



Casting Emission Reduction Program

Prepared by:

TECHNIKON LLC

5301 Price Avenue ▶ McClellan, CA, 95652 ▶ (916) 929-8001

www.technikonllc.com

US Army Task N256 ASAM Validation Test 1

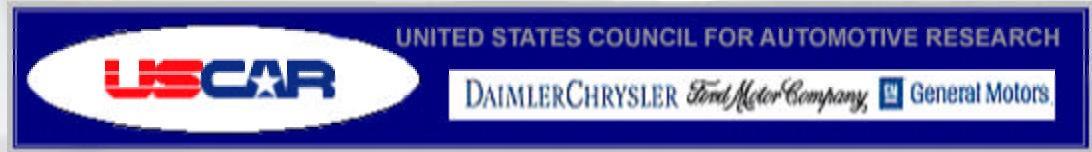
Technikon # 1256-333

This document was revised for unlimited public distribution and published on
6 May 2003

 United States
Environmental Protection Agency




AMERICAN FOUNDRY SOCIETY, INC.



Technikon LLC

**ASAM VALIDATION TEST
SYSTEM INTEGRATION AND VALIDATION LABORATORY (SIVL)
EBENCH VALIDATION TEST
ENGINEERING REPORT**

WBS 3.3.3

21 March 2001

INTRODUCTION

This report provides a technical summary of the System Integration and Validation Laboratory (SIVL), Association for Standardization of Automation and Measuring systems (ASAM) Validation Test 1.

TASK OBJECTIVE

This task was a validation test of a Standard Commands for Programmable Instruments (SCPI) Ebenc, using an Ebenc model together with an instrument simulator to simulate the SCPI Ebenc. A test suite was developed to send commands to the Ebenc, and receive responses. The responses were evaluated against the expected results. The objectives of this task were two-fold: to validate the operation of the SCPI Ebenc (model), and to evaluate the testing approach and test cases which were used in the test.

SYSTEM INTEGRATION AND VALIDATION LABORATORY (SIVL)

Technikon performed work on the SIVL that included developing instrument validation software that could be used to validate a simple SCPI instrument. The instrument that was chosen for this task is an instrument model of a SCPI Ebenc, which was developed to use with the CERP instrument simulator.

TEST PROGRAMS

The validation test program is the SIVL test executive that is used to run a validation test of a real or simulated instrument. It has the capability of reading a test data file containing the test steps, and then running the test on an instrument. The validation test program (User Interface) is shown in Figure 1-1. The test step file contains SCPI commands and expected responses for an instrument. As each test step is performed, the test program displays the SCPI command to be sent, the expected result, and the actual result. If a response is not within a defined tolerance of the expected result, an error is written to the test result log file. A summary of the results including the total errors is presented at the end of the test results log file and also in the test results summary file. Upon receipt of data channel initialization commands, the Ebenc model sends data channel values asynchronously. The validation test program receives and validates the data channel values. If data values are outside the valid range, the validation test program displays an error message and increments the count of data channel errors. A graphic representation of Ebenc data channel data is provided via the test_grafix executable shown in Figure 1-1.

The Ebenc model contains two data files that are used with the SIVL instrument simulator to perform a simulation of a SCPI emission bench instrument. One of the files defines the SCPI commands, which are accepted by the model, and the other file defines the model behavior. A copy of the Ebenc model is included in Appendix I EBENCH MODEL Version 1.4. The Ebenc instrument simulator (Engine-Ebench) is shown in Figure 1-1.

TEST EXECUTION

The validation test was performed using the test step file ebench_cmd.txt. The test step file is included in Appendix II, Ebenech Validation Test Files. This file contains a comprehensive list of valid Ebenech SCPI commands. The test was intended to validate that the Ebenech model accepts valid SCPI commands and that the Ebenech model responds with valid responses to those SCPI commands.

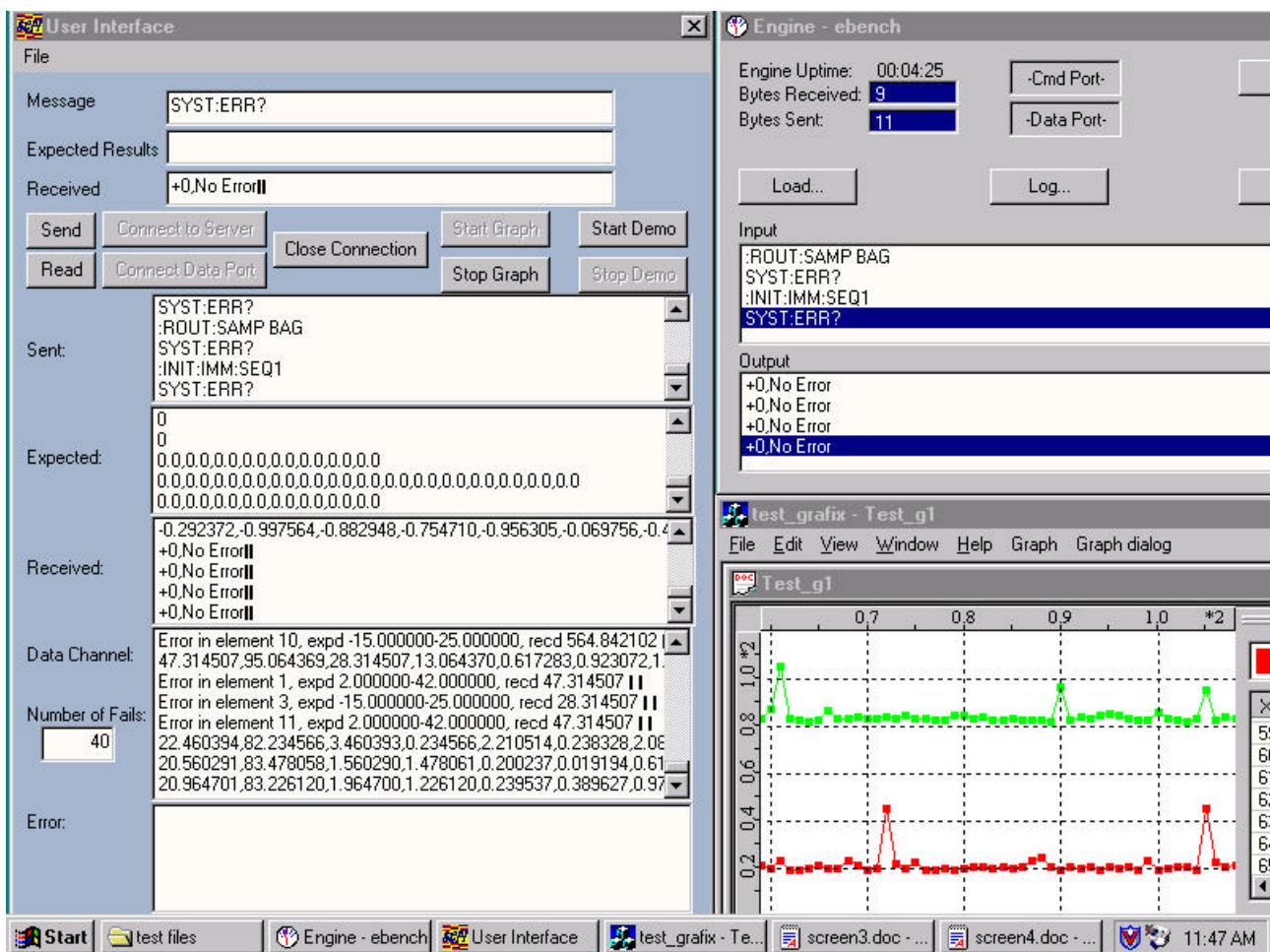


Figure 1-1 Ebench Validation Test Program and Instrument Simulator

Test Results

The results of the validation test were logged into the test results log file ebench_cmd_res.txt and the test results summary file ebench_cmd_sum.txt. The test results files are included in Appendix I, EBench Validation Test Files. These files show 40 data channel errors, 71 passed test steps, and 3 test step failures, for the following commands:

```
:SYST:TIME?  
:SYST:DATE?  
:STAT:OPER:INST:ISUM1?
```

The results of the time and date are variable and must be analyzed by hand. The results of the commands are as follows:

```
Command ::SYST:TIME?  
Received :11:50:59.00
```

```
Command : :SYST:DATE?  
Received :2001/02/12
```

These results are consistent with the time and date of the validation test.

In order to validate error handling and reporting, the EBench model was designed to respond to :STAT:OPER and :STAT:QUES commands with a one or a zero depending upon the value of a randomly generated seed value. Therefore it is expected that at least one of the :STAT commands will fail due to receiving a value of 1, when 0 is the expected result.

Similarly, data channel values returned by the EBench model are generated at random and will randomly exceed the range of valid data. Therefore, the forty data channel errors are expected results of this validation test.

The enumeration of test steps passed and test steps failed only refers to Pass/Fail test steps. Pass/Fail test steps are defined for those SCPI commands to which a response is expected. If no response is expected, validation of the test step involves ensuring that the EBench model accepts the SCPI command and any accompanying command parameters. When the EBench model encounters an illegal EBench command, a pop-up error message is produced, indicating that the received command is invalid. No pop-up error messages were generated by this validation test.

Test Conclusions

The validation test was performed with expected results including expected data anomalies to validate error handling and reporting. However, the validation test was performed on an Instrument Simulator EBench model. The EBench model was designed with fewer command-order dependencies. It is expected that actual EBench operation will include a far greater degree of command-order dependencies and will produce actual test results rather than random data values. Therefore, it is expected that this test will require modifications for command order and expected results values if it is to be run on a real (hardware) Ebench.

