

Referenz

QIK Quality Improvement Keeper

Produkt:	QIK - Quality Improvement Keeper
Dokument:	QIK Referenz
Version Dokument:	V2.20 Doc#: P0405R
Version SW:	V2.20
Klassifizierung:	Ausschliesslich in Verwendung mit der Anwendung QIK
Dateiname:	QIK_Referenz.doc, 03.02.2025 08:58:00

Alle Marken-, Firmen- und Produktnamen in dieser Dokumentation werden nur für Erklärungen verwendet und sind in der Regel durch deren Inhaber geschützt.

Diese Dokumentation oder Teile daraus dürfen nicht ohne schriftliche Genehmigung von EDX Software Design in irgendeiner Form verarbeitet, kopiert, vervielfältigt oder verbreitet werden.

© EDX Software Design. Alle Rechte vorbehalten.

Änderungswesen

Version	Datum	Beschreibung der Änderung
2.20	03.02.25	EDX Software Design, E.Furrer + Neues Operator Konzept mit Abfrage beim Start und angepasste Testabschlussdialoge + Neue Möglichkeiten mit dem Error Report Dialog + Dialog <i>Extras</i> erweitert mit <i>Show Error Report dialog</i> + Dialog <i>Error Report</i> erweitert zu den neuen Möglichkeiten mit Sys.ShowErrorReport + Prozesssteuerung erweitert zum Ausführen oder Deaktivieren von Testfunktionen/fällen + Neu werden Definitionen in #Defines aufgelöst + Neue Direktiven #ShowDefineDoc, #if, #elif, #ifdef, #ifndef, #elifdef, #else und #endif mit bedingtem Laden + Neu QDF-Funktionen __PrinterInstalled, __ShowWarning und __AbortWithMessage + Neu Set-Befehl für DecimalSeparator und ThousandSeparator. + Neue SystemControl Funktionen: SetTestCaseStatus, Goto, AssignValueLog, AssignVariable AssignVariableEx und AssignVarConditionally + Device Document Display "Test Case aus der Doku" erweitert mit Hinweisen und Warnungen + Neu Device Port WaitForResponse mit NoResponse Label erweitert + Neu Device Port Monitor zur Beobachtung der Schnittstelle zwischen QIK und Prüfling + Device Instrument Reset Funktion ist veraltet und wird nicht mehr benötigt + ANHANG: QIK Doku mit MS Word" erweitert mit Tipps zur Formatierung und Ansicht
2.11	07.08.23	EDX Software Design, E.Furrer + Konventionen mit IP erweitert + ValueLog erweitert mit StartDateTimeFmt und FileDateFormat + Document Display Doku aktualisiert + UDP mit neuer Adress-Definition + TCP.Client Doku aktualisiert + Device.Instruct erweitert für UDP und TCP Adress-Redefinition + Device Port SendJson und WaitForJson
2.10	13.03.23	EDX Software Design, E.Furrer + Standard Dokument Anzeige Ready.html und ScriptError.html + Neu: Set DefaultDoc für Standard Dokumentation der Testdatei + Device RS232 erweitert mit SBI - Send Byte Intervall + Neu: GPCC Device + Anpassungen für Windows 10 respektive Windows 11
2.08	30.04.20	EDX Software Design, E.Furrer + Neu: TCP Client Device zu Testzwecken
2.07	01.09.16	EDX Software Design, E.Furrer + Neu: SystemControl erweitert mit Calculate und Evaluate + Neu: Port erweitert mit WaitForResponse + Neu: RS232 Port Bereich neu 1..255 + Neu: UDP neu implementiert
2.06	02.07.16	EDX Software Design, E.Furrer + Neu: SystemControl.Execute erweitert mit der Möglichkeit Variablen mit % einzubinden + Neu: QDF Definition für Notepad++
2.05	29.07.14	EDX Software Design, E.Furrer + Neu: Audio Input Device mit Start.AVG.Check erweitert + Neu: Audio Input Device Start.RMS.Measurements anstelle von Start.Analyzer + Neu: Audio Input Device Check.RMS mit Berechnungsformel erweitert + Neu: FTDI Device.Create mit dem Hinweis wegen Win7 und USB3.0 ergänzt
...		

Inhaltsverzeichnis

1.	Einleitung	6
2.	Bedingungen.....	6
2.1	Lizenz.....	6
2.1.1	Bestimmungen	6
2.1.2	Berechtigung.....	6
2.1.3	Registrieren	6
2.2	Garantie	6
2.3	Haftung.....	6
3.	Installation.....	7
3.1	Betriebssystemanforderungen.....	7
3.2	Dateien	7
3.3	Installation	7
3.3.1	Portable Executable	7
3.3.2	Windows UAC (User Account Control)	7
3.4	Programmparameter	7
4.	Registrierung	8
5.	Benutzerschnittstelle.....	9
5.1	Übersicht.....	9
5.2	Prozesssteuerung.....	10
6.	Dialoge.....	11
6.1	Operator.....	11
6.2	Extras	11
6.3	Error Report	12
6.4	Testabschluss	12
7.	Testablaufdefinition	13
7.1	Syntax Übersicht	13
7.1.1	Testmodule.....	13
7.1.2	Deklarationen.....	13
7.1.3	Geräte	13
7.1.4	Testfälle	13
7.1.5	Testschritte.....	13
7.1.6	Testfunktionen	13
7.2	Interpreter.....	13
7.3	Konventionen dieser Beschreibung	14
8.	Präprozessor	15
8.1	Direktiven.....	15
8.1.1	#Include "FileName"	15
8.1.2	#Define Identifier TokenString	15
8.1.3	#ShowDefineDoc FileName[URL Parameter]	15
8.1.4	Bedingtes Laden.....	16
8.1.4.1	#if Condition, #elif Condition, #else und #endif	17
8.1.4.2	#ifdef Identifier, #ifndef Identifier, #elifdef Identifier, #else und #endif	17
8.2	QDF-Funktionen.....	17
8.2.1	__PrinterInstalled("Printer Name")	17
8.2.2	__ShowWarning("Message")	17
8.2.3	__AbortWithMessage("Message")	17
9.	Deklarationen.....	18
9.1	Set(FileInfo; "Name"; "Version")	18
9.2	Set(DefaultDoc; "FileName").....	18
9.3	Set(DecimalSeparator; "Character")	18
9.4	Set(ThousandSeparator; "Character").....	19
9.5	Set(KeyCodeToFunction; KeyCode; "Function").....	19
9.6	Set(ValueLog; "FileName"[]; "StartDateTimeFmt"[]; "FileDateFormat[]")	19
9.7	Set(ErrorReport; "Template"; "Printer"; Orientation; Left; Top; Right; Bottom)	20
9.8	Set(DataType; VariableName).....	21
10.	Testfälle	22
10.1	Test.Case("Name"[]; Appearance))	22
10.2	Test.Final("Name"[]; Appearance))	22

11.	Device: System Control.....	23
11.1	Device.Create(SystemControl; Naming).....	23
11.2	Funktionen.....	23
11.2.1	Process("Name"; Option).....	23
11.2.2	ShowErrorReport("Name").....	23
11.2.3	SetTestCaseStatus("Name"; "TestCase"; Status).....	24
11.2.4	Goto("Name"; "TestCase"; Condition).....	24
11.2.5	Delay("Name"; Time).....	24
11.2.6	Pause("Name").....	24
11.2.7	Break("Name").....	24
11.2.8	Beep("Name").....	25
11.2.9	Dialog("Name"; "Title"; "Text"; Button).....	25
11.2.10	Dialog.Input.String("Name"; "Title"; "Text"; "Default"; "Expected"; Variable).....	25
11.2.11	Dialog.Input.Value("Name"; "Title"; "T1"; "T2"; "Default"; Min; Max; Variable).....	26
11.2.12	AssignValueLog("Name" §Spalte; Value; Option).....	26
11.2.13	AssignVariable("Name"; Value; VariableName).....	26
11.2.14	AssignVariableEx("Name"; Value; Variable).....	27
11.2.15	AssignVarConditionally("Name"; Condition, TrueValue; FalseValue; Variable).....	27
11.2.16	VarIntToStr("Name"; iVariable; "Format"; sVariable).....	27
11.2.17	VarStrToStr("Name"; sVariable1; "Format"; sVariable2).....	27
11.2.18	Calculate("Name"; Term; Variable).....	28
11.2.19	Evaluate("Name"; Condition).....	28
11.2.20	GetNextID("Name"; "FileName"; Variable).....	28
11.2.21	RegisterID("Name"; "FileName"; Variable).....	28
11.2.22	Print("Name"; " Template"; "Printer"; Orientation; L; T; R; B).....	29
11.2.23	OpenQDF("Name"; "FileName"; Action).....	29
11.2.24	Execute("Name"; "File"; Parameter; Options; WaitTime).....	29
11.2.25	ExistWindow("Name"; "WindowTitle").....	30
11.2.26	SendKey("Name"; KeyCode; "WindowTitle"; Monitoring)).....	30
12.	Device: Document Display.....	31
12.1	Device.Create(HTMLViewer; Naming).....	31
12.2	Funktionen.....	31
12.2.1	Open("Name"; "FileName").....	31
12.2.2	Goto("Name"; "Anchor").....	31
12.3	Test Case aus der Doku.....	32
13.	Device: Indicator.....	33
13.1	Device.Create(Indicator; Naming; "Text"; Digits; Color).....	33
13.2	Funktionen.....	33
13.2.1	Indicate("Name"; "Text"; Variable).....	33
14.	Device: Port.....	34
14.1	Device.Create(RS232; Naming; Timeout; Port; BPS; SBI).....	34
14.2	Device.Create(FTDI; Naming; Timeout; Port; BPS [;DBits; Parity; SBit; Mode]).....	34
14.3	Device.Create(UDP; Naming; Timeout; Address; ListeningPort).....	35
14.4	Device.Create(TCP.Client; Naming; Timeout; Address).....	35
14.5	Funktionen.....	35
14.5.1	Open("Name").....	35
14.5.2	Close("Name").....	35
14.5.3	Instruct("Name"; Statement).....	36
14.5.4	Timeout("Name"; Milliseconds).....	36
14.5.5	Send("Name"; "Data"; "STag"; "ETag"; "Expected" [; Variable]).....	36
14.5.6	WaitForResponse("Name"; "STag"; "ETag"; Expected[; Variable]).....	37
14.5.7	Evaluate.Value("Name"; "Cmd"; "STag"; "ETag"; Min; Max [; Variable]).....	37
14.5.8	SendJson("Name"; JSON; "STag"; "ETag"; "Expected" [; Variable]).....	37
14.5.9	WaitForJson("Name"; "STag"; "ETag"; "Expected" [; Variable]).....	38
14.5.10	Monitor.Initialize("Name"; Port; Adress[; Listening Port[; Options]]).....	38
15.	Device: GPCC.....	40
15.1	Device.Create(GPCC; Naming; Timeout).....	40
15.2	Funktionen.....	40
15.2.1	Open("Name").....	40
15.2.2	Close("Name").....	40

15.2.3	Timeout("Name"; Milliseconds)	40
15.2.4	Execute("Name"; "FunctionCall"; "Expected" [; Variable])	40
16.	Device: Instrument	42
16.1	DMM Mastech MS8218 - Bedienungshinweis	42
16.2	Device.Create(Instrument; Naming; "Type"; "Configuration")	42
16.3	Funktionen	42
16.3.1	Open("Name")	42
16.3.2	Close("Name")	42
16.3.3	Reset("Name") – [Deprecated!]	43
16.3.4	Check("Name"; "Mode"; Timeout; Min; Max [; Variable])	43
17.	Device: Audio	44
17.1	Allgemeine Funktionen	44
17.1.1	SetVolume("Name"; ValueType; Level[; Level])	44
17.1.2	MuteOn("Name")	44
17.1.3	MuteOff("Name")	44
17.1.4	SetAsDefault("Name")	44
17.2	Device: Audio Output	45
17.2.1	Device.Create(AudioOutput; Naming; "DeviceName")	45
17.2.2	Funktionen	45
17.2.2.1	Start.Sine("Name"; Frequency)	45
17.2.2.2	Start.File("Name"; "FileName")	45
17.2.2.3	Stop("Name")	45
17.3	Device: Audio Input	46
17.3.1	Device.Create(AudioInput; Naming; "DeviceName")	46
17.3.2	Funktionen	46
17.3.2.1	Start.AVG.Check("Name"; Terminate; Channel1[; Channel N])	46
17.3.2.2	Start.RMS.Measurements("Name")	46
17.3.2.3	Check.RMS("Name"; Min:Max [; Min:Max])	47
17.3.2.4	Start.Recorder("Name"; "FileName")	47
17.3.2.5	Stop("Name")	47
18.	ANHANG: Dokumentation / Benutzerführung mit MS Word	48
19.	ANHANG: RTF Vorlagen	50
20.	ANHANG: ID Dateien	50
21.	ANHANG: Syntax in Farbe	51
21.1	Notepad++	51
22.	ANHANG: Einstellungen zum Protokollfenster	51

1. Einleitung

QIK (Quality Improvement Keeper) ist eine Anwendung, die komplexe Testumgebungen in einem Testablauf zusammenführen. Basierend auf einer Ablaufdatei werden für ein Testobjekt Testfälle mit Funktionen definiert, die den Anwender durch den Testablauf führen.

In der Produktion ist das Testen selbsterklärend einfach, und bei Problemanalysen dienen die detaillierten Testschritt-Aufzeichnungen der Fehlerfindung.

2. Bedingungen

2.1 Lizenz

2.1.1 Bestimmungen

QIK ist eine lizenzpflichtige Software und darf ohne rechtmässig registrierte Lizenz nicht eingesetzt werden. Zur Evaluation kann die Anwendung ohne Lizenz geprüft werden.

2.1.2 Berechtigung

EDX Software Design überträgt dem Lizenznehmer das einfache und nicht übertragbare Recht, die Software für eigene Zwecke zu benutzen. Dem Lizenznehmer ist es nicht gestattet, die Software und zugehöriges Material an Dritte zu übergeben oder die Software auf EDV-Anlagen anderer Eigentümer zu verwenden.

2.1.3 Registrieren

Lizenzen können bei EDX Software Design (info@TheEDX.ch) angefordert werden.

