Performance Studio

Test Application Build Reference

Document	PerformanceStudio_1006_ApplicationBuildReference
Author	Richard Todd
Date	9/05/2022
Version	.95

1 Overview

This document defines the build and deployment process for test application developed using the ‘Uniconsole Framework’. It does not provide information on any package(s) that were not specifically developed by Performance Studio.

The following information is provided:

- Required software tools
- Required developer knowledge
- The standard deployment framework and usage
- Project build process
- The framework configuration process

2 Required Software Tools

The following software platforms and tools are required to build the test application(s):

1. Visual Studio 2022 or greater
2. DB Browser for sqlite version 3.12.2 or greater

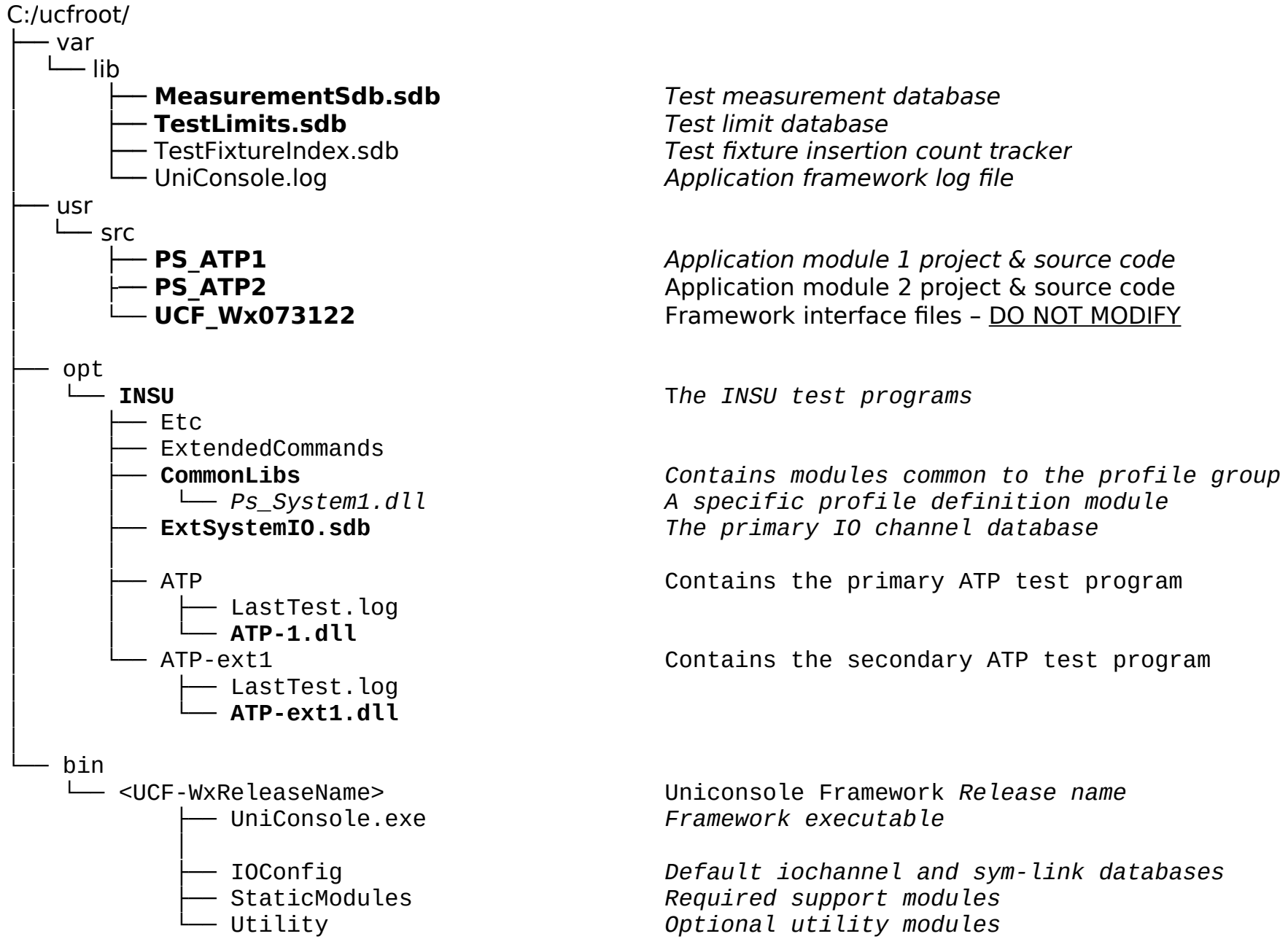
3 Required Developer Knowledge