Appendix II

**EBENCH Model
Version 1.4**

EBench SDF data file. This file defines all of the SCPI commands which are supported by the EBench model, and maps them to simulator routines (defined in the Ebench DDL file).

```
:CAL:LIN:ACC->NoResponse
:CAL:LIN:ACQ->NoResponse
:CAL:LIN:AUTO->GetAuto
:CAL:LIN:CALC->CALCALC
:CAL:LIN:CURV->CALCURVTYP
:CAL:LIN:CURV:TYPE->CALCURVTYP
:CAL:LIN:CURV:ZFOR->CALCURVZFOR
:CAL:LIN:VER:ACQ->NoResponse
:CAL:LIN:VER:TOL->SetTolerance
:CAL:LIN:VER:TYPE->CALCVERTYP
:CONT:EBEN:CLE->SetClean
:CONT:EBEN:CLE:DUR->SetCleanTime
:CONT:EBEN:CLE:DUR?->GetCleanTime
:CONT:EBEN:CLE:INIT->SetClean
:DIAG:HUP:ACQ->NoResponse
:DIAG:HUP:CALC->NoResponse
:DIAG:LEAK:ACQ->NoResponse
:DIAG:LEAK:CALC->NoResponse
:DIAG:NEFF:ACQ->NoResponse
:DIAG:NEFF:CALC->NoResponse
:INIT:CONT->SetInitiate
:INIT:CONT:ON->SystCommSockFeedStart
:INIT:CONT:ALL->SetInitiate
:INIT:CONT:OFF->SystCommSockFeedStop
:INIT:CONT:SEQ1->SetInitiate
:INIT:CONT:SEQ2->SetInitiate
:INIT:CONT:SEQ3->SetInitiate
:INIT->Initiate
:INIT:SEQ1->Initiate
:INIT:SEQ2->Initiate
:INIT:SEQ3->Initiate
:INIT:IMM->Initiate
:INIT:IMM:ALL->Initiate
:INIT:IMM:SEQ1->Initiate
:INIT:IMM:SEQ2->Initiate
:INIT:IMM:SEQ3->Initiate
:INST:CAT?->SendInstList
:INST:CAT:FULL?->SendInstList
:INST:DEF?->SendInstList
:INST:DEF->SetInstrument
:INST:DEF:GRO->SetInstGroup
:INST:DEF:NAME->SetInstrument
:INST:DEL->DeleteInst
:INST:DEL:ALL->DeleteInst
:INST:DEL:NAME->DeleteInst
:INST:SEL->SelectInst
:INST:SEL:GRO->SelectInst
:MEM:CLE->DeleteTable
:MEM:CLE:NAME->DeleteTable
:MEM:CLE:TABLE->DeleteTable
```

```
:MEM:DEL->DeleteTable
:MEM:DEL:NAME->DeleteTable
:MEM:DEL:ALL->DeleteTable
:MEM:DATA->EnterTable
:MEM:DATA?->ReturnTable
:MEM:TABL:BNUM->SetTable
:MEM:TABL:BNUM:POIN?->GetEntries
:MEM:TABL:CCUR->SetTable
:MEM:TABL:CCUR:POIN?->GetEntries
:MEM:TABL:CONC->SetTable
:MEM:TABL:CONC:POIN?->GetEntries
:MEM:TABL:CPO->SetTable
:MEM:TABL:CPO:POIN?->GetEntries
:MEM:TABL:DEFine->DefineTable
:MEM:TABL:DFAC->SetTable
:MEM:TABL:DFAC:POIN?->GetEntries
:MEM:TABL:DLAS->SetTable
:MEM:TABL:DLAS:POIN?->GetEntries
:MEM:TABL:DLIN->SetTable
:MEM:TABL:DLIN:POIN?->GetEntries
:MEM:TABL:EXP->SetTable
:MEM:TABL:EXP:POIN?->GetEntries
:MEM:TABL:LAB->SetTable
:MEM:TABL:LAB:POIN?->GetEntries
:MEM:TABL:LOG->SetTable
:MEM:TABL:LOG:POIN?->GetEntries
:MEM:TABL:NCUR->SetTable
:MEM:TABL:NCUR:POIN?->GetEntries
:MEM:TABL:RAW->SetTable
:MEM:TABL:RAW:POIN?->GetEntries
:MEM:TABL:SEL->SelectTable
:MEM:TABL:TOL->SetTable
:MEM:TABL:TOL:POIN?->GetEntries
:MEM:TABL:WFAC->SetTable
:MEM:TABL:WFAC:POIN?->GetEntries
:ROUT:SAMP->SetSamplePoint
:ROUT:SAMP:OPEN->SetSamplePoint
:ROUT:SAMP:CAT?->SendRouteList
:SENS:AVER:COUN->SetAvgCount
:SENS:AVER:TCON->SetTCOn
:SENS:AVER:TYPE->SetAvgType
:SENS:AVER:STAT->SetAvgState
:SENS:AVER1:COUN->SetAvgCount
:SENS:AVER1:TCON->SetTCOn
:SENS:AVER1:TYPE->SetAvgType
:SENS:AVER1:STAT->SetAvgState
:SENS:CONC:CSET->SetCSET
:SENS:CONC:CSET?->ReturnData
:SENS:CONC:LOW->SetLower
:SENS:CONC:LOW?->GetLower
:SENS:CONC:LSET->SetLSET
:SENS:CONC:LSET?->ReturnData
:SENS:CONC:RANG->SetRange
:SENS:CONC:RANG?->ReturnRange
```

```
:SENS:CONC:RANG:AUTO->SetConcState
:SENS:CONC:RANG:AUTO:LOW->SetLower
:SENS:CONC:RANG:AUTO:LOW?->ReturnRange
:SENS:CONC:RANG:AUTO:STATE->SetConcState
:SENS:CONC:RANG:AUTO:UPP->SetUpper
:SENS:CONC:RANG:AUTO:UPP?->ReturnRange
:SENS:CONC:RANG:FIX->SetRange
:SENS:CONC:RANG:FIX?->ReturnRange
:SENS:CONC:TAL->SetConcTAL
:SENS:CONC:UPP->SetUpper
:SENS:CORR:AUTO->SetCorrAuto
:SENS:CORR:CALC->SetCorrCalc
:SENS:CORR:SPO:ACQ->SetSPOAcq
:SENS:CORR:SPO:DTOL->SetSPODTOL
:SENS:CORR:SPAN:ACQ->SetSPOAcq
:SENS:CORR:SPAN:DTOL->SetSPODTOL
:SENS:CORR:STAT->SetCorrState
:SENS:CORR:STAT?->GetCorrState
:SENS:CORR:ZERO:ACQ->SetZeroAcq
:SENS:CORR:ZERO:DTOL->SetZeroDTOL
:SENS:DATA?->ReturnData
:SENS:FUNC->SetFunction
:SENS:FUNC:CONC->SetConcurrent
:SENS:STAB->SetSTABState
:SENS:STAB:NTOL->SetSTABNTOL
:SENS:STAB:NTOL?->GetSTABNTOL
:SENS:STAB:STAT->SetSTABState
:SENS:STAB:TIME1->SetSTABTime
:SENS:STAB:TIME1?->GetSTABTime
:SENS:STAB:TIME2->SetSTABTime
:SENS:STAB:TIME2?->GetSTABTime
:SENS:STAB:TIME3->SetSTABTime
:SENS:STAB:TIME3?->GetSTABTime
:SENS:STAB:TIME4->SetSTABTime
:SENS:STAB:TIME4?->GetSTABTime
:SENS:STAB:TIME5->SetSTABTime
:SENS:STAB:TIME5?->GetSTABTime
:SYST:COMM:SOCK:ADDR->SetSocketAddress
:SYST:COMM:SOCK:PORT->SetSocketPort
:SYST:COMM:SOCK:TYPE->SetSocketType
:SYST:COMM:SOCK:CONN->ConnectSocket
:SYST:COMM:SOCK:DISC->DisconnectSocket
:SYST:COMM:SOCK:FEED->SetSocketData
:SYST:COMM:SOCK:FEED:OCON->SetSocketCondition
:SYST:COMM:SOCK:FEED:SCON->SetSocketCondition
:SYST:COMM:SOCK:LIST->NoResponse
:SYST:COMM:SOCK1:ADDR->SetSocketAddress
:SYST:COMM:SOCK1:PORT->SetSocketPort
:SYST:COMM:SOCK1:TYPE->SetSocketType
:SYST:COMM:SOCK1:CONN->ConnectSocket
:SYST:COMM:SOCK1:DISC->DisconnectSocket
:SYST:COMM:SOCK1:FEED->SetSocketData
:SYST:COMM:SOCK1:FEED:OCON->SetSocketCondition
:SYST:COMM:SOCK1:FEED:SCON->SetSocketCondition
```

:SYST:COMM:SOCK1:LIST->NoResponse
:SYST:DATE->SetDate
:SYST:DATE?->BenchDate
SYST:ERR?->ReturnNoError
:SYST:ERR?->ReturnNo Error
:SYST:ERR:ALL?->ReturnNoError
:SYST:ERR:CODE?->ReturnNoError
:SYST:ERR:CODE:ALL?->ReturnNoError
:SYST:ERR:CODE:NEXT?->ReturnNoError
:SYST:ERR:ENAB->SetErrList
:SYST:ERR:ENAB:ADD->SetErrList
:SYST:ERR:ENAB:DEL->SetErrList
:SYST:ERR:ENAB:LIST->SetErrList
:SYST:ERR:NEXT?->ReturnNoError
:SYST:LOCK:REL->NoResponse
:SYST:LOCK:REQ?->ReturnOne
:SYST:LOCK:OWN?->ReturnOwner
:SYST:TIME?->BenchTime
:SYST:TIME->SetTime
:SYST:TIME:TIM->SetTimerState
:SYST:TIME:TIM:COUN->SetTimerCount
:SYST:TIME:TIM:STAT->SetTimerState
:TRIG->NoResponse
:TRIG:ECO->SetTriggerCount
:TRIG:LINK->SetTriggerLink
:TRIG:SOUR->SetTriggerSource
:TRIG:TIM->SetTriggerTime
:TRIG:SEQ1:ECO->SetTriggerCount
:TRIG:SEQ2:ECO->SetTriggerCount
:TRIG:SEQ3:ECO->SetTriggerCount
:TRIG:SEQ1:LINK->SetTriggerLink
:TRIG:SEQ2:LINK->SetTriggerLink
:TRIG:SEQ3:LINK->SetTriggerLink
:TRIG:SEQ1:SOUR->SetTriggerSource
:TRIG:SEQ2:SOUR->SetTriggerSource
:TRIG:SEQ3:SOUR->SetTriggerSource
:TRIG:SEQ1:TIM->SetTriggerTime
:TRIG:SEQ2:TIM->SetTriggerTime
:TRIG:SEQ3:TIM->SetTriggerTime
:STAT:OPER?->StatusOperation
:STAT:OPER:BIT9?->StatusOperation
:STAT:OPER:BIT9:COND?->StatusOperation
:STAT:OPER:BIT9:ENAB->SetOEnableList
:STAT:OPER:BIT9:EVEN?->StatusOperation
:STAT:OPER:BIT9:NTR->SetOEnableList
:STAT:OPER:BIT9:PTR->SetOEnableList
:STAT:OPER:COND?->StatusOperation
:STAT:OPER:ENAB->SetOEnableList
:STAT:OPER:EVEN?->StatusOperation
:STAT:OPER:INST?->StatusOperation
:STAT:OPER:INST:COND?->StatusOperation
:STAT:OPER:INST:ENAB->SetOEnableList
:STAT:OPER:INST:EVEN?->StatusOperation
:STAT:OPER:INST:ISUM0?->StatusOperation

```

*STB?->SetOneorZero
*ESR?->SetOneorZero
:STAT:OPER:INST:ISUM0?->StatusOperation
:STAT:OPER:INST:ISUM0:BIT10?->StatusOperation
:STAT:OPER:INST:ISUM0:BIT10:CAL?->SetOneorZero
:STAT:OPER:INST:ISUM0:BIT10:CAL:COND?->StatusOperation
:STAT:OPER:INST:ISUM0:BIT10:CAL:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM0:BIT10:CAL:EVEN?->SetOneorZero
:STAT:OPER:INST:ISUM0:BIT10:CAL:NTR->SetOEnableList
:STAT:OPER:INST:ISUM0:BIT10:CAL:PTR->SetOEnableList
:STAT:OPER:INST:ISUM0:BIT10:COND?->StatusOperation
:STAT:OPER:INST:ISUM0:BIT10:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM0:BIT10:EVEN?->StatusOperation
:STAT:OPER:INST:ISUM0:NTR->SetOEnableList
:STAT:OPER:INST:ISUM0:PTR->SetOEnableList
:STAT:OPER:INST:ISUM0:BIT10?->StatusOperation
:STAT:OPER:INST:ISUM0:COND?->StatusOperation
:STAT:OPER:INST:ISUM0:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM0:EVEN?->StatusOperation
:STAT:OPER:INST:ISUM0:NTR->SetOEnableList
:STAT:OPER:INST:ISUM0:PTR->SetOEnableList
:STAT:OPER:INST:ISUM1?->StatusOperation
:STAT:OPER:INST:ISUM1:BIT10?->StatusOperation
:STAT:OPER:INST:ISUM1:BIT10:CAL?->SetOneorZero
:STAT:OPER:INST:ISUM1:BIT10:CAL:COND?->StatusOperation
:STAT:OPER:INST:ISUM1:BIT10:CAL:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM1:BIT10:CAL:EVEN?->SetOneorZero
:STAT:OPER:INST:ISUM1:BIT10:CAL:NTR->SetOEnableList
:STAT:OPER:INST:ISUM1:BIT10:CAL:PTR->SetOEnableList
:STAT:OPER:INST:ISUM1:BIT10:COND?->StatusOperation
:STAT:OPER:INST:ISUM1:BIT10:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM1:BIT10:EVEN?->StatusOperation
:STAT:OPER:INST:ISUM1:BIT10:NTR->SetOEnableList
:STAT:OPER:INST:ISUM1:BIT10:PTR->SetOEnableList
:STAT:OPER:INST:ISUM1:COND?->StatusOperation
:STAT:OPER:INST:ISUM1:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM1:EVEN?->StatusOperation
:STAT:OPER:INST:ISUM1:NTR->SetOEnableList
:STAT:OPER:INST:ISUM1:PTR->SetOEnableList
:STAT:OPER:INST:ISUM2?->StatusOperation
:STAT:OPER:INST:ISUM2:BIT10?->StatusOperation
:STAT:OPER:INST:ISUM2:BIT10:CAL?->SetOneorZero
:STAT:OPER:INST:ISUM2:BIT10:CAL:COND?->StatusOperation
:STAT:OPER:INST:ISUM2:BIT10:CAL:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM2:BIT10:CAL:EVEN?->SetOneorZero
:STAT:OPER:INST:ISUM2:BIT10:CAL:NTR->SetOEnableList
:STAT:OPER:INST:ISUM2:BIT10:CAL:PTR->SetOEnableList
:STAT:OPER:INST:ISUM2:BIT10:COND?->StatusOperation
:STAT:OPER:INST:ISUM2:BIT10:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM2:BIT10:EVEN?->StatusOperation
:STAT:OPER:INST:ISUM2:BIT10:NTR->SetOEnableList
:STAT:OPER:INST:ISUM2:BIT10:PTR->SetOEnableList
:STAT:OPER:INST:ISUM2:COND?->StatusOperation
:STAT:OPER:INST:ISUM2:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM2:EVEN?->StatusOperation