2.2 Garantie

Die Software und Beispiele werden "wie besehen" ("AS IS"), ohne jede Gewähr bereitgestellt. EDX Software Design übernimmt keine Garantie dafür, dass die Software und Beispiele fehlerfrei laufen oder dass die Benutzung der Software den Anforderungen des Lizenznehmers genügen. Der Lizenznehmer ist verpflichtet, vorab die Eignung der Software für seine Zwecke zu prüfen.

2.3 Haftung

Jegliche Haftung für Schäden und Folgeschäden jeder Art, die dem Lizenznehmer direkt oder indirekt aus der Benutzung der Software entstehen, ist vollständig ausgeschlossen.

3. Installation

3.1 Betriebssystemanforderungen

Die Anwendung ist für Microsoft Windows Desktop und Server Betriebssysteme erstellt worden und wird bei Bedarf an die aktuellen Betriebssystem Versionen angepasst.

3.2 Dateien

Datei	Beschreibung
ReadMe_Vnnnnn.txt	Textdatei mit Informationen zu der gelieferten Anwendung und den Dateien.
QIK.exe	Anwendung
Registration.exe	Anwendung zur Registrierung
QDF_...Notepad++.xml	Notepad++ Konfigurationsdatei zur Syntax Einfärbung (siehe 21.1 Notepad++)
Ui Verzeichnis Ui\Ready.html Ui\ScriptError.html	Standard Dokument Anzeigen mit optionalen Musterdateien zum Anpassen: Ready.html wird beim Start der Applikation oder wenn eine neue Testdatei geladen worden ist angezeigt, sofern in der Testdatei keine Standard Dokumentation definiert ist. (Siehe 9.2 Set(DefaultDoc; "FileName")) ScriptError.html wird angezeigt, wenn eine Testdatei nicht erfolgreich geladen werden konnte.

3.3 Installation

QIK besteht aus einer ausführbaren Datei und muss nicht im System installiert werden. Schnittstellen und die Installation von Treibern sind nicht Bestandteil vom QIK. Dazu sind die Instruktionen des Hardware- und/oder Treiber-Herstellers zu berücksichtigen.

3.3.1 Portable Executable

Wenn Windows UAC nicht eingehalten werden muss, können die QIK Dateien in ein beliebiges Verzeichnis kopiert werden und das `QIK.exe` daraus gestartet werden. (z.B. `C:\Test`)

3.3.2 Windows UAC (User Account Control)

Nach Microsoft Windows Benutzerkontensteuerung sind ab Windows VISTA die ausführbaren Dateien von den Datendateien zu trennen. Ausführbare Dateien sind im geschützten Verzeichnis `C:\Program Files\` und Datendateien sind im offenen `C:\ProgramData\` oder in Benutzerverzeichnissen anzulegen. Die Bezeichnung der Verzeichnisse ist vom System und der Sprache abhängig. 32 Bit Applikationen auf 64 Bit Systemen werden beispielsweise unter `C:\Program Files (x86)\` abgelegt.

3.4 Programmparameter

Parameter	Beschreibung
+File:	Mit diesem Parameter wird die definierte QDF Datei beim Start der Anwendung geladen. Verzeichnisse und Dateinamen mit Leerzeichen müssen in Anführungszeichen definiert werden. Beispiele: <code>C:\Test\QIK.EXE +File:Test1.QDF</code> <code>C:\Test\QIK.EXE +File:"Test 1.QDF"</code> <code>C:\Test\QIK.EXE +File:"..\..\Test1.QDF"</code>
+AutoStart	Wenn mit +File eine QDF Datei beim Start geladen wird, kann mit diesem Parameter bestimmt werden, dass der Ablauf unmittelbar nach dem Laden gestartet wird. Beispiel: <code>C:\Test\QIK.EXE +File:"Test1.QDF" +AutoStart</code>

4. Registrierung

Die Lizenz wird mit der beiliegenden Applikation `Registration.exe` registriert. Die Applikation benötigt Administratorenrechte, da die Daten in der Windows Registry Datenbank für alle Benutzer erreichbar angelegt werden.

5. Benutzerschnittstelle

5.1 Übersicht



1. Dateibezeichnung

In der Testdatei kann eine Bezeichnung definiert werden. Die Version des Testablaufs wird in der Statuszeile angezeigt, wenn der Mauszeiger über der Testbezeichnung steht. Siehe dazu unter: 9.1 Set(FileInfo; "Name"; "Version")

2. Testschritte mit Statusanzeige

- Offen
- ◐ In Bearbeitung
- Ausgeführt
- ✓ Erfolgreich ausgeführt
- ✗ Antwort ausserhalb der definierten Bereichsgrenzen oder Fehler beim Ausführen
- Deaktiviert
- Funktionsattrappe eines nicht gefundenen Gerätes (Menü *Options*, *Skip Errors* aktiviert)

3. Testfallbezeichnung

Die Bezeichnung des aktuellen Testfalls in dem Testschritte ausgeführt werden.

4. Dokumentationsfenster

Beschreibung zum jeweiligen Testfall. Die Anwendung greift für die Anzeige der Dokumentation auf den im Windows System integrierten Browser zurück. Durch die System Browser Version sind die Möglichkeiten bestimmt.

Unter Windows 10 Pro 22H2 ist das immer noch der Internet Explorer IE 7.0.







(Siehe \2_Document\HtmlInterpreterInfo.qdf)

5. Protokoll


Aufzeichnung der Aktionen, Resultate und Auswertungen.

5.2 Prozesssteuerung

Im Menü und in der Funktionsleiste stehen verschiedene Funktionen zur Prozesssteuerung zur Verfügung.

Funktion	Beschreibung
 Reset	Testablauf zurücksetzen.
 Start	Testablauf zurücksetzen und beim ersten Testfall/Testschritt den Ablauf starten.
 Step	Der selektierte Testschritt wird ausgeführt. Konnte der Testschritt erfolgreich ausgeführt werden, wird der nächste Testschritt selektiert.
 Pause	Der aktuelle Testschritt wird abgebrochen und der Ablauf angehalten.
 Continue	Der Ablauf wird vom selektierten Testschritt gestartet.
 Stop	Der selektierte Testschritt wird abgebrochen und der Ablauf wird beim nächsten Final-Testfall fortgesetzt.

Mit **Ctrl+Left Click** auf Testfunktionen/Testfälle im Testschrittfenster können einzelne Testfunktionen und ganze Testfälle ausgeführt werden.

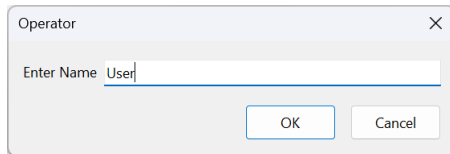
Mit  oder **Shift+Left Click** auf Testfunktionen/Testfälle im Testschrittfenster können einzelne Testfunktionen und ganze Testfälle deaktiviert respektive aktiviert werden.



Step-Modus wird beendet, wenn manuell ein Testpunkt selektiert wird.

6. Dialoge

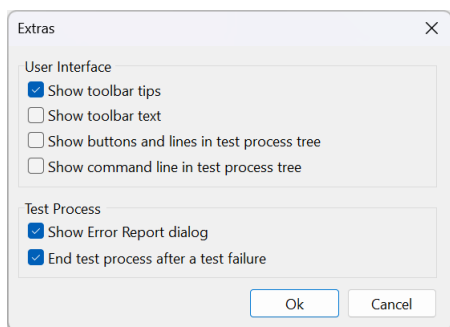
6.1 Operator



Menü: *Options, Set Operator Name...*

Der Operator wird beim Start des ersten Testablaufs abgefragt. Danach kann der Operator über das Menü *Options, Set Operator Name...* geändert werden.

6.2 Extras



Menü: *Options, Extras...*

Dialog zur Konfiguration der Benutzerschnittstelle und des Testablaufs. Die Einstellungen werden im Windows Benutzerkonto gespeichert.

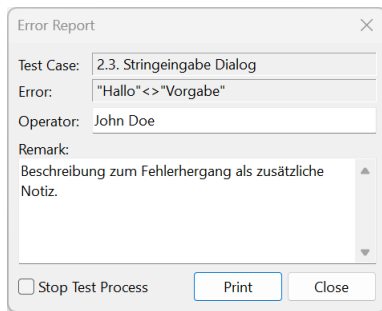
User Interface

- ☒ *Show toolbar tips*
In der Symbolleiste werden zusätzliche Hinweise zu den Funktionen unter dem Mauszeiger angezeigt.
- ☒ *Show toolbar text*
Wenn die Box angewählt ist, wird die Symbolleiste mit grossen Tasten und Text angezeigt.
- ☒ *Show button and lines in test process tree*
Wenn die Box angewählt ist, wird der Testablaufbaum mit Tasten und Gliederungslinien angezeigt.
- ☒ *Show command line in test process tree*
Wenn die Box angewählt ist, wird anstelle der definierten Namen der Testfunktionen die vollständige Definitionszeile angezeigt. Die Anzeige wird erst beim nächsten Laden der Definitionsdatei entsprechend aktualisiert.

Test Process

- ☒ *Show Error Report dialog*
Wenn die Box angewählt ist, wird unmittelbar bei einem Fehler der Error Dialog angezeigt. Ist die Box nicht angewählt, wird bei einem Fehler der Error Dialog nicht angezeigt.
Siehe dazu auch 11.2.2 ShowErrorReport("Name").
- ☒ *End test process after a test failure*
Wenn die Box angewählt ist, wird bei einem Fehler der Testablauf beendet, respektive den nächst folgenden Schlusstestfall `Test.Final` ausgeführt.
Diese Einstellung kann in der Definitionsdatei überschrieben werden mit 11.2.1 Process("Name"; Option).

6.3 Error Report



Menü: *View, Show Last Error Report*

Im Dialog *Error Report* können Details zum Fehler angezeigt, ergänzt und ausgedruckt werden.

Mit der Checkbox *Stop Test Process* kann der Testablauf beim Schliessen des Dialogs abgebrochen werden.

Der Dialog mit dem letzten Fehlerrapport des Testablaufs kann auch über das Menü: *View, Show Last Error Report* wieder geöffnet werden.

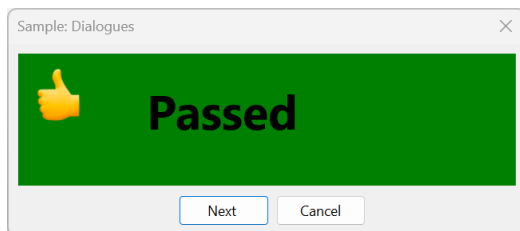
In der QDF Datei wird der *Error Report* mit dem Befehl `Set (ErrorReport...)` konfiguriert. Im Weiteren muss im Dialog *Extras* die Checkbox *Show Error Report dialog* aktiviert sein oder in der QDF Datei wird der Dialog konkret zum gewünschten Zeitpunkt aufgerufen mit `ShowErrorReport`.

Siehe dazu das Beispiel `..\System\ProcessTerms&ErrorReport.qdf` und die entsprechenden Schreibungen:

9.7 `Set(ErrorReport; "Template"; "Printer"; Orientation; Left; Top; Right; Bottom)`

11.2.2 `ShowErrorReport("Name")`.

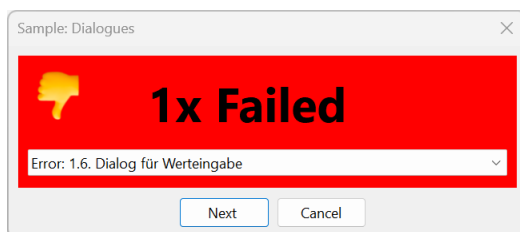
6.4 Testabschluss



Nachdem der letzte Testschritt ausgeführt worden ist, wird der Abschlussdialog angezeigt.

Next: Nächsten Testablauf starten.

Cancel: Das Testen beenden.



Wenn ein Testdurchlauf nicht einwandfrei beendet werden konnte, wird eine Einblendliste mit den bemängelten Testschritten angezeigt.

Next: Nächsten Testablauf starten.

Cancel: Das Testen beenden.

7. Testablaufdefinition

7.1 Syntax Übersicht

7.1.1 Testmodule

In einer Testdatei können weitere Testdateien wie Textbausteine eingebettet werden. Es besteht so die Möglichkeit wieder verwendbare Testabläufe in Test-Module auszulagern.

7.1.2 Deklarationen

In Testablaufdateien können verschiedene Angaben zur Datei, globale Variablen und Funktionalitäten deklariert werden.

7.1.3 Geräte

Eine Testumgebung besteht aus verschiedenen Geräten, die den Prüfling mit Stimulsignalen oder Daten konditioniert um Messungen durchzuführen oder Daten abzufragen.

In der Testablaufdefinition wird die Testumgebung mit Geräteobjekten so genannten *Devices* aufgebaut. Jedes im Testablauf verwendete Gerät (Ausgabe, Schnittstellen, Peripherie Geräte, usw.) wird durch ein Geräteobjekt repräsentiert. In der Deklaration eines Gerätes wird ein eindeutiger Name zugewiesen, über den die Funktionen des Gerätes im ganzen Testablauf aufgerufen werden.

7.1.4 Testfälle

In Testfällen werden einzelne Testschritte oder weitere Testfälle in geschweiften Klammern zusammengefasst.

7.1.5 Testschritte


Testschritte sind einzelne Testfunktionen oder mehrere Testfunktionen in einem Testfall zusammengefasst.

7.1.6 Testfunktionen

Die für den Testablauf deklarierten Geräte stellen verschiedene Funktionen zur Verfügung. Über den definierten Namen des Gerätes werden die Funktionen aufgerufen.

7.2 Interpreter

Beim Laden einer Definitionsdatei werden im QIK Objekt Strukturen aufgebaut, assoziiert und initialisiert. Der Interpreter unterscheidet grundsätzlich zwischen Gross-/Kleinschreibung wobei fehlerhafte oder unbekannte Befehle nach dem Einlesen in einem Fehlerdialog und im Protokollfenster angezeigt werden.

Eine Definitionsdatei kann nur vollständig geladen und ausgeführt werden, wenn keine Fehlermeldungen nach dem Einlesen vorhanden sind. Wenn das Menü *Options, Skip Errors* aktiviert ist, werden Funktionen von unbekannten Geräten ohne Fehlermeldung mit  gekennzeichnet und im Ablauf ignoriert. Unbekannte oder fehlerhaft definierte Funktionen werden mit einer Fehlermeldung im Protokoll ausgewiesen aber nicht in den Ablauf geladen.

