

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Gopel VSC 1149.1-A* at our website: *Click HERE*

Software Description

VXI Scan Commander

VSC 1149.1

User Manual

GOEPEL electronic GmbH
Goeschwitzer Str. 58/60
D-07745 Jena
Phone: +49-3641-6896-0
Fax: +49-3641-6896-44
E-Mail: support@goepel.com
http://www.goepel.com

Intelligent Boundary Scan Solutions®

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

© 2003 GOEPEL electronic GmbH. All rights reserved.

The software described in this manual as well as the manual itself are supplied under license and may be used or copied only in accordance with the terms of the license.
The customer may make one copy of the software for safety purposes.
The content of the manual is subject to change without notice and is supplied for information only.

In case of inaccuracies or errors appearing in this manual, GOEPEL electronic GmbH assumes no liability or responsibility.

Without the prior written permission of GOEPEL electronic GmbH, no part of this documentation may be transmitted, reproduced or stored in a retrieval system in any form or by any means as well as translated into other languages (except as permitted by the license).

Printed: 09.10.2003
Authors: Ekkehard Behr

CASCON GALAXY, SCANPLUS and SCANTURY are registered trademarks of GOEPEL electronic. ADYCS, BSDL Wizard, DUALSCAN, SCANPLUS Device Checker, SPACE, SYSTEM CASCON and POLARIS are trademarks of GOEPEL electronic. All other product and company names are trade names or registered trade names of their respective owners.

Issue: 06/2003

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1 VXI SCAN COMMANDER VSC 1149.1 1-1

1.1 SCOPE OF SUPPLY 1-2
1.2 MODEL TYPES 1-2
1.3 FEATURES OF THE SCAN COMMANDER 1-3

2 CONFIGURATION, INSTALLATION AND GETTING STARTED 2-1

2.1 CONFIGURATION 2-1
2.2 HARDWARE INSTALLATION 2-2
2.3 SOFTWARE INSTALLATION 2-3
2.3.1 Scan Commander Firmware 2-3
2.3.2 Host-PC CASCON Driver 2-3
2.3.3 CASCON Remote Control 2-5
2.4 GETTING STARTED 2-6
2.4.1 Power-On 2-6
2.4.2 Dynamic Configuration 2-6
2.5 SCAN COMMANDER FIRMWARE UPDATE 2-7

3 DESCRIPTION OF HARDWARE UNITS 3-1

3.1 BOUNDARY SCAN CONTROLLER (BSC) 3-1
3.1.1 Technical Data 3-1
3.1.1.1 TCK Frequency 3-1
3.1.1.2 Operating Conditions 3-2
3.1.2 Hardware Interface 3-3
3.1.3 BScan Controller Programming 3-4
3.2 PARALLEL INTERFACE PORT (PIP) 3-4
3.3 TRIGGER UNIT (C-BUS) 3-4
3.3.1 Hardware Interface 3-5
3.3.2 C-Bus Programming 3-6

4 CASCON REMOTE OPERATING PROGRAM 4-1

4.1 INITIAL SCREEN AND MAIN MENU 4-1
4.2 TEST EXECUTION SECTION 4-2
4.2.1 UUT and Test Selection 4-2
4.2.2 Starting Test Execution 4-3
4.3 INTERACTIVE SECTION 4-4
4.4 TRANSFER OF ARCHIVES AND FILES 4-5

5 SCPI PROGRAMMING 5-1

5.1 COMMAND HEADER 5-1
5.2 PARAMETER DATA TYPES 5-2
5.2.1 String Program Data 5-2
5.2.2 Character Program Data 5-2
5.2.3 Arbitrary ASCII Data 5-2
5.2.4 NR1 Numeric Data 5-2
5.2.5 Binary Numeric Data 5-3
5.2.6 Hexadecimal Numeric Data 5-3
5.2.7 Arbitrary Block Program Data 5-3
5.3 MMEMEMORY SUBSYSTEM 5-4
5.4 PROGRAM SUBSYSTEM 5-10
5.5 OUTPUT, TRIGGER AND CBUS SUBSYSTEMS 5-37
5.6 SYSTEM SUBSYSTEM 5-44
5.7 STATUS SUBSYSTEM 5-45
*5.7.1 Commands *SRE and *STB? 5-46*

6 PROGRAMMING EXAMPLES 6-1

6.1 WRITE FILE 6-1
6.2 READ FILE 6-1
6.3 BOUNDARY SCAN TEST EXECUTION 6-2

7 REMOTE SUPPORT IN CASLAN 7-1

7.1 HOSTREAD INSTRUCTION 7-2
7.2 MANUAL DIAGNOSIS SUPPORT 7-3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8 INTERFACE FOR THE USER APPLICATION 8-1

8.1 FUNCTION TREE 8-2
8.2 INIT/ CLOSE FUNCTIONS 8-4
8.2.1 Init 8-4
8.2.2 Init With Options 8-5
8.2.3 Close 8-5
8.3 APPLICATION FUNCTIONS 8-6
8.3.1 Run Test 8-6
8.3.2 Launch Test 8-7
8.3.3 Check Test End 8-7
8.3.4 Receive Protocol File 8-7
8.3.5 Receive Any File 8-8
8.3.6 Transmit Any File 8-8
8.3.7 Transmit Archive 8-9
8.3.8 Transmit Update 8-9
8.4 DATA FUNCTIONS 8-10
8.4.1 Write Instrument Data 8-10
8.4.2 Read Instrument Data 8-10
8.4.3 Query Instrument Data 8-11
8.4.4 Transmit File Data 8-11
8.4.5 Receive File Data 8-11
8.5 UTILITY FUNCTIONS 8-12
8.5.1 Error Message 8-12
8.5.2 Error Query 8-12
8.5.3 Revision Query 8-13
8.6 CONFIGURATION FUNCTIONS 8-14
8.7 ATTRIBUTES 8-15

9 COMPARISON TO SCAN COMMANDER 3 9-1

9.1 VSC 1149.1 AND SYSTEM CASCON 9-2
9.2 COMPATIBILITY OF THE INTERFACE TO THE USER APPLICATION 9-3
9.3 MODIFICATIONS IN THE OPERATING PROGRAM 9-4
9.3.1 Transferring Archives 9-4
9.3.2 File Transfer Section 9-6

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1 VXI Scan Commander VSC 1149.1

The VXI Scan Commander VSC 1149.1 is a VXI bus based Boundary Scan controller fully compatible with IEEE 1149.1 (JTAG) standard and VXI bus system specifications of revision 1.4.

In order to use the Scan Commander, a host PC is required, were SYSTEM CASCON runs. The entire configuration, consisting of the Scan Commander and the software modules required on the host PC, is shown in the figure below.

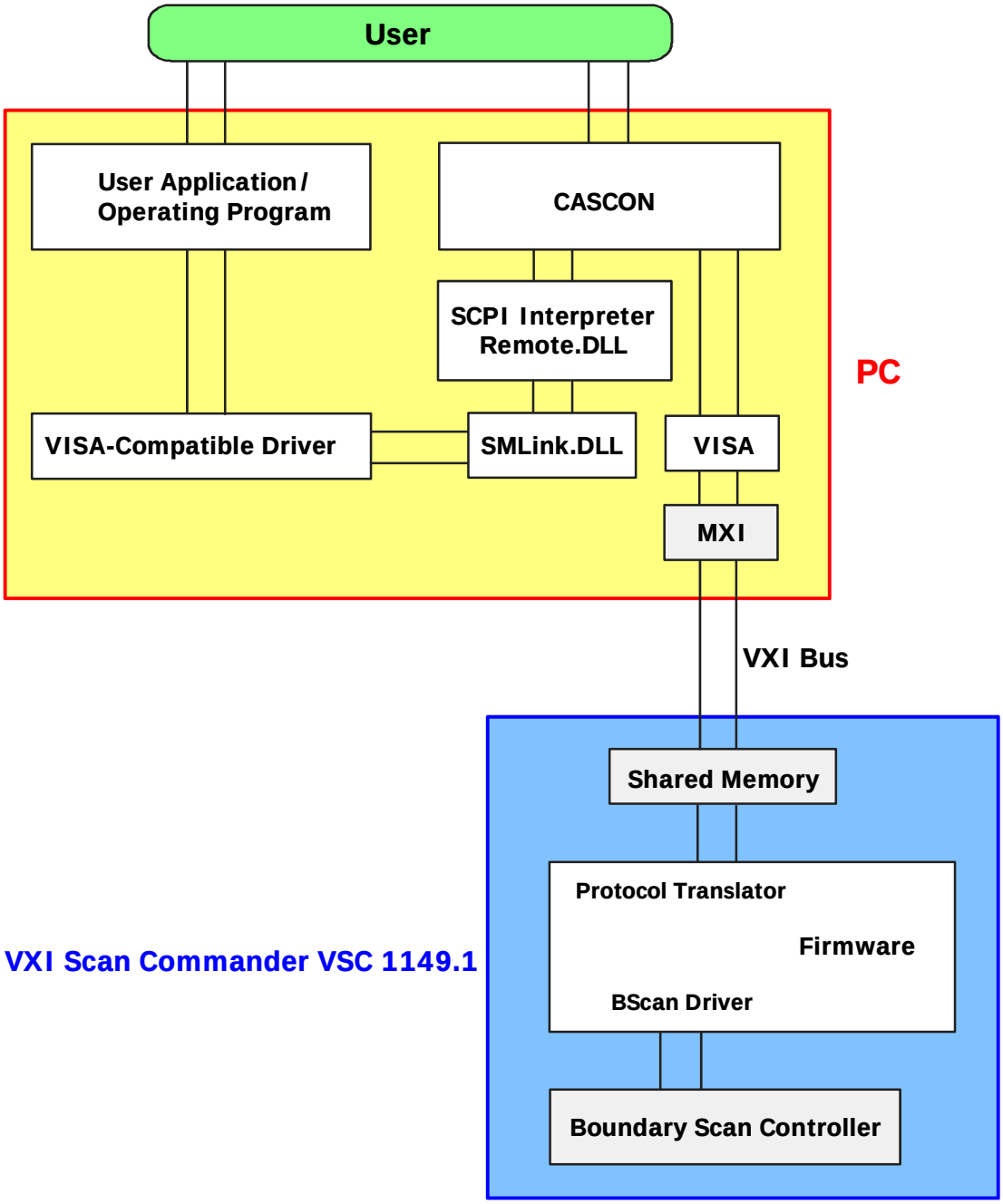


Figure 1-1: Position of Scan Commander 4 in the Entire System

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Essentially, the Scan Commander VSC 1149.1 consists of the VXI interface, designed to shared memory, the actual Boundary Scan controller and the firmware.