```

:STAT:OPER:INST:ISUM2:NTR->SetOEnableList
:STAT:OPER:INST:ISUM2:PTR->SetOEnableList
:STAT:OPER:INST:ISUM3?->StatusOperation
:STAT:OPER:INST:ISUM3:BIT10?->StatusOperation
:STAT:OPER:INST:ISUM3:BIT10:CAL?->SetOneorZero
:STAT:OPER:INST:ISUM3:BIT10:CAL:COND?->StatusOperation
:STAT:OPER:INST:ISUM3:BIT10:CAL:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM3:BIT10:CAL:EVEN?->SetOneorZero
:STAT:OPER:INST:ISUM3:BIT10:CAL:NTR->SetOEnableList
:STAT:OPER:INST:ISUM3:BIT10:CAL:PTR->SetOEnableList
:STAT:OPER:INST:ISUM3:BIT10:COND?->StatusOperation
:STAT:OPER:INST:ISUM3:BIT10:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM3:BIT10:EVEN?->StatusOperation
:STAT:OPER:INST:ISUM3:BIT10:NTR->SetOEnableList
:STAT:OPER:INST:ISUM3:BIT10:PTR->SetOEnableList
:STAT:OPER:INST:ISUM3:COND?->StatusOperation
:STAT:OPER:INST:ISUM3:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM3:EVEN?->StatusOperation
:STAT:OPER:INST:ISUM3:NTR->SetOEnableList
:STAT:OPER:INST:ISUM3:PTR->SetOEnableList
:STAT:OPER:INST:ISUM4?->StatusOperation
:STAT:OPER:INST:ISUM4:BIT10?->StatusOperation
:STAT:OPER:INST:ISUM4:BIT10:CAL?->SetOneorZero
:STAT:OPER:INST:ISUM4:BIT10:CAL:COND?->StatusOperation
:STAT:OPER:INST:ISUM4:BIT10:CAL:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM4:BIT10:CAL:EVEN?->SetOneorZero
:STAT:OPER:INST:ISUM4:BIT10:CAL:NTR->SetOEnableList
:STAT:OPER:INST:ISUM4:BIT10:CAL:PTR->SetOEnableList
:STAT:OPER:INST:ISUM4:BIT10:COND?->StatusOperation
:STAT:OPER:INST:ISUM4:BIT10:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM4:BIT10:EVEN?->StatusOperation
:STAT:OPER:INST:ISUM4:BIT10:NTR->SetOEnableList
:STAT:OPER:INST:ISUM4:BIT10:PTR->SetOEnableList
:STAT:OPER:INST:ISUM4:COND?->StatusOperation
:STAT:OPER:INST:ISUM4:ENAB->SetOEnableList
:STAT:OPER:INST:ISUM4:EVEN?->StatusOperation
:STAT:OPER:INST:ISUM4:NTR->SetOEnableList
:STAT:OPER:INST:ISUM4:PTR->SetOEnableList
:STAT:OPER:INST:NTR->SetOEnableList
:STAT:OPER:INST:PTR->SetOEnableList
:STAT:QUES?->StatusOperation
:STAT:QUES:BIT9?->StatusOperation
:STAT:QUES:BIT9:COND?->StatusOperation
:STAT:QUES:BIT9:ENAB->SetQEnableList
:STAT:QUES:BIT9:EVEN?->StatusOperation
:STAT:QUES:BIT9:NTR->SetQEnableList
:STAT:QUES:BIT9:PTR->SetQEnableList
:STAT:QUES:COND?->StatusOperation
:STAT:QUES:ENAB->SetQEnableList
:STAT:QUES:EVEN?->StatusOperation
:STAT:QUES:INST?->StatusOperation
:STAT:QUES:INST:COND?->StatusOperation
:STAT:QUES:INST:ENAB->SetQEnableList
:STAT:QUES:INST:EVEN?->StatusOperation

```
:STAT:QUES:INST:ISUM0?->StatusOperation
:STAT:QUES:INST:ISUM0:BIT10?->StatusOperation
:STAT:QUES:INST:ISUM0:BIT10:CAL?->SetOneorZero
:STAT:QUES:INST:ISUM0:BIT10:CAL:COND?->StatusOperation
:STAT:QUES:INST:ISUM0:BIT10:CAL:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM0:BIT10:CAL:EVEN?->SetOneorZero
:STAT:QUES:INST:ISUM0:BIT10:CAL:NTR->SetQEnableList
:STAT:QUES:INST:ISUM0:BIT10:CAL:PTR->SetQEnableList
:STAT:QUES:INST:ISUM0:BIT10:COND?->StatusOperation
:STAT:QUES:INST:ISUM0:BIT10:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM0:BIT10:EVEN?->StatusOperation
:STAT:QUES:INST:ISUM0:BIT10:NTR->SetQEnableList
:STAT:QUES:INST:ISUM0:BIT10:PTR->SetQEnableList
:STAT:QUES:INST:ISUM0:COND?->StatusOperation
:STAT:QUES:INST:ISUM0:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM0:EVEN?->StatusOperation
:STAT:QUES:INST:ISUM0:NTR->SetQEnableList
:STAT:QUES:INST:ISUM0:PTR->SetQEnableList
:STAT:QUES:INST:ISUM1?->StatusOperation
:STAT:QUES:INST:ISUM1:BIT10?->StatusOperation
:STAT:QUES:INST:ISUM1:BIT10:CAL?->SetOneorZero
:STAT:QUES:INST:ISUM1:BIT10:CAL:COND?->StatusOperation
:STAT:QUES:INST:ISUM1:BIT10:CAL:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM1:BIT10:CAL:EVEN?->SetOneorZero
:STAT:QUES:INST:ISUM1:BIT10:CAL:NTR->SetQEnableList
:STAT:QUES:INST:ISUM1:BIT10:CAL:PTR->SetQEnableList
:STAT:QUES:INST:ISUM1:BIT10:COND?->StatusOperation
:STAT:QUES:INST:ISUM1:BIT10:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM1:BIT10:EVEN?->StatusOperation
:STAT:QUES:INST:ISUM1:BIT10:NTR->SetQEnableList
:STAT:QUES:INST:ISUM1:BIT10:PTR->SetQEnableList
:STAT:QUES:INST:ISUM1:COND?->StatusOperation
:STAT:QUES:INST:ISUM1:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM1:EVEN?->StatusOperation
:STAT:QUES:INST:ISUM1:NTR->SetQEnableList
:STAT:QUES:INST:ISUM1:PTR->SetQEnableList
:STAT:QUES:INST:ISUM2?->StatusOperation
:STAT:QUES:INST:ISUM2:BIT10?->StatusOperation
:STAT:QUES:INST:ISUM2:BIT10:CAL?->SetOneorZero
:STAT:QUES:INST:ISUM2:BIT10:CAL:COND?->StatusOperation
:STAT:QUES:INST:ISUM2:BIT10:CAL:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM2:BIT10:CAL:EVEN?->SetOneorZero
:STAT:QUES:INST:ISUM2:BIT10:CAL:NTR->SetQEnableList
:STAT:QUES:INST:ISUM2:BIT10:CAL:PTR->SetQEnableList
:STAT:QUES:INST:ISUM2:BIT10:COND?->StatusOperation
:STAT:QUES:INST:ISUM2:BIT10:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM2:BIT10:EVEN?->StatusOperation
:STAT:QUES:INST:ISUM2:BIT10:NTR->SetQEnableList
:STAT:QUES:INST:ISUM2:BIT10:PTR->SetQEnableList
:STAT:QUES:INST:ISUM2:COND?->StatusOperation
:STAT:QUES:INST:ISUM2:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM2:EVEN?->StatusOperation
:STAT:QUES:INST:ISUM2:NTR->SetQEnableList
:STAT:QUES:INST:ISUM2:PTR->SetQEnableList
```

```

:STAT:QUES:INST:ISUM3?->StatusOperation
:STAT:QUES:INST:ISUM3:BIT10?->StatusOperation
:STAT:QUES:INST:ISUM3:BIT10:CAL?->SetOneorZero
:STAT:QUES:INST:ISUM3:BIT10:CAL:COND?->StatusOperation
:STAT:QUES:INST:ISUM3:BIT10:CAL:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM3:BIT10:CAL:EVEN?->SetOneorZero
:STAT:QUES:INST:ISUM3:BIT10:CAL:NTR->SetQEnableList
:STAT:QUES:INST:ISUM3:BIT10:CAL:PTR->SetQEnableList
:STAT:QUES:INST:ISUM3:BIT10:COND?->StatusOperation
:STAT:QUES:INST:ISUM3:BIT10:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM3:BIT10:EVEN?->StatusOperation
:STAT:QUES:INST:ISUM3:BIT10:NTR->SetQEnableList
:STAT:QUES:INST:ISUM3:BIT10:PTR->SetQEnableList
:STAT:QUES:INST:ISUM3:COND?->StatusOperation
:STAT:QUES:INST:ISUM3:ENAB->SetEnableList
:STAT:QUES:INST:ISUM3:EVEN?->StatusOperation
:STAT:QUES:INST:ISUM3:NTR->SetQEnableList
:STAT:QUES:INST:ISUM3:PTR->SetQEnableList
:STAT:QUES:INST:ISUM4?->StatusOperation
:STAT:QUES:INST:ISUM4:BIT10?->StatusOperation
:STAT:QUES:INST:ISUM4:BIT10:CAL?->SetOneorZero
:STAT:QUES:INST:ISUM4:BIT10:CAL:COND?->StatusOperation
:STAT:QUES:INST:ISUM4:BIT10:CAL:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM4:BIT10:CAL:EVEN?->SetOneorZero
:STAT:QUES:INST:ISUM4:BIT10:CAL:NTR->SetQEnableList
:STAT:QUES:INST:ISUM4:BIT10:CAL:PTR->SetQEnableList
:STAT:QUES:INST:ISUM4:BIT10:COND?->StatusOperation
:STAT:QUES:INST:ISUM4:BIT10:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM4:BIT10:EVEN?->StatusOperation
:STAT:QUES:INST:ISUM4:BIT10:NTR->SetQEnableList
:STAT:QUES:INST:ISUM4:BIT10:PTR->SetQEnableList
:STAT:QUES:INST:ISUM4:COND?->StatusOperation
:STAT:QUES:INST:ISUM4:ENAB->SetQEnableList
:STAT:QUES:INST:ISUM4:EVEN?->StatusOperation
:STAT:QUES:INST:ISUM4:NTR->SetQEnableList
:STAT:QUES:INST:ISUM4:PTR->SetQEnableList
:STAT:QUES:INST:NTR->SetQEnableList
:STAT:QUES:INST:PTR->SetQEnableList
*RST->Reset
*WAI->WaitForComplete

```

EBench DDL file. This file defines the functionality supported by the EBench model. It uses the Instrument Definition Language (InDL) defined in the Instrument Simulator InDL reference document.