The following skill-sets are required for test application development and maintenance:

1. Intermediate level knowledge of C++ development (C++ 11 and later).
2. Working knowledge of MS Windows 10 code development and system administration

4 The Standard Deployment Framework

This section defines the standard directory structure used by all deployed test applications. The directories and files of primary interest are highlighted.



5 Starting a Test Application

5.1 The test application is started using a desktop launcher (shortcut) using the following command line.

C:\ucfroot\bin\<UCF-WxReleaseName>\UniConsole.exe -p c:\ucfroot\opt\INSU\ATP

Where:

a) UniConsole.exe is the test executive executable.
b) c:\ucfroot\opt\INSU\ATP' contains the test module(s) to be loaded.

6 Project Build Procedure

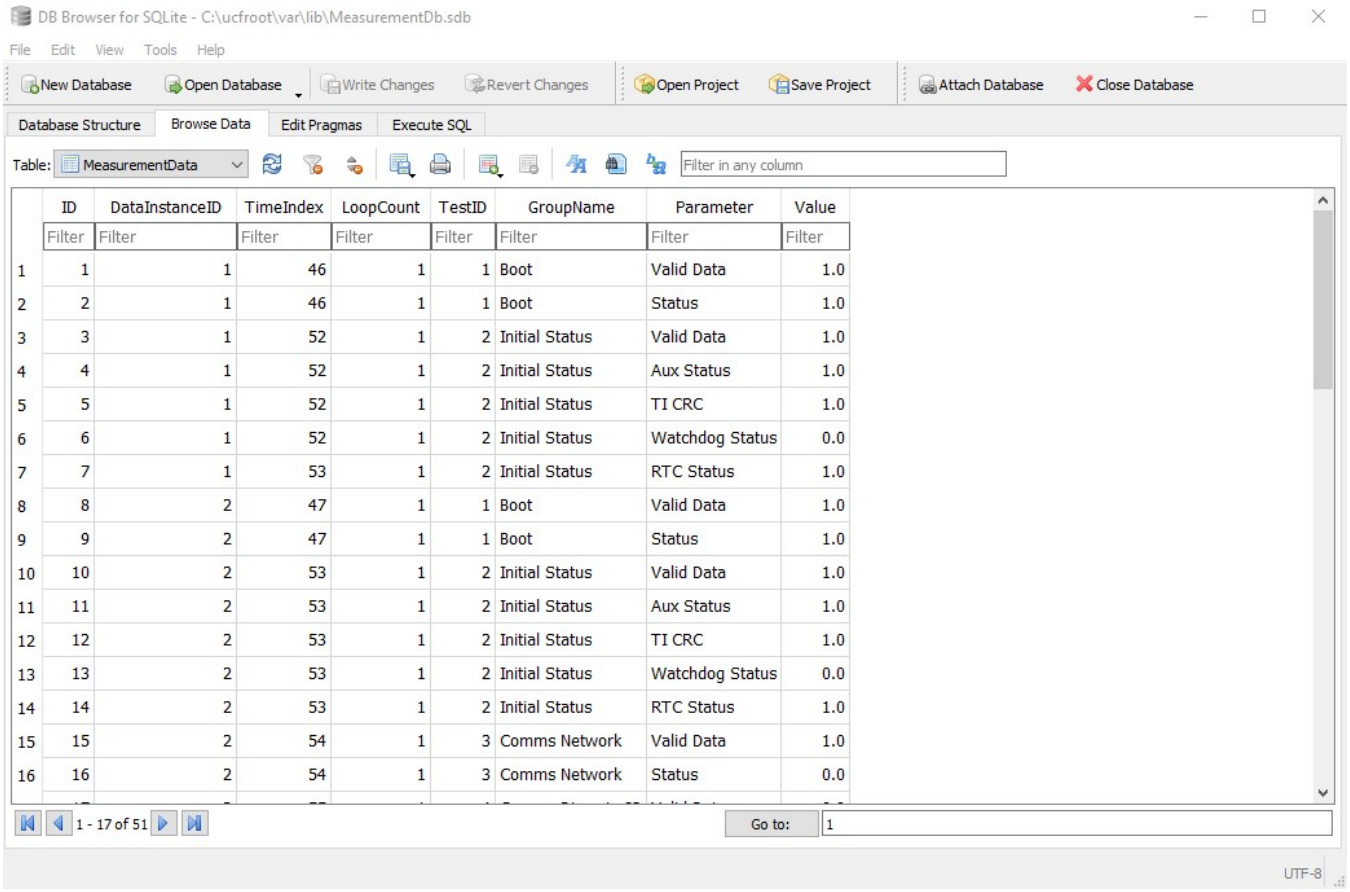
6.1 Prerequisites.