Besides SYSTEM CASCON, the VISA driver of the VSC 1149.1 as well as a MXI controller with VISA driver interface are needed on the host PC.

Optionally, this system can be remote-controlled by means of two CASCON DLLs and a VISA-compatible driver. An operating program (*SCC4HOST.EXE*, included in the delivery) or an user application can be used for this remote control.

1.1 Scope of Supply

The delivery also contains (in addition to the device VSC 1149.1, including firmware) software to be installed on the host PC:

- the DLLs to be linked with SYSTEM CASCON,
- the VISA-compatible driver, to be linked with the remote control program (user application),
- the example operating program *SCC4HOST.EXE*.

The delivery does **not** contain CASCON/ CASCON GALAXY, which has to be purchased separately in a suitable package, the MXI hardware including accessory drivers as well as the user application to be provided by the user himself.

1.2 Model Types

VSC 1149.1 is available in two types. It is indicated on the respective label on the instrument.

type	description
VSC1149.1-A	2 TAP controllers
VSC1149.1-B	2 TAP controllers with SPACE™ architecture

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.3 Features of the Scan Commander

The Scan Commander VSC 1149.1 has the following features:

- A CPU is integrated in VSC 1149.1 that runs the Boundary Scan controller driver and a Protocol Translator (firmware).
- The Boundary Scan controller integrated in VSC 1149.1-B(A) achieves performance and speed of the PSC 1149.1-B(A) card for PC-based Boundary Scan systems.
- As triggering is possible (external and via VXI backplane) to synchronize test vectors of the Scan Commander with other VXI modules (e.g. PIO), the Scan Commander is especially suited for application in fast manufacturing test systems.
- SYSTEM CASCON (on the host system) can be controlled by means of text commands complying to SCPI standard and a VISA-compatible driver.
- In contrast to conventional VISA- or SCPI-controlled devices, the commands are interpreted by SYSTEM CASCON on the host PC. However, the compatibility to already existing user applications is ensured in this way.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2 Configuration, Installation and Getting Started

- Configuration
- Hardware Installation
- Software Installation
- Getting Started
- Scan Commander Firmware Update

2.1 Configuration

Logic Address

Like all VXI devices, the VXI Scan Commander needs a logic address too. The VSC 1149.1 can be configured both statically and dynamically. The logic address can be defined between 1 and 255. Default address after manufacturing is LA=255 (dynamic configuration).

The logic address (LA) is set by means of an 8-pin DIL switch located on the upper side of the instrument near the VXI connectors.

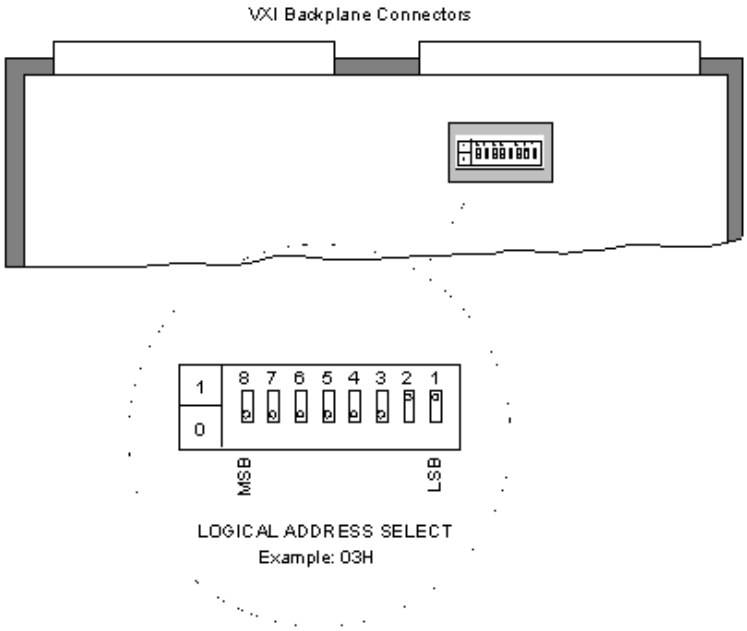


Figure 2-1: Logical Address



Do not use address 0 though this switch position is possible. This address is reserved for the VXI rack's system controller.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Trigger Lines

VSC 1149.1 is provided with a so-called C-bus (2-wire handshake bus) to synchronize the TAP controller with other devices.

Signals of the C-bus are provided at external connector XH3 on the front panel of VSC 1149.1. C-bus signals can be connected both to the TTL trigger lines and LBUSA or LBUSC lines on the VXI backplane. This connection is programmable.

2.2 Hardware Installation

An interface MXI II from National Instruments is used for communication, otherwise a compatible VXI interface with belonging drivers is needed. Due to memory-based operation, an GPIB slot 0 controller cannot be used.

If the manufacturer's setting of the logic address (255) is not modified, then it can be fit into any free slot of the VXI rack immediately without considering the addresses of existing devices. The valid logical address will be assigned by the Resource Manager later.

If a logical address between 1 and 254 (static configuration) is defined, make sure that no other instrument uses this address. Otherwise the Resource Manager cannot correctly initialize the VXI system, and hardware damages at the respective devices are possible.

After successful definition of the logical address install the module into the VXI rack after selecting a free slot (C-size). Note that VSC 1149.1 cannot be used as system controller.



Do not install the device in slot 0. Otherwise hardware damages on both module and VXI backplane are possible.

Make sure the operating voltage of the VXI device is switched off (POWER OFF) before inserting the module in the selected slot.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.3 Software Installation

- Scan Commander Firmware
- Host-PC CASCON Driver
- CASCON Remote Control

2.3.1 Scan Commander Firmware

The Scan Commander firmware is pre-configured. It consists of the following components:

file	call / notes	parameters
SCCBOOT.EXE	in *AutoExec.bat*	none
CASVXI.BAT	in *AutoExec.bat*	
SCC4INIT.EXE	in *CASVXI.BAT* parameter determines VSC1149A/B	none
SCC4EXEC.EXE	in *CASVXI.BAT*	none
D32_PXI_A.DLL	loaded in *SCC4EXEC.EXE*, if SCC4INIT was 'PXI1149_A'	none
D32_PXI_B.DLL	loaded in *SCC4EXEC.EXE*, if SCC4INIT was 'PXI1149_B'	none
Borland DPMI-Runtime 16/32 **32RTM.EXE** **RTM.EXE** **DPMI16BI.OVL**	SccBoot, Scc4Init under DPMI16, Scc4Exec under Dpmi32	none

(*) Additionally, the device contains a complete CASCON 3 Scan Commander configuration which is started via *CASVXI.BAT*. This CASVXI.BAT.is overwritten during transferring the update.

2.3.2 Host-PC CASCON Driver

Software Assumptions:

In order that SYSTEM CASCON is able to work together with the Scan Commander, the following files are needed:

- *VSC1149A.DRL* and
- *VSC1149B.DRL* respectively (Driver DLL)
- *SCC3HDRV.DLL* (VISA DLL)

These files have to be located in the directory <cascon>\SYS\BIN.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Drivers:

The driver DLL (DRL) communicates to the VXI device of the host PC (VXI-MXI) via VISA DLL.

There are the following driver DLLs, to be loaded under User Defined if necessary:

- *VSC1149A.DRL*
- *VSC1149B.DRL*

The selection of DRL must correspond to the used Scan Commander. A VISA init string has to be handed over as parameter.

The parameter is taken over by the driver DLL from the registry from the following path:

[HKEY_LOCAL_MACHINE\SOFTWARE\Goepel\CASCON\DRIVERS\drive_name]
"VisaInitString"="VXIboard_num::vxi_addr::INSTR"

<driver_name>	VSC1149-A | VSC1149-B
<board_num>	number of interface card, beginning with 0
<vix_addr>	VXI address, mostly taken by dynamic configuration

Example: `"VXI0::2::INSTR"`

The special parameters can be found out by means of the T&M Explorer during the use of NI software. In this connection, it has to be searched for devices with Vendor-ID 0xEC5 and Device-ID 0x0135 (memory based device).

As no real WINDOWS driver is used, no driver needs to be installed. The Scan Commander drivers are **not** considered in the driver manager of SYSTEM CASCON.

As the driver communicates to the Scan Commander via VISA interface, it is essential to install the corresponding run-time engines. They do **not** belong to the CASCON installation and must be provided by the manufacturer of VXI transmission medium.

These run-time engines are e.g. for VXI-MXI of National Instruments:

- NI-VXI software (tested for V3.2)
- NI-VISA software (tested for V2.60)
- NI-IVI software (tested for V1.83)

All three components have to be installed.

Additionally, LabWindows CVI runtime is needed. However, it is installed within the normal CASCON installation.

Here, the files needed for remote-control are still not considered and will be explained in the next section.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.3.3 CASCON Remote Control

The installation of the remote software is optional (in contrast to CASCON 3.x) because SYSTEM CASCON can be used "directly".

The remote environment is linked, whereas x is a current number:

- *CASCON.CFG*:
  under [Add_Ons] an entry Load_DLL_x = 'remote'
  The file *REMOTE.DLL* in <cascon>\SYS\BIN is required.
- *REMOTE.CFG* (cascon\SYS\MISC):
  under [Remote_Interface] an entry Load_Client_x = 'smlink'
  The file SMLINK.DLL in <cascon>\SYS\BIN is required.

The communication is made via shared memory which has to be created and initialized by *SMLINK.DLL*. That means, SYSTEM CASCON must be started before the remote application. The remote application can be left and started again any time after SYSTEM CASCON has been started.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.4 Getting Started

A two-color LED on the front panel indicates the operating state of VSC 1149.1. After initialization the LED lights up green.

While waiting for data/ write permission, the LED is red. After successful termination the LED returns to green. Continuous red light indicates unsuccessful read/ write.

2.4.1 Power-On

After all devices have been installed properly, switch on the VXI rack power (POWER ON). The following procedures will be executed in VSC 1149.1:

- Initialization of the programmable VXI interface registers by the integrated microprocessor unit
- Execution of self test procedure
- Start of the Update tool software

If VXI initialization and self test were successful, the LED will light up green. Time between power on or VXI mainframe reset until the end of the self test is about one minute.

2.4.2 Dynamic Configuration

If VSC 1149.1 is dynamically configured, then the assigned logic address is detected as follows after closing the resource manager (examples are valid only for use of system controllers from National Instruments):

Reading of resource manager messages:

The logical address has to be read out by a corresponding tool (e.g. Measurement Instrument Explorer or VIC). In this connection, it has to be looked for a memory-based device with the vendor ID EC5H.