Instrument BagBench

Var

```
CONC1 : NUMERIC;  
CONC2 : NUMERIC;  
CONC3 : NUMERIC;  
CONC4 : NUMERIC;  
SDEV1 : NUMERIC;  
SDEV2 : NUMERIC;  
SDEV3 : NUMERIC;  
SDEV4 : NUMERIC;  
DATA1 : NUMERIC;  
DATA2 : NUMERIC;  
DATA3 : NUMERIC;  
DATA4 : NUMERIC;  
TAB1 : NUMERIC;  
TAB2 : NUMERIC;  
TAB3 : NUMERIC;  
TAB4 : NUMERIC;  
TEMPF : NUMERIC;  
CSETX : NUMERIC;  
CSETY : NUMERIC;  
LSETX : NUMERIC;  
LSETY : NUMERIC;  
CONCLower : NUMERIC;  
CONCUpper : NUMERIC;  
CONCRRange : NUMERIC;  
CONCTAL : NUMERIC;  
CORRZeroDTOL : NUMERIC;  
CORRSPDTOL : NUMERIC;  
STABNTOL : NUMERIC;  
STABTime : NUMERIC;  
CONCState : BOOLEAN;  
CORRState : BOOLEAN;  
CORRZeroAcq : BOOLEAN;
```

```
CORRSPOAcq : BOOLEAN;
CORRAuto : BOOLEAN;
CORRCalc : BOOLEAN;
SenseConc : BOOLEAN;
STABState : BOOLEAN;
SenseFunc : INTEGER;
LSETType : STRING;
NUMDATA : INTEGER;
NUMPARAM : INTEGER;
NUMENTRIES : INTEGER;
NULLSTRING : STRING;
STATUS : NUMERIC;
WORKING : BOOLEAN;
SamplePoint : STRING;
TimerCount : INTEGER;
TriggerCount : INTEGER;
TriggerTime : INTEGER;
TriggerSource : STRING;
TriggerLink : STRING;
ReturnCount : INTEGER;
SelTable : STRING;
TableStruct : STRING;
TableMax : INTEGER;
ErrEnableList : STRING;
StatQEnableList : INTEGER;
StatOEnableList : INTEGER;
DateString : STRING;
TimeString : STRING;
TimerState : INTEGER;
SocketAddress : STRING;
SocketPort : INTEGER;
SocketType : STRING;
SocketCondition : STRING;
SocketData : STRING;
SocketNUM : INTEGER;
InstName1 : STRING;
InstName2 : STRING;
```

```
InstName3 : STRING;
InstName4 : STRING;
InstName5 : STRING;
InstName6 : STRING;
InstNum1 : INTEGER;
InstNum2 : INTEGER;
InstNum3 : INTEGER;
InstNum4 : INTEGER;
InstNum5 : INTEGER;
InstNum6 : INTEGER;
InstCount1 : INTEGER;
InstCount2 : INTEGER;
InstCount3 : INTEGER;
InstCount4 : INTEGER;
InstCount5 : INTEGER;
InstCount6 : INTEGER;
InstCount : INTEGER;
InstGroup : STRING;
InstGroup1 : STRING;
InstGroup2 : STRING;
InstGroup3 : STRING;
InstGroup4 : STRING;
InstGroup5 : STRING;
InstGroup6 : STRING;
SelectGroup : INTEGER;
GroupNum1 : INTEGER;
GroupNum2 : INTEGER;
GroupNum3 : INTEGER;
GroupNum4 : INTEGER;
GroupNum5 : INTEGER;
GroupNum6 : INTEGER;
GroupCount : INTEGER;
SelGroupNum : INTEGER;
InputGroup : STRING;
AvgCount : INTEGER;
AvgTCON : STRING;
AvgType : STRING;
```

```
AvgState : STRING;  
SocketConnect : BOOLEAN;  
Clean : BOOLEAN;  
CleanTime : INTEGER;  
OK : INTEGER;
```

```
End Var
```

```
Event BenchDate
```

```
    RESPOND DATE;
```

```
End Event
```

```
Event SetDate
```

```
(
```

```
    term1 : STRING
```

```
)
```

```
    DateString := term1;
```

```
End Event
```

```
Event SetTime
```

```
(
```

```
    term1 : STRING
```

```
)
```

```
    TimeString := term1;
```

```
End Event
```

```
Event SetTimerCount
```

```
(
```

```
    term1 : INTEGER
```

```
)
```

```
    TimerCount := term1;
```

```
End Event
```

```
Event BenchTime
```

```
    RESPOND TIME;
```

```
End Event
```

```
Event SetTimerState
```

```

(
    term1 : STRING
)
IF term1 == "ON" THEN
    TimerState := 1;
ELSE
    TimerState := 0;
END IF
End Event

```

```

Event NoResponse
    WORKING :=TRUE;
End Event

```

```

Event UnknownResponse
    RESPOND 0;
End Event

```

```

Event ReturnZero
    RESPOND 0;
End Event

```

```

Event ReturnOne
    RESPOND 1;
End Event

```

```

Event StatusOperation
    Var
        Seed : INTEGER;
    End Var
    Seed := RANDOM;
    IF Seed > 1000 THEN
        RESPOND "10101";
    ELSE
        RESPOND 0;
    END IF

```

End Event

Event SetOneorZero

Var

 Seed : INTEGER;

End Var

 Seed := RANDOM;

 IF Seed > 1000 THEN

 RESPOND 1;

 ELSE

 RESPOND 0;

 END IF

End Event

Event ReturnOwner

 RESPOND "AIGER";

End Event

Event ReturnNoError

 RESPOND "+0,No Error";

End Event

Event WaitForComplete

 WORKING :=TRUE;

End Event

Event ReturnData

Var

 Seed : INTEGER;

End Var

 Seed := RANDOM;

 CONC1 := SIN(Seed);

 CONC2 := SIN(Seed + 45);

 CONC3 := SIN(Seed + 90);

```

CONC4 := SIN(Seed + 135);

Seed := RANDOM;

SDEV1 := SIN(Seed);
SDEV2 := SIN(Seed + 45);
SDEV3 := SIN(Seed + 90);
SDEV4 := SIN(Seed + 135);

IF SelectGroup == 6 THEN
    SelGroupNum := GroupNum6;
END IF

IF SelectGroup == 5 THEN
    SelGroupNum := GroupNum5;
END IF

IF SelectGroup == 4 THEN
    SelGroupNum := GroupNum4;
END IF

IF SelectGroup == 3 THEN
    SelGroupNum := GroupNum3;
END IF

IF SelectGroup == 2 THEN
    SelGroupNum := GroupNum2;
END IF

IF SelectGroup == 1 THEN
    SelGroupNum := GroupNum1;
END IF

IF SelGroupNum == 0 THEN
    RESPOND CONC1;
END IF

IF SelGroupNum == 1 THEN
    RESPOND CONC1;
END IF

IF SelGroupNum == 2 THEN
    RESPOND CONC1,SDEV1;
END IF

IF SelGroupNum == 3 THEN
    RESPOND CONC1,SDEV1,CONC2;

```

```

END IF
IF SelGroupNum == 4 THEN
RESPOND CONC1,SDEV1,CONC2,SDEV2;
END IF
IF SelGroupNum == 5 THEN
RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3;
END IF
IF SelGroupNum == 6 THEN
RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3,SDEV3;
END IF
IF SelGroupNum == 7 THEN
RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3,SDEV3,CONC4;
END IF
IF SelGroupNum == 8 THEN
RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3,SDEV3,CONC4,SDEV4;
END IF
IF SelGroupNum == 9 THEN
RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3,SDEV3,CONC4,SDEV4,CONC1;
END IF
IF SelGroupNum == 10 THEN
RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3,SDEV3,CONC4,SDEV4,CONC2,SDEV1;
END IF

```

End Event

Event SetClean

Clean :=TRUE;

End Event

Event SetCleanTime

(

term1 : INTEGER

)

CleanTime := term1;

End Event

Event GetCleanTime

 RESPOND CleanTime;

End Event

Event SetInstrument

(

 term1 : STRING,

 term2 : INTEGER

)

 IF InstCount == 0 THEN

 InstName1 := term1;

 InstNum1 := term2;

 END IF

 IF InstCount == 1 THEN

 InstName2 := term1;

 InstNum2 := term2;

 END IF

 IF InstCount == 2 THEN

 InstName3 := term1;

 InstNum3 := term2;

 END IF

 IF InstCount == 3 THEN

 InstName4 := term1;

 InstNum4 := term2;

 END IF

 IF InstCount == 4 THEN

 InstName5 := term1;

 InstNum5 := term2;

 END IF

 IF InstCount == 5 THEN

 InstName6 := term1;

 InstNum6 := term2;

 END IF

 InstCount := InstCount + 1;

End Event

```

Event SetInstGroup
(
    term1 : STRING,
    term2 : STRING,
    term3 : STRING,
    term4 : STRING,
    term5 : STRING,
    term6 : STRING,
    term7 : STRING,
    term8 : STRING,
    term9 : STRING,
    term10 : STRING,
    term11 : STRING
)
NUMDATA := 10;
InstGroup := term1;
IF term1 == NULLSTRING THEN
    NUMDATA := 9;
    InstGroup := term2;
END IF
IF term2 == NULLSTRING THEN
    NUMDATA := 8;
    InstGroup := term3;
END IF
IF term3 == NULLSTRING THEN
    NUMDATA := 7;
    InstGroup := term4;
END IF
IF term4 == NULLSTRING THEN
    NUMDATA := 6;
    InstGroup := term5;
END IF
IF term5 == NULLSTRING THEN
    NUMDATA := 5;
    InstGroup := term6;
END IF

```

```

IF term6 == NULLSTRING THEN
    NUMDATA := 4;
    InstGroup :=term7;
END IF

IF term7 == NULLSTRING THEN
    NUMDATA := 3;
    InstGroup :=term8;
END IF

IF term8 == NULLSTRING THEN
    NUMDATA := 2;
    InstGroup :=term9;
END IF

IF term9 == NULLSTRING THEN
    NUMDATA := 1;
    InstGroup :=term10;
END IF

IF term10 == NULLSTRING THEN
    NUMDATA := 0;
    InstGroup :=term11;
END IF

IF GroupCount == 6 THEN
    GroupCount := 0;
END IF

IF GroupCount == 5 THEN
    InstGroup6 := InstGroup;
    InstCount6 := NUMDATA;
    GroupCount := 6;
END IF

IF GroupCount == 4 THEN
    InstGroup5 := InstGroup;
    InstCount5 := NUMDATA;
    GroupCount := 5;
END IF

IF GroupCount == 3 THEN
    GroupCount := 4;
    InstCount4 := NUMDATA;
    InstGroup4 := InstGroup;

```

```

END IF
IF GroupCount == 2 THEN
    GroupCount := 3;
    InstCount3 := NUMDATA;
    InstGroup3 := InstGroup;
END IF
IF GroupCount == 1 THEN
    GroupCount := 2;
    InstCount2 := NUMDATA;
    InstGroup2 := InstGroup;
END IF
IF GroupCount == 0 THEN
    GroupCount := 1;
    InstCount1 := NUMDATA;
    InstGroup1 := InstGroup;
END IF
term1 := "";
term2 := "";
term3 := "";
term4 := "";
term5 := "";
term6 := "";
term7 := "";
term8 := "";
term9 := "";
term10 := "";
term11 := "";

```

End Event

```

Event DeleteInst
(
    term1 : STRING
)
InstCount := 0;
End Event

```

```

Event SelectInst
(
    term1 : STRING
)
    InputGroup := term1;
    IF InputGroup == InstGroup1 THEN
        SelectGroup := 1;
    END IF
    IF InputGroup == InstGroup2 THEN
        SelectGroup := 2;
    END IF
    IF InputGroup == InstGroup3 THEN
        SelectGroup := 3;
    END IF
    IF InputGroup == InstGroup4 THEN
        SelectGroup := 4;
    END IF
    IF InputGroup == InstGroup5 THEN
        SelectGroup := 5;
    END IF
    IF InputGroup == InstGroup6 THEN
        SelectGroup := 6;
    END IF

```

End Event

```

Event SendInstList
    RESPOND "CONC1,CONC2,CONC3,CONC4,CONC5,CONC6";
End Event

```

```

Event DeleteInst
    InstCount := 0;
End Event

```

```

Event SetTolerance
(

```

```
term1 : NUMERIC
)
    WORKING := TRUE;
End Event
```

```
Event SetSocketPort
(
    term1 : INTEGER
)
    SocketPort := term1;
End Event
```

```
Event SetSocketAddress
(
    term1 : STRING
)
    SocketAddress := term1;
End Event
```

```
Event SetSocketType
(
    term1 : STRING
)
    SocketType := term1;
End Event
```

```
Event ConnectSocket
    SocketConnect := TRUE;
End Event
```

```
Event DisconnectSocket
    SocketConnect := FALSE;
End Event
```

```
Event SetSocketData
(
    term1 : STRING,
```

```

term2 : STRING,
term3 : STRING,
term4 : STRING,
term5 : STRING,
term6 : STRING,
term7 : STRING,
term8 : STRING
)
IF term1 == NULLSTRING THEN
    SocketNUM := 7;
ELSE
    SocketNUM := 8;
END IF
IF term2 == NULLSTRING THEN
    SocketNUM := 6;
END IF
IF term3 == NULLSTRING THEN
    SocketNUM := 5;
END IF
IF term4 == NULLSTRING THEN
    SocketNUM := 4;
END IF
IF term5 == NULLSTRING THEN
    SocketNUM := 3;
END IF
IF term6 == NULLSTRING THEN
    SocketNUM := 2;
END IF
IF term7 == NULLSTRING THEN
    SocketNUM := 1;
END IF
IF term8 == NULLSTRING THEN
    SocketNUM := 0;
END IF
term1 := "";
term2 := "";
term3 := "";

