limit (fine leak). For tests with positive pressure a slightly negative pressure change should always be allowed because of temperature effects.	Floating point number	Pascal
Pressure.LowerLimit2	Second lower limit (gross leak).	Floating point number	Pascal
PressureAbs.UpperLimit1	First upper limit (fine leak). Corresponds to the specified leak.	Positive floating point number	Pascal

PressureAbs.UpperLimit2	Second upper limit (gross leak). Parts, where reworking is possible, can still be reworked within this limit.	Positive floating point number	Pascal
PressureAbs.LowerLimit1	First lower limit (fine leak). For tests with positive pressure a slightly negative pressure change should always be allowed because of temperature effects.	Positive floating point number	Pascal
PressureAbs.LowerLimit2	Second lower limit (gross leak).	Positive floating point number	Pascal
PressureChange.DetailChart.Min	Smallest value on the y-axis.	Floating point number	Pascal
PressureChange.DetailChart.Max	Biggest value on the y-axis.	Floating point number	Pascal

3.2.5 Offset

Name	Description	Type	Unit
Offset.Active	Activation of the results correction by offset.	Boolean	-
OffsetPressure.Fixed	Offset value. Will be added to the result continuously.	Floating point number	Pascal
OffsetLeakage.Fixed	Offset value. Will be added to the result continuously.	Floating point number	Pa*m³/s

3.2.6 Temperature compensation

Name	Description	Type	Unit
TempCo.Active	Activation of the temperature compensation.	Boolean	-
Phase.SoftwareTempCoVenting	Time for venting the test setup before measuring the value for the temperature compensation.	Positive floating point number	Seconds
Phase.SoftwareTempCoBalancing	Time for stabilizing the test setup before measuring the value for the temperature compensation.	Positive floating point number	Seconds
Phase.SoftwareTempCoMeasuring	Time for measuring the value for the temperature compensation.	Positive floating point number	Seconds

TempCo.SW.Coefficient	Optional coefficient. Is used for the optimization of the correction value of the temperature compensation.	Floating point number	-

3.2.7 Process monitor

3.2.7.1 Start temperature check

Name	Description	Type	Unit
StartTemperatureCheck .Mode	The type of the start temperature monitoring. The temperature value is recorded after the pre run phase.	Enumeration	Temperature check mode
StartTemperatureCheck .Part.LowerLimit	Lower limit of the part temperature while monitoring the start temperature.	Positive floating point number	Kelvin
StartTemperatureCheck .Part.UpperLimit	Upper limit of the part temperature while monitoring the start temperature.	Positive floating point number	Kelvin
StartTemperatureCheck .Ambient.LowerLimit	Lower limit of the ambient temperature while monitoring the start temperature.	Positive floating point number	Kelvin
StartTemperatureCheck .Ambient.UpperLimit	Upper limit of the ambient temperature while monitoring the start temperature.	Positive floating point number	Kelvin

3.2.7.2 Prefilling pressure check

Name	Description	Type	Unit
Pressure.PreFillingCapa cityCheck.Active	Enables the prefilling pressure monitoring, which can detect too small a volume due to an early analysis of the pressure.	Boolean	-
Pressure.PreFillingCapa cityCheck.BalancingOffs etTime	The time interval from the beginning of the phase prefilling. This allows to perform the prefill pressure monitoring analysis at any position within the prefill phase.	Positive floating point number	Seconds
Pressure.PreFillingCapa cityCheck.BalancingTim e	The balancing time before the pressure will be analyzed.	Positive floating point number	Seconds
Pressure.PreFillingCapa cityCheck.Tolerance.Ne gative	The minimum allowed value of pressure at this time.	Floating point number	Pascal
Pressure.PreFillingCapa cityCheck.Tolerance.Po sitive	The maximum allowed value of pressure at this time.	Floating point number	Pascal