The logic address has to be submitted to the VISA driver or it has to be set in the initializing options ***Options/ Settings*** of the operating program.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.5 Scan Commander Firmware Update

In order to generate an update of the Scan Commander firmware, the firmware update tool *SCCUpdate.EXE* must be started.

Assumptions:

The VXI controller, the VISA environment and the IVI Runtime engine have to be installed (they are not contained in the Scan Commander Software and have to be delivered by the VXI controller manufacturer).

Presettings:

Under *Options/ Settings,* some presettings have to be made:

Remote Control	select the type of VXI Slot 0 Controller
VXI-LA	find out the logical address of Scan Commander (by means of Measurement & Instrument Explorer or another tool)
Update Files Default Directory	directory with update files (this will be the initial directory of the file select dialog)

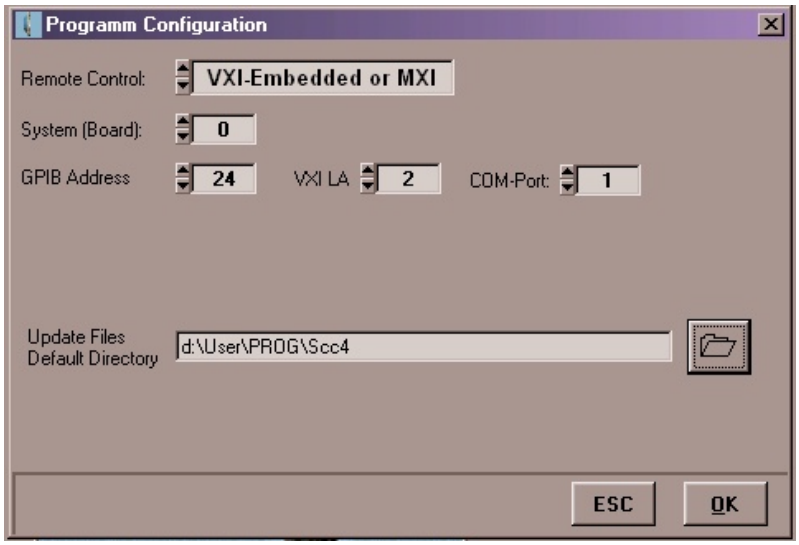


Figure2-2:
Update Presettings

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Operation:

The Scan Commander must be rebooted in order to run the *SCCUpdate.EXE* there. The resource manager (on the host) must have been run. SYSTEM CASCON must not have been started.

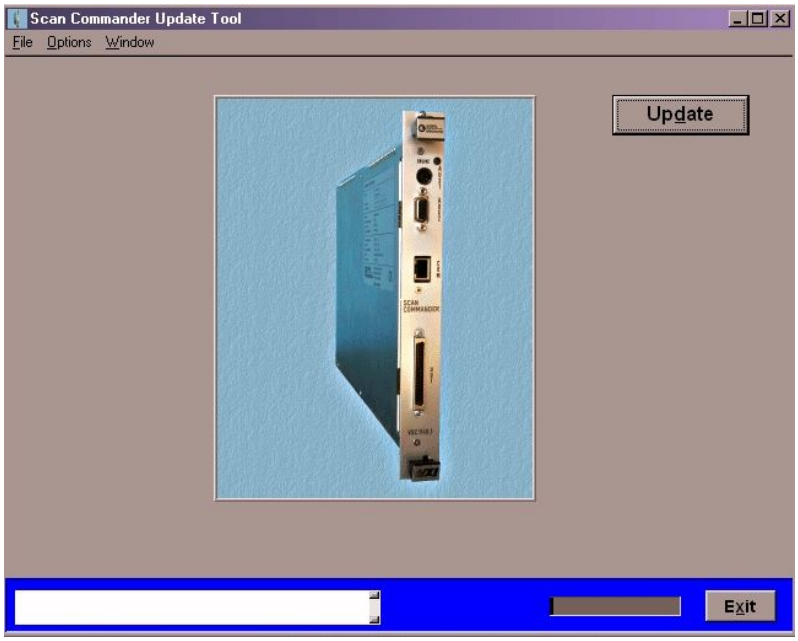


Figure 2-3: Scan Commander Update Tool

If an error is displayed in the lower small terminal window after the start of SCCUpdate.EXE, one or more of the upper preassumptions are not fulfilled and updating is impossible.

The ***Update*** button must not be disabled (if it is disabled, either CASCON 3 or the protocol translator runs on the Scan Commander and not the update tool).

Pressing ***Update*** causes the selection of the update archive in a file dialog. This archive has the extension *UPD*.

Confirming by Select transfers the selected file to the Scan Commander. A progress bar is refreshed.

After file transfer, unpacking is started automatically. Then SCCUpdate.EXE is paused until the protocol translator can be contacted. Afterwards, SYSTEM CASCON can be started.

Before the update can be repeated, the program has to be left and the initial condition has to be restored on the Scan Commander (by RESET and recalling the resource manager).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3 Description of Hardware Units

- Boundary Scan Controller (BSC)
- Parallel Interface Port (PIP)
- Trigger Unit (C-Bus)

3.1 Boundary Scan Controller (BSC)

VSC 1149.1 allows Boundary Scan tests compliant with IEEE 1149.1 standard.

The Boundary Scan Controller consists of a serializer that transform data coming from Boundary Scan test and diagnosis software within VSC 1149.1 into a serial data format, intermediately stores it and outputs it in form of a burst via the testbus interface.

3.1.1 Technical Data

- two TAP controllers
- parallel test of eight equal-type test objects
- SCP-bus support
- 2 x 16 bit PIO (Parallel Input Output)
- C-bus
- TCK burst
- C-bus connection via trigger bus (TTLTRG 0-7), local bus left (LBUSA 00-11), local bus right (LBUSC 00-11) or externally.

3.1.1.1 TCK Frequency

VSC 1149.1-A

- TCK frequency is programmable within range 61Hz..16MHz. It is derived from a 64 MHz clock by dividing and is calculated according to the following formula: $f_{TCK} = f_{Base} / [2 \cdot (x+1)]$.
- Frequency f_{Base} is a frequency of 64 MHz if the selected TCK frequency is ≥ 2MHz. If the selected TCK frequency is below 2MHz, then frequency f_{Base} is switched to 8 MHz. Value x is a register value. It has a value of 0 to 15 (for 64 MHz) and 0 to 65535 (for 8 MHz) depending on the reference frequency.

VSC 1149.1-B

- TCK frequency is programmable within in range 500Hz..30MHz.
- TCK frequency is generated according to the following formula: (12MHz+n*250kHz)/ 2m

Max. [MHz]	Min. [MHz]	Resolution [MHz]	Accuracy [%]	Value for m	Frequency range	Value for n
30.0	12.0	0.25	2	0	0	0..72
12.0	6	0.125	2	1	1	0..47
6	0.0005	$0.25/2^m$	2	2..15	2..15	0..47

Values m and n are calculated by the CASCON software depending on the set TCK frequency.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1.1.2 Operating Conditions

VSC 1149.1-A

Vcc	3,3 V
signal level	Fixed TTL level. Input and output drivers are provided by two 74LVT244A.
input voltage range	0 V – 5.5 V
max. output voltage	Vcc – 0.2 V (3.1 V)
max. output voltage	± 24 mA
max. high-level output current	- 32 mA
max. low level output current	+64 mA (current duty cycle ≤ 50%, f ≥ 1 kHz)

VSC 1149.1-B

The driver for the output level is provided by a 74CVX16244. Level programmability is achieved by varying the operating voltage of the driver device. Thus the output current is varied. The following extract from the data sheet illustrates this.

output current in I_{OH} / I_{OL}	Vcc=3.0V to 3.6V	± 24mA
	Vcc=2.3V to 2.7V	± 18mA
	Vcc=1.8V	± 6mA
signal level	programmable TTL level for inputs and outputs	

Electrical characteristics (2.7V < Vcc ≤ 3.6V):

symbol	condition	Vcc(V)	min.	max.	unit
V_{OH}	I_{OH}=-100µA	2.7-3.6	Vcc-0.2		V
V_{OH}	I_{OH}=-12mA	2.7	2.2		V
V_{OH}	I_{OH}=-18mA	3.0	2.4		V
V_{OH}	I_{OH}=-24mA	3.0	2.2		V
V_{OL}	I_{OL}=100µA	2.7-3.6		0.2	V
V_{OL}	I_{OL}=12mA	2.7		0.4	V
V_{OL}	I_{OL}=18mA	3.0		0.4	V
V_{OL}	I_{OL}=24mA	3.0		0.55	V

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Electrical characteristics (23V < Vcc ≤ 2.7V):

symbol	condition	Vcc(V)	min.	max.	unit
V_{OH}	I_{OH}=-100µA	2.3-2.7	Vcc-0.2		V
V_{OH}	I_{OH}=-6mA	2.3	2.0		V
V_{OH}	I_{OH}=-12mA	2.3	1.8		V
V_{OH}	I_{OH}=-18mA	2.3	1.7		V
V_{OL}	I_{OL}=100µA	2.3-2.7		0.2	V
V_{OL}	I_{OL}=12mA	2.3		0.4	V
V_{OL}	I_{OL}=18mA	2.3		0.6	V

Electrical characteristics (1.8V ≤ Vcc < .3V):

symbol	condition	Vcc(V)	min.	max.	unit
V_{OH}	I_{OH}=-100µA	1.8	Vcc-0.2		V
V_{OH}	I_{OH}=-6mA	1.8	1.4		V
V_{OL}	I_{OL}=100µA	1.8		0.2	V
V_{OL}	I_{OL}=6mA	1.8		0.3	V

V_{OH} HIGH level output voltage
V_{OL} LOW level output voltage

3.1.2 Hardware Interface

Testbus signals are supplied at connector XH3 → see Appendix C:

TAP1	TCK1	Test clock
	TMS1	Test Mode Select
	TDO1	TDO (test data output) of the controller
	TDI1_1	TDI (test data input 1) of the controller
	TDI1_2	TDI (test data input 2) of the controller
	TDI1_3	TDI (test data input 3) of the controller
	TDI1_4	TDI (test data input 4) of the controller
TAP2	TCK2	Test clock
	TMS2	Test Mode Select
	TDO2	TDO (test data output) of the controller
	TDI2_1	TDI (test data input 1) of the controller
	TDI2_2	TDI (test data input 2) of the controller
	TDI2_3	TDI (test data input 3) of the controller
	TDI2_4	TDI (test data input 4) of the controller
both TAPs	/TRST	Testbus reset

Connect the controller's TDI to the TDO of the UUT. Connect the controller's TDO to the TDI of the UUT.



We recommend to twist the signals with GND before supplying them to the UUT.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1.3 BScan Controller Programming

Programming of the Boundary Scan controller is very extensive. Therefore, we will not mention most important instructions and illustrate them with a programming example here. For information about the programming please refer to the description of the programming language CASLAN (Reference Manual).