```

```

    term4 := "";
    term5 := "";
    term6 := "";
    term7 := "";
    term8 := "";

    SocketData := term1;
End Event

Event SetSocketCondition
(
    term1 : STRING
)
    SocketCondition := term1;
End Event

Event SetTriggerCount
(
    term1 : INTEGER
)
    TriggerCount := term1;
End Event

Event SetTriggerSource
(
    term1 : STRING
)
    TriggerSource := term1;
End Event

Event SetTriggerLink
(
    term1 : STRING
)
    TriggerLink := term1;
End Event

```

```

Event SetTriggerTime
(
    term1 : INTEGER
)
    TriggerTime := term1;
End Event

Event Initiate
    ReturnCount := TriggerCount;
End Event

Event TriggerDelay
(
    term1 : NUMERIC
)
    WORKING :=TRUE;
End Event

Event SetInitiate
(
    term1 : STRING
)
    IF term1 == NULLSTRING THEN
        term1 := "ON";
    END IF
    IF term1 == "OFF" THEN
        CONTINUOUS;
    ELSE
        CONTINUOUS SystCommSockFeed;
    END IF
End Event

Event SystCommSockFeedStart
    IF SocketConnect == TRUE THEN
        CONTINUOUS SystCommSockFeed;
    END IF
End Event

```

```

Event SystCommSockFeedStop
    CONTINUOUS;
End Event

Event SystCommSockFeed
    Var
        SEED1 : NUMERIC;
        SEED2 : NUMERIC;
        SEED3 : NUMERIC;
        SEED4 : NUMERIC;
        SEED5 : NUMERIC;
        SEED6 : NUMERIC;
        SEED7 : NUMERIC;
        SEED8 : NUMERIC;
        PARM1 : NUMERIC;
        PARM2 : NUMERIC;
    End Var
    SEED1 := RANDOM / RANDOM;
    SEED2 := RANDOM / RANDOM;
    SEED3 := RANDOM / RANDOM;
    SEED4 := RANDOM / RANDOM;
    SEED5 := RANDOM / RANDOM;
    SEED6 := RANDOM / RANDOM;
    SEED7 := RANDOM / RANDOM;
    SEED8 := RANDOM / RANDOM;
    PARM1 := SEED1 +19;
    PARM2 := SEED2 +82;

    IF SocketNUM == 1 THEN
        RESPOND PARM1,PARM2,SEED1,SEED2,SEED3,SEED4,SEED5,SEED6,SEED7,SEED8;
    END IF
    IF SocketNUM == 2 THEN
        RESPOND PARM1,PARM2,SEED1,SEED2,SEED3,SEED4,SEED5,SEED6,SEED7,SEED8,
        PARM1,PARM2,SEED1,SEED2,SEED3,SEED4,SEED5,SEED6,SEED7,SEED8;
    END IF
    IF SocketNUM == 3 THEN

```



```
PARM1,PARM2,SEED1,SEED2,SEED3,SEED4,SEED5,SEED6,SEED7,SEED8,  
PARM1,PARM2,SEED1,SEED2,SEED3,SEED4,SEED5,SEED6,SEED7,SEED8,  
PARM1,PARM2,SEED1,SEED2,SEED3,SEED4,SEED5,SEED6,SEED7,SEED8,  
PARM1,PARM2,SEED1,SEED2,SEED3,SEED4,SEED5,SEED6,SEED7,SEED8,  
PARM1,PARM2,SEED1,SEED2,SEED3,SEED4,SEED5,SEED6,SEED7,SEED8,  
PARM1,PARM2,SEED1,SEED2,SEED3,SEED4,SEED5,SEED6,SEED7,SEED8;
```

```
END IF
```

```
End Event
```

```
Event SendRouteList
```

```
    RESPOND "BAG,DIL,PRE,POST,MID,CEFF,NONE,ZERO,SPAN,VER,MAN";
```

```
End Event
```

```
Event Reset
```

```
    WORKING := TRUE;
```

```
End Event
```

```
Event SetErrList
```

```
(
```

```
    term1 : STRING
```

```
)
```

```
    ErrEnableList := term1;
```

```
End Event
```

```
Event SetQEnableList
```

```
(
```

```
    term1 : INTEGER
```

```
)
```

```
    StatQEnableList := term1;
```

```
End Event
```

```
Event SetOEnableList
```

```
(
```

```
    term1 : INTEGER
```

```
)
```

```
    StatOEnableList := term1;
```

```
End Event
```

```
Event GetEnableList
    RESPOND StatOEnableList;
End Event
```

```
Event CALCALC
(
    term1 : STRING
)
WORKING := TRUE;
```

```
End Event
```

```
Event GetAuto
(
    term1 : STRING
)
WORKING := TRUE;
End Event
```

```
Event CALCURVTYP
(
    term1 : STRING,
    term2 : NUMERIC,
    term3 : NUMERIC,
    term4 : NUMERIC,
    term5 : NUMERIC,
    term6 : NUMERIC
)
WORKING := TRUE;
```

```
End Event
```

```
Event CALCVERTYP
(
    term1 : STRING
```

```
)  
    WORKING := TRUE;  
End Event
```

```
Event CALCURVZFOR  
(  
    term1 : BOOLEAN  
)  
    WORKING := TRUE;
```

```
End Event
```

```
Event SetSamplePoint  
(  
    term1 : STRING  
)  
    SamplePoint := term1;  
End Event
```

```
Event DeleteTable  
(  
    term1 : STRING  
)  
    WORKING := TRUE;
```

```
End Event
```

```
Event SelectTable  
(  
    term1 : STRING  
)  
    SelTable := term1;  
End Event
```

```
Event DefineTable  
(  
    term1 : STRING,  
    term2 : INTEGER  
)
```

```

TableStruct := term1;
TableMax := term2;
End Event

Event SetTable
(
    term1 : STRING,
    term2 : STRING,
    term3 : STRING,
    term4 : STRING,
    term5 : STRING,
    term6 : STRING,
    term7 : STRING,
    term8 : STRING,
    term9 : STRING,
    term10 : STRING
)
IF term1 == NULLSTRING THEN
    NUMENTRIES := 9;
ELSE
    NUMENTRIES := 10;
END IF
IF term2 == NULLSTRING THEN
    NUMENTRIES := 8;
END IF
IF term3 == NULLSTRING THEN
    NUMENTRIES := 7;
END IF
IF term4 == NULLSTRING THEN
    NUMENTRIES := 6;
END IF
IF term5 == NULLSTRING THEN
    NUMENTRIES := 5;
END IF
IF term6 == NULLSTRING THEN
    NUMENTRIES := 4;
END IF

```

```

IF term7 == NULLSTRING THEN
    NUMENTRIES := 3;
END IF
IF term8 == NULLSTRING THEN
    NUMENTRIES := 2;
END IF
IF term9 == NULLSTRING THEN
    NUMENTRIES := 1;
END IF
term1 := "";
term2 := "";
term3 := "";
term4 := "";
term5 := "";
term6 := "";
term7 := "";
term8 := "";
term9 := "";
term10 := "";

```

End Event

```

Event GetEntries
    RESPOND NUMENTRIES;
End Event

```

```

Event EnterTable
(
    term1 : STRING,
    term2 : NUMERIC,
    term3 : NUMERIC,
    term4 : NUMERIC,
    term5 : NUMERIC,
    term6 : NUMERIC,
    term7 : NUMERIC,
    term8 : NUMERIC,
    term9 : NUMERIC,

```

```

        term10 : NUMERIC,
        term11 : NUMERIC,
        term12 : NUMERIC,
        term13 : NUMERIC,
        term14 : NUMERIC,
        term15 : NUMERIC,
        term16 : NUMERIC
    )
    WORKING := TRUE;

```

End Event

```

Event ReturnTable
(
    term1 : STRING
)

```

```

Var
    SEED1 : NUMERIC;
    SEED2 : NUMERIC;
    SEED3 : NUMERIC;
    SEED4 : NUMERIC;
    SEED5 : NUMERIC;
    SEED6 : NUMERIC;
    SEED7 : NUMERIC;
    SEED8 : NUMERIC;

End Var
SEED1 := RANDOM / RANDOM;
SEED2 := RANDOM / RANDOM;
SEED3 := RANDOM / RANDOM;
SEED4 := RANDOM / RANDOM;
SEED5 := RANDOM / RANDOM;
SEED6 := RANDOM / RANDOM;
SEED7 := RANDOM / RANDOM;
SEED8 := RANDOM / RANDOM;

```

```

RESPOND SEED1,SEED2,SEED3,SEED4,SEED5,SEED6,SEED7,SEED8,
        SEED2,SEED4,SEED6,SEED8,SEED1,SEED3,SEED5;

```

End Event

```
Event SetAvgCount
(
    term1 : INTEGER
)
AvgCount := term1;
End Event
```

```
Event SetTCON
(
    term1 : STRING
)
AvgTCON := term1;
End Event
```

```
Event SetAvgType
(
    term1 : STRING
)
AvgType := term1;
End Event
```

```
Event SetAvgState
(
    term1 : STRING
)
AvgState := term1;
End Event
```

```
Event SetCSET
(
    term1 : NUMERIC,
    term2 : NUMERIC
)
CSETEX := term1;
CSETY := term2;
End Event
```

```
Event GetCSET
RESPOND CSETEX, CSETY;
End Event
```

```

Event SetLSET
(
    term1 : STRING,
    term2 : NUMERIC,
    term3 : NUMERIC,
    term4 : STRING,
    term5 : NUMERIC,
    term6 : NUMERIC,
    term7 : STRING,
    term8 : NUMERIC,
    term9 : NUMERIC,
    term10 : STRING,
    term11 : NUMERIC,
    term12 : NUMERIC
)
LSETX := term2;
LSETY := term3;

```

End Event

Event GetLSET

```
RESPOND LSETX, LSETY;
```

End Event

Event SetLower

```

(
    term1 : NUMERIC,
    term2 : NUMERIC,
    term3 : NUMERIC,
    term4 : NUMERIC,
    term5 : NUMERIC,
    term6 : NUMERIC,
    term7 : NUMERIC,
    term8 : NUMERIC,
    term9 : NUMERIC,
    term10 : NUMERIC
)
CONCLower := term10;

```

End Event

Event GetLower

 RESPOND CONCLower,CONCLower,CONCLower;

End Event

Event SetUpper

(
 term1 : NUMERIC,
 term2 : NUMERIC,
 term3 : NUMERIC,
 term4 : NUMERIC,
 term5 : NUMERIC,
 term6 : NUMERIC,
 term7 : NUMERIC,
 term8 : NUMERIC,
 term9 : NUMERIC,
 term10 : NUMERIC

)

 CONCUpper := term1;

End Event

Event SetRange

(
 term1 : NUMERIC,
 term2 : NUMERIC,
 term3 : NUMERIC,
 term4 : NUMERIC,
 term5 : NUMERIC,
 term6 : NUMERIC,
 term7 : NUMERIC,
 term8 : NUMERIC,
 term9 : NUMERIC,
 term10 : NUMERIC

)

 CONCRange := term1;

End Event

```

Event ReturnRange
Var
    Seed : INTEGER;
    Num : INTEGER;
End Var

Seed := RANDOM;

CONC1 := SIN(Seed);
CONC2 := SIN(Seed + 45);
CONC3 := SIN(Seed + 90);
CONC4 := SIN(Seed + 135);

IF SelectGroup == 6 THEN
    Num := InstCount6;
END IF

IF SelectGroup == 5 THEN
    Num := InstCount5;
END IF

IF SelectGroup == 4 THEN
    Num := InstCount4;
END IF

IF SelectGroup == 3 THEN
    Num := InstCount3;
END IF

IF SelectGroup == 2 THEN
    Num := InstCount2;
END IF

IF SelectGroup == 1 THEN
    Num := InstCount1;
END IF

IF Num == 0 THEN
    RESPOND CONC1;
END IF

IF Num == 1 THEN
    RESPOND CONC1;
END IF

```

```

IF Num == 2 THEN
    RESPOND CONC1,SDEV1;
END IF

IF Num == 3 THEN
    RESPOND CONC1,SDEV1,CONC2;
END IF

IF Num == 4 THEN
    RESPOND CONC1,SDEV1,CONC2,SDEV2;
END IF

IF Num == 5 THEN
    RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3;
END IF

IF Num == 6 THEN
    RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3,SDEV3;
END IF

IF Num == 7 THEN
    RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3,SDEV3,CONC4;
END IF

IF Num == 8 THEN
    RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3,SDEV3,CONC4,SDEV4;
END IF

IF Num == 9 THEN
    RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3,SDEV3,CONC4,SDEV4,CONC1;
END IF

IF Num == 10 THEN
    RESPOND CONC1,SDEV1,CONC2,SDEV2,CONC3,SDEV3,CONC4,SDEV4,CONC2,SDEV1;
END IF