Kommentarbereiche in `/*...*/` eingeschlossen oder Zeilenkommentare mit `//` werden vom Interpreter ignoriert. Kommentarzeichen werden auch in Zeichenketten erkannt und kommentieren den entsprechenden Inhalt aus.

7.3 Konventionen dieser Beschreibung

Typ	Beschreibung								
Str	Zeichenkette. Eine Zeichenkette wird immer in Anführungszeichen (") angeben. Die Länge solcher Zeichenketten ist nicht explizit definiert.								
Str[N]	Zeichenkette mit N Zeichen. Eine Zeichenkette wird immer in Anführungszeichen (") angeben, wobei die Länge genau der Anzahl N Zeichen entsprechen muss.								
StrExt	<p>Zeichenkette in der ESC-Sequenzen interpretiert werden können. Eine ESC-Sequenz wird durch ein <i>BackSlash</i> eingeleitet, gefolgt von einem hexadezimalen Wert für ein Byte oder einem der vordefinierten Tags:</p> <p>Definierte Tags und HEX-Beispiele:</p> <ul style="list-style-type: none"> • \STX = \02 = 0x02 • \ETX = \03 = 0x03 • \EOT = \04 = 0x04 • \ENQ = \05 = 0x05 • \AKN = \06 = 0x06 • \NAK = \07 = 0x07 • \r = \0D = 0x0D • \n = \0A = 0x0A 								
Pfad	<p>Absoluter Pfad: Vollständiger Pfad mit Laufwerk, Pfad und Dateiname. "c:\Dir1\Datei.xyz"</p> <p>Relativer Pfad: Standard Pfadangabe relative zur Anwendungsdatei</p> <table> <tr> <td>Anwendung</td> <td>c:\Dir1\Dir2\Dir3\QIK.exe</td> </tr> <tr> <td>"\Datei.xyz"</td> <td>c:\Datei.xyz</td> </tr> <tr> <td>".\Datei.xyz"</td> <td>c:\Dir1\Dir2\Dir3\Datei.xyz</td> </tr> <tr> <td>"..\Datei.xyz"</td> <td>c:\Dir1\Dir2\Datei.xyz</td> </tr> </table> <p>Relativer QIK Pfad: Diese Definition entspricht dem relativen Standard Pfad oben "QIK:.\Datei.xyz"</p> <p>Relativer QDF Pfad: Standard Pfadangabe relative zur Hauptdefinitionsdatei QDF "QDF:.\Datei.xyz"</p>	Anwendung	c:\Dir1\Dir2\Dir3\QIK.exe	"\Datei.xyz"	c:\Datei.xyz	".\Datei.xyz"	c:\Dir1\Dir2\Dir3\Datei.xyz	"..\Datei.xyz"	c:\Dir1\Dir2\Datei.xyz
Anwendung	c:\Dir1\Dir2\Dir3\QIK.exe								
"\Datei.xyz"	c:\Datei.xyz								
".\Datei.xyz"	c:\Dir1\Dir2\Dir3\Datei.xyz								
"..\Datei.xyz"	c:\Dir1\Dir2\Datei.xyz								
IP	<p>IP-Adresse als Name oder in IPv4 Notation mit Port</p> <p>Syntax: Host IPv4:Port Beispiele: raspberrypi:55001 192.168.1.23:55001</p>								
Param	Parameter als Folge von Bedingungen von Drittanwendungen die Zeichen oder Zeichenketten mit oder ohne Anführungszeichen durch Separatoren getrennt. (Strichpunkt kann nicht als Separator genutzt werden)								
Label	Schlüsselwörter und Definitionen der Anwendung oder deklarierte Bezeichnungen in der Definitionsdatei, die die Anwendung kennt und zuordnen kann. (z.B. Gerätetypen, Funktionsarten, Gerätebezeichnungen, Variablen usw.).								
Wert	Zahl								
Wert[d]	dezimale Zahl								
Wert[h]	hexadezimale Zahl								
N	Natürliche Zahl $N = \{0, 1, 2, \dots\}$								
G	Ganze Zahl $G = \{\dots, -2, -1, 0, 1, 2, \dots\}$								
R	Rationale Zahl $R = a/b$ ($a, b \in G$)								
Term	Ausdruck, der Zahlen, Variablen und Symbole einer mathematischen Verknüpfung								

8. Präprozessor

Das Laden von Testdateien kann mit Direktiven beginnend mit "#" und QDF Funktionen beginnend mit zwei Unterstrichen "__" beeinflusst werden.

8.1 Direktiven

Bei den Schlüsselwörtern der Direktiven wird die Gross-/Kleinschreibung vom Interpreter ignoriert.

8.1.1 #Include "FileName"

Bezeichnung	Typ	Beschreibung
FileName	Pfad	Dateiname einer Datei im Verzeichnis oder Unterverzeichnis ausgehen von der Haupt-Definitionsdatei

Mit der Direktive `#Include` können bestehende Definitionsdateien wie Textbausteine eingefügt werden. Die Verschachtelungstiefe ist auf 20 begrenzt.

Beispiel: `..\X_PreProcessor\Include0.qdf`

8.1.2 #Define Identifier TokenString

Bezeichnung	Typ	Beschreibung
Identifier	Label	Platzhalter
TokenString	-	Zeichenkette die durch den Platzhalter in der QDF Datei eingesetzt wird. Ohne Zeichenkette wird der Platzhalter nicht ersetzt.

Direktiven mit `#Define` sind ab der Definition für alle zu ladenden Dateien gültig und werden beim Einlesen durch die Zeichenketten ersetzt. Definitionen mit gleichem Platzhalter werden überschreiben. Definitionen ohne *TokenString* werden nicht ersetzt, sondern dienen dem bedingten Laden mit `#ifdef`.



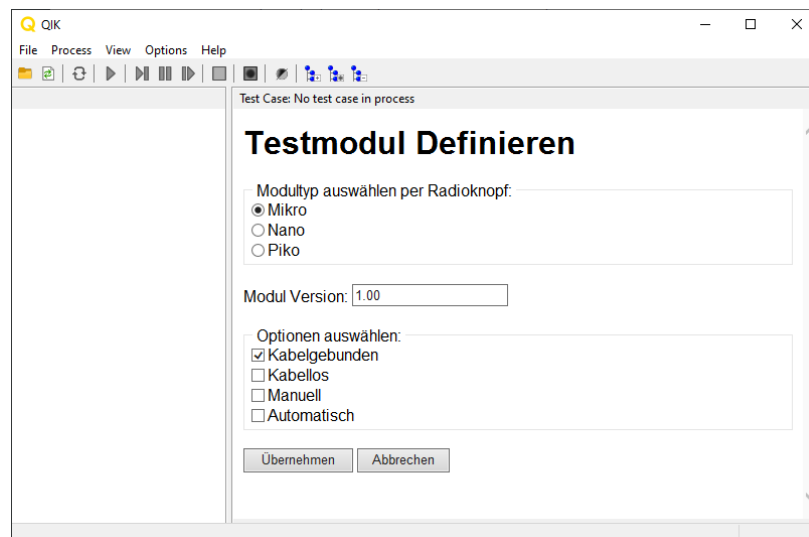
Im Gegensatz zu Programmiersprachen wie C oder C++ sind die Platzhalter respektive die TokenStrings an keine Syntax gebunden und können frei eingesetzt werden.

Beispiel: `..\X_PreProcessor\Defines.qdf`

8.1.3 #ShowDefineDoc FileName[URL Parameter]

Bezeichnung	Typ	Beschreibung
FileName	~Pfad	Pfad mit HTML Dateiname ohne Anführungszeichen mit optionalem URL ähnlichen Parametrierung nach Fragezeichen. Beispiele: <code>#ShowDefineDoc QDF:..\Doc\DefinesFromDoc.html</code> <code>#ShowDefineDoc QDF:..\Doc\DefinesFromDoc.html?version=1.00&manual=true</code>

Mit dieser Direktive können während dem Laden der QDF Datei Definitionen ähnlich wie mit `#Define` angelegt werden. In HTML stehen Testboxen, Aufklapplisten, Checkboxes, Radio-Buttons und weitere Möglichkeiten zur Verfügung.



..\X_PreProcessor\Doc\DefinesFromDoc.html

Das Laden der QDF Datei wird angehalten und die definierte Datei wird im Dokumentationsfenster angezeigt. Über einen HTML Link mit dem Präfix `>InitDefineValues>` und Schlüsselwort wird der Ablauf fortgesetzt.

Schlüsselwort Cancelled

Mit diesem Schlüsselwort wird das Laden der QDF Datei mit einer entsprechenden Meldung abgebrochen.

`>InitDefineValues>Cancelled`

Schlüsselwort Parameter

Mit diesem Schlüsselwort werden die neuen Definitionen übergeben. Eine Definition besteht aus einem Identifier und optional mit Gleichheitszeichen zugewiesenem Wert oder Token String. Mehrere Definitionen werden mit "&" getrennt aufgelistet.

`>InitDefineValues>Parameter:ModulTyp=Nano&ModulVersion=1.00&ModulOptionA&ModulOptionC`



Die Übergabe der Definitionen basiert auf dem URL Link Konzept. URL spezifische Zeichen, Umlaute und Sonderzeichen sollten in den Definitionen nicht verwendet werden.

Das Fragezeichen "?" kann nicht eingesetzt werden!

Beispiel: `..\X_PreProcessor\DefinesFromDoc.qdf`

8.1.4 Bedingtes Laden

Die mit `#Define` und `#ShowDefineDoc` erstellten Identifier und Token, können für das bedingte Laden von QDF Dateien genutzt werden. Im Protokollfenster werden die aufgelösten Bedingungen als Trace ausgegeben.

```
DefinesFromDoc.qdf, 38: Condition TRUE for: «#if 'QDF_ModuleTypeMicro_' = 'DOC_ModuleType_'» resolved «#if 'Mikro' = 'Mikro'»
DefinesFromDoc.qdf, 40: Condition FALSE for: «#if DOC_ModuleVersion_ < QDF_ModuleVersion100_» resolved «#if 1.00 < 1.00»
DefinesFromDoc.qdf, 43: Condition TRUE for: «#if DOC_ModuleVersion_ < QDF_ModuleVersion150_» resolved «#if 1.00 < 1.50»
DefinesFromDoc.qdf, 53: Condition TRUE for: «#ifdef DOC_ModuleOptionWired_»
DefinesFromDoc.qdf, 56: Condition TRUE for: «#ifdef DOC_ModuleOptionWireless_»
DefinesFromDoc.qdf, 59: Condition FALSE for: «#ifdef DOC_ModuleOptionManual_»
DefinesFromDoc.qdf, 62: Condition FALSE for: «#ifdef DOC_ModuleOptionAuto_»
```


8.1.4.1 #if Condition, #elif Condition, #else und #endif

Als Konditionen können einfache Terme mit folgenden Operatoren ausgewertet werden. Auch Zeichenketten in einfachen Anführungszeichen können ausgewertet werden.

Rechenoperatoren: * / + -

Vergleichsoperatoren: > < <> = <= >=

Beispiel: ..\X_PreProcessor\ConditionalLoad.qdf
 ..\X_PreProcessor\DefinesFromDoc.qdf

8.1.4.2 #ifdef Identifier, #ifndef Identifier, #elifdef Identifier, #else und #endif

Wenn nur auf die Existenz eines Identifier einer Definition zu prüfen ist, kann dies mit #ifdef erledigt werden.

Beispiel: ..\X_PreProcessor\ConditionalLoad.qdf
 ..\X_PreProcessor\DefinesFromDoc.qdf

8.2 QDF-Funktionen

Diese Funktionen werden vor den Direktiven für bedingtes Laden ausgeführt und können für diese genutzt werden.

8.2.1 __PrinterInstalled("Printer Name")

In der Liste der installierten Drucker, die online sind, wird nach dem übergebenen *Printer Namen* gesucht. Wenn der Drucker vorhanden ist, wird "true" retourniert, das mit der Direktive #if ausgewertet werden kann.

Beispiel: ..\X_PreProcessor\PrinterInstalled.qdf

8.2.2 __ShowWarning("Message")

Warnmeldung im Protokoll Fenster ausgeben. Beim bedingten Laden kann beispielsweise auf nicht erfüllte Konditionen hingewiesen werden.

Beispiel: ..\X_PreProcessor\PrinterInstalled.qdf

8.2.3 __AbortWithMessage("Message")

Laden der Testdatei mit der übergebenen Meldung abbrechen. Beim bedingten Laden können beispielsweise unabdingbare Voraussetzungen durchgesetzt werden.

Beispiel: ..\X_PreProcessor\PrinterInstalled.qdf

9. Deklarationen

9.1 Set(FileInfo; "Name"; "Version")

Bezeichnung	Typ	Beschreibung
FileInfo	Label	Schlüsselwort der Anwendung
Name	Str	Bezeichnung des Testablaufs
Version	Str	Version des Testablaufs

Mit der Deklaration `FileInfo` kann dem Testablauf eine Bezeichnung und Version zugewiesen werden. Die Bezeichnung wird in der Benutzeroberfläche über den Baum mit Testfällen angezeigt. Die Version wird in der Statuszeile angezeigt, wenn der Mauszeiger über der Testbezeichnung steht.

9.2 Set(DefaultDoc; "FileName")

Bezeichnung	Typ	Beschreibung
DefaultDoc	Label	Schlüsselwort der Anwendung
FileName	Pfad	Dateiname der Standard Dokumentation der Testdatei.

Wenn definiert, wird sobald die Testdatei geladen ist, die Standard Dokumentation der Testdatei angezeigt.

Beispiel: `..\2_Document\DocumentDisplay.qdf`

9.3 Set(DecimalSeparator; "Character")

Bezeichnung	Typ	Beschreibung
DecimalSeparator	Label	Schlüsselwort der Anwendung
Character	Str[1]	Zeichen das in einer Zahl den ganzzahligen vom fraktionalen Teil trennt. Punkt "." oder Komma ",".

Die Anwendung übernimmt grundsätzlich die Einstellung vom Betriebssystem. Mit der Deklaration `DecimalSeparator` kann das Dezimaltrennzeichen in der Anwendung bestimmt werden. Das ermöglicht, QDF Dateien und Geräteschnittstellen unabhängig von den Regionaleinstellungen des Betriebssystems einsetzen zu können.