3.2 Parallel Interface Port (PIP)

number of I/Os	32
formatability	4 x 8 or 2 x 16
Vcc	3,3 V
signal level	TTL
input voltage range	0 V – 5.5 V
output voltage	max. Vcc – 0.2 V (3.1 V)
recommended output current	±24 mA
max. high-level output current	-32 mA
max. low-level output current	+64 mA (current duty cycle ≤ 50%, f ≥ 1 kHz)

The Parallel Interface Port (PIP) consists of 2 low voltage ICs of type 74LVT16652. One of these 74LVT16652 ICs realizes the function of two 8-bit wide I/O ports on Boundary Scan controller PSC 1149.1-B. To realize a 32-bit wide port as well, two of these devices have been implemented. Connect them individually or together as input or output in groups of 8 bits. (PA1..8, PB1..8, PC1..8, PD1..8).

There are no handshake signals for the PIP devices. Safe data transmission has to be ensured from user side.

For PIP complex pin assignment refer to section 5.3:"MMEMory Subsystem".

3.3 Trigger Unit (C-Bus)

VSC 1149.1 is equipped with an external control for the Boundary Scan controller. It is a 2-wire handshake bus called C-bus (control bus). The C-bus can be used to synchronize Boundary Scan test vectors with vectors or measurements of other VXI modules.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3.1 Hardware Interface

C-bus signals can be supplied at connector XH3 (see Appendix C), or at the trigger lines and the local bus lines of connector P2 (see Appendix C). The lines used for the C-bus are selected with Boundary Scan Software SYSTEM CASCON.

The C-bus consists of a ready line (BSRDY) of VSC 1149.1 and a trigger line (BSTRG) driven by the superior test system. Further description refers to the case that the superior system is an in-circuit tester (ICT). It might be useful to adapt the C-bus signals via the ICT's pin electronics.

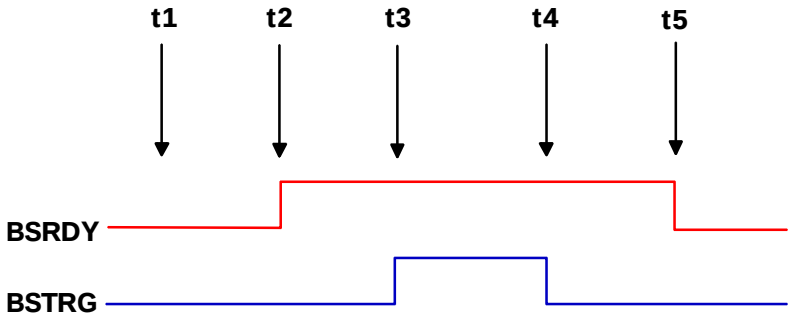


Figure 3-1: C-Bus Signal Behavior

t1	The ICT (host system) transmits the request for interactive test vector synchronization via IEEE 488.2 or RS232 to the CASCON Boundary Scan software by means of a SCPI command. CASCON then runs the preset test program. This activity has to be executed at the beginning of a test program only.
t2	The Boundary Scan controller (BSC) supplies the current test data (driver values) at the UUT, sets BSRDY line from LOW to HIGH, and stops testbus control after TAP controller state Update.
t3	The ICT drives its test vectors and sets the BSTRG line from LOW to HIGH. Thus the Boundary Scan controller is triggered and testbus control is continued. There are no requirements of a minimal or maximal delay between t2 and t3.
t4	As soon as the ICT has measured the line levels, it resets the BSTRG line to LOW. The ICT may not deactivate its driver pins (floating), unless BSRDY line is set to low as well.
t5	The Boundary Scan controller again stops testbus control, after the UUT has measured the line levels with its Boundary Scan input cells (Capture state). This is indicated for the ICT by setting BSRDY line from HIGH to LOW The Boundary Scan controller can continue testbus control and apply the next test vector, if ICTRDY line is set to LOW. Steps t2 through t5 will be executed until test program end then.

Despite the application of the C-bus for interactive connection of the Boundary Scan test system to an in-circuit tester described above, the BSRDY line can be used as trigger signal for several measuring devices. Program any desired TAP controller state. BSTRG line does not have to be connected in this mode.

For information on programming refer to the description of the programming language CASLAN in the Reference Manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3.2 C-Bus Programming

C-bus programming is divided into two sections:

- C-bus control within the Boundary Scan test program
- Connection of C-bus signals to TTL trigger lines on the VXI backplane

C-Bus Control

C-bus control is programmed within a Boundary Scan test program by means of a special set of instructions integrated in the CASCON software package. The following table contains instructions needed for programming.

C_Bus_On	Switches C-bus control on; all following DRShift instructions contain breaks
C_Bus_Off	Switches C-bus control off
C_Bus_On / NoWait	Switches C-bus control on; all following DRShift instructions cause a low-high-low change on the TRG OUT line (BSRDY)
C_Bus_On / Wait	Same as **C_Bus_On**

Programming Example: (Extract from CASLAN Program)

```
DH (IC1:#25);
DrShift;        normal testbus control
C_Bus_On;       C-bus control is switched on
DL (IC3:#3);
DrShift;        Testbus control is stopped after
                passing TAP state Update or Capture
C_Bus_On/NoWait;     Change of C-bus control
DL (IC1:#25);
DrShift;        Testbus control triggers low-high-
                low signal on the TRG OUT line
C_Bus_Off;      C-bus control is switched off
```

Connection on the VXI Backplane

The C-bus signals are connected to the TTL trigger lines of the VXI backplane by means of the controller settings within CASCON.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4 CASCON Remote Operating Program

SYSTEM CASCON is fully controllable via SCPI instructions, i.e. no special registers have to be addressed. These instructions are supplied with the interface program for communication. Make sure to install the appropriate version for your Windows system (Windows NT, Windows 95/98).

The interface program uses a VISA-compatible library.

The program runs on the host PC and communicates by means of SCPI commands via CASCON Remote DLL to SYSTEM CASCON. Before, SYSTEM CASCON must have been started.

4.1 Initial Screen and Main Menu

The interface program allows the following main functions:

- Certain files can be transferred from host system to VSC 1149.1 and vice versa;
- Single SCPI instructions can be send interactively and possible responses can be read. This is an additional support for the development of own remote programs;
- Trigger types of the VXI system (TRG-IN, TRG-OUT, trigger lines on the backplane) are easily set;
- UUTs and belonging test programs as well as batch files can be selected and run,

Switching between the sections is done via the application's main menu.

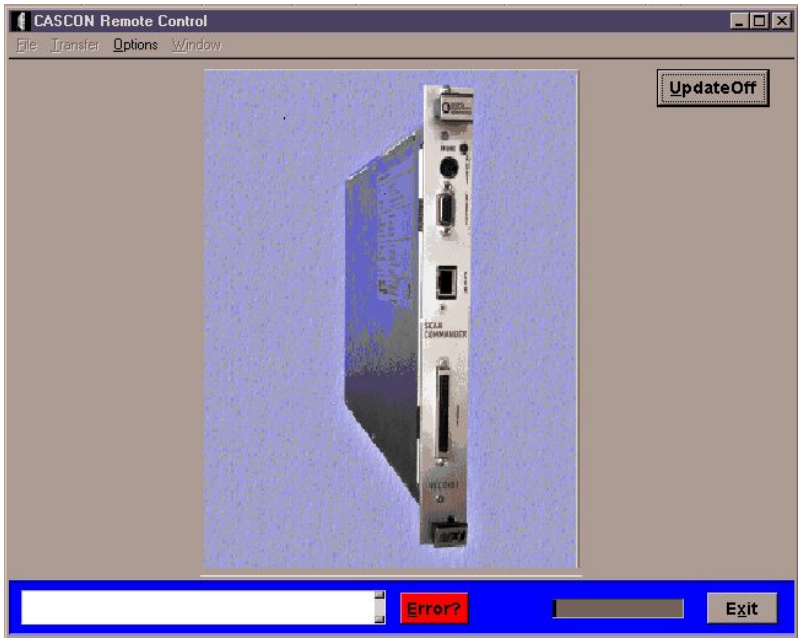


Figure 4-1: Initial Screen

Use the button ***Update Off*** for leaving the initial screen.

The blue status line is active in all sections. It contains a message field for error messages, a button for error query (**syst:err?**), a progress bar for file transfers and an exit button to quit the program. The operating program can be quit and restarted on running VSC 1149.1 unless is just executed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.2 Test Execution Section

After quitting the Update Tools the test section is automatically opened if SYSTEM CASCON is set as servant. Switching to the test section from other sections is possible via the *Window* menu. The panel shown here is similar to the surface of CASCON.

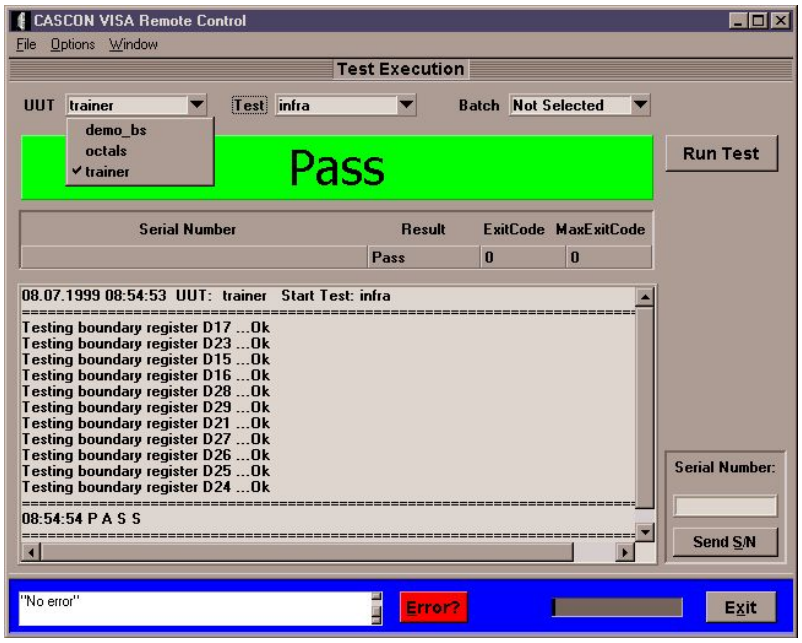


Figure 4-2: Test Section Panel

4.2.1 UUT and Test Selection

UUTs, tests and batches are selected in the upper three fields. Available and current items are loaded on starting CASCON. Click on the button to the right of the field to open a listbox. The selected line is entered into the entry field, and the UUT/ test is set by pressing ***OK***. Use menu point *Options / Get UUTs and Tests* to read anew.