```

End Event

```

Event SetConcState
(
    term1 : BOOLEAN
)
    CONCState := term1;

```

End Event

```

Event SetConcTAL
(
    term1 : NUMERIC

```

```

)
CONCTAL := term1;
End Event

Event SetCorrState
(
    term1 : STRING
)
IF term1 == "ON" THEN
    CORRState := TRUE;
ELSE
    CORRState := FALSE;
END IF
End Event

Event GetCorrState
    RESPOND CORRState;
End Event

Event SetSPODTOL
(
    term1 : NUMERIC,
    term2 : NUMERIC,
    term3 : NUMERIC,
    term4 : NUMERIC,
    term5 : NUMERIC,
    term6 : NUMERIC,
    term7 : NUMERIC
)
CORRSPODTOL := term1;
End Event

Event SetZeroDTOL
(
    term1 : NUMERIC,
    term2 : NUMERIC,
    term3 : NUMERIC,
    term4 : NUMERIC,
    term5 : NUMERIC,
    term6 : NUMERIC,

```

```

term7 : NUMERIC
)
CORRZeroDTOL := term1;
End Event

Event SetSPOAcq
CORRSPOAcq := TRUE;
End Event

Event SetZeroAcq
CORRZeroAcq := TRUE;
End Event

Event SetCorrAuto
CORRAuto := TRUE;
End Event

Event SetCorrCalc
CORRCalc := TRUE;
End Event

Event SetConcurrent
(
    term1 : BOOLEAN
)
SenseConc := term1;
End Event

Event SetFunction
(
    term1 : STRING,
    term2 : STRING,
    term3 : STRING,
    term4 : STRING,
    term5 : STRING,
    term6 : STRING,
    term7 : STRING,
    term8 : STRING,

```

```

term9 : STRING,
term10 : STRING
)
IF term1 == NULLSTRING THEN
    NUMPARAM := 9;
ELSE
    NUMPARAM := 10;
END IF
IF term2 == NULLSTRING THEN
    NUMPARAM := 8;
END IF
IF term3 == NULLSTRING THEN
    NUMPARAM := 7;
END IF
IF term4 == NULLSTRING THEN
    NUMPARAM := 6;
END IF
IF term5 == NULLSTRING THEN
    NUMPARAM := 5;
END IF
IF term6 == NULLSTRING THEN
    NUMPARAM := 4;
END IF
IF term7 == NULLSTRING THEN
    NUMPARAM := 3;
END IF
IF term8 == NULLSTRING THEN
    NUMPARAM := 2;
END IF
IF term9 == NULLSTRING THEN
    NUMPARAM := 1;
END IF
IF term10 == NULLSTRING THEN
    NUMPARAM := 1;
END IF

IF SelectGroup == 6 THEN

```

```

    GroupNum6 := NUMPARAM;
END IF
IF SelectGroup == 5 THEN
    GroupNum5 := NUMPARAM;
END IF
IF SelectGroup == 4 THEN
    GroupNum4 := NUMPARAM;
END IF
IF SelectGroup == 3 THEN
    GroupNum3 := NUMPARAM;
END IF
IF SelectGroup == 2 THEN
    GroupNum2 := NUMPARAM;
END IF
IF SelectGroup == 1 THEN
    GroupNum1 := NUMPARAM;
END IF

term1 := "";
term2 := "";
term3 := "";
term4 := "";
term5 := "";
term6 := "";
term7 := "";
term8 := "";
term9 := "";
term10 := "";

SenseFunc := NUMPARAM;
End Event

```

```

Event SetSTABNTOL
(
    term1 : NUMERIC,
    term2 : NUMERIC,
    term3 : NUMERIC,
    term4 : NUMERIC,

```

```

        term5 : NUMERIC,
        term6 : NUMERIC,
        term7 : NUMERIC,
        term8 : NUMERIC      )
STABNTOL := term8;

End Event

Event GetSTABNTOL
    RESPOND STABNTOL,STABNTOL,STABNTOL;
End Event

Event SetSTABTime
(      term1 : NUMERIC,
       term2 : NUMERIC,
       term3 : NUMERIC,
       term4 : NUMERIC,
       term5 : NUMERIC,
       term6 : NUMERIC,
       term7 : NUMERIC,
       term8 : NUMERIC
)
STABTime := term8;

End Event

Event GetSTABTime
    RESPOND STABTime,STABNTOL,STABTime;
End Event

Event SetSTABState
(      term1 : BOOLEAN
)
STABState := term1;

End Event

End Instrument

```

Appendix II

EBench Validation Test Files

EBench Test Step file. This file defines the SCPI commands that are to be issued to the EBench model and the validation to be performed on the data returned.

```
#% E-Bench test step file %
#% Header record %
HEADER
%
TEST ID: TEST-001
TEST DESCRIPTION: EBench Command Test
%
#% Configuration record %
CONFIG
%
PORTNO: 701
IPADDRESS: 127.0.0.1
SPECIFIER:EBENCH
%
#% Test step records%
#% COMMAND~EXPECTED~TOLERANCE~DELAY~ONPASS~ONFAIL~ONERROR%
05% *RST %
10%:SYST:LOCK:REQ?~1~0 %
20%:CONT:EBEN:CLE:DUR 60 %
30%:CONT:EBEN:CLE %
40%:INST:DEF:GRO "Bag_Analyzers" "CONC1" "CONC2" "CONC3" "CONC4" %
50%:INST:SEL "Bag_Analyzers" %
60%.SENS:FUNC:CONC ON %
70%:SENS:FUNC "CONC1" "CONC1:SDEV" "CONC2" "CONC2:SDEV" "CONC3" "CONC3:SDEV" "CONC4" %
"CONC4:SDEV" %
80%:TRIG:SEQ1:SOUR TIMER %
90%:TRIG:SEQ1:TIM 0.1%
91%:TRIG:SEQ2:LINK "TRIGger:SEQence1" %
92%:TRIG:SEQ1:ECO 2%
93%:SYST:COMM:SOCK1:ADDR "123.456.789.1" %
94%:SYST:COMM:SOCK1:PORT 702 %
95%:SYST:COMM:SOCK1:TYPE TCP %
96%:SYST:COMM:SOCK1:FEED:OCON "TRIG:SEQ1" %
97%:SYST:COMM:SOCK1:FEED:SCON "TRIG:SEQ2" %
98%:SYST:COMM:SOCK1:CONN %
```

100%:SENS:STAB:STAT ON %
101%:SENS:STAB:TIME1 60 50 45 55 %
102%:SENS:STAB:TIME2 15 15 15 15 %
103%:SENS:STAB:TIME3 5 5 5 5 %
104%:SENS:STAB:TIME4 120 120 120 120 %
105%:SENS:STAB:NTOL 4 4 4 4 %
106%:SENS:AVER1:COUN 10 %
107%:SENS:AVER1:TCON MOV %
108%:SENS:AVER1:TYPE SCALAR %
109%:SENS:AVER1:STAT ON %
110%:SENS:CONC:RANG:AUTO ON %
120% WAIT ~Select Bag on Sampling System %
130%:ROUT:SAMP BAG%
133%:SYST:COMM:SOCK1:FEED "TlMer:COUnT" "CONC1:TAL" "TEMP1" %
135%DATACHANNEL~22.0,83.0,5.0,5.0,5.0,5.0,5.0,5.0,5.0,5.0,5.0,22.0,83.0~
20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0%
140%:INIT:CONT:SEQ1 ON %
150%:STAT:OPER:BIT9?~0~0 %
160%:STAT:QUES:BIT9?~0~0 %
170%:SENS:DATA?~0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0~
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0 %
180%:SENS:CONC:RANG:AUTO OFF%
190%:SENS:CONC:RANG:FIX?~0.0,0.0,0.0,0.0~1.0,1.0,1.0,1.0 %
200%:ROUT:SAMP NONE%
210%:SENS:CORR:ZERO:DTOL 5 %
220%:SENS:CORR:SPO:DTOL 5 %
230%:SENS:CORR:STAT OFF%
240%:SENS:CORR:ZERO:ACQ%
250%:STAT:OPER:INST:ISUM1:BIT10:CAL? ~0~0~ ~250~250 %
260%:STAT:OPER:INST:ISUM2:BIT10:CAL? ~0~0~ ~260~260 %
270%:STAT:OPER:INST:ISUM3:BIT10:CAL? ~0~0~ ~270~270 %
280%:STAT:OPER:INST:ISUM4:BIT10:CAL? ~0~0~ ~280~280 %
290%:STAT:QUES:INST:ISUM1:BIT10:CAL? ~0~0 %
300%:STAT:QUES:INST:ISUM2:BIT10:CAL? ~0~0 %
310%:STAT:QUES:INST:ISUM3:BIT10:CAL? ~0~0 %
320%:STAT:QUES:INST:ISUM4:BIT10:CAL? ~0~0 %
330%:SENS:DATA? ~0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0~

1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0 %
340%:MEM:DATA? ZDRift ~0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0~
15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0 %
350%:SENS:CORR:SPO:ACQ %
360%:STAT:OPER:INST:ISUM1:BIT10:CAL? ~0~0 %
370%:STAT:OPER:INST:ISUM2:BIT10:CAL? ~0~0 %
380%:STAT:OPER:INST:ISUM3:BIT10:CAL? ~0~0 %
390%:STAT:OPER:INST:ISUM4:BIT10:CAL? ~0~0 %
400%:STAT:QUES:INST:ISUM1:BIT10:CAL? ~0~0 %
410%:STAT:QUES:INST:ISUM2:BIT10:CAL? ~0~0 %
420%:STAT:QUES:INST:ISUM3:BIT10:CAL? ~0~0 %
430%:STAT:QUES:INST:ISUM4:BIT10:CAL? ~0~0 %
440%:SENS:DATA? ~0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0~
1.0,1.0,1.0,1.0,1.0,1.0,1.0 %
450%:MEM:DATA? SDRift~0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0~
15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0,15.0 %
460%:SENS:CORR:CALC %
640%:SENS:CONC:CSET? ~0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0~
1.0,1.0,1.0,1.0,1.0,1.0,1.0 %
645%:INIT:CONT OFF %
660%:ROUT:SAMP BAG %
670%:INIT:IMM:SEQ1 %
685%:INIT:CONT OFF %
686%:CAL:LIN:ACC %
687%:CAL:LIN:ACQ %
688%:CAL:LIN:AUTO ONCE %
689%:CAL:LIN:CALC POLY1 %
690%:CAL:LIN:CURV SRAT2 0.25 0.75 1.00 %
691%:CAL:LIN:CURV:ZFOR ON %
692%:CAL:LIN:VER:ACQ %
693%:CAL:LIN:VER:TOL 5.0 %
694%:CAL:LIN:VER:TYPE CCUR %
695%:CONT:EBEN:CLE:INIT %
701%:DIAG:HUP:ACQ %
702%:DIAG:HUP:CALC %
703%:DIAG:LEAK:ACQ %
704%:DIAG:LEAK:CALC %