3.2.7.3 Process monitor

Name	Description	Type	Unit
Pressure.Global.Tolerance.Active	Activation of the global tolerance limits for the pressure monitoring.	Boolean	-
Pressure.Global.Tolerance.Positive	Global upper tolerance range for the pressure.	Positive floating point number	Pascal
Pressure.Global.Tolerance.Negative	Global lower tolerance range for the pressure.	Positive floating point number	Pascal
Pressure.PreFilling.Tolerance.Positive	Upper tolerance range for the pre-filling pressure.	Positive floating point number	Pascal
Pressure.PreFilling.Tolerance.Negative	Lower tolerance range for the pre-filling pressure.	Positive floating point number	Pascal
Pressure.Filling.Tolerance.Positive	Upper tolerance range for the filling pressure.	Positive floating point number	Pascal
Pressure.Filling.Tolerance.Negative	Lower tolerance range for the filling pressure.	Positive floating point number	Pascal
Pressure.Measuring.Tolerance.Positive	Upper tolerance range of the measuring pressure.	Positive floating point number	Pascal
Pressure.Measuring.Tolerance.Negative	Lower tolerance range of the measuring pressure.	Positive floating point number	Pascal
Flow.Global.Tolerance.Active	Activation of the global tolerance limits for the pressure monitoring.	Boolean	-
Flow.Global.Tolerance.Positive	Global upper tolerance range for the flow.	Positive floating point number	$Pa*m^3/s$
Flow.Global.Tolerance.Negative	Global lower tolerance range for the flow.	Positive floating point number	$Pa*m^3/s$
Flow.PreFilling.Tolerance.Positive	Upper tolerance range for the pre-filling flow.	Positive floating point number	$Pa*m^3/s$
Flow.PreFilling.Tolerance.Negative	Lower tolerance range for the pre-filling flow.	Positive floating point number	$Pa*m^3/s$
Flow.Filling.Tolerance.Positive	Upper tolerance range for the filling flow.	Positive floating point number	$Pa*m^3/s$
Flow.Filling.Tolerance.Negative	Lower tolerance range for the filling flow.	Positive floating point number	$Pa*m^3/s$

Pressure.ChangeDetection.Minimum	Minimum opening pressure.	Floating point number	Pascal
Pressure.ChangeDetection.Maximum	Maximum opening pressure.	Floating point number	Pascal
PressureAbs.ChangeDetection.Minimum	Minimum opening pressure.	Positive floating point number	Pascal
PressureAbs.ChangeDetection.Maximum	Maximum opening pressure.	Positive floating point number	Pascal

3.2.8 Ramps

Name	Description	Type	Unit
Ramps.Active	Activation of the pressure ramps. Pressure ramps are used to raise or lower the pressure by ramp functions.	Boolean	-
Ramp.PreFilling	The pre-filling pressure will be reached within the specified percentage of the pre-filling time. Example: A percentage of 50% and a pre-filling time of 6 s means that the pre-filling pressure will be reached within 3 s.	Positive floating point number (0…100)	Percent
Ramp.Filling	The filling pressure will be reached within the specified percentage of the filling time. Example: A percentage of 50% and a filling time of 6 s means that the filling pressure will be reached within 3 s.	Positive floating point number (0…100)	Percent
Ramp.Measuring	The measuring pressure will be reached within the specified percentage of the measuring time. Example: A percentage of 50% and a measuring time of 6 s means that the measuring pressure will be reached within 3 s.	Positive floating point number (0…100)	Percent
Ramp.Venting	The test setup will be vented within the specified percentage of the venting time. Example: A percentage of 50% and a venting time of 6 seconds means that the test setup will be vented within 3 s.	Positive floating point number (0…100)	Percent

3.2.9 Units

Name	Description	Type	Unit
DefaultUnits.Active	Activation of the standard units. If the default units are enabled, the default units that are configured in the settings will be used.	Boolean	-
Unit.LeakRate	Unit of the leak rate.	Enumeration	Flow
Unit.LeakRate.DecimalPlaces	Number of decimal places for the leak rate.	Positive integer	Amount

Unit.PressureChange	Unit of the pressure change.	Enumeration	Pressure
Unit.PressureChange.DecimalPlaces	Number of decimal places for the pressure change.	Positive integer	Amount
Unit.Flow	Unit of the flow. The flow is shown in units of volume-flow in standard configuration (0°C / 101325 Pa).	Enumeration	Flow
Unit.Flow.DecimalPlaces	Number of decimal places for the flow.	Positive integer	Amount
Unit.Pressure	Unit of the pressure.	Enumeration	Pressure
Unit.Pressure.DecimalPlaces	Number of decimal places for the pressure.	Positive integer	Amount
Unit.Volume	Unit of the volume.	Enumeration	Volume
Unit.Volume.DecimalPlaces	Number of decimal places for the volume.	Positive integer	Amount
Unit.Temperature	Unit of the temperature.	Enumeration	Temperature
Unit.Temperature.DecimalPlaces	Number of decimal places for the temperature.	Positive integer	Amount
Unit.TemperatureDifference	Unit of the temperature difference.	Enumeration	Temperature
Unit.TemperatureDifference.DecimalPlaces	Number of decimal places for the temperature difference.	Positive integer	Amount
Unit.Time	Unit of the time.	Enumeration	Time
Unit.Time.DecimalPlaces	Number of decimal places for the time.	Positive integer	Amount
Unit.DimensionlessQuantity	Unit of the percentage.	Enumeration	Dimensionless Quantity
Unit.DimensionlessQuantity.DecimalPlaces	Number of decimal places for the percentage.	Positive integer	Amount