The selection whether a CASCON batch or a CASLAN test is started is done in the batch list. If a CASLAN test has to be started, then select Not Selected for the batch.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.2.2 Starting Test Execution

Press button ***Run Test*** to run the test/ batch. Depending on the settings, the following actions are executed:

- Setting of test timeout
- Sending of correct start command
- Waiting for test end or HostRead
- Query and display of the test result
- Reading, displaying and storing of the test protocol
- Setting of default timeout

Before running the test, a serial number can be entered and send. It is indicated in the test protocol.

If a **HOSTREAD** instruction is reached, it will be indicated (Paused) and the ***Run*** button is modified to Continue. Now you can switch to the interactive section, e.g. to query variables or to input data. Then test execution can be continued.

The test protocol is stored under the name *protocol.txt* and the previous one is overwritten. The protocol is shown in the large scrolling field up to the length of 16 Kbytes. Further information is cut, but stored in the file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 Interactive Section

This section allows access to all SCPI instructions SYSTEMCASCON (or possibly another servant) can interpret. This might be helpful when own operating programs are developed on the basis of the supplied VISA driver.

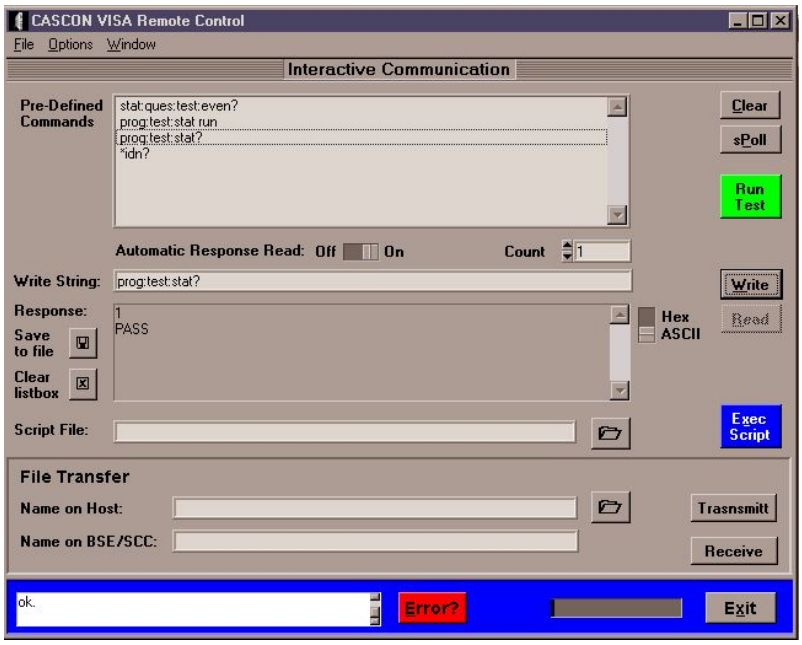


Figure 4-3: Panel Interactive Communication

The upper part contains a list of commands or partial commands. They are read from a text file on program start. The file is set under *Options/ Settings* for Command File. Double click a line to enter it into the Write String field. Here they can be edited. Press ***Write*** to send this string. Press ***Read*** to read the response. If Automatic Response Read is switched ***on***, the response is read automatically if the string contains an '?'. Activities are repeated as set in the Count field.

Responses are either shown in ASCII or hexadecimal format. Settings will be valid after reading only.

Clicking ***ClearListbox*** the response window is cleared. All read responses are listed in this window. Use ***SaveToFile*** to save the entire contents of the window in a file. The file name is set under *Options/ Settings* for ***SaveResponse***.

This panel also allows the execution of simple scripts. They only contain a sequence of SCPI commands. Every command containing '?' is regarded as query, and a response will be read.

Select the script file for this (use the button next to the field for browsing) and press ***ExecScript***. Script files will be executed only one time, not depending on Count settings.

Finally, files can be transferred via this panel. Select the file name on host side and enter the file name on VSC 1149.1/ BSE 1149.1 (no browsing available).

Respective VISA functions are directly used for file transfer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4 Transfer of Archives and Files

The complete archive work and file transfer are carried out within SYSTEM CASCON on the host computer. However, it is still possible to proceed as in CASCON 3.x (refer to section 9.3: "Modifications in the Operating Program") because of compatibility.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5 SCPI Programming

VSC 1149.1 is programmed with text commands complying to SCPI standard. Commands are available within subsystems **MMEMory**, **PROGram**, **STATus**, **OUTPut** and **SYSTem**. The **MMEMory** subsystem supports the features of the Scan Commander's mass storage (hard disk). Commands of **PROGram** subsystem refer to Boundary Scan test execution using the Boundary Scan test and diagnosis software installed on VSC 1149.1. The **OUTPut** subsystem allows the selection of TTL trigger lines on the VXI backplane. Transferring SCPI commands and reading responses on SCPI commands respectively is served by means of the data functions of the VISA compatible driver (see section 8.4: "Data Functions").



If an application function for the requested action is provided, it should be passed on SCPI commands!

5.1 Command Header

All commands consist of a composition of single keywords. Most of the keywords have long and short form. The short form is indicate by capitalization. Short and long form of single keywords may be mixed within one command. Yet do not use any forms other than long or short form of the keyword.

Example:

```
MMEMory:CATalog?      Only long form ->correct
MMEM:CAT?             Only short form ->correct
MMEMory:CAT?          Short and long form ->correct
MMEM:CATalog?         -"-
MMEMo:CATal?          Intermediate forms → not permitted
```

If several commands are used together in one message, then separate the commands by a semicolon.

Example:

```
MMEMory:CDIRectory "C:\CASVS";SYSTem:ERRor?
```

In case commands of the same subsystem or the same node in the tree structure of commands are combined to a message, then the superior keywords may be omitted from the second command on.