705%:DIAG:NEFF:ACQ %
706%:DIAG:NEFF:CALC %
710%:INST:DEF:NAME CONC 1%
711%:INST:DEL %
712%:INST:DEL:ALL %
713%:INST:DEL:NAME CONC1%
720%:MEM:CLE:TABLE ZDrift%
721%:MEM:DEL:ALL %
722%:MEM:DATA ZDrift 0.25 0.26 0.27 0.28 %
723%:MEM:TABL:BNUM ZDrift 1 %
724%:MEM:TABL:BNUM:POIN? ~1.0~1.0 %
725%:MEM:TABL:CCUR 2.0 2.0 2.0 %
726%:MEM:TABL:CCUR:POIN? ~3.0~1.0 %
727%:MEM:TABL:CONC 3.0 3.0 3.0 3.0 %
728%:MEM:TABL:CONC:POIN? ~4.0~1.0 %
729%:MEM:TABL:CPO 5.0 5.0 5.0 5.0 %
740%:MEM:TABL:CPO:POIN? ~4.0~1.0 %
741%:MEM:TABL:DEFine FREQ 10 %
742%:MEM:TABL:DFAC 1.0 2.0 3.0 4.0 5.0 %
743%:MEM:TABL:DFAC:POIN? ~5.0~1.0 %
744%:MEM:TABL:DLAS 10.0 10.0 10.0 %
745%:MEM:TABL:DLAS:POIN? ~3.0~1.0 %
750%:MEM:TABL:DLIN 1.0 1.0 2.0 2.0 %
751%:MEM:TABL:DLIN:POIN? ~4.0~1.0 %
752%:MEM:TABL:EXP 0.0 0.0 0.0 0.0 0.0 %
753%:MEM:TABL:EXP:POIN? ~4.0~1.0 %
754%:MEM:TABL:LAB Bottle1 Bottle2 %
755%:MEM:TABL:LAB:POIN? ~2.0~1.0 %
760%:MEM:TABL:LOG Name Bottle %
761%:MEM:TABL:LOG:POIN? ~2.0~1.0 %
762%:MEM:TABL:NCUR 2.0 3.0 2.0 3.0 %
763%:MEM:TABL:NCUR:POIN? ~4.0~1.0 %
764%:MEM:TABL:RAW 0.0 0.5 0.25 %
765%:MEM:TABL:RAW:POIN? ~3.0~1.0 %
771%:MEM:TABL:TOL 5.0 5.0 5.0 5.0 %
772%:MEM:TABL:TOL:POIN? ~4.0~1.0 %
773%:MEM:TABL:WFAC 1.0 2.0 3.0 4.0 5.0 %

774%:MEM:TABL:WFAC:POIN? ~5.0~1.0 %
780%:SENS:CONC:CSET 3.0 3.0 %
781%:SENS:CONC:LOW 0.5 0.25 1.0 %
782%:SENS:CONC:LOW? ~0.0,0.0,0.0~1.0,1.0,1.0 %
783%:SENS:CONC:LSET 0.1 0.2 0.3 0.4 %
784%:SENS:CONC:LSET? ~0.0,0.0,0.0,0.0~1.0,1.0,1.0,1.0 %
785%:SENS:CONC:RANG:AUTO ON %
786%:SENS:CONC:RANG:AUTO:LOW 0.0 0.0 0.0 0.0 %
787%:SENS:CONC:RANG:AUTO:LOW? ~0.0,0.0,0.0,0.0~1.0,1.0,1.0,1.0 %
790%:SENS:CONC:RANG:AUTO:UPP 10.0 10.0 10.0 10.0 10.0 %
791%:SENS:CONC:RANG:AUTO:UPP? ~0.0,0.0,0.0,0.0~1.0,1.0,1.0,1.0 %
792%:SENS:CONC:RANG:FIX 10.0 10.0 10.0 10.0 10.0 %
793%:SENS:CONC:TAL 2.25 %
794%:SENS:CONC:UPP 10.0 10.0 10.0 10.0 10.0 %
795%:SENS:CORR:AUTO %
796%:SENS:CORR:CALC %
797%:SENS:CORR:STAT ON %
810%:SYST:COMM:SOCK:DISC %
811%:SYST:COMM:SOCK:FEED:OCON 0.5 %
812%:SYST:COMM:SOCK:FEED:SCON 1 %
%hh:mm:ss.sss %
813% :SYST:TIME? ~NA%
% yyyy/mm/dd %
814% :SYST:DATE? ~NA%
820%:SYST:LOCK:OWN? ~ AIGER %
821%:SYST:TIME:TIM ON %
822%:SYST:TIME:TIM:COUN 1.0 %
823%:TRIG %
900%:STAT:OPER? ~0~0 %
901%:STAT:OPER:BIT9? ~0~0 %
902%:STAT:OPER:BIT9:COND? ~0~0 %
903%:STAT:OPER:BIT9:ENAB 32767 %
904%:STAT:OPER:BIT9:NTR 32767 %
905%:STAT:OPER:BIT9:PTR 32767 %
906%:STAT:OPER:COND? ~0~0 %
907%:STAT:OPER:EVEN? ~0~0 %
908%:STAT:OPER:INST? ~0~0 %

909%:STAT:OPER:INST:COND? ~0~0 %
910%:STAT:OPER:INST:ENAB 32767 %
911%*STB? ~0~0 %
912%*ESR? ~0~0 %
920%:STAT:OPER:INST:ISUM1? ~0~0 %
921%:STAT:OPER:INST:ISUM1:BIT10? ~0~0 %
922%:STAT:OPER:INST:ISUM1:BIT10:CAL? ~0~0 %
923%:STAT:OPER:INST:ISUM1:BIT10:CAL:COND? ~0~0 %
924%:STAT:OPER:INST:ISUM1:BIT10:CAL:ENAB 32767 %
925%:STAT:OPER:INST:ISUM1:BIT10:CAL:NTR 32767 %
926%:STAT:OPER:INST:ISUM1:BIT10:CAL:PTR 32767 %
927%:STAT:OPER:INST:ISUM1:BIT10:COND? ~0~0 %
928%:STAT:OPER:INST:ISUM1:BIT10:ENAB 32767 %
929%:STAT:OPER:INST:ISUM1:BIT10:NTR 32767 %
930%:STAT:OPER:INST:ISUM1:BIT10:PTR 32767 %
931%:STAT:OPER:INST:ISUM1:COND? ~0~0 %
932%:STAT:OPER:INST:ISUM1:ENAB 32767 %
933%:STAT:OPER:INST:ISUM1:NTR 32767 %
934%:STAT:OPER:INST:ISUM1:PTR 32767 %
935%:STAT:OPER:INST:NTR 32767 %
936%:STAT:OPER:INST:PTR 32767 %
1000%:STAT:QUES? ~0~0 %
1001%:STAT:QUES:BIT9? ~0~0 %
1002%:STAT:QUES:BIT9:COND? ~0~0 %
1003%:STAT:QUES:BIT9:ENAB 32767 %
1004%:STAT:QUES:BIT9:NTR 32767 %
1005%:STAT:QUES:BIT9:PTR 32767 %
1006%:STAT:QUES:COND? ~0~0 %
1010%:STAT:QUES:INST? ~0~0 %
1011%:STAT:QUES:INST:COND? ~0~0 %
1012%:STAT:QUES:INST:ENAB 32767 %
1013%:STAT:QUES:INST:ISUM1? ~0~0 %
1014%:STAT:QUES:INST:ISUM1:BIT10? ~0~0 %
1015%:STAT:QUES:INST:ISUM1:BIT10:CAL? ~0~0 %
1016%:STAT:QUES:INST:ISUM1:BIT10:CAL:COND? ~0~0 %
1017%:STAT:QUES:INST:ISUM1:BIT10:CAL:ENAB 32767 %
1018%:STAT:QUES:INST:ISUM1:BIT10:CAL:NTR 32767 %

1019%:STAT:QUES:INST:ISUM1:BIT10:CAL:PTR 32767 %
1020%:STAT:QUES:INST:ISUM1:BIT10:COND? ~0~0 %
1021%:STAT:QUES:INST:ISUM1:BIT10:ENAB 32767 %
1022%:STAT:QUES:INST:ISUM1:BIT10:NTR 32767 %
1023%:STAT:QUES:INST:ISUM1:BIT10:PTR 32767 %
1024%:STAT:QUES:INST:ISUM1:COND? ~0~0 %
1025%:STAT:QUES:INST:ISUM1:ENAB 32767 %
1026%:STAT:QUES:INST:ISUM1:NTR 32767 %
1027%:STAT:QUES:INST:ISUM1:PTR 32767 %
1028%:STAT:QUES:INST:NTR 32767 %
1029%:STAT:QUES:INST:PTR 32767 %
1060% *WAI %
1090%:SYST:LOCK:REL %

EBench Test Results Log file. This file describes the test steps performed and the results of test step validation.

ebench_cmd

Monday, February 12, 2001, 11:43:16

HEADER

%

TEST ID: TEST-001

TEST DESCRIPTION: : EBench Command Test

%

CONFIG

%

PORTNO: 701

IPADDRESS: 127.0.0.1

SPECIFIER:EBENCH

%

Command : *RST

Command : :SYST:LOCK:REQ?

Expected :1

Received :1

Test Step : 10 Passed

Command : :CONT:EBEN:CLE:DUR 60

Command : :CONT:EBEN:CLE

Command : :INST:DEF:GRO "Bag_Analyzers" "CONC1" "CONC2" "CONC3" "CONC4"

Command : :INST:SEL "Bag_Analyzers"

Command : :SENS:FUNC:CONC ON

Command : :SENS:FUNC "CONC1" "CONC1:SDEV" "CONC2" "CONC2:SDEV" "CONC3" "CONC3:SDEV" "CONC4" "CONC4:SDEV"

Command : :TRIG:SEQ1:SOUR TIMER

Command : :TRIG:SEQ1:TIM 0.1

Command : :TRIG:SEQ2:LINK "TRIGger:SEQence1"

Command : :TRIG:SEQ1:ECO 2

Command : :SYST:COMM:SOCK1:ADDR "123.456.789.1"

Command : :SYST:COMM:SOCK1:PORT 702

Command : :SYST:COMM:SOCK1:TYPE TCP

Command : :SYST:COMM:SOCK1:FEED:OCON "TRIG:SEQ1"

Command : :SYST:COMM:SOCK1:FEED:SCON "TRIG:SEQ2"

Command : :SYST:COMM:SOCK1:CONN
Command : :SENS:STAB:STAT ON
Command : :SENS:STAB:TIME1 60 50 45 55
Command : :SENS:STAB:TIME2 15 15 15 15
Command : :SENS:STAB:TIME3 5 5 5 5
Command : :SENS:STAB:TIME4 120 120 120 120
Command : :SENS:STAB:NTOL 4 4 4 4
Command : :SENS:AVER1:COUN 10
Command : :SENS:AVER1:TCON MOV
Command : :SENS:AVER1:TYPE SCALAR
Command : :SENS:AVER1:STAT ON
Command : :SENS:CONC:RANG:AUTO ON
Command : :ROUT:SAMP BAG
Command : :SYST:COMM:SOCK1:FEED "TMR:COUNT" "CONC1:TAL" "TEMP1"
Command : :INIT:CONT:SEQ1 ON
Command : :STAT:OPER:BIT9?
Expected :0
Received :0
Test Step : 150 Passed
Command : :STAT:QUES:BIT9?
Expected :0
Received :0
Test Step : 160 Passed
Command : :SENS:DATA?
Expected :0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Received :0.069756,-0.891007,-0.656059,-0.951057,-0.997564,-0.453990,-0.754710,0.309017
Test Step : 170 Passed
Command : :SENS:CONC:RANG:AUTO OFF
Command : :SENS:CONC:RANG:FIX?
Expected :0,0,0,0,0,0,0
Received :0.358368,-0.891007,0.913545,-0.951057
Test Step : 190 Passed
Command : :ROUT:SAMP NONE
Command : :SENS:CORR:ZERO:DTOL 5
Command : :SENS:CORR:SPO:DTOL 5
Command : :SENS:CORR:STAT OFF
Command : :SENS:CORR:ZERO:ACQ

Command : :STAT:OPER:INST:ISUM1:BIT10:CAL?

Expected :0

Received :0

Test Step : 250 Passed

Command : :STAT:OPER:INST:ISUM2:BIT10:CAL?

Expected :0

Received :0

Test Step : 260 Passed

Command : :STAT:OPER:INST:ISUM3:BIT10:CAL?

Expected :0

Received :0

Test Step : 270 Passed

Command : :STAT:OPER:INST:ISUM4:BIT10:CAL?

Expected :0

Received :0

Test Step : 280 Passed

Command : :STAT:QUES:INST:ISUM1:BIT10:CAL?

Expected :0

Received :0

Test Step : 290 Passed

Command : :STAT:QUES:INST:ISUM2:BIT10:CAL?

Expected :0

Received :0

Test Step : 300 Passed

Command : :STAT:QUES:INST:ISUM3:BIT10:CAL?

Expected :0

Received :0

Test Step : 310 Passed

Command : :STAT:QUES:INST:ISUM4:BIT10:CAL?

Expected :0

Received :0

Test Step : 320 Passed

Command : :SENS:DATA?

Expected :0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

Received :0.809017,0.681998,0.987688,-0.034899,0.587785,-0.731354,-0.156434,-0.999391

Test Step : 330 Passed

Command : :MEM:DATA? ZDRift

Expected :0,0

Received

:0.863667,3.182998,2.393942,1.590819,0.186630,1.829160,1.082601,2.684046,3.182998,1.590819,1.829160,2.684046,0.863667,2.39
3942,0.186630

Test Step : 340 Passed

Command : :SENS:CORR:SPO:ACQ

Command : :STAT:OPER:INST:ISUM1:BIT10:CAL?