3.2.10 Testing configs

3.2.10.1 NOK Acknowledgment

Name	Description	Type	Unit
NOK.Active	Activation of the NOK acknowledgment.	Boolean	-

3.3 Enumerations

3.3.1 Channel mode

Value	Description
LeakTest	Normal measuring mode to detect a leak rate or pressure change.
LeakDetection	Continuous pressure mode to find a leak on the test set-up or on the test object.

3.3.2 Measuring type

Value	Description
PressureChangeGauge	Pressure change (RD)
PressureChangeGaugeLeakage	Leak rate based on pressure change (RD)
PressureChangeDifferential	Pressure change (DD)
PressureChangeDifferentialLeakage	Leak rate based on pressure change (DD)
MassflowLeakage	Mass flow leakage
PressureControlledFlow	Pressure controlled flow (DF)
StagnationPressure	Stagnation pressure
PressureChangeDetection	Pressure change detection
VolumeCheck	Volume check
FlowChangeDetection	Flow detection
PressureChangeAndVolumeCalculation	Leak rate with volume calculation
PressureChangeDifferentialAndVolumeCalculation	Leak rate with volume calculation (DD)
PressureChangeGaugeAbs	Pressure change (AD)
PressureChangeGaugeLeakageAbs	Leak rate based on pressure change (AD)
PressureChangeDifferentialAbs	Pressure change (DD)
PressureChangeDifferentialLeakageAbs	Leak rate based on pressure change (DD)
MassflowLeakageAbs	Mass flow leakage
PressureControlledFlowAbs	Pressure controlled flow (DF)
StagnationPressureAbs	Stagnation pressure
PressureChangeDetectionAbs	Pressure change detection
VolumeCheckAbs	Volume check
FlowChangeDetectionAbs	Flow detection
PressureChangeAndVolumeCalculationAbs	Leak rate with volume calculation
PressureChangeDifferentialAndVolumeCalculationAbs	Leak rate with volume calculation (DD)

3.3.3 Venting mode

Value	Description
InternalVenting	The test item is vented via the internal venting valve.
ExternalVenting	The test item is vented via the external venting valve.
Both	The test item is vented via the internal and the external venting valve.

3.3.4 Temperature check mode

Value	Description
None	No temperature check.
Part	Temperature check via the temperature of the test item.
Ambient	Temperature check via the temperature of the ambient.
PartAndAmbient	Temperature check via the temperature of the test item and the ambient.

3.3.5 Flow

Value	Description
Pa*m³/s	Pascal * Cubic meters / Second
mbar*l/s	Millibar * Liter / Second
m³/s	Cubic meters / Second
ml/s	Milliliter / Second
ml/h	Milliliter / Hour
cm³/min	Cubic centimeter / Minute
cm³/s	Cubic centimeter / Second
l/min	Liter / Minute
l/h	Liter / Hour
mm³/s	Cubic millimeter / Second
US gpm	US gallons per Minute

3.3.6 Pressure

Value	Description
Pa	Pascal
mbar	Millibar
bar	Bar
psi	Pound-force per square inch
mmWS	Millimeter water

3.3.7 Volume

Value	Description
m³	Cubic meter
hl	Hectoliter
l	Liter

dl	Deci liter
cl	Centiliter
ml	Milliliter
cm³	Cubic centimeter
µl	Microliter
mm³	Cubic millimeter

3.3.8 Temperature

Value	Description
K	Kelvin
°F	Degree Fahrenheit
°C	Degree Celsius
°R	Degree Rankine
°Ré	Degree Réaumur

3.3.9 Time

Value	Description
ms	Millisecond
s	Second
min	Minute
h	Hour
d	Day

3.3.10 Dimensionless Quantity

Value	Description
%	Percent
‰	Per thousand
ppm	Parts per million

ZELTWANGER Dichtheits- und
Funktionsprüfsysteme GmbH
Maltschachstraße 32
72144 Dußlingen

DEUTSCHLAND

Telefon: +49 7072 92897-501
Telefax: +49 7072 92897-555

E-Mail: dichtheit@zeltwanger.de
Internet: www.zeltwanger.de