a) MS Windows 10
b) Visual Studio 2022 or later

6.2 All projects under c:\usr\src\PS_ATP1 and PS_ATP2 are opened and built with Visual Studio. No additional steps are needed.

7 Framework Configuration

The framework is configured using the ‘sqlitebrowser’ database manager to modify local configuration databases.



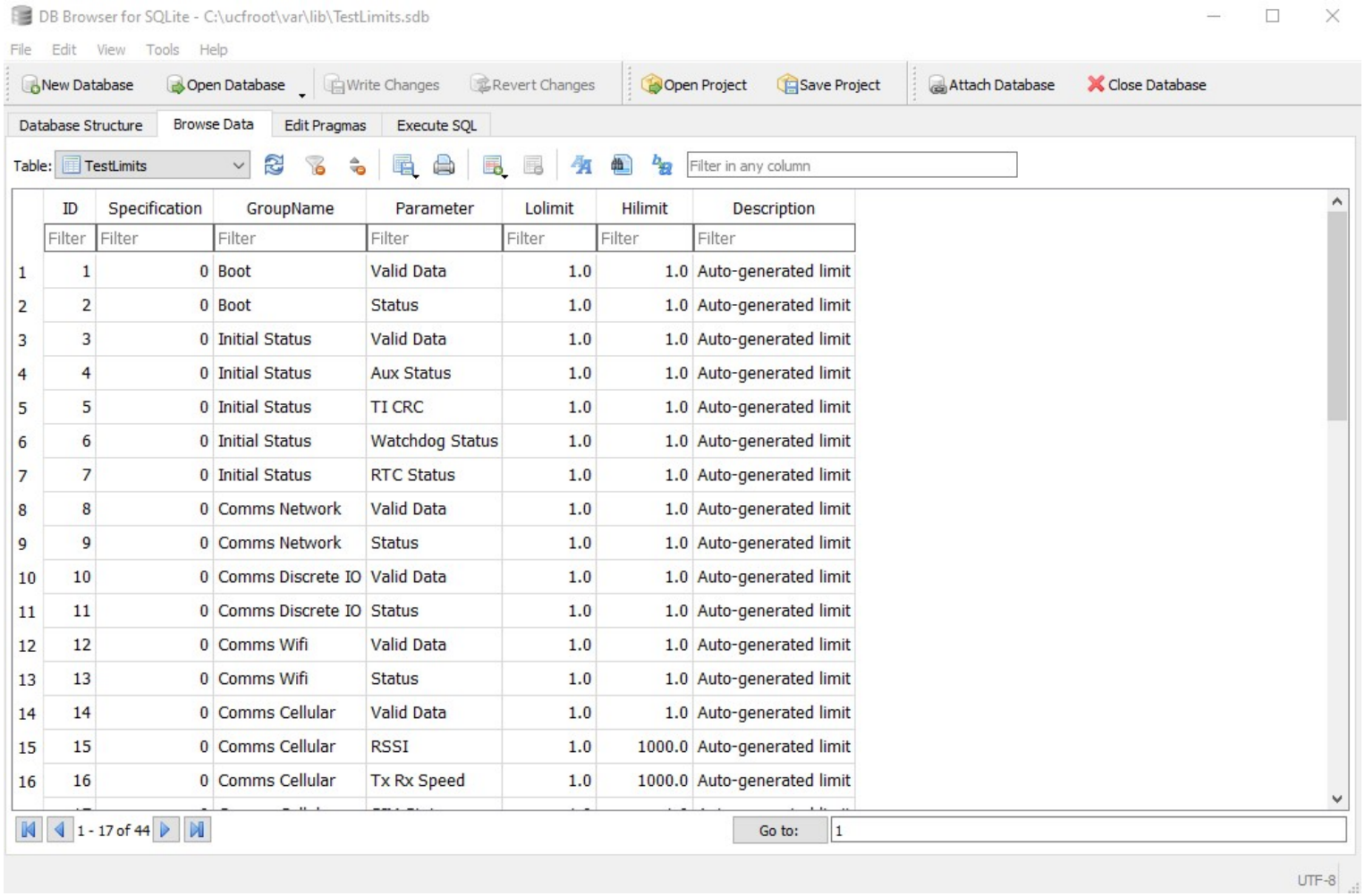
	ID	DataInstanceID	TimeIndex	LoopCount	TestID	GroupName	Parameter	Value
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	1	46	1	1	Boot	Valid Data	1.0
2	2	1	46	1	1	Boot	Status	1.0
3	3	1	52	1	2	Initial Status	Valid Data	1.0
4	4	1	52	1	2	Initial Status	Aux Status	1.0
5	5	1	52	1	2	Initial Status	TI CRC	1.0
6	6	1	52	1	2	Initial Status	Watchdog Status	0.0
7	7	1	53	1	2	Initial Status	RTC Status	1.0
8	8	2	47	1	1	Boot	Valid Data	1.0
9	9	2	47	1	1	Boot	Status	1.0
10	10	2	53	1	2	Initial Status	Valid Data	1.0
11	11	2	53	1	2	Initial Status	Aux Status	1.0
12	12	2	53	1	2	Initial Status	TI CRC	1.0
13	13	2	53	1	2	Initial Status	Watchdog Status	0.0
14	14	2	53	1	2	Initial Status	RTC Status	1.0
15	15	2	54	1	3	Comms Network	Valid Data	1.0
16	16	2	54	1	3	Comms Network	Status	0.0

The following items can be viewed or modified in the local sqlite databases:

1. ‘Test Limit Values’ which defines the pass/fail thresholds for each parameter saved to the database.
2. Measurement Data from each test sequence.
3. ‘Registered IO Channels’ which define IO ports and configurations to be used.

7.1 Test Limit Values

Test limits are stored in the ‘Test Limit’ table of the database located in **c:\ucfroot\var\lib\TestLimits.sdb**’.