Beispiel: `..\1_System\NumberSeparator.qdf`



Die Deklaration setzt die Einstellung in der Anwendung, bis die Einstellung überschrieben oder die Anwendung geschlossen wird.



Weitergehende Informationen:

<https://de.wikipedia.org/wiki/Dezimaltrennzeichen>

https://de.wikipedia.org/wiki/Schreibweise_von_Zahlen

9.4 Set(ThousandSeparator; "Character")

Bezeichnung		Typ	Beschreibung
ThousandSeparator		Label	Schlüsselwort der Anwendung
Character		Str[1]	Zeichen das zum Aufteilen von Tausenderwerten in Zahlen mit mehr als drei links vom Dezimalzeichen stehenden Ziffern verwendet wird. Apostroph "'", Leerzeichen " ", Komma "," oder Punkt ".".

Die Anwendung übernimmt grundsätzlich die Einstellung vom Betriebssystem. Mit der Deklaration `ThousandSeparator` kann die Zifferngruppierung in der Anwendung bestimmt werden. Das ermöglicht, QDF Dateien und Geräteschnittstellen unabhängig von den Regionaleinstellungen des Betriebssystems einsetzen zu können.

Beispiel: `..\1_System\NumberSeparator.qdf`



Die Deklaration setzt die Einstellung in der Anwendung, bis die Einstellung überschrieben oder die Anwendung geschlossen wird.



Weitergehende Informationen:

<https://de.wikipedia.org/wiki/Zifferngruppierung>

https://de.wikipedia.org/wiki/Schreibweise_von_Zahlen

9.5 Set(KeyCodeToFunction; KeyCode; "Function")

Bezeichnung	Typ	Beschreibung
KeyCodeToFunction	Label	Schlüsselwort der Anwendung
Name	N	Tastencode Der Tastencode kann aus dem Protokollfenster entnommen werden, wenn im Menü <i>View, Show Key Codes</i> aktiviert ist.
Function	Str	Funktionsbezeichnung entsprechend der im Menüpunktamen oder der folgende: "Select Tree" Baumansicht aktivieren für das Navigieren mit den Pfeilen

Beispiel: `..\1_System\KeyboardKeyAssignment.qdf`

9.6 Set(ValueLog; "FileName"[]; "StartDateTimeFmt"; "FileDateFormat"])

Bezeichnung	Typ	Beschreibung
ValueLog	Label	Schlüsselwort der Anwendung
FileName	Pfad	Pfad und Name der Protokolldatei. Im Dateinamen kann mit dem Symbollabel %DATE das Datum integriert werden.
StartDateTimeFmt	Str	Datum-Zeit-Format der Start-Spalte. (optional) Standard: <code>yyyy-mm-dd hh:nn:ss</code>
FileDateFormat	Str	Datum-Format für das Symbollabel %DATE im ValueLog Dateinamen. (optional) Standard: <code>YYMMDD</code>

Mit der Deklaration `ValueLog` kann eine Protokolldatei zum Testablauf für die Werte vom Prüfling definiert werden. Bei den folgenden Testfunktionen kann der Protokollspaltenname mit einem § zum Testfunktionsnamen definiert werden.

- Device SystemControl: AssignValuelog
- Device SystemControl: Dialog.Input.String
- Device SystemControl: Dialog.Input.Value
- Device SystemControl: Calculate
- Device SystemControl: GetNextID

- Device SystemControl: RegisterID
- Device Port: Send
- Device Port: WaitForResponse
- Device Port: SendJson
- Device Port: WaitForJson
- Device Port: Port.Evaluate.Value
- Device Instrument: Check
- Device AudioInput: Check.RMS
- Device AudioInput: Start.AVG.Check



Um die CSV Struktur in der Protokolldatei sicherstellen zu können, werden Steuerzeichen (Ordinalwerte < 0x20) durch Fragezeichen und Strichpunkt durch Komma ersetzt.

Beispiel: ..\1_System\ValueLog.qdf
 ..\1_System\ValueLogEx.qdf

9.7 Set(ErrorReport; "Template"; "Printer"; Orientation; Left; Top; Right; Bottom)

Bezeichnung	Typ	Beschreibung
ErrorReport	Label	Schlüsselwort der Anwendung
Template	Pfad	Dateiformatvorlage siehe 19. ANHANG: RTF Vorlagen Spezielle Platzhalter: %FILENAME QDF Dateiname vollständig mit Laufwerk und Pfad %FILEINFO Bezeichnung des Testablaufs gemäss Set FileInfo %FILEVERSION Version des Testablaufs gemäss Set FileInfo %FUNCTION Nummer und Bezeichnung der Testfunktion %ERRORTEXT Fehlertext der fehlgeschlagenen Testfunktion %DATE Datum %TIME Zeitstempel %OPERATOR Bediener %REMARK Im Fehlerdialog erfasste Bemerkung
Printer	Str	Name des Druckers
Orientation	Label	Portrait Ausrichtung Hochformat Landscape Ausrichtung Querformat
Left Top Right Bottom	R	Ränder in Millimeter innerhalb des vom Drucker bestimmten Druckbereichs.

Mit der Deklaration `ErrorReport` wird der Fehlerdialog eingerichtet und die Druckereinstellungen initialisiert. Wenn eine Testfunktion einen Wert ausserhalb des erwarteten Wertebereichs erhält, wird der Fehlerdialog angezeigt. Siehe 6.3 Error Report. Der Bediener kann eine Bemerkung erfassen und den Fehlerrapport basierend auf der definierten Formatvorlage drucken.

Beispiel: ..\1_System>ErrorReport.qdf

9.8 Set(DataType; VariableName)

Bezeichnung	Typ	Beschreibung
DataType	Label	<code>Integer</code> Ganze Zahl (Initialisiert mit 0)
		<code>Float</code> Fließkommazahl (Initialisiert mit 0)
		<code>String</code> Zeichenkette (Initialisiert mit einem Leerstring)
VariableName	Label	Beliebiger, eindeutiger Name der Variable. (Gross- Kleinschreibung wird ignoriert)

Mit der Deklaration `DataType` wird eine Variable vom entsprechenden Datentyp eingerichtet.

Beispiele: `..\1_System\DialoguesAndVariables.qdf`
`..\1_System\ID_LoadAndRegister.qdf`
`..\4_Port\Extended.qdf`

10. Testfälle

Testfälle werden mit einer beliebigen Bezeichnung in Klammern definiert. In geschweiften Klammern werden anschliessend untergeordnete Testfälle und Testfunktionen zusammengefasst.

10.1 Test.Case("Name"[; Appearance])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung des Testfalls
Appearance	Label	Optionale Definition ob die enthaltenen Testfälle und Testfunktionen in der Standardansicht der Testfallliste angezeigt werden. <ul style="list-style-type: none">ExpandedCollapsed (Standard)

10.2 Test.Final("Name"[; Appearance])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung des Testfalls
Appearance	Label	Optionale Definition ob die enthaltenen Testfälle und Testfunktionen in der Standardansicht der Testfallliste angezeigt werden. <ul style="list-style-type: none">ExpandedCollapsed (Standard)

Wenn die Ablaufkondition `BreakOnFailure` aktiviert ist, wird bei einem Fehler der Testablauf abgebrochen und beim nächsten `Test.Final` fortgesetzt.

Beispiel: `..\1_System\ProcessTerms.qdf`

11. Device: System Control

11.1 Device.Create(SystemControl; Naming)

Bezeichnung	Typ	Beschreibung
SystemControl	Label	Schlüsselwort der Anwendung
Naming	Label	Eine im ganzen Testablauf eindeutige Benennung für dieses Gerät

11.2 Funktionen

Allgemeine Funktionen der Applikation werden unter dem Gerät `SystemControl` zur Verfügung gestellt. Die Funktionen werden einleitend mit `Naming` Label aufgerufen: z.B. `Sys`.

```
Device.Create(SystemControl; Sys)

Sys.Dialog("Dialog Ok"; "Dialog Ok"; "Dialog ..."; ButtonOk)
Sys.Execute("Batch asynchron starten"; "QDF:\TestBatch.bat"; ; Normal; -1)
Sys.Pause("Weiter mit ENTER, abbrechen mit ESC")
Sys.Delay("Warten bis offen"; 500);
```

11.2.1 Process("Name"; Option)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Option	Label	<code>ContinueOnFailure</code> Die Testschritte werden Schritt für Schritt ausgeführt auch bei einem Fehler.
		<code>BreakOnFailure</code> Bei einem Fehler wird der Testablauf abgebrochen respektive beim nächsten Final-Testfall fortgesetzt.
		<code>ExitAfterFailure</code> Falls im Testablauf ein Fehler auftritt, wird der Testablauf an dieser Stelle beendet.

Beispiel: `..\1_System\ProcessTerms.qdf`

11.2.2 ShowErrorReport("Name")



Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

Wenn im Dialog Extras die Checkbox *Show Error Report dialog* angewählt ist, wird der *Error Report* Dialog unmittelbar bei einem Fehler angezeigt. Ist die Checkbox nicht angewählt, kann mit *ShowErrorReport* bestimmt werden, wo im Testablauf der *Error Report* Dialog angezeigt werden soll. Falls ein Fehler noch nicht angezeigt worden ist, wird dieser sicher vor dem Abschlussfenster angezeigt.

Siehe dazu auch die Beschreibungen unter
 6.2 Extras Checkbox *Show Error Report dialog*
 6.3 Error Report

Beispiel: `..\1_System\ProcessTerms&ErrorReport.qdf`

11.2.3 SetTestCaseStatus("Name"; "TestCase"; Status)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
TestCase	Str*	Die Bezeichnung des Testfalls dessen Status zu setzen ist.
Status	Label	<div>Enabled Testschritt auf Status "offen"  setzen.</div> <div>Disabled Testschritt auf Status "deaktiviert"  setzen.</div>

* Möglichkeit Variablen mit % einzusetzen



Testfälle können auch manuell mit **Shift+Left Click** deaktiviert respektive aktiviert werden.

Beispiel: ..\1_System\SetTestCaseStatus.qdf

11.2.4 Goto("Name"; "TestCase"[; Condition])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
TestCase	Str	Bezeichnung des Testfalls der angesprungen werden soll.
Condition	Term*	Optionale Kondition zur Ausführung des Sprungbefehls: <div>✓ Kondition erfüllt und TestCase angesprungen.</div> <div>● Kondition nicht erfüllt, nächster Testschritt wird ausgeführt.</div> <div>✗ Testschrittbezeichnung nicht gefunden.</div>

* Möglichkeit Variablen mit % einzusetzen

Beispiel: ..\2_Dokument\TestCaseSelection.qdf

11.2.5 Delay("Name"; Time)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Time	N	Testablauf für die definierte Zeit in Millisekunden anhalten

11.2.6 Pause("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion

Der Testablauf wird angehalten, bis der Anwender eine der definierten Tasten auf dem Keyboard drückt.

ENTER: Die Eingabetaste setzt den Ablauf fort.

ESC: Die Abbruchtaste bricht den Ablauf ab.

Beispiel: ..\2_Document\DocumentDisplay.qdf

11.2.7 Break("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion

Der Ablauf wird abgebrochen um beispielsweise via Link aus dem Dokument einen spezifischen Testfall anzuspringen.

Beispiel: ..\2_Document\SelectTestItem.qdf

11.2.8 Beep("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion

Akustischer Hinweis um die Aufmerksamkeit des Bedieners zu erreichen.

11.2.9 Dialog("Name"; "Title"; "Text"; Button)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Title	Str	Dialog Titel
Text	Str	Dialog Text und der Möglichkeit Variablen mit % anzuzeigen
Button	Label	Tasten im Dialog <ul style="list-style-type: none"> • ButtonOk • ButtonOkCancel • ButtonYesNo

Der Ablauf wird angehalten und der definierte Dialog wird angezeigt. Nach dem Bestätigen des Dialogs wird der Ablauf fortgesetzt respektive abgebrochen.

Ok: Ablauf fortsetzen

Ok, Cancel: Ablauf fortsetzen, Ablauf abbrechen (ENTER-Taste = Ok, ESC-Taste = Cancel)

Yes, No: Ablauf fortsetzen, Ablauf abbrechen

Beispiele: ..\1_System\Dialogues.qdf
 ..\1_System\DialoguesAndVariables.qdf

11.2.10 Dialog.Input.String("Name"; "Title"; "Text"; "Default"; "Expected"[; Variable])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Title	Str	Dialog Titel
Text	Str*	Dialog Text
Default	Str*	Vorgabe in der Textbox
Expected	Str	Erwartete Zeichenkette Die Zeichenkette kann mit Wildcard * und ? ausmaskiert werden.
Variable	Label	Optional: Name der Integer, Float oder String Variable in der die Eingabe gespeichert werden soll.

* Möglichkeit Variablen mit % einzusetzen

Der Ablauf wird angehalten, so dass im Dialog eine Eingabe gemacht werden kann.

Ok: Eingabe mit der Expected Zeichenkette vergleichen und auswerten.

Cancel: Standardzeichenkette wird mit der Expected Zeichenkette verglichen und auswerten.

Beispiele: ..\1_System\Dialogues.qdf
 ..\1_System\DialoguesAndVariables.qdf

11.2.11 Dialog.Input.Value("Name"; "Title"; "T1"; "T2"; "Default"; Min; Max; [, Variable])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Title	Str	Dialog Titel
T1	Str*	Dialog Text
T2	Str*	Dialog Text der nach einer nicht interpretierbaren Eingabe ausgegeben wird
Default	Str*	Vorgabe in der Textbox
Min	G R	Minimal Wert der Eingabe (Erfolgreich = Eingabe ≥ Min)
Max	G R	Maximal Wert der Eingabe (Erfolgreich = Eingabe ≤ Max)
Variable	Label	Optional: Name der Float-Variable in der die Eingabe gespeichert werden soll.

* Möglichkeit Variablen mit % einzusetzen

Der Ablauf wird angehalten, so dass im Dialog ein Wert eingegeben werden kann.

Ok: Eingabewert mit Min und Max vergleichen und auswerten.

Cancel: Vorgabewert mit Min und Max vergleichen und auswerten.