Example:

```
MMEMory:NAME "C:\TEST.TXT";OPEN;FEED?;CLOSe
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.2 Parameter Data Types

The following data types are used for parameters:

- String Program Data
- Character Program Data
- .Arbitrary ASCII Data
- NR1 Numeric Data
- Binary Numeric Data
- Hexadecimal Numeric Data
- Arbitrary Block Program Data

5.2.1 String Program Data

The STRING PROGRAM DATA data type is used for parameters in form of strings included in double or single inverted commas. All ASCII characters may be used in this string. Capitalization is kept (case sensitivity depends on the program processing the instructions).

Example: Command: **PROG:UUT:SNAM "OCTALS"**

5.2.2 Character Program Data

The CHARACTER PROGRAM DATA data type is used for parameter information (e.g. program names) in form of short alphanumeric strings. The alphanumeric string has to begin with a letter and may contain the following ASCII coded characters (decimal):

65 to 90 ('A' to 'Z'), 97 to 122 ('a' to 'z'), 48 to 57 ('0' to '9') and 95 ('_').

Blanks are not allowed. Characters are capitalized for interpretation.

Example: Command: **PROGRAM:UUT: NAME "oCtALs"**

5.2.3 Arbitrary ASCII Data

This data type is used only as ARBITRARY ASCII RESPONSE DATA for responses to instructions ***idn?** and **SYSTem:CTYPe?**. It is similar to type CHARACTER DATA; yet, blanks may be used within parameters, parameters may begin with numbers, and characters except for ';' and NL may be contained.

Example: Command: ***idn?**
Response: **Goepel Electronic GmbH, BSE, 4711, V3.0**

5.2.4 NR1 Numeric Data

This data type contains integer numeric values consisting of a double sharp followed by letter B. Binary numbers (0 or 1) will be indicated behind without blanks. The numbers have to evaluated unsigned always. Leading zeros are not necessary.

Example: **STAT:QUES:TEST:ENAB 3**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.2.5 Binary Numeric Data

This data type contains integer numeric values consisting of a double sharp followed by letter B. Binary numbers (0 or 1) will be indicated behind without blanks. The numbers have to evaluated unsigned always. Leading zeros are not necessary.

Example:
```
Command: CBUS:DEST #B11110000, #B01101001
```

5.2.6 Hexadecimal Numeric Data

This data type contains integer numeric values consisting of a double sharp followed by letter H. Hexadecimal numbers will be indicated behind ('0' to '9', 'A', 'B'., 'C', 'D', E', 'F'). There is no case sensitivity. The numbers have to be evaluated unsigned and hexadecimal (numeric basis is 16) always.

Example:
```
Command:  PROG:TEST:DIAG:FGR? 1, DELETE
Response: "ADDRESS", 3, #H0007, #H0000
```

5.2.7 Arbitrary Block Program Data

Any 8 bit characters (including extended ASCII code) may be transmitted in a message with the ARBITRARY BLOCK PROGRAM DATA data type. This data type is especially useful to send high amounts of data, and is recommended for binary data transmission. Send all files to be transferred from host system to BSE 1149.1 or Scan Commander in this format. Here, generally, variant DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA is used. This is a special type where the length of data is pre-defined.



#xyyyy<DAB>

<x>	length of length (number of digits)
<yyyy>	number of following data bytes (x-digits)
<DAB>	data bytes to be send

Example:
```
Command:  PROG:RES:DEF?
Response: #14Fail
```

On sending data in this format, no other commands should be given on the same line especially for BSE 1149.1 (send last character of the block data with EOI).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.3 MMEMemory Subsystem

The **MMEMory** subsystem contains all functions that are necessary for file handling. Several functions should be combined for the creation of own drivers on host side to reduce error pronity.

The application functions **CRG_TransmitAnyFile** and **CRG_ReceiveAnyFile** should be used for all file transfers.

The following table gives an overview of the keywords and their parameter forms:

	parameter form	notes
MMEMory		
:CATalog?		query only
:CDIRectory	[<directory_name>]	
:CLOSe		no query
:COPY	<file_source>, <file_destination>	no query
:DELete	<file_name>	no query
:FEED	<data_handle>	
:NAME	<file_name>	
:OPEN		no query
:MOVE	<file_source>, <file_destination>	no query

MMEMory:CATalog?

Syntax **MMEMory:CATalog?**

form	query only	
parameter	none	
response	<num_value>**,** <num_value> {**,**<file_entry>}	see description
<file_entry>	**"**<file_name>**,** <file_type>**,** <file_size>**"**	

Description: Use **MMEMory:CATalog** command to query the directory structure of the Scan Commander's/ BSE 11149.1's hard disk.

The first numeric value indicates the covered memory of the hard disk, and the second numeric value indicates the total free memory of the hard disk in bytes. Then as many file entries will be stated as files are contained in the directory. The file entry is of STRING PROGRAM DATA type and contains file name with extension, file type and size of the file in bytes separated by commas. As the file type cannot be drawn from directory entries under DOS, the file type remains empty, except in case of sub-directories. Should a sub-directory be contained as directory entry, then *DIR* is stated as file type.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MMEMory:CDIRectory



MMEMory:CDIRectory

form	query and command		
parameter	<directory_name>		STRING PROGRAM DATA (command only)
response	<directory_name>		STRING PROGRAM DATA (query only)
errors	-250	"mass storage error; directory not found"	given directory is not valid

Description: Use **MMEMory:CDIRectory** command to change the currently set directory to the one specified by <directory_name>. The parameter given here has to comply to the format of path indications under DOS. In case no directory is stated, then the CASCON-directory will be switched to. If **MMEMory:CDIRectory?** is used as query command, then the currently set directory is sent back.

MMEMory:CLOSe



MMEMory:CLOSe

form	command only		
parameter	none		STRING PROGRAM DATA (command only)
errors	-256	"file not found; <file_name> not open"	file to be closed is not open

Description: Use **MMEMory:CLOSe** to close the file specified with **NAME**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MMEMory:COPY

Syntax **MMEMory:COPY**

form	command only		
parameter	<file_source>,<file_destination>		STRING PROGRAM DATA
errors	-256	"file name not found; <file_name>"	Source file does not exist.
	-257	"file name error; <file_name> already exists"	Destination file already exists.
	-257	"file name error; <file_name> directory not found"	Name of destination file contains a directory not existing.
	-257	"file name error; <file_name> open"	Source or destination file is open.

Description: Use **MMEMory:COPY** command to copy an existing file to a new file. The given parameters are of STRING PROGRAM DATA type and have to comply to the format of path indications under DOS. The first string is the source file, and the second parameter is the destination file. Enter the complete path name for both source and destination file.

MMEMory:DELete

Syntax **MMEMory:DELete** <file_name>

form	command only		
parameter	<file_name>		STRING PROGRAM DATA
errors	-256	"file name not found; <file_name>"	File to be deleted does not exist.
	-257	"file name error; <file_name> open"	File to be deleted is open.

Description: Use **MMEMory:DELete** command to delete a file on the hard disk. The parameter <file_name> is a STRING PROGRAM DATA and indicates the file to be deleted. Enter the complete path name. It has to comply to the format of path indications under DOS.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MMEMory:FEED

Syntax **MMEMory:FEED**

form	command (write) and query (read)		
parameter	<data_handle>		DEFINITE ARBITRARY BLOCK PROGRAM DATA(writing only)
response	<data_handle>		DEFINITE ARBITRARY BLOCK PROGRAM DATA (reading only)
errors	-256	"file name not found; <file_name> not open"	File to be written/ read is not open.

Description: Use **MMEMory:FEED** command to transfers data into the file specified with **NAME** and opened with **OPEN** before. Here, the contents of a file already existing is overwritten. Use query command **MMEMory:FEED?** to send the contents of the file specified with **NAME** and opened with **OPEN** to the host controller. The transmission of the file contents(<data_handle>) is made as DEFINITE ARBITRARY BLOCK PROGRAM DATA. No further commands should be given on the same line after sending the data.

Each data block can be transferred with a feed command on writing to VSC 1149.1 with CASCON version 3.2 or higher. The blocks are added to the opened file. This increases transfer safety.

Example Writing File:
```
MMEM:NAM "HALLO.TXT";OPEN;FEED #16Hallo!
MMEM:CLOSE
```

Example Reading File:
```
Command:   MMEM:NAM "HALLO.TXT";OPEN;FEED?;CLOSE
Response:  #16Hallo!
```

An additional character 0A is transmitted together with EOI by SYSTEM CASCON on ARBITRARY BLOCK DATA Responses. This does not belong to the file data and is not contained in the length. It has to be overread on host side. If the VISA driver is used, this is ensured. Here block data without the additional character are accepted as well.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MMEMory:NAME

Syntax **MMEMory:NAME**

form	command and query	
parameter	<file_name>	STRING PROGRAM DATA (command only)
response	<file_name>	STRING PROGRAM DATA (query only)

Description: Use **MMEMory:NAME** command to specify a file for commands **OPEN** and **CLOSE**. Enter the complete path name. It has to comply to the format of path indications under DOS. If the path is not entered, then file is searched for or created automatically in the UUT directory. Use query form **MMEMory:NAME?** To have the name of the file just being specified send back.

MMEMory:OPEN

Syntax **MMEMory:OPEN**

form	command only		
parameter	none		
errors	-257	"file name error; <file_name> open"	File to be opened is already open.
	-257	"file name error; <file_name> could not open"	File was not found, creation of file failed.

Description: Use **MMEMory:OPEN** command to open the file specified with **NAME** before. In case the file to be opened does not exist, then it is created as new file if possible. There is no query form of this command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MMEMory:MOVE

Syntax **MMEMory:MOVE**

form	command only		
parameter	<file_source>,<file_destination>		STRING PROGRAM DATA
errors	-256	"file name not found; <file_name>"	Source file does not exist.
	-257	"file name error;<file_name> already exists"	Destination file already exists.
	-257	"file name error; <file_name> directory not found"	Name of destination file contains a directory not existing.
	-257	"file name error; <file_name> open"	Source or destination file is open.

Description: Use **MMEMory:MOVE** command to move an existing file to a new file. The first string is the source file, and the second parameter is the destination file. Enter the complete path name for both source and destination file. It has to comply to the format of path indications under DOS.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4 PROGram Subsystem

The commands will be described in detail in the following, whereas commands for same functions (context) are explained together. The first group of commands can be used generally (general), i.e. before program start as well as in HostRead state.

Use the second group only if no test has been started or after a run test has been finished (Idle state). If the program is in HostRead, these commands might generate error messages. No responses will be given to queries then.

The third group of commands can be used only if the program is in HostRead mode. The program has to contain respective CASLAN instructions for diagnosis and necessary variable settings to use diagnosis queries. These commands are not available for **PROGram:EXPLicit**:

The following table gives an overview of the keywords, their parameter forms and the context they are used in:

	parameter form	context	notes
PROGram			
:UUT			
:CATalog?		Idle	query only
[:SELected]			
:NAME	{<uutname>}	Idle	
:SNAMe	{"<uutsname>"}	Idle	
:DEFine	<uut>	Idle	no query
:DELete		Idle	no query
:OPTion			
:SNUMber		Idle	no query
:EXPLicit			
:DEFine	<uutname>, <uut>	Idle	no query
:DELete	<uutname>	Idle	no query