Expected :0

Received :0

Test Step : 360 Passed

Command : :STAT:OPER:INST:ISUM2:BIT10:CAL?

Expected :0

Received :0

Test Step : 370 Passed

Command : :STAT:OPER:INST:ISUM3:BIT10:CAL?

Expected :0

Received :0

Test Step : 380 Passed

Command : :STAT:OPER:INST:ISUM4:BIT10:CAL?

Expected :0

Received :0

Test Step : 390 Passed

Command : :STAT:QUES:INST:ISUM1:BIT10:CAL?

Expected :0

Received :0

Test Step : 400 Passed

Command : :STAT:QUES:INST:ISUM2:BIT10:CAL?

Expected :0

Received :0

Test Step : 410 Passed

Command : :STAT:QUES:INST:ISUM3:BIT10:CAL?

Expected :0

Received :0

Test Step : 420 Passed

Command : :DIAG:LEAK:CALC

Command : :DIAG:NEFF:ACQ

Command : :DIAG:NEFF:CALC

Command : :INST:DEF:NAME CONC 1

Command : :INST:DEL

Command : :INST:DEL:ALL

Command : :INST:DEL:NAME CONC1

Command : :MEM:CLE:TABLE ZDrift

Command : :MEM:DEL:ALL

Command : :MEM:DATA ZDrift 0.25 0.26 0.27 0.28

Command : :MEM:TABL:BNUM ZDrift 1

Command : :MEM:TABL:BNUM:POIN?

Expected :1.0

Received :2

Test Step : 724 Passed

Command : :MEM:TABL:CCUR 2.0 2.0 2.0

Command : :MEM:TABL:CCUR:POIN?

Expected :3.0

Received :3

Test Step : 726 Passed

Command : :MEM:TABL:CONC 3.0 3.0 3.0 3.0

Command : :MEM:TABL:CONC:POIN?

Expected :4.0

Received :4

Test Step : 728 Passed

Command : :MEM:TABL:CPO 5.0 5.0 5.0 5.0

Command : :MEM:TABL:CPO:POIN?

Expected :4.0

Received :4

Test Step : 740 Passed

Command : :MEM:TABL:DEFine FREQ 10

Command : :MEM:TABL:DFAC 1.0 2.0 3.0 4.0 5.0

Command : :MEM:TABL:DFAC:POIN?

Expected :5.0

Received :5

Test Step : 743 Passed

Command : :MEM:TABLE:DLAS 10.0 10.0 10.0

Command : :MEM:TABLE:DLAS:POIN?

Expected :3.0

Received :3

Test Step : 745 Passed

Command : :MEM:TABLE:DLIN 1.0 1.0 2.0 2.0

Command : :MEM:TABLE:DLIN:POIN?

Expected :4.0

Received :4

Test Step : 751 Passed

Command : :MEM:TABLE:EXP 0.0 0.0 0.0 0.0

Command : :MEM:TABLE:EXP:POIN?

Expected :4.0

Received :4

Test Step : 753 Passed

Command : :MEM:TABLE:LAB Bottle1 Bottle2

Command : :MEM:TABLE:LAB:POIN?

Expected :2.0

Received :2

Test Step : 755 Passed

Command : :MEM:TABLE:LOG Name Bottle

Command : :MEM:TABLE:LOG:POIN?

Expected :2.0

Received :2

Test Step : 761 Passed

Command : :MEM:TABLE:NCUR 2.0 3.0 2.0 3.0

Command : :MEM:TABLE:NCUR:POIN?

Expected :4.0

Received :4

Test Step : 763 Passed

Command : :MEM:TABLE:RAW 0.0 0.5 0.25

Command : :MEM:TABLE:RAW:POIN?

Expected :3.0

Received :3

Test Step : 765 Passed

Command : :MEM:TABL:TOL 5.0 5.0 5.0 5.0

Command : :MEM:TABL:TOL:POIN?

Expected :4.0

Received :4

Test Step : 772 Passed

Command : :MEM:TABL:WFAC 1.0 2.0 3.0 4.0 5.0

Command : :MEM:TABL:WFAC:POIN?

Expected :5.0

Received :5

Test Step : 774 Passed

Command : :SENS:CONC:CSET 3.0 3.0

Command : :SENS:CONC:LOW 0.5 0.25 1.0

Command : :SENS:CONC:LOW?

Expected :0.0,0.0,0.0

Received :1.000000,1.000000,1.000000

Test Step : 782 Passed

Command : :SENS:CONC:LSET 0.1 0.2 0.3 0.4

Command : :SENS:CONC:LSET?

Expected :0.0,0.0,0.0,0.0

Received :-0.987688,0.891007,-0.809017,0.951057,-0.156434,0.453991,0.587785,-0.309017

Test Step : 784 Passed

Command : :SENS:CONC:RANG:AUTO ON

Command : :SENS:CONC:RANG:AUTO:LOW 0.0 0.0 0.0 0.0

Command : :SENS:CONC:RANG:AUTO:LOW?

Expected :0.0,0.0,0.0,0.0

Received :0.939693,0.891007,0.422618,0.951057

Test Step : 787 Passed

Command : :SENS:CONC:RANG:AUTO:UPP 10.0 10.0 10.0 10.0 10.0

Command : :SENS:CONC:RANG:AUTO:UPP?

Expected :0.0,0.0,0.0,0.0

Received :-0.190809,0.891007,-0.829038,0.951057

Test Step : 791 Passed

Command : :SENS:CONC:RANG:FIX 10.0 10.0 10.0 10.0 10.0

Command : :SENS:CONC:TAL 2.25
Command : :SENS:CONC:UPP 10.0 10.0 10.0 10.0 10.0
Command : :SENS:CORR:AUTO

Command : :SENS:CORR:CALC
Command : :SENS:CORR:STAT ON
Command : :SYST:COMM:SOCK:DISC
Command : :SYST:COMM:SOCK:FEED:OCON 0.5
Command : :SYST:COMM:SOCK:FEED:SCON 1
Command : :SYST:TIME?
Expected :NA
Received :11:50:59.00
Test Step : 813 : :SYST:TIME? : Failed
Command : :SYST:DATE?
Expected :NA
Received :2001/02/12
Test Step : 814 : :SYST:DATE? : Failed
Command : :SYST:LOCK:OWN?
Expected :AIGER
Received :AIGER
Test Step : 820 Passed
Command : :SYST:TIME:TIM ON
Command : :SYST:TIME:TIM:COUN 1.0
Command : :TRIG
Command : :STAT:OPER?
Expected :0
Received :0
Test Step : 900 Passed
Command : :STAT:OPER:BIT9?
Expected :0
Received :0
Test Step : 901 Passed
Command : :STAT:OPER:BIT9:COND?
Expected :0
Received :0

Test Step : 902 Passed

Command : :STAT:OPER:BIT9:ENAB 32767

Command : :STAT:OPER:BIT9:NTR 32767

Command : :STAT:OPER:BIT9:PTR 32767

Command : :STAT:OPER:COND?

Expected :0

Received :0

Test Step : 906 Passed

Command : :STAT:OPER:EVEN?

Expected :0

Received :0

Test Step : 907 Passed

Command : :STAT:OPER:INST?

Expected :0

Received :0

Test Step : 908 Passed

Command : :STAT:OPER:INST:COND?

Expected :0

Received :0

Test Step : 909 Passed

Command : :STAT:OPER:INST:ENAB 32767

Command : *STB?

Expected :0

Received :0

Test Step : 911 Passed

Command : *ESR?

Expected :0

Received :0

Test Step : 912 Passed

Command : :STAT:OPER:INST:ISUM1?

Expected :0

Received :10101

Test Step : 920 : :STAT:OPER:INST:ISUM1? : Failed

Command : :STAT:OPER:INST:ISUM1:BIT10?

Expected :0
Received :0
Test Step : 921 Passed

Command : :STAT:OPER:INST:ISUM1:BIT10:CAL?
Expected :0
Received :0
Test Step : 922 Passed
Command : :STAT:OPER:INST:ISUM1:BIT10:CAL:COND?
Expected :0
Received :0
Test Step : 923 Passed
Command : :STAT:OPER:INST:ISUM1:BIT10:CAL:ENAB 32767
Command : :STAT:OPER:INST:ISUM1:BIT10:CAL:NTR 32767
Command : :STAT:OPER:INST:ISUM1:BIT10:CAL:PTR 32767
Command : :STAT:OPER:INST:ISUM1:BIT10:COND?
Expected :0
Received :0
Test Step : 927 Passed
Command : :STAT:OPER:INST:ISUM1:BIT10:ENAB 32767
Command : :STAT:OPER:INST:ISUM1:BIT10:NTR 32767
Command : :STAT:OPER:INST:ISUM1:BIT10:PTR 32767
Command : :STAT:OPER:INST:ISUM1:COND?
Expected :0
Received :0
Test Step : 931 Passed
Command : :STAT:OPER:INST:ISUM1:ENAB 32767
Command : :STAT:OPER:INST:ISUM1:NTR 32767
Command : :STAT:OPER:INST:ISUM1:PTR 32767
Command : :STAT:OPER:INST:NTR 32767
Command : :STAT:OPER:INST:PTR 32767
Command : :STAT:QUES?
Expected :0
Received :0

Test Step : 1000 Passed

Command : :STAT:QUES:BIT9?

Expected :0

Received :0

Test Step : 1001 Passed

Command : :STAT:QUES:BIT9:COND?

Expected :0

Received :0

Test Step : 1002 Passed

Command : :STAT:QUES:BIT9:ENAB 32767

Command : :STAT:QUES:BIT9:NTR 32767

Command : :STAT:QUES:BIT9:PTR 32767

Command : :STAT:QUES:COND?

Expected :0

Received :0

Test Step : 1006 Passed

Command : :STAT:QUES:INST?

Expected :0

Received :0

Test Step : 1010 Passed

Command : :STAT:QUES:INST:COND?

Expected :0

Received :0

Test Step : 1011 Passed

Command : :STAT:QUES:INST:ENAB 32767

Command : :STAT:QUES:INST:ISUM1?

Expected :0

Received :0

Test Step : 1013 Passed

Command : :STAT:QUES:INST:ISUM1:BIT10?

Expected :0

Received :0

Test Step : 1014 Passed

Command : :STAT:QUES:INST:ISUM1:BIT10:CAL?

Expected :0

Received :0

Test Step : 1015 Passed

Command : :STAT:QUES:INST:ISUM1:BIT10:CAL:COND?

Expected :0

Received :0

Test Step : 1016 Passed

Command : :STAT:QUES:INST:ISUM1:BIT10:CAL:ENAB 32767

Command : :STAT:QUES:INST:ISUM1:BIT10:CAL:NTR 32767

Command : :STAT:QUES:INST:ISUM1:BIT10:CAL:PTR 32767

Command : :STAT:QUES:INST:ISUM1:BIT10:COND?

Expected :0

Received :0

Test Step : 1020 Passed

Command : :STAT:QUES:INST:ISUM1:BIT10:ENAB 32767

Command : :STAT:QUES:INST:ISUM1:BIT10:NTR 32767

Command : :STAT:QUES:INST:ISUM1:BIT10:PTR 32767

Command : :STAT:QUES:INST:ISUM1:COND?

Expected :0

Received :0

Test Step : 1024 Passed

Command : :STAT:QUES:INST:ISUM1:ENAB 32767

Command : :STAT:QUES:INST:ISUM1:NTR 32767

Command : :STAT:QUES:INST:ISUM1:PTR 32767

Command : :STAT:QUES:INST:NTR 32767

Command : :STAT:QUES:INST:PTR 32767

Command : *WAI

Command : :SYST:LOCK:REL

Steps Passed : 71, Steps Failed : 3, Errors : 0

Data Channel Errors : 40

EBench Test Results Summary file. This file contains a summary of the EBench validation test.

ebench_cmd

Monday, February 12, 2001, 11:43:16

HEADER

%

TEST ID: TEST-001

TEST DESCRIPTION: Bag Read Dialog

%

CONFIG

%

PORNO: 701

IPADDRESS: 127.0.0.1

SPECIFIER:EBENCH

%

Test Step : 813 : :SYST:TIME? : Failed

Test Step : 814 : :SYST:DATE? : Failed

Test Step : 920 : :STAT:OPER:INST:ISUM1? : Failed

Steps Passed : 71, Steps Failed : 3, Errors : 0

Data Channel Errors : 40