Beispiele: ..\1_System\Dialogues.qdf
 ..\1_System\DialoguesAndVariables.qdf

11.2.12 AssignValueLog("Name" §Spalte; Value; Option)

Bezeichnung	Typ	Beschreibung
"Name" §Spalte	Str	Bezeichnung der Funktion im Testablauf in Anführungszeichen und Spaltenname im ValueLog mit Paragraphenzeichen.
Value	Str* Wert*	Text oder Wert der im ValueLog gesetzt werden soll.
Option	Label	MultipleUse Für alternative Testabläufe können ValueLog Spalten mehrfach definiert werden ohne eine Warnmeldung auszulösen.

* Möglichkeit Variablen mit % einzusetzen

Beispiel: ..\1_System\ValueLog.qdf
 ..\2_Dokument\TestCaseSelection.qdf

11.2.13 AssignVariable("Name"; Value; VariableName)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Value	Str* Wert*	Text oder Wert der in der definierten Variable gesetzt werden soll. Die "Value" muss mit dem Datentyp der Variable übereinstimmen.
VariableName	Label	Name der Variable in der die "Value" gespeichert werden soll.

* Möglichkeit Variablen mit % einzusetzen

Beispiel: ..\1_System\AssignVariable.qdf
 ..\1_System\DialoguesAndVariables.qdf

11.2.14 AssignVariableEx("Name"; Value; Variable)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Value	StrExt* Wert*	Text, Zeichenkette mit ESC-Sequenzen oder Wert der in der definierten Variable gesetzt werden soll. Die "Value" muss mit dem Datentyp der Variable übereinstimmen.
Variable	Label	Name der Variable in der die "Value" gespeichert werden soll.

* Möglichkeit Variablen mit % einzusetzen

Beispiel: ..\1_System\AssignVariableEx.qdf

11.2.15 AssignVarConditionally("Name"; Condition, TrueValue; FalseValue; Variable)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Condition	Term*	Kondition mit logischen Verknüpfungen Vergleichsoperatoren: > < <> = <= >= Verknüpfungen and or xor not
TrueValue	Str* Wert*	Text oder Wert der in die definierte Variable gesetzt werden soll. Die "TrueValue" muss mit dem Datentyp der Variable übereinstimmen.
FalseValue	Str* Wert*	Text oder Wert der in die definierte Variable gesetzt werden soll. Die "FalseValue" muss mit dem Datentyp der Variable übereinstimmen.
Variable	Label	Name der Variable in der "TrueValue" respektive "FalseValue" zugewiesen wird.

* Möglichkeit Variablen mit % einzusetzen

Beispiel: ..\1_System\AssignVarConditionally.qdf

11.2.16 VarIntToStr("Name"; iVariable; "Format"; sVariable)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
iVariable	Label	Name einer deklarierten Ganzzahl Variable (Quelle)
Format	Str	Definition der Konvertierung %d Zahl im dezimalen Format %x Zahl im hexadezimalen Format %.4x 4 Hex Zeichen mit führenden Nullen %.5d 5 Dez Digits mit führenden Nullen ...
sVariable	Label	Name der String Variable in der die konvertierte Zahl gespeichert werden soll. (Ziel)

Integer Variable in eine String Variable konvertieren nach der Definition im Format.

Beispiel: ..\1_System\DialoguesAndVariables.qdf

11.2.17 VarStrToStr("Name"; sVariable1; "Format"; sVariable2)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
sVariable1	Label	Name einer deklarierten String Variable (Quelle)
Format	Str	Formatierung der Zeichenkette Die Zeichen der sVariable1 werden an den Platzhaltern # eingesetzt.
sVariable2	Label	Name der String Variable in der die formatierte Zeichenkette gespeichert werden soll. (Ziel)

String Variable in eine String Variable nach der Definition im Format an den Platzhaltern abfüllen.

Beispiel: ..\1_System\DialoguesAndVariables.qdf

11.2.18 Calculate("Name"; Term; Variable)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Term	Term*	Arithmetischer Ausdruck Rechenoperatoren: * / div mod + -
Variable	Label	Name einer Integer-, Float- oder String-Variable in der das Resultat gespeichert werden soll. Für Integer-Variablen wird das Resultat mathematisch gerundet.

* Möglichkeit Variablen mit % einzusetzen

Beispiele in: ..\1_System\CalculationAndEvaluation.qdf

11.2.19 Evaluate("Name"; Condition)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Condition	Term*	Ausdrücke zum Vergleich und mit logischen Verknüpfungen zur Auswertung auf "wahr" ✓ und "falsch" ✗. Vergleichsoperatoren: > < <> = <= >= Verknüpfungen and or xor not

* Möglichkeit Variablen mit % einzubinden

Beispiele in: ..\1_System\CalculationAndEvaluation.qdf

11.2.20 GetNextID("Name"; "FileName"; Variable)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
FileName	Pfad	Pfad und Name der ID Quelldatei siehe 20. ANHANG: ID Dateien
Variable	Label	Name einer deklarierten Integer Variable

Aus der in FileName definierten ID Datei wird die nächste ID Zahl ermittelt und in die Variable geschrieben.

Beispiel: ..\1_System\ID_LoadAndRegister.qdf

11.2.21 RegisterID("Name"; "FileName"; Variable)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
FileName	Pfad	Pfad und Name der ID Log-Datei siehe 20. ANHANG: ID Dateien
Variable	Label	Name einer deklarierten Integer oder String Variable

Die ID aus der Variable wird in der Log-Datei auf Eindeutigkeit geprüft und gespeichert.

Beispiel: ..\1_System\ID_LoadAndRegister.qdf

11.2.22 Print("Name"; "Template"; "Printer"; Orientation; L; T; R; B)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Template	Pfad	Dateiformatvorlage siehe 19. ANHANG: RTF Vorlagen
Printer	Str	Name des Druckers
Orientation	Label	Portrait Ausrichtung Hochformat Landscape Ausrichtung Querformat
L Left T Top R Right B Bottom	R	Ränder in Millimeter innerhalb des vom Drucker bestimmten Druckbereichs

Beispiel: ..\1_System\Print.qdf

11.2.23 OpenQDF("Name"; "FileName"; Action)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
FileName	Pfad	QIK Definitionsdatei QDF
Action	Label	OpenOnly Der Ablauf muss nach dem Laden manuell gestartet werden. AutoStart Der Ablauf wird nach dem Laden der QDF Datei automatisch gestartet.

Beispiel: ..\1_System\OpenQdfFile.qdf

11.2.24 Execute("Name"; "File"; Parameter; Options; WaitTime)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
File	Pfad	Dateiname der Ausführungsdatei
Parameter	Param*	Parameter zur Ausführungsdatei
Options	Label	Die folgenden Optionen können durch Komma oder Leerzeichen getrennt aufgelistet werden. Normal Ausführungsdatei normal starten NotActivated Ausführungsdatei starten, Fenster öffnen aber nicht aktivieren (Dieses Attribut wird vom Kommandozeilenfenster ignoriert) Minimized Ausführungsdatei minimiert in der Taskleiste starten ErrorReport Fehlerdialog bei einem Fehler öffnen
WaitTime	-1, 0, N	-1 Nicht auf das Schliessen der gestarteten Ausführungsdatei warten. 0 Warten bis die Ausführungsdatei geschlossen wird, ohne Time out. N Warten bis die Ausführungsdatei geschlossen wird, maximal jedoch die definierte Wartezeit in Sekunden.

* Möglichkeit Variablen mit % einzubinden

Für den QIK kann zur Ausführungsdatei *File* eine Exit Code Informationsdatei QEC erstellt werden. Die QEC Datei ist eine CSV Datei im Verzeichnis der Ausführungsdatei mit gleichem Namen und der Dateierweiterung QEC.

Die folgenden drei Spalten müssen in der Datei definiert werden:

1. Exit Code dezimal, hexadezimal oder mit einem "-" für nicht spezifizierte Exit Codes
2. Kondition Success oder Failed für die Execute Funktion
3. Text für das Protokoll und/oder die Fehlermeldung in Anführungszeichen

Beispiele: ..\1_System\CommandLine.qdf
 ..\1_System\CommandLineWithExitCode.qdf

11.2.25 ExistWindow("Name"; "WindowTitle")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
WindowTitle	Str	Zeichenkette im Fenstertitel

Beispiel: ..\1_System\ExistWindow.qdf

11.2.26 SendKey("Name"; KeyCode[; "WindowTitle"[; Monitoring]])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
KeyCode	N	Wenn <i>WindowTitle</i> definiert ist, wird das Zeichen entsprechend dem Ordinalwert von <i>KeyCode</i> an das Fenster gesendet. (Nur wenn das Fenster auch der Anwendung entspricht wie beispielsweise bei der Command Shell). Wenn <i>WindowTitle</i> nicht definiert oder leer ist, wird der virtuelle Tastencode an die aktive Anwendung gesendet.
WindowTitle	Str	Optional: Fenstertitel zur Zieldefinition des <i>KeyCodes</i>
Monitoring	Label	Optional: Definition zur Überwachung der Funktionsausführung. NotMonitored Einfache Ausführung mit der Antwort ● (Standard) Monitored Wenn der <i>KeyCode</i> an das definierte Fenster gesendet werden konnte: ✓ sonst ✗.

Beispiel: ..\1_System\SendKey.qdf

12. Device: Document Display

Die Anwendung greift für die Anzeige der Dokumente auf den im Windows System integrierten Browser zurück. Durch die System Browser Version sind die Möglichkeiten bestimmt.



Unter Windows 11 Pro 23H2 ist das noch immer der Internet Explorer MSIE 7.0.
(Prüfen mit \2_Document\HtmlInterpreterInfo.qdf)



Eine einfache Test-Doku kann beispielsweise mit MS Word erstellt werden.
Siehe: 18. ANHANG:

12.1 Device.Create(HTMLViewer; Naming)

Bezeichnung	Typ	Beschreibung
HTMLViewer	Label	Schlüsselwort der Anwendung
Naming	Label	Eine im ganzen Testablauf eindeutige Benennung für dieses Gerät

12.2 Funktionen

Die Anzeige und Steuerung der Testdokumentation kann über das Gerät `HTMLViewer` bedient werden. Die Funktionen werden einleitend mit Naming Label aufgerufen: z.B. `Doc`.

```
Device.Create(HTMLViewer; Doc)

Doc.Open("Dokument öffnen"; "QDF:\SimpleDocument.htm")
Doc.Goto("Teill anspringen"; "SystemDelayTeill")
```

12.2.1 Open("Name"; "FileName")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
FileName	Pfad	Dateiname der zu öffnenden Dokumentation

Kann die Anwendung die Seite nicht öffnen, wird dies mit einer der folgenden Meldung angezeigt:

- "Die Seite kann nicht angezeigt werden."
- "Aktion abgebrochen"

Beispiel: \2_Document\DocumentDisplay.qdf

12.2.2 Goto("Name"; "Anchor")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Anchor	Str	Name des Anker-Tags das angesprungen werden soll. (Anker String = "Test1" wenn in HTML

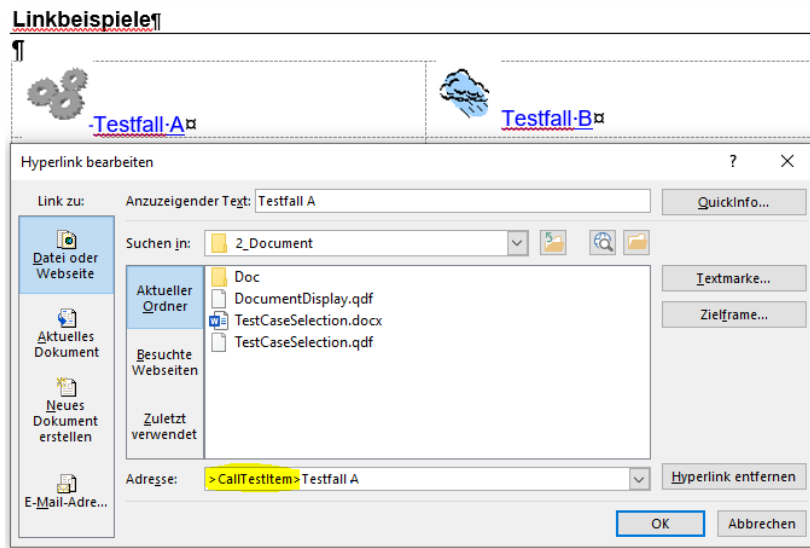


Die Anwendung kann nicht feststellen, ob der definierte Anker vorhanden ist.
Die Funktion wird bei einem unbekannten Anker ignoriert.

Beispiel: \2_Document\DocumentDisplay.qdf

12.3 Test Case aus der Doku

Aus der Testdokumentation können mit dem Link Präfix `>CallTestItem>` Testschritte im QDF gestartet werden.



Beispiel: \2_Document\TestCaseSelection.qdf (TestCaseSelection.docx)



Falls der hinterlegte **Test Case** nicht gefunden werden kann, wird eine Fehlermeldung im Protokollfenster ausgegeben.



In der HTML Datei wird der Link URL codiert in Prozentdarstellung gespeichert.

```
<src="TestCaseSelection-Dateien/image001.jpg"><a  
<href="..\..\>CallTestItem%3eTestfall%20A"><span lang=EN-US  
<style='font-size:12.0pt'>Testfall A</span></a></p>
```



HTML Datei vom System gesperrt!

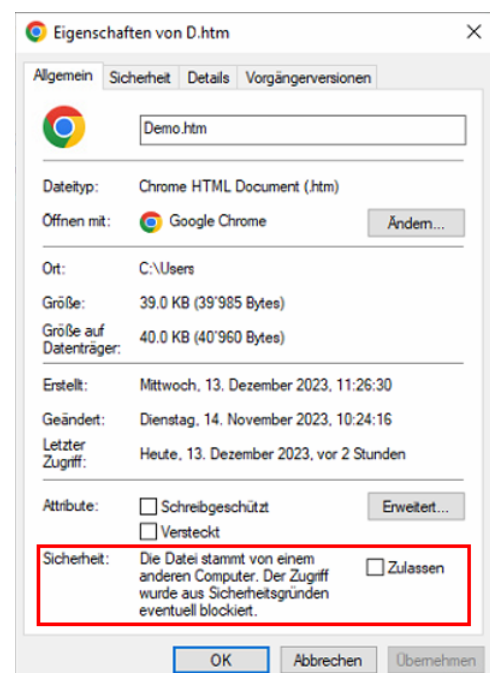
Aufgrund einer Windows Sicherheitsmassnahme kann das System Dateien, die von einem anderen Rechner stammen sperren. Auf das Anklicken des Links folgt keine Reaktion, keine Fehlermeldung, nichts!