:ARCHive	<uut_archive_name>	Idle	no query
:UNPack	<uut_archive_name>	Idle	no query
:TEST			
:CATalog?		Idle	query only
[:SELected]			
:NAME	{<test>}	Idle	
:DEFine	<test>	Idle	no query
:DELete		Idle	no query
:STATe	{ **RUN** | **CONTinue** }	general	
:VALue?	<caslanvarname>	HostRead	query only
:EXECute	"<caslancmd>"	HostRead	no query

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram			
:TEST			
[:SELected]			
:DIAG			query only
:FAIL			
:GROup?	<groupfailnr>, { **KEEP** | **DELete** }	HostRead	query only
:GROup			
:COUNt?		HostRead	query only
:INDex?	<groupname>, <pinfailnr>, { **KEEP** | **DELete** }	HostRead	query only
:PIN?	<groupname>, <pinfailnr>, { **KEEP** | **DELete** }	HostRead	query only
:PIN?	"<icname>", <failnr>, { **KEEP** | **DELete** }	HostRead	query only
:PIN			
:COUNt?	"<icname>"	HostRead	query only
:FGCount?	<groupname>	HostRead	query only
:FGINdex?	<groupname>, <pinfailnr>, { **KEEP** | **DELete** }	HostRead	query only
:FPCount?	"<icname>"	HostRead	query only
:FGPin	<groupname>, <pinfailnr>, { **KEEP** | **DELete** }	HostRead	query only
:FGRoup?	<groupfailnr>, { **KEEP** | **DELete** }	HostRead	query only
:FPIN?	"<icname>", <failnr>, { **KEEP** | **DELete** }	HostRead	query only
:GPIN?	<groupname>, <gridx>, { **IN** | **OUT** | **CONTrol** | **ANY** }	HostRead	query only
:DIAG			
:GROup			
:PIN?	<groupname>, <gridx>, { **IN** | **OUT** | **CONTrol** | **ANY** }	HostRead	query only
:SIZE?	<groupname>	HostRead	query only
:GSIZe?	<groupname>	HostRead	query only

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram			
....:TEST			
........:EXPLizit			
:DEFine	<testname>, <batch>	Idle	no query
:DELete	<testname>	Idle	no query
:STATe	<testname>, { **RUN** }	Idle	
:ARCHive	<test_archive_name>	Idle	no query
:UNPack	<test_archive_name>	Idle	no query
:BATCh			
:CATalog?		Idle	query only
[:SELected]			
:NAME	{<batchname>}	Idle	
:DEFine	<batch>	Idle	no query
:DELete		Idle	no query
:STATe	{ **RUN** | **CONTinue** }	Idle	
:EXPLicit			
:DEFine	<batchname>, <batch>	Idle	no query
:DELete	<batchname>	Idle	no query
:STATe	<batchname>, { **RUN** }	Idle	
:ARCHive	<batchname>	Idle	no query
:RESult			
:DEFine?		Idle	query only
:MESSage			
:DEFine?		Idle	query only
:EXPLicit			
:EXEC	<cmd_string>	Idle	no query
:DIAlog			
:NAME?		HostRead	query only
:CATalog?		HostRead	query only
:VALue	<varname>, <value>	HostRead	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST[:SELected]:STATe

Syntax **PROGram:TEST[:SELected]:STATe**

context	general	
form	command and query	
parameter	**RUN** **CONTinue**	CHARACTER PROGRAM DATA; set only one parameter (command only)
response	**PASS**	faultless test program execution
	FAIL	faulty test program
	PAUSED	test program in HostRead
	NOT STARTED	not started
		type CHARACTER PROGRAM DATA; query only

Description: Use **PROGram:TEST[:SELected]:STATe** command to control execution of the test program selected with **PROGram:TEST[:SELected]:NAME** before. Query the state of test program execution with **PROGram:TEST[:SELected]:STATe?**.

If a program is run and is not in HostRead state, SYSTEM CASCON cannot generate responses. Then the state has to be query another way.

As alternative the application functions **CRG_RunTest**, **CRG_LaunchTest** and **CRG_CheckTestEnd** can be used.

PROGram:UUT:CATalog?

Syntax **PROGram:UUT:CATalog?**

context	Idle	
form	query only	
parameter	none	
response	<uutname>[,<uutname.>...]	CHARACTER PROGRAM DATA

Description: Use **PROGram:UUT:CATalog?** command to query all available UUT configurations (extension *UCF*; see CASCON documentation). Names will be sent back without extension. UUT names will be separated by commas.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:UUT[:SELected]:NAME / SNAME

Syntax **PROGram:UUT[:SELected]:NAME**
PROGram:UUT[:SELected]:SNAMe

context	Idle	
form	command and query	
parameter	<uutname>	CHARACTER PROGRAM DATA for "...**:NAME**"
	<uutsname>	STRING PROGRAM DATA for "...:**SNAMe**"
		(command only)
response	<uutname>	CHARACTER PROGRAM DATA for "...**:NAME**"
	<uutsname>	STRING PROGRAM DATA for "...**:SNAMe**"
	UUT_EMPTY	No valid UUT selected.
		(all responses are valid for query only)

Description: Use **PROGram:UUT[:SELected]:NAME** command to specify a UUT configuration to use **PROGram:UUT[:SELected]** commands.

Simultaneously, the UUT configuration given with <uutname> will be set as valid UUT in the Boundary Scan test and diagnosis software. All commands for the use of test programs, test protocols, diagnosis files, net lists and user libraries as well as commands for test execution will refer to this UUT.

If **PROGram:UUT[:SELected]:NAME?** is used, the name of the currently selected UUT configuration will be sent back. The given configuration name is of CHARACTER PROGRAM DATA type. In case no valid UUT configuration file has been selected, then string **UUT_EMPTY** is sent back.

Commands with **SNAMe** (string name) have the same effect, yet UUT names are given as string in double inverted commas (<uutsname>). Use these commands if the UUT name does not comply to SCPI requirements of CHARACTER PROGRAM DATA (e.g. starting with a number). Make sure that UUT name has maximally 8 characters and a name complying to DOS name conventions for file or directory names are adhered to.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:UUT[:SELected]:DEFine

Syntax **PROGram:UUT[:SELected]:DEFine**

context	Idle	
form	command only	
parameter	<uut_handle>	DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA

Description: Use **PROGram:UUT[:SELected]:DEFine** command to download and unpack UUT archive files selected with **PROGram:UUT[:SELected]:NAME** before. There is no query form of this command.

If a UUT with the same name exists, then this UUT will be overwritten during download. The transmitted UUT archive file is the parameter.

PROGram:UUT[:SELected]:DELete

Syntax **PROGram:UUT[:SELected]:DELete**

context	Idle		
form	command only		
parameter	none		
errors	-280	"program error;<file_name> UUT not found"	UUT was not found

Description: Use **PROGram:UUT[:SELected]:DELete** command to delete the UUT selected with **PROGram:UUT[:SELected]:NAME** before. All files of the selected UUT will be deleted.

PROGram:UUT[:SELected]:OPTion:SNUMber

Syntax **PROGram:UUT[:SELected]:OPTion:SNUMber**

context	Idle	
form	command and query	
parameter	<serialnumber>	STRING PROGRAM DATA (command only)
response	<serialnumber>	STRING PROGRAM DATA (command only)

Description: give a serial number for each UUT. This serial number will be indicated in the test protocol, in case the respective placeholder has been given in the run messages. Refer to the CASCON documentation for run message definition.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:UUT:EXPLicit:DEFine

Syntax **PROGram:UUT:EXPLicit:DEFine**

context	Idle	
form	command only	
parameter	**"**<uut_name>**",** <uut_handle>	<uut_name>: CHARACTER PROGRAM DATA <uut_handle>: DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA

Description: Use **PROGram:UUT:EXPLicit:DEFine** command to download and unpack the UUT archive file given with <uutname>. If a UUT with the same name exists then this UUT will be overwritten during download. The transmitted UUT archive file is the second parameter.

PROGram:UUT:EXPLicit:DELete

Syntax **PROGram:UUT:EXPLicit:DELete**

context	Idle		
form	command only		
parameter	**"**<uut_name>**"**		STRING PROGRAM DATA
errors	-280	"program error;<file_name> UUT not found"	UUT was not found

Description: Use **PROGram:UUT:EXPLicit:DELete** command to delete the UUT configuration selected with <uutname>. All files of this UUT will be deleted.

PROGram:UUT:EXPLicit:ARCHive

Syntax **PROGram:UUT:EXPLicit:ARCHive**

context	Idle	
form	command only	
parameter	**"**<uut_archive_name>**"**	STRING PROGRAM DATA

Description: This SCPI command is available with CASCON version 3.2 or higher. Use this command to prepare the UUT given with <uut_archive_name> for archive transfer . The command has the same function as **MMEM:NAME**. The archive then is transferred with commands **MMEM:OPEN** / **MMEM:FEED** / **MMEM:CLOSE** as any other file. It is unpacked with **PROG:UUT:EXPL:UNP**. Unpacking now is possible at the end after transferring several archives.

As alternative, the application function **CRG_TransmitArchive** can be used.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:UUT:EXPLicit:UNPack

Syntax **PROGram:UUT:EXPLicit:UNPack**

context	Idle	
form	command only	
parameter	**"**<uut_archive_name>**"**	STRING PROGRAM DATA

Description: This SCPI command is available with CASCON version 3.2 or higher. Use this command to unpack a transferred archive file named <uut_archive_name>. Use this command only to unpack uncompressed archives.

As alternative, the application function **CRG_TransmitArchive** can be used.

PROGram:TEST:CATalog?

Syntax **PROGram:TEST:CATalog?**

context	Idle	
form	query only	
parameter	none	
response	<uut_name> [**,**<uut_name> ...]	CHARACTER PROGRAM DATA

Description: Use **PROGram:TEST:CATalog?** command to query all available CASCON test program configurations (extension *TCF*, see CASCON documentation) assigned to the current UUT. Test program names are sent back without extension. Test names will be separated by commas.

PROGram:TEST[:SELected]:NAME

Syntax **PROGram:TEST[:SELected]:NAME**

context	Idle	
form	command and query	
parameter	<testname>	CHARACTER PROGRAM DATA (command only)
response	<testname> **TEST_EMPTY**	CHARACTER PROGRAM DATA (query only)

Description: Use **PROGram:TEST[:SELected]:NAME** command to select the CASLAN test program given with <testname> as valid test program. If it does not exist yet, the test will created. All following **PROGram:TEST[:SELected]** commands for transfer (**DEFine**), deletion (**DELete**) and test program run will refer to this test program.

If **PROGram:TEST[:SELected]:NAME?** is used, then the currently set test program is queried. In case no valid test program file is selected at the moment, then string **TEST_EMPTY** will be sent back.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST[:SELected]:DEFine

Syntax **PROGram:TEST[:SELected]:DEFine**

context	Idle	
form	command only	
parameter	<test_handle>	DEFINITE LENGTH ARBITRARY BLOCK PROGRAM DATA

Description: Use **PROGram:TEST[:SELected]:DEFine** command to download and unpack the test archive file selected with **PROGram:TEST[:SELected]:NAME** before. The send test archive file is assigned to the currently set UUT. If a test with the same name exists, then it will be overwritten during download. The transmitted test archive file is the parameter.

PROGram:TEST[:SELected]:DELete

Syntax **PROGram:TEST[:SELected]:DELete**

context	Idle		
form	command only		
parameter	none		
errors	-280	"program error;<file_name> test not found"	test program was not found

Description: Use **PROGram:TEST[:SELected]:DELete** command to delete the test program selected with **PROGram:TEST[:SELected]:NAME** before in the UUT directory. All files of the selected CASCON test will be deleted.

PROGram:TEST:EXPLicit:DEFine

Syntax **PROGram:TEST:EXPLicit:DEFine**

context	Idle	
form	command only	
parameter	**"**<test_name>**",** <test_handle>	<test_name>: CHARACTER PROGRAM DATA <test_handle>: DEFINITE LENGTH ARBITRARY BLOCK PROGRAM DATA

Description: Use **PROGram:TEST:EXPLicit:DEFine** command to download and unpack the test archive file given with <testname>. The send test archive file is assigned to the currently set UUT. If a test with the same name exists, then it will be overwritten during download. The transmitted test archive file is the second parameter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST:EXPLicit:ARCHive



PROGram:TEST:EXPLicit:ARCHive

context	Idle	
form	command only	
parameter	**"**<test_archive_name>**"**	STRING PROGRAM DATA

Description: This SCPI command is available with CASCON version 3.2 or higher. Use this command to prepare the test given with <test_archive_name> for archive transfer . The test is assigned to the currently set UUT. The command has the same function as **MMEM:NAME**. The archive then is transferred with commands **MMEM:OPEN** / **MMEM:FEED** / **MMEM:CLOSE** as any other file. It is unpacked with **PROG:UUT:EXPL:UNP**. Unpacking now is possible at the end after transferring several archives. However, all test archives have to belong to the same UUT.