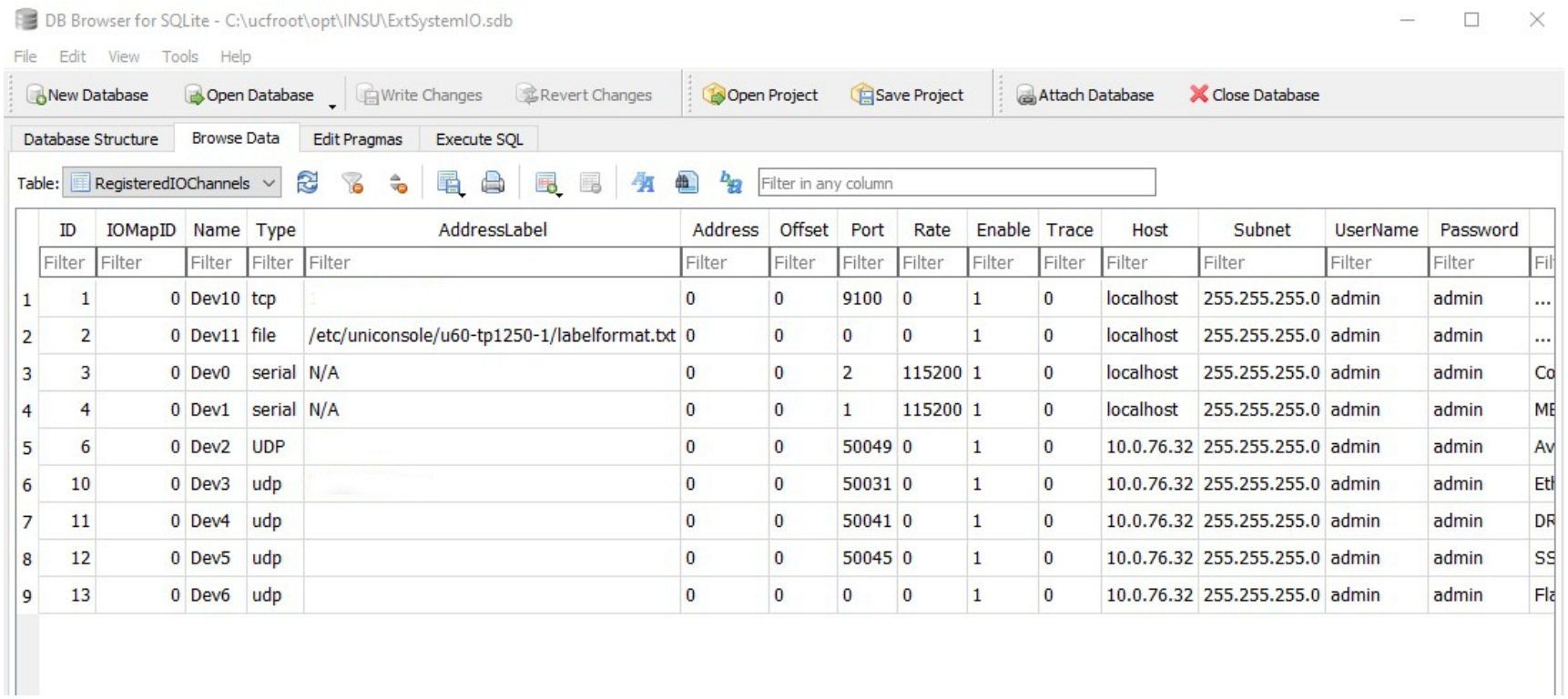
	ID	Specification	GroupName	Parameter	Lolimit	Hilimit	Description
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	0	Boot	Valid Data	1.0	1.0	Auto-generated limit
2	2	0	Boot	Status	1.0	1.0	Auto-generated limit
3	3	0	Initial Status	Valid Data	1.0	1.0	Auto-generated limit
4	4	0	Initial Status	Aux Status	1.0	1.0	Auto-generated limit
5	5	0	Initial Status	TI CRC	1.0	1.0	Auto-generated limit
6	6	0	Initial Status	Watchdog Status	1.0	1.0	Auto-generated limit
7	7	0	Initial Status	RTC Status	1.0	1.0	Auto-generated limit
8	8	0	Comms Network	Valid Data	1.0	1.0	Auto-generated limit
9	9	0	Comms Network	Status	1.0	1.0	Auto-generated limit
10	10	0	Comms Discrete IO	Valid Data	1.0	1.0	Auto-generated limit
11	11	0	Comms Discrete IO	Status	1.0	1.0	Auto-generated limit
12	12	0	Comms Wifi	Valid Data	1.0	1.0	Auto-generated limit
13	13	0	Comms Wifi	Status	1.0	1.0	Auto-generated limit
14	14	0	Comms Cellular	Valid Data	1.0	1.0	Auto-generated limit
15	15	0	Comms Cellular	RSSI	1.0	1000.0	Auto-generated limit
16	16	0	Comms Cellular	Tx Rx Speed	1.0	1000.0	Auto-generated limit