Nur in den Dateieigenschaften wird folgendes angezeigt:

"Die Datei stammt von einem anderen Computer. der Zugriff wurde aus Sicherheitsgründen eventuell blockiert"

Die Datei kann mit "Zulassen" freigegeben werden oder es kann die Gruppenrichtlinie im System angepasst werden.

https://okidk.de/fix-diese-datei-stammt-von-einem-anderen-computer-und-ist-moeglicherweise-blockiert/#google_vignette



13. Device: Indicator

13.1 Device.Create(Indicator; Naming; "Text"; Digits; Color)

Bezeichnung	Typ	Beschreibung
Indicator	Label	Schlüsselwort der Anwendung
Naming	Label	Eine im ganzen Testablauf eindeutige Benennung für dieses Gerät
Text	Str	Beschreibung in der Anzeige
Digits	N	Anzahl Stellen die der Anzeige zur Verfügung stehen werden. Durch diesen Wert wird implizit die Breite der Anzeige bestimmt.
Color	Wert[h]	Hex-Wert zur Definition der Farbe in der folgenden Form: BBGGRRh BB Blauanteil Wertebereich von 00h-FFh GG Grünanteil Wertebereich von 00h-FFh RR Rotanteil Wertebereich von 00h-FFh Beispiele • Cyan = FFFF00 • Grün = 00FF00

13.2 Funktionen

Die Funktionen werden einleitend mit Naming Label aufgerufen: z.B. DMM.

```
Device.Create(Indicator; DMM; "Multimeter 1"; 6; FFFF00)

DMM.Indicate("Anzeige 1"; "Konstante 1 [mV]"; 0.8798)
```

13.2.1 Indicate("Name"; "Text"; Variable)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Text	Str	Beschreibung der Anzeige (z.B. Messeinheit)
Variable	Label	Name der Variable deren Wert angezeigt werden soll. Anstelle einer Variablen können auch Konstanten angezeigt werden.

Beispiel: \3_Indicator\DmmIndicator.qdf

14. Device: Port

Die Anwendung stellt mehrere Kommunikationsschnittstellen zur Verfügung: RS232, USB VCP und USB FTDI D2xx wie auch UDP und TCP Socket.

Die Kommunikation auf den Schnittstellen basiert grundsätzlich auf Bytes ohne Textkodierung respektive auf ANSI entsprechend den QDF Dateien. Aufgrund der internen Zeichenkettenverarbeitung werden die empfangenen Bytes zwischen 0x80-0x9F direkt als Byte-Zeichen übernommen. In der Darstellung werden diese Zeichen als Escape Sequenzen \80...\9F angezeigt.

14.1 Device.Create(RS232; Naming; Timeout; Port; BPS; SBI)

Bezeichnung	Typ	Beschreibung
RS232	Label	Schlüsselwort der Anwendung
Naming	Label	Eine im ganzen Testablauf eindeutige Benennung für dieses Gerät
Timeout	N	Antwortwartezeit in Millisekunden. Falls die Gegenstelle keine vollständige Antwort innerhalb dieser Zeit zurücksendet, wird eine Zeitüberschreitung ausgelöst.
Port	1..255	Port Nummer der seriellen Schnittstelle.
BPS	Wert	Übertragungsgeschwindigkeit in Bit pro Sekunde Werte = 110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000, 57600, 115200
SBI	N	Send Byte Intervall - Zeit in Millisekunden zwischen den einzelnen Bytes <ul style="list-style-type: none"> 0: Daten in einem Stream senden 1: Byte für Byte senden N: Zeit in Millisekunden zwischen den einzelnen Bytes Windows ist keine Echtzeitbetriebssystem. Die Zeiten können abhängig von Leistung und Auslastung stark abweichen.



Die weiteren Schnittstelleneinstellungen:

Datenbits 8, Parity Bit NONE, Stopp Bit 1, Protokoll NONE

14.2 Device.Create(FTDI; Naming; Timeout; Port; BPS [;DBits; Parity; SBit; Mode])

Bezeichnung	Typ	Beschreibung
FTDI	Label	Schlüsselwort der Anwendung
Naming	Label	Eine im ganzen Testablauf eindeutige Benennung für dieses Gerät
Timeout	N	Antwortwartezeit in Millisekunden. Falls die Gegenstelle keine vollständige Antwort innerhalb dieser Zeit zurücksendet, wird eine Zeitüberschreitung ausgelöst.
Port	Label	Die Schnittstelle kann über die folgenden drei Möglichkeiten definiert werden: <ul style="list-style-type: none"> USB Location Id: USBxxxx (Nicht unterstützt in Win7 mit USB3.0) FTDI Chip Seriennummer: SN:"SerialNo" FTDI Chip Beschreibung: DS:"Description"
BPS	Wert	Übertragungsgeschwindigkeit in Bit pro Sekunde.
DBits*	7, 8	Anzahl Datenbits in einem Datenwort. (Standard = 8)
Parity*	Label	Parity Bit <ul style="list-style-type: none"> NONE (Standard) EVEN ODD
SBit*	1, 2	Stopp Bit Standard = 1
Mode*	Label	Datenflusskontrolle <ul style="list-style-type: none"> NONE (Standard) RTS/CTS DTR/DSR XON/XOFF

* Die optionalen Definitionen werden zurzeit ignoriert. Es gelten die Default Werte.

14.3 Device.Create(UDP; Naming; Timeout; Address; ListeningPort)

Bezeichnung	Typ	Beschreibung
UDP	Label	Schlüsselwort der Anwendung
Naming	Label	Eine im ganzen Testablauf eindeutige Benennung für dieses Gerät
Timeout	N	Antwortwartezeit in Millisekunden. Falls die Gegenstelle keine vollständige Antwort innerhalb dieser Zeit zurücksendet, wird eine Zeitüberschreitung ausgelöst.
Address	IP	IP-Zieladresse Host oder IPv4 Adresse mit Port Nummer
Listening Port	Wert	Empfangs Port Nummer

14.4 Device.Create(TCP.Client; Naming; Timeout; Address)

Bezeichnung	Typ	Beschreibung
TCP.Client	Label	Schlüsselwort der Anwendung
Naming	Label	Eine im ganzen Testablauf eindeutige Benennung für dieses Gerät
Timeout	N	Antwortwartezeit in Millisekunden. Falls die Gegenstelle keine vollständige Antwort innerhalb dieser Zeit zurücksendet, wird eine Zeitüberschreitung ausgelöst.
Address	IP	IP-Zieladresse Host oder IPv4 Adresse mit Port Nummer

14.5 Funktionen

Die Schnittstellen Funktionen werden einleitend mit Naming Label aufgerufen: z.B. Port.

```
Device.Create(RS232; Port; 5000; 3; 9600; 0)

Port.Open("COM öffnen")
Port.Send("Send: Hallo ohne eine Antwort zu erwarten"; "Hallo"; ""; ""; "")
Port.Close("COM schliessen")
```


14.5.1 Open("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

14.5.2 Close("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

14.5.3 Instruct("Name"; Statement)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Statement	Label* oder IP*	<p>FTDI</p> <p>SetRTS ClrRTS SetDTR ClrDTR</p> <p>Die Kontrollsignale können als Ausgänge benutzt werden, sofern diese nicht anderweitig im Einsatz sind. Das Verhalten der Signale des eingesetzten Chips sollte aber auf jeden Fall vorab geprüft werden.</p> <p>CBBB[0xMW]</p> <p>CBUS Bit Bang für FTDI FT232R Im Bit Bang Modus können die CBUS IOs als Ausgänge benutzt werden. Wert Bitmaskiert: CBUS0 = Bit 1, CBUS1 = Bit 2, CBUS2 = Bit 3, CBUS3 = Bit 4 Modus Bitmaskiert: CBUS0 = Bit 5, CBUS1 = Bit 6, CBUS2 = Bit 7, CBUS3 = Bit 8</p> <p>CBBB[0xMW] W Wert am Pin: Low=0, High=1 (Bitmaskiert) M Modus am Pin: Input=0, Output=1 (Bitmaskiert)</p> <p>UDP und TCP.Client Redefinition der IP Adresse mit Host, IPv4 und Port oder nur Port.</p> <p>Syntax: [Host IPv4] [:Port] Beispiele: raspberrypi:55001 192.168.1.23:55001 192.168.1.23 :55001</p> <p> Die Änderung bleibt auch nach einem Test-Prozess Reset.</p>

* Möglichkeit Variablen mit % einzubinden

14.5.4 Timeout("Name"; Milliseconds)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Milliseconds	N	Antwortwartezeit in Millisekunden. Falls die Gegenstelle keine vollständige Antwort innerhalb dieser Zeit zurücksendet, wird eine Zeitüberschreitung ausgelöst.

Mit dieser Funktion kann die im *Device* definierte Antwortzeit oder vorgängige Definitionen mit `Timeout` überschrieben werden.

14.5.5 Send("Name"; "Data"; "STag"; "ETag"; "Expected" [; Variable])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Data	StrExt*	Die zu sendenden Daten oder Kommando. Durch ein Leer-String ("") wird die Schnittstelle für den Datenempfang vorbereitet ohne ein Kommando zu senden.
STag	StrExt	Antwort Start-Tag. Durch einen Leer-String ("") wird das Start-Tag ignoriert.
ETag	StrExt	Antwort End-Tag. Durch einen Leer-String ("") wird das End-Tag ignoriert.
Expected	StrExt*	Erwartete Antwortdaten Die Antwort kann mit Wildcard * und ? ausmaskiert werden. Bei einem Leer-String ("") wird keine Antwort abgewartet.
	Label	NoResponse Bis zum Timeout wird keine Antwort erwartet. Falls trotzdem Daten empfangen werden, wird ein Fehler ausgelöst.
Variable	Label	Optional: Name String Variable in der die Antwort gespeichert werden soll.

* Möglichkeit Variablen mit % einzubinden

Beispiele: ..\4_Port\Basics.qdf
 ..\4_Port\Variables.qdf

14.5.6 WaitForResponse("Name"; "STag"; "ETag"; Expected[; Variable])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
STag	StrExt	Antwort Start-Tag. Durch einen Leer-String ("") wird das Start-Tag ignoriert.
ETag	StrExt	Antwort End-Tag. Durch einen Leer-String ("") wird das End-Tag ignoriert.
Expected	StrExt*	Erwartete Antwortdaten Die Antwort kann mit Wildcard * und ? ausmaskiert werden.
	Label	NoResponse Bis zum Timeout wird keine Antwort erwartet. Falls trotzdem Daten empfangen werden, wird ein Fehler ausgelöst.
Variable	Label	Optional: Name einer String Variable in der die Antwort gespeichert werden soll.

* Möglichkeit Variablen mit % einzubinden

Beispiele: ..\4_Port\Basics.qdf
 ..\4_Port\Variables.qdf
 ..\4_Port\WaitForResponse.qdf

14.5.7 Evaluate.Value("Name"; "Cmd"; "STag"; "ETag"; Min; Max [; Variable])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Cmd	StrExt	Kommando String der die Gegenstelle veranlasst einen Wert zurückzusenden.
STag	StrExt	Antwort Start-Tag. Durch einen Leer-String ("") wird das Start-Tag ignoriert.
ETag	StrExt	Antwort End-Tag. Durch einen Leer-String ("") wird das End-Tag ignoriert.
Min	G R Term	Minimal Wert der Antwort. (Term = Variable ±G R)
Max	G R Term	Maximal Wert der Antwort. (Term = Variable ±G R)
Variable	Label	Optional: Name der Float-Variable in der die Antwort gespeichert werden soll.

Beispiel: ..\4_Port\Extended.qdf

14.5.8 SendJson("Name"; JSON; "STag"; "ETag"; "Expected" [; Variable])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
JSON	StrExt	Gültiges JSON Objekt zum Senden (www.json.org)
STag	StrExt	Antwort Start-Tag. Durch einen Leer-String ("") wird das Start-Tag ignoriert.
ETag	StrExt	Antwort End-Tag. Durch einen Leer-String ("") wird das End-Tag ignoriert.
Expected	StrExt	Vereinfachte JSON Notation ohne Anführungszeichen einer oder mehrerer zwingenden Namen/Werte Paare. Die Werte können mit Wildcard * und ? ausmaskiert werden.
Variable	Label	Optional: Name der Variable in der der Wert des im "Expected" als letztes definierten Name/Werte Paares.

Beispiel: ..\4_Port\Json.qdf



Die Kommunikation auf den Schnittstellen basiert grundsätzlich auf Bytes ohne Textkodierung respektive auf ANSI entsprechend den QDF Dateien. Deshalb sind die Zeichenketten entgegen dem JSON Standards nicht in UTF-8.

14.5.9 WaitForJson"Name"; "STag"; "ETag"; "Expected" [, Variable])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
STag	StrExt	Antwort Start-Tag. Durch einen Leer-String ("") wird das Start-Tag ignoriert.
ETag	StrExt	Antwort End-Tag. Durch einen Leer-String ("") wird das End-Tag ignoriert.
Expected	StrExt	Vereinfachte JSON Notation ohne Anführungszeichen einer oder mehrerer zwingenden Namen/Werte Paare. Die Werte können mit Wildcard * und ? ausmaskiert werden.
Variable	Label	Optional: Name der Variable in der der Wert des im "Expected" als letztes definierten Name/Werte Paares.

Beispiel: ..\4_Port\Json.qdf



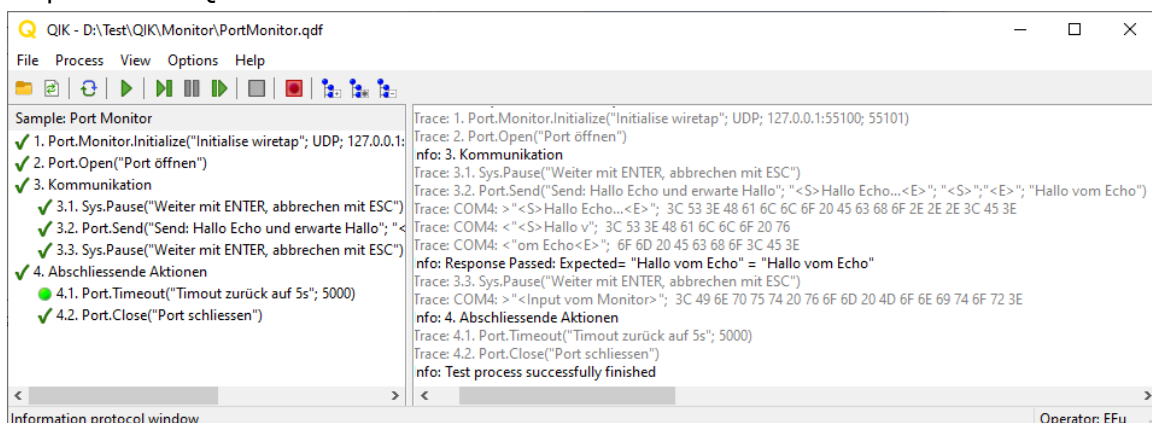
Die Kommunikation auf den Schnittstellen basiert grundsätzlich auf Bytes ohne Textkodierung respektive auf ANSI entsprechend den QDF Dateien. Deshalb sind die Zeichenketten entgegen dem JSON Standards nicht in UTF-8.

14.5.10 Monitor.Initialize("Name"; Port; Address[, Listening Port[, Options]])

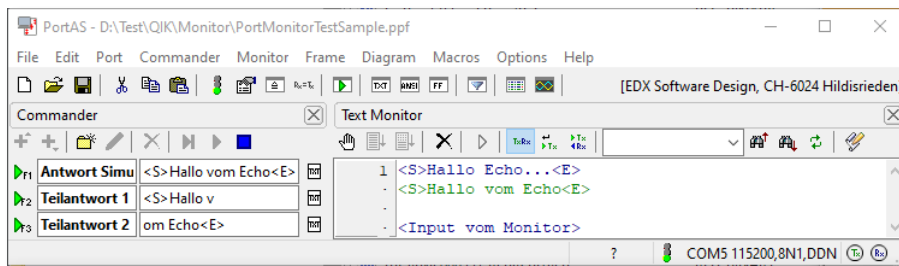
Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Port	Label	Art der Schnittstelle an die die Daten zur Beobachtung gesendet werden. Aktuell wird nur UDP unterstützt.
Address	IP	IP-Zieladresse Host oder IPv4 Adresse mit Port Nummer
Listening Port	Wert	Optional kann der Listening Port definiert werden, um Daten von der Monitor Schnittstelle direkt dem Prüfling senden zu können.
Options	Labels	Optional ein oder mehrere Zusätze getrennt durch ein Komma <div>Verbose</div> Wenn definiert, werden im QIK Protokoll Fenster die Daten zur und von der Monitor Schnittstelle ausgegeben. <div>Silent</div> Wenn definiert, werden im QIK Protokoll Fenster keine Fehlermeldungen der Monitor Schnittstelle ausgegeben.

Monitor Schnittstelle zur Beobachtung der Schnittstelle zwischen QIK und Prüfling. Mit der Initialisierung des Monitors auf dem Port wird beim Öffnen des Ports die Schnittstelle zum Monitor geöffnet. Wenn der Port geschlossen wird, wird auch die Schnittstelle zum Monitor geschlossen.

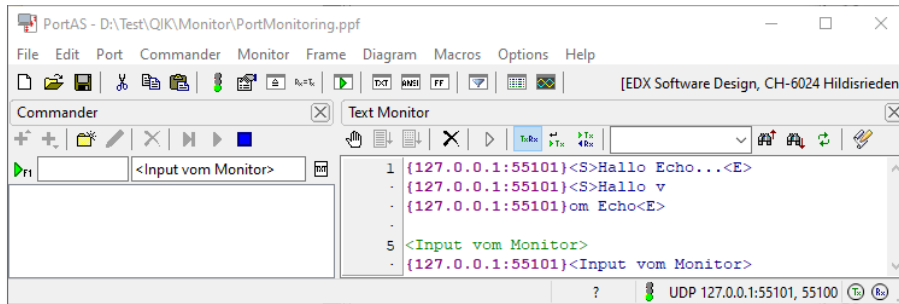
Beispielablauf in QIK



Prüfling als PortAS Gegenstelle



Monitoring in PortAS



Beispiel: ..\4_Port\Monitor\PortMonitor.qdf
und PortAS Dateien PortMonitoring.ppf und PortMonitorTestSample.ppf

15. Device: GPCC

Mit dem GPCC (General Purpose Component Object Model Client) wird eine Möglichkeit zur Verfügung gestellt, einen COM Server einzubinden. Auf Anfrage kann die COM Schnittstelle erläutert und ein COM Server Wrapper Konzept angeboten werden.

15.1 Device.Create(GPCC; Naming; Timeout)

Bezeichnung	Typ	Beschreibung
GPCC	Label	Schlüsselwort der Anwendung
Naming	Label	Eine im ganzen Testablauf eindeutige Benennung für dieses Gerät
Timeout	N	Antwortwartezeit in Millisekunden. Die Zeit wird im <i>Device</i> und im COM Server gesetzt. Siehe 15.2.3 Timeout("Name"; Milliseconds)

15.2 Funktionen

15.2.1 Open("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

15.2.2 Close("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

15.2.3 Timeout("Name"; Milliseconds)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Milliseconds	N	Antwortwartezeit in Millisekunden. Die Zeit wird im <i>Device</i> und im COM Server gesetzt.

Mit dieser Funktion kann die im *Device* und im COM Server definierte Antwortzeit oder vorgängige Definitionen mit `Timeout` überschrieben werden.

Zeitüberschreitungen können im *Device* und im COM Server ausgelöst werden:

1. Falls der COM Server keine Antwort innerhalb dieser Zeit zurücksenden kann, wird im COM Server die Zeitüberschreitung ausgelöst, die die Funktion abbricht und dem Device beispielsweise mit "Timeout" meldet.
2. Falls im Device vom COM Server keine Antwort innerhalb dieser Zeit geliefert worden ist, wird im Device die Zeitüberschreitung ausgelöst und die Funktion wird abgebrochen.

15.2.4 Execute("Name"; "FunctionCall"; "Expected" [; Variable])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
FunctionCall	Str*	Zeichenkette beispielsweise Funktionsaufruf der auf dem COM Server auszuführen ist.
Expected	Str*	Erwartete Antwortzeichenkette Die Antwort kann mit Wildcard * und ? ausmaskiert werden.
Variable	Label	Optional: Name einer Integer-, Float- oder String-Variable in der die Antwort gespeichert werden soll.

* Möglichkeit Variablen mit % einzubinden

`Execute` ruft via COM Interface die `ExecuteFunction` auf dem COM Server auf. Es gelten die folgenden Return Codes:

=0: Funktion ausgeführt - synchron. Auswertung von `Expected`.

>0: Funktion ausgeführt - asynchron. Die Antwort wird abgewartet, dann `Expected` ausgewertet.

<0: Fehler beim Ausführen der Funktion.

16. Device: Instrument

Über dieses Device können DMM und andere Messgeräte mit PC Schnittstelle eingebunden werden. So vielfältig die Gerätegruppen so vielfältig sind die Schnittstellen und Protokolle. Integrationen werden anhand konkreter Geräte implementiert.

16.1 DMM Mastech MS8218 - Bedienungshinweis

Multimeter einschalten, *WAKE* Taste antippen, *HOLD* Taste gedrückt halten bis *RS232* im Display erscheint. Jetzt werden Daten via *RS232* ausgegeben. Das Gerät schaltet nach einigen Minuten selbständig aus (*Powersave*). Wird die *WAKE* Taste während dem Einschalten gedrückt gehalten, wird der *Powersave* deaktiviert.

Beispiel: ..\5_Instrument\MS8218_Sample.qdf

16.2 Device.Create(Instrument; Naming; "Type"; "Configuration")

Bezeichnung	Typ	Beschreibung
Instrument	Label	Schlüsselwort der Anwendung
Naming	Label	Eine im ganzen Testablauf eindeutige Benennung für dieses Gerät
Type	Label	Instrumenten Typ • MS8218 DMM MASTECH MS8218
Configuration	Str	MS8218 • COMn RS232 Port Nummer (1..9) • USBxxxx USB Location Id • SN:"SerialNo" FTDI Chip Seriennummer • DS:"Description" FTDI Chip Beschreibung



Unter Windows 7 kann das Instrument nur über USB 2.0 via FTDI angeschlossen werden!

16.3 Funktionen

Die Funktionen werden einleitend mit Naming Label aufgerufen: z.B. DMM1.

```
Device.Create(Instrument; DMM1; MS8218; COM3)

DMM1.Open("Verbindung zum Messgerät öffnen");
DMM1.Reset("Reset Messresultate");
DMM1.Check("Messdaten lesen"; MS8218_VAC; 0; -10.0001; +11.00)
```

16.3.1 Open("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

Die definierte Schnittstelle wird geöffnet und der Datenfluss wird aktiviert.

16.3.2 Close("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

Die definierte Schnittstelle wird geschlossen.

16.3.3 Reset("Name") – [Deprecated!]

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

Instrument: MS8218

Die Daten werden kontinuierlich vom DMM gelesen. Mit der Funktion `Check` wird das letzte gelesene Messresultat geprüft. Durch die Funktion `Reset` wird das Messresultat gelöscht, so dass erst nach einer gültigen Messung ein Resultat zur Verfügung steht.



Die Funktion ist veraltet und nur noch aufgrund der Retour Kompatibilität vorhanden. Ab QIK V2.20 wird der Messinstrument-Reset implizit im `Device.Check` ausgeführt.

16.3.4 Check("Name"; "Mode"; Timeout; Min; Max [; Variable])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Mode	Str	Gerätespezifischer Definitionsstring für Betriebsmodus, Messbereich und weitere Definitionen. • MS8218 Datei <i>Instrument_MS8218.QDF</i>
Timeout	0, N	Wert in Sekunden während dem die Messdaten geprüft werden. Beim ersten Messwert der im definierten Bereich ist, wird der Ablauf fortgesetzt. Mit Timeout = 0 gilt die laufende Messung für die Prüfung.
Min	R Term	Mindestwert (Term = Variable ±G R)
Max	R Term	Maximalwert (Term = Variable ±G R)
Variable	Label	Optional: Name der Variablen in der die Antwort gespeichert werden soll.

Erfolgreich wenn der gelesene $\text{Min} \leq \text{Wert} \leq \text{Max}$ ist.

17. Device: Audio

Der Audiobereich basiert auf einer in Windows 7 neu eingeführten API und ist mit Vorgängersystemen nicht kompatibel.

Grundsätzlich sind Möglichkeiten und Einstellungen von der Audio Hardware und den Treibern abhängig. Es muss aber beachtet werden, dass die Audio Geräte dieser Anwendung zurzeit ausschliesslich 16 Bit pro Sample unterstützen.

Beispiele: ..\6_Audio\AudioControl.qdf
 ..\6_Audio\AudioOutput.qdf
 ..\6_Audio\AudioInput.qdf

17.1 Allgemeine Funktionen

17.1.1 SetVolume("Name"; ValueType; Level[; Level])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
ValueType	Label	Scalar Pegel normalisiert von 0.0 ... 1.0 dB Pegel in dB
Level	R	Abhängig von der Anzahl Kanäle werden die Pegel mit Strichpunkt getrennt aufgelistet.

Die Anzahl der Kanäle, die Pegel in dB wie auch die kleinstmögliche Schrittgrösse sind von der Hardware und/oder Treiber abhängig und werden bei der Initialisierung des Gerätes im Protokollfenster angezeigt.

17.1.2 MuteOn("Name")

17.1.3 MuteOff("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

17.1.4 SetAsDefault("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

Oft werden von Audio Einheiten nicht alle angebotenen Ein- respektive Ausgabegeräte parallel unterstützt. Einige Geräte können aber durch das Setzen als Standardgerät aktiviert werden. Das Verhalten ist Hersteller und Treiber abhängig und muss geprüft werden.

Die Hersteller empfehlen im Allgemeinen nicht benötigte Geräte zu deaktivieren.

17.2 Device: Audio Output

17.2.1 Device.Create(AudioOutput; Naming; "DeviceName")

Bezeichnung	Typ	Beschreibung
AudioOutput	Label	Schlüsselwort der Anwendung
Naming	Label	Eine im ganzen Testablauf eindeutige Benennung für dieses Gerät
DeviceName	Str	Vollständige Bezeichnung oder erster Teil der Bezeichnung des Ausgangsgerätes

17.2.2 Funktionen

17.2.2.1 Start.Sine("Name"; Frequency)

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Frequency	N	Sinus Frequenz in Hz im Rahmen des Datenformates

17.2.2.2 Start.File("Name"; "FileName")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
FileName	Pfad	Audiodatei die abgespielt wird.

Diese Funktion ist nicht für das kontinuierliche Abspielen von Dateien ausgelegt. Die Datei wird vor dem Abspielen vollständig eingelesen und muss auf die Einstellungen des Gerätetreibers passen: Anzahl der Kanäle, Sample Rate und Bits per Sample. Es wird ausschliesslich das unkomprimierte RIFF Wave Dateiformat unterstützt.

Das Abspielen der Datei kann mit `stop` beendet werden.

17.2.2.3 Stop("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

17.3 Device: Audio Input

17.3.1 Device.Create(AudioInput; Naming; "DeviceName")

Bezeichnung	Typ	Beschreibung
AudioInput	Label	Schlüsselwort der Anwendung
Naming	Label	Eine im ganzen Testablauf eindeutige Benennung für dieses Gerät
DeviceName	Str	Vollständige Bezeichnung oder erster Teil der Bezeichnung des Eingang-Gerätes

17.3.2 Funktionen

17.3.2.1 Start.AVG.Check("Name"; Terminate; Channel1[; Channel N])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Terminate	Label	Manual Überschreiten von Maximalwerten werden als Warnung im Protokollfenster ausgegeben ohne die Testfunktion zu beenden. AutoError Überschreitung eines Maximalwertes beendet die Testfunktion automatisch.
Channel	N:N:N:N	Abhängig von der Anzahl unterstützter Kanäle werden die Werte für einen Kanal mit Strichpunkt getrennt aufgelistet. Für einen Kanal werden <i>Offset</i> , <i>Threshold</i> , <i>Max AVG</i> und <i>Max Peak</i> mit Doppelpunkten getrennt definiert. Offset: Wandler Kompensation (Sample = Sample -Offset) Threshold: Rauschschwelle (Noise Threshold) Max AVG: Maximaler Durchschnittswert. Wenn 0 wird der Grenzwert nicht geprüft. Max Peak: Maximaler Abtastwert. Wenn 0 wird der Grenzwert nicht geprüft.