Note:

The initial test limit values are auto-generated for each measured parameter by the test executive the first time it executes the test program.

7.2 Registered IO Channels

Registered IO channels define and configure the framework IO such as serial and ethernet IO and are stored in a sqlite database.



	ID	IOMapID	Name	Type	AddressLabel	Address	Offset	Port	Rate	Enable	Trace	Host	Subnet	UserName	Password	
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Fil
1	1	0	Dev10	tcp		0	0	9100	0	1	0	localhost	255.255.255.0	admin	admin	...
2	2	0	Dev11	file	/etc/uniconsole/u60-tp1250-1/labelformat.txt	0	0	0	0	1	0	localhost	255.255.255.0	admin	admin	...
3	3	0	Dev0	serial	N/A	0	0	2	115200	1	0	localhost	255.255.255.0	admin	admin	Co
4	4	0	Dev1	serial	N/A	0	0	1	115200	1	0	localhost	255.255.255.0	admin	admin	MB
5	6	0	Dev2	UDP		0	0	50049	0	1	0	10.0.76.32	255.255.255.0	admin	admin	Av
6	10	0	Dev3	udp		0	0	50031	0	1	0	10.0.76.32	255.255.255.0	admin	admin	Eth
7	11	0	Dev4	udp		0	0	50041	0	1	0	10.0.76.32	255.255.255.0	admin	admin	DR
8	12	0	Dev5	udp		0	0	50045	0	1	0	10.0.76.32	255.255.255.0	admin	admin	SS
9	13	0	Dev6	udp		0	0	0	0	1	0	10.0.76.32	255.255.255.0	admin	admin	Fla

Notes:

1. The database is located in c:\ucfroot\opt\INSU\ExtSystemIO.sdb
2. The names of the IO Channels are DEV0 to DEVn.
3. The ‘Type’ name is case insensitive.
4. In the function call ‘TheSystem->RegisterIOChannelDevice(**1**,"The MBC serial interface")’, ‘1’ refers to DEV1.

7.2.1 Registered IO Channel Table Definitions

The following table defines the possible IO Channel configurations.

Database Field	File IO	Serial IO	TCP / UDP IO	Notes
ID				Unused.
IOMapID				Unused.
Name	DEV0-n	DEV0-n	DEV0-n	Required channel name used by the application program(s).
Type	file	serial	tcp	Required type specification determines the type of IO object to be created. Case insensitive.
AddressLabel	<filename>	ttyUSBn / com n		Text device name for Linux. Unused on Windows.
Address				Optional numeric address info.
Offset				Optional numeric address offset info
Port		1-n	<nn>	Numeric port specification. For Windows, the value translates to COM1-n.
Rate		<baud rate>		Optional numeric communication speed.
Enable	<0|1>	<0|1>	<0|1>	1 to enable the IO channel, else 0.
Trace	<0|1>	<0|1>	<0|1>	1 to optionally enable trace monitoring of the channel. Currently unused.
Host	localhost	localhost	<ip address>	Optional host specification
Subnet	255.255.255.0	255.255.255.0	255.255.255.0	Optional host subnet.
UserName	admin	admin	admin	Optional host user name.
Password	admin	admin	admin	Optional host password.