AVG *Average* oder Durchschnitt ist hier das arithmetische Mittel über die absoluten Abtastwerte und wird wie folgt berechnet (http://de.wikipedia.org/wiki/Arithmetisches_Mittel):

$$\bar{x}_{\text{arithm}} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + \dots + x_n}{n}$$

Nach dem Start der Testfunktion werden kontinuierlich *AVG* und *Peak* Werte ermittelt. Wenn ein *Max AVG* oder *Max Peak* überschritten wird, wird dies im Protokollfenster mit einem *Warning* angezeigt. Mit *AutoError* wird die Auswertung automatisch eingeleitet und die Testfunktion beendet.

Der Anwender kann die Testfunktion manuell mit den folgenden Tastaturtasten beschliessen:

ENTER: Die Grenzwerte werden geprüft und die Testfunktion wird beendet mit ✓ oder ✗.

ESC: Die Testfunktion wird mit ✗ abgebrochen.

Beispiele: ..\ 6_Audio\AudioInput_Start.AVG.Check.qdf

17.3.2.2 Start.RMS.Measurements("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

Kontinuierliche RMS Messungen die in das Protokollfenster ausgegeben werden. (Ohne Auswertung)

Details zur RMS Messung siehe: 17.3.2.3 Check.RMS("Name"; Min:Max [;Min:Max])

17.3.2.3 Check.RMS("Name"; Min:Max [,Min:Max])

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
Min:Max	N:N	Abhängig von der Anzahl unterstützter Kanäle werden die Pegel mit Strichpunkt getrennt aufgelistet. Ist der Minimalwert \geq dem Maximalwert, wird für den Kanal der Mittelwert nicht ausgewertet.

Effektivwert der Samples über 400ms. RMS (Root Mean Square) ist das quadratische Mittel und wird wie folgt berechnet (<http://de.wikipedia.org/wiki/Effektivwert>):

$$U_{\text{eff}} \approx \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} = \sqrt{\frac{1}{n} (x_1^2 + x_2^2 + x_3^2 \cdots + x_n^2)}$$

Beispiel: ..\6_Audio\ AudioTestSequence.qdf

17.3.2.4 Start.Recorder("Name"; "FileName")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf
FileName	Pfad	Audiodatei die aufgenommen wird.

Diese Funktion erstellt eine unkomprimierte RIFF Wave Datei mit den Einstellungen des Gerätetreibers: Anzahl der Kanäle, Sample Rate und Bits per Sample. Sie ist nicht für das kontinuierliche Aufnehmen von Audiosignalen ausgelegt. Die Sample Daten werden erst beim Speichern in eine Datei geschrieben. Die Dateigröße ist auf ca. 25MB respektive 2¼ Minuten begrenzt.

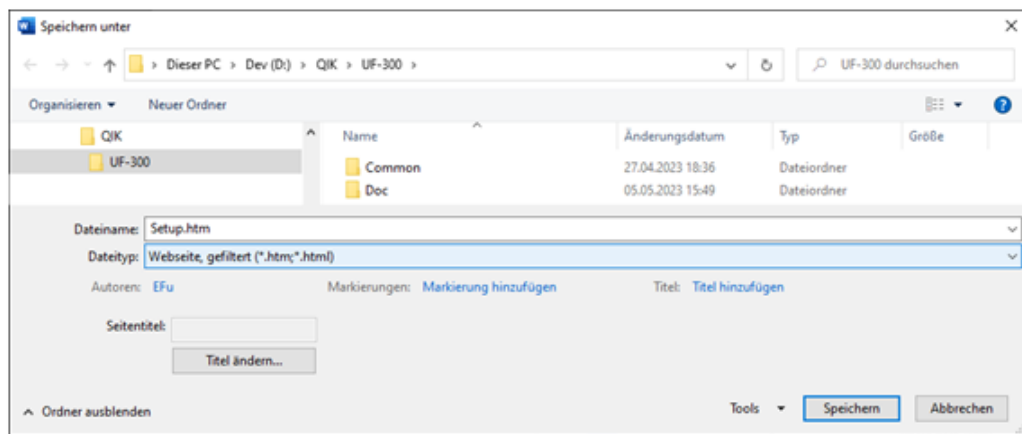
17.3.2.5 Stop("Name")

Bezeichnung	Typ	Beschreibung
Name	Str	Bezeichnung der Funktion im Testablauf

18. ANHANG: Dokumentation / Benutzerführung mit MS Word

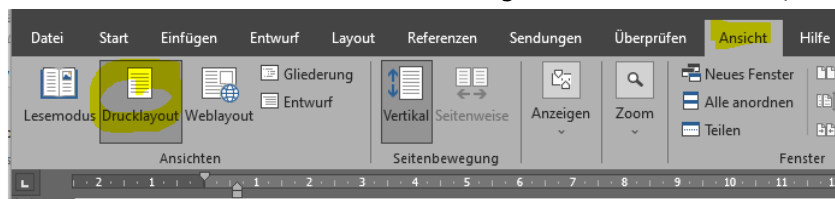
Einfache Testablaufbeschreibungen können mit Microsoft Word erstellt werden. Im Word können Dokumente auch im HTML Format gespeichert und so im QIK genutzt werden. Mit den folgenden Hinweisen können einige Fallstricke umgangen werden.

1. Bearbeitung ausschliesslich im docx Word Dokument
(Im HTML gehen Formatierungen verloren und die Zuordnung der Bilder kann beispielsweise durcheinanderkommen)
2. HTML Datei durch speichern als "HTML gefiltert".
(Übersichtliche und kompakte Dateien)



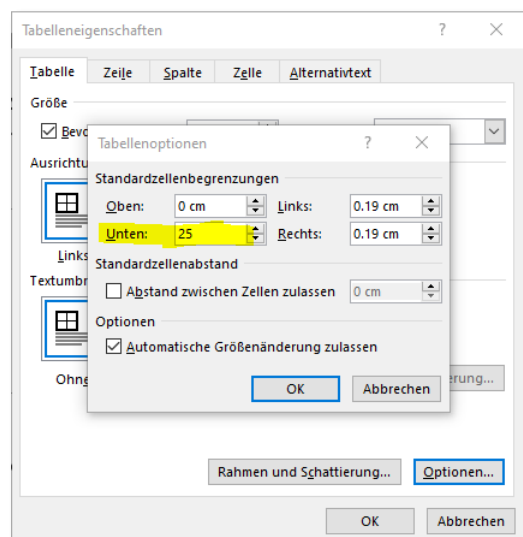
Achtung, nach dem Speichern ist das Dokument als HTML offen.
HTM Datei schliessen und docx Dokument für weitere Änderungen öffnen.

3. Wenn eine Datei als HTML gespeichert wird, schaltet Word die Ansicht auf Weblayout um.
Die "Normal" Ansicht muss manuell zurückgesetzt werden: Ansicht, Drucklayout.



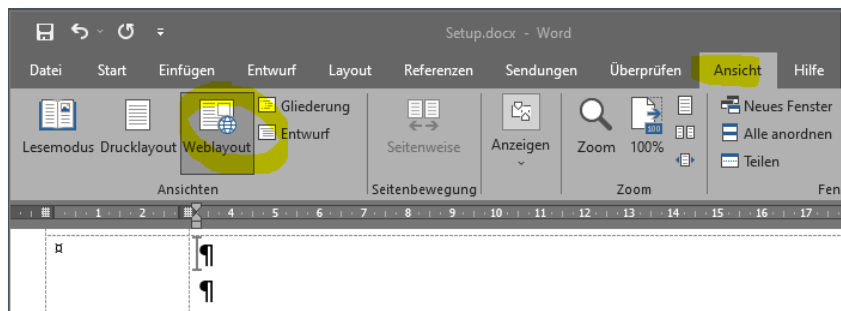
4. MS Word Dokument Formatierung
Mit einer Tabelle im MS Word Dokument kann die HTML Formatierung einfacher gehandhabt werden.

Mit dem Zeilenabstand "Unten" kann beispielsweise die nächste Dokumenten-Anker-Position mit genügend Abstand für alle Testfälle über das ganze Dokument eingestellt werden.



5. Dokumentenansicht Weblayout

Wenn mit Tabellen Zeilenabstand gearbeitet wird, empfiehlt es sich mit der Ansicht Weblayout zu arbeiten. (Keine Seitenumbrüche)



19. ANHANG: RTF Vorlagen

Zum Drucken von Daten werden Dateivorlagen im RTF Format mit Platzhaltern erstellt. Als Platzhalter stehen alle deklarierten Variablen sowie Datum und Zeit Platzhalter zur Verfügung. Zu den allgemeinen Platzhaltern sind auch funktionsspezifische Platzhalter möglich.

%Variable	Variablen werden mit einem % und den Variablen Namen eingesetzt
%DATE	Datum
%W %WW	Kalenderwoche ein- oder zweistellig nach ISO 8601
%D %DD	Tag ein- oder zweistellig
%M %MM	Monat ein- oder zweistellig
%YY %YYYY	Jahr zwei- oder vierstellig
%TIME	Zeitstempel

Hinweis:

Mit dem Windows WordPad können schlanke und übersichtliche Formatvorlagen erstellt werden. MS Word wird nicht empfohlen, da dies die Dateien unnötig gross macht. Der Vergleich der Vorlage `Error-ReportTemplate.rtf` zeigt die Grössenordnungen:

- WordPad 360 Byte
- Word2002 4 KB!
- Word2010 32 KB!
- Word2016 44 KB!
- ...

20. ANHANG: ID Dateien

Identifikationsnummern für Serie, Geräte oder andere eindeutige Nummern werden in Textdateien verwaltet und dienen als Quell- und Log-Datei.

Im Ablauf kann mit der `SystemControl` Funktion `GetNextId` die nächste Nummer eine Integer Variable geladen werden. Die nächste Nummer wird anhand der letzten Nummer und der optionalen Bereichsdefinition ermittelt. (Bereich $1..2^{63}-1$). Über die Funktion `RegisterId` wird die Identifikationsnummer in der Quell- und Log-Datei auf Eindeutigkeit geprüft und gespeichert.

Definitionen

#Ranges	Eine oder mehrere Bereichsdefinitionen möglicher Identifikationsnummern (Optional)
#Sorted	Wenn definiert, wird die Liste der Nummern sortiert, bevor die nächste Nummer ermittelt wird.
#Start	Start der Nummernliste

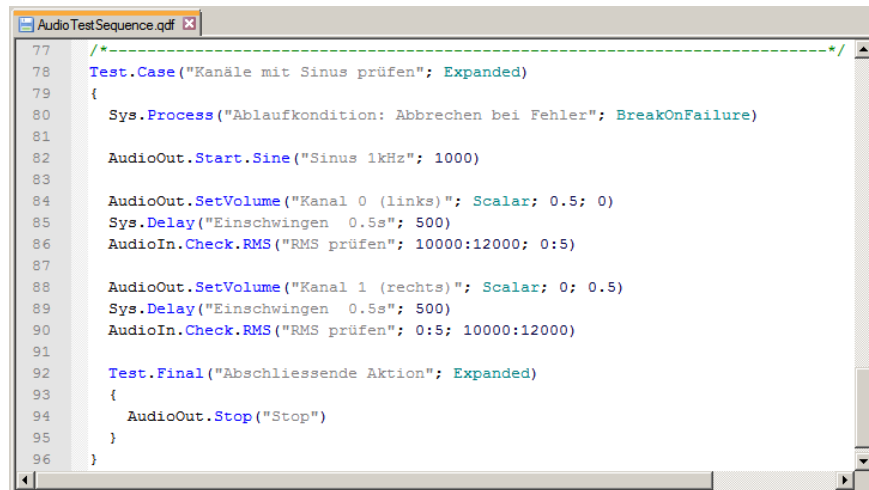
Beispiele: `..\1_System\ID_Number.dat`
`..\1_System\ID_NumberHex.dat`
`..\1_System\ID_NumberSorted.dat`

21. ANHANG: Syntax in Farbe

21.1 Notepad++

<<Notepad++ is a free source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by [GNU General Public License](https://notepad-plus-plus.org/).>>
<https://notepad-plus-plus.org/>

In der Datei QDF_SyntaxColoring_Notepad++.xml ist ein Definitionsvorschlag für die Einfärbung der QIK Syntax. Die Datei kann im Dialog *User Defined Language* importiert werden (Menu: *Languages, Define your Language...*)



```

77  /*-----*/
78  Test.Case("Kanäle mit Sinus prüfen"; Expanded)
79  {
80      Sys.Process("Ablaufkondition: Abbrechen bei Fehler"; BreakOnFailure)
81
82      AudioOut.Start.Sine("Sinus 1kHz"; 1000)
83
84      AudioOut.SetVolume("Kanal 0 (links)"; Scalar; 0.5; 0)
85      Sys.Delay("Einschwingen 0.5s"; 500)
86      AudioIn.Check.RMS("RMS prüfen"; 10000:12000; 0:5)
87
88      AudioOut.SetVolume("Kanal 1 (rechts)"; Scalar; 0; 0.5)
89      Sys.Delay("Einschwingen 0.5s"; 500)
90      AudioIn.Check.RMS("RMS prüfen"; 0:5; 10000:12000)
91
92      Test.Final("Abschliessende Aktion"; Expanded)
93      {
94          AudioOut.Stop("Stop")
95      }
96  }
  
```

22. ANHANG: Einstellungen zum Protokollfenster

Die visuellen Effekte im Windows verzögern den Bildlauf des Protokollfensters beim Scrollen mit dem Mausrad. Falls dies stört kann dieser Effekt explizit deaktiviert werden mit "Smooth-scroll list boxes" (Optimierten Bildlauf für Listenfelder)