As alternative, the application function **CRG_TransmitArchive** can be used.

PROGram:TEST:EXPLicit:UNPack



PROGram:TEST:EXPLicit:UNPack

context	Idle	
form	command only	
parameter	**"**<test_archive_name>**"**	STRING PROGRAM DATA

Description: This SCPI command is available with CASCON version 3.2 or higher. Use this command to unpack a transferred archive file named <test_archive_name>. Use this command only to unpack uncompressed archives.

As alternative, the application function **CRG_TransmitArchive** can be used.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST:EXPLicit:STATe

Syntax **PROGram:TEST:EXPLicit:STATe**

context	Idle		
form	command and query		
parameter	<test_name>, **RUN** <test_name>		CHARACTER PROGRAM DATA; (**RUN** for command only)
response	**PASS** **FAIL** **NOT STARTED**		test program execution faultless/ faulty/ not started CHARACTER PROGRAM DATA; query only
errors	-280	"Program error; <file_name> test not found"	test program was not found

Description: Use **PROGram:TEST:EXPLicit:STATe** command to control the execution of the test program given with <testname>. There are the following parameters:

Query the state of test program execution with **PROGram:TEST:EXPLicit:STATe?**.

As alternative the application functions **CRG_RunTest**, **CRG_LaunchTest** and **CRG_CheckTestEnd** can be used.

PROGram:TEST:EXPLicit:DELete

Syntax **PROGram:TEST:EXPLicit:DELete**

context	Idle		
form	command only		
parameter	<test_name>		CHARACTER PROGRAM DATA
errors	-280	"program error;<file_name> test not found"	test program was not found

Description: Use **PROGram:TEST:EXPLicit:DELete** command to delete the test selected with <test_name> in the UUT directory. All files of this test will be deleted.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:BATCh:CATalog?

Syntax **PROGram:BATCh:CATalog?**

context	Idle	
form	query only	
parameter	none	
response	<batch_name> [,<batch_name>...]	CHARACTER PROGRAM DATA

Description: Use **PROGram:BATCh:CATalog?** command to query all batch test programs (extension *CBT*, see CASCON documentation) assigned to the currently set UUT. Batch test program names are sent back without extension. The program names will be separated by commas.

PROGram:BATCh[:SELected]:NAME

Syntax **PROGram:BATCh[:SELected]:NAME**

context	Idle	
form	command and query	
parameter	<batch_name>	CHARACTER PROGRAM DATA (command only)
response	<batch_name> **BATCH_EMPTY**	CHARACTER PROGRAM DATA (command only)

Description: Use **PROGram:BATCh[:SELected]:NAME** command to select the batch test program given with <batch_name> as valid batch program. All following **PROGram:BATCh[:SELected]** commands for transfer (**DEFine**), deletion (**DELete**) and batch program run will refer to this batch program.

Use **PROGram:BATCh[:SELected]:NAME?** command to query the currently set batch program. In case no valid batch program file has been selected, then string **BATCH_EMPTY** is sent back.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:BATCh[:SELected]:DEFine



PROGram:BATCh[:SELected]:DEFine

context	Idle	
form	command only	
parameter	<batch_handle>	DEFINITE LENGTH ARBITRARY BLOCK PROGRAM DATA

Description: Use **PROGram:BATCh[:SELected]:DEFine** command to download the batch program selected with **PROGram:BATCh:SELected:NAME** before. The batch program will be written into the batch directory of the currently set UUT. If a batch program with the same name exists, then it will be overwritten during download. The sent batch program is the parameter.

PROGram:BATCh[:SELected]:DELete



PROGram:BATCh[:SELected]:DELete

context	Idle		
form	command only		
parameter	none		
errors	-280	"pogram error; <file_name> test not found"	CASCON batch program was not found.
	-280	"program error; <file_name> directory not found"	Directory of batch programs is not valid.
	-280	"program error; <file_name> could not delete"	CASCON batch program could not be deleted (e.g. it is write-protected).

Description: Use **PROGram:BATCh[:SELected]:DELete** command to delete the batch program selected with **PROGram:BATCh[:SELected]:NAME** before in the UUT directory.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:BATCh[:SELected]:STATe

Syntax **PROGram:BATCh[:SELected]:STATe**

context	Idle	
form	command and query	
parameter	**RUN** **CONTinue**	CHARACTER PROGRAM DATA; set only one parameter (command only)
response	**PASS**	faultless batch program execution
	FAIL	faulty batch program execution
	PAUSED	batch program in HostRead
	NOT STARTED	batch program not started
		type CHARACTER PROGRAM DATA; query only

Description: Use **PROGram:BATCh[:SELected]:STATe** command to control the execution of the batch program selected with **PROGram:BATCh[:SELected]:NAME** before. Set parameter CONTinue to continue program execution from HostRead.

Query the batch program execution state with **PROGram:BATCh[:SELected]:STATe?**

As alternative the application functions **CRG_RunTest**, **CRG_LaunchTest** and **CRG_CheckTestEnd** can be used.

PROGram:BATCh:EXPLicit:DEFine

Syntax **PROGram:BATCh:EXPLicit:DEFine**

context	Idle	
form	command only	
parameter	<batch_name>, <batch_handle>	<batch_name>: CHARACTER PROGRAM DATA <batch_handle>: DEFINITE LENGTH ARBITRARY BLOCK PROGRAM DATA

Description: Use **PROGram:BATCh:EXPLicit:DEFine** command to download the batch program given with <batch_name>. The sent batch program is assigned to currently set UUT. If a batch program with the same name exists, then it will be overwritten during download.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:BATCh:EXPLicit:DELete

Syntax **PROGram:BATCh:EXPLicit:DELete**

context	Idle		
form	command only		
parameter	<batch_name>		CHARACTER PROGRAM DATA
errors	-280	"pogram error; <batch_name> test not found"	CASCON batch program was not found.
	-280	"program error; <batch_name> directory not found"	Directory of batch programs is not valid.
	-280	"program error; <batch_name> could not delete"	CASCON batch program could not be deleted (e.g. it is write-protected).

Description: Use **PROGram:BATCh:EXPLicit:DELete** command to delete the batch program given with <batch_name> in the UUT directory.

PROGram:BATCh:EXPLicit:ARCHive

Syntax **PROGram:BATCh:EXPLicit:ARCHive**

context	Idle	
form	command only	
parameter	**"**<batch_name>**"**	STRING PROGRAM DATA

Description: This SCPI command is available with CASCON version 3.2 or higher. Use this command to prepare the CASCON batch file given with <batch_name> for archive transfer . The test is assigned to the currently set UUT. The command has the same function as **MMEM:NAME**. The batch file then is transferred with commands **MMEM:OPEN** / **MMEM:FEED** / **MMEM:CLOSE** as any other file.

As alternative, the application function **CRG_TransmitArchive** can be used.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:BATCh:EXPLicit:STATe



PROGram:BATCh:EXPLicit:STATe

context	Idle		
form	command and query		
parameter	<batch_name>, **RUN** <batch_name>		CHARACTER PROGRAM DATA; (**RUN** for command only)
response	**PASS** **FAIL** **NOT STARTED**		batch program execution faultless/ faulty/ not started CHARACTER PROGRAM DATA; query only
errors	-280	"program error; <batch_name> test not found"	test program was not found

Description: Use **PROGram:BATCh:EXPLicit:STATe** command to control the execution of the batch program given with <batch_name>.

Query the state of batch program execution with **PROGram:BATCh:EXPLicit:STATe?**.

As alternative, the functions **CRG_RunTest**, **CRG_LaunchTest** and **CRG_CheckTestEnd** can be used.

PROGram:RESult:DEFine?



PROGram:RESult:DEFine?

context	Idle		
form	query only		
parameter	none		
response	<result_handle>		DEFINITE LENGTH ARBITRARY BLOCK PROGRAM DATA
errors	-280	"program error;<file_name> test not found"	Result file was not found. (Upload)
	-280	program error;<file_name> not found"	Directory of result file is not valid.
	-280	Program error,<file_name> already exists"	Result file already exists. (Download)

Description: Use **PROGram:RESult:DEFine?** command to upload the result file of the CASCON test last run. The result file is transferred as response result file.

As alternative, **CRG_ReceiveProtocolFile** function can be used.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:MESSage:DEFine?

Syntax **PROGram:MESSage:DEFine?**

context	Idle	
form	query only	
parameter	none	
response	<result_handle>	STRING DATA

Description: Use **PROGram:MESSage:DEFine?** to display messages that usually are displayed in CASCON dialog windows. A response is given in any case. As these messages cannot be assigned to a SCPI command they are send the same way as test end or HostRead, i.e. VXI interrupt is possible. The messages are managed by SYSTEM CASCON in an own queue similar to the error queue.

PROGram:EXPLicit:EXEC

Syntax **PROGram:EXPLicit:EXEC**

context	Idle	
form	command only	
parameter	<cmd_string>	STRING DATA

Description: Use P**ROGram:EXPLicit:EXEC** to execute programs. Program name and necessary parameters are given in <cmd_string>. List DOS programs only that automatically return to CASCON after finishing. Unless further SCPI communication is not possible.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST[:SELected]:VALue?

Syntax **PROGram:TEST[:SELected]:VALue?**

context	HostRead	
form	query only	
parameter	<caslan_varname>	CHARACTER PROGRAM DATA
response	<value>	depends on the type of the CASLAN variable

Description: Use **PROGram:TEST[:SELected]:VALue?** command to query CASLAN variables. The responses are of the following types:

- STRING PROGRAM DATA if <caslan_varname> is a string variable
- NR1 NUMERIC RESPONSE DATA (decimal), if <caslan_varname> is a variable or group with less than 31 bits
- HEXADECIMAL NUMERIC RESPONSE DATA, if <caslan_varname> is a variable or group with more than 31 bits

PROGram:TEST[:SELected]:EXECute

Syntax **PROGram:TEST[:SELected]:EXECute**

context	HostRead	
form	command only	
parameter	<caslan_cmd>	STRING PROGRAM DATA; any command that can be used within the debugger may be used

Description: Use **PROGram:TEST[:SELected]:EXECute** command to execute a single CASLAN **DEBUG** command if a CASLAN program has been interrupted by **HOSTREAD**.

PROGram:DIALog:NAME?

Syntax **PROGram:DIALog:NAME?**

context	HostRead	
form	query only	
parameter	none	
response	**"**<dialog_name >**"**	STRING PROGRAM DATA

Description: Use **PROGram:DIALog:NAME?** command to query the name of the **HOSTREAD** dialog that interrupted a run CASLAN program.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:DIALog:CATalog?

Syntax **PROGram:DIALog:CATalog?**

context	HostRead	
form	query only	
parameter	none	
response	**"**<var_name>**", "**<value>**"**	STRING PROGRAM DATA (always in pairs)

Description: Use **PROGram:DIALog:CATalog?** command to query all variables of the dialogue opened by **HOSTREAD** for a running CASLAN program and belonging values. Responses are pairs of variable name and its contents. The data is always of STRING PROGRAM DATA type not depending on variable type.

PROGram:DIALog:VALue

Syntax **PROGram:DIALog:VALue**

context	HostRead	
form	command and query	
parameter	**"**<var_name>**", "**<value>**"** **"**<var_name>**"**	command (for setting) query (for query); all are of STRING PROGRAM DATA type
response	**"**<value>**"**	STRING PROGRAM DATA

Description: Use **PROGram:DIALog:VALue** command the defined setting of **HOSTREAD** dialog variable if the program is given in it. Query single variables in this state by means of **PROGram:DIALog:VALue?** command. Enter the variable name as parameter. Query only variables used in the **HOSTREAD** command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST[:SELected]:DIAG:FAIL:PIN:COUNt? / FPCount?



PROGram:TEST[:SELected]:DIAG:FAIL:PIN:COUNt?
PROGram:TEST[:SELected]:DIAG:FPCount?

context	HostRead	
form	query only	
parameter	**"**<ic_name>**"**	STRING PROGRAM DATA
response	**"**<failpin_count>**"**	NR1 NUMERIC RESPONSE DATA

Description: Use **PROGram:TEST[:SELected]:DIAG:FAIL:PIN:COUNt?** and **PROGram:TEST[:SELected]:DIAG:FPCount?** commands to query results of the equivalent CASLAN function **GetFailPinCount** if the program is in HostRead. The IC name is the parameter, "*" for all ICs or CASLAN string variable if a string variable was assigned in the program. Both types are identical and serve the query of the number of faulty pin list entries. No response or answer 0 is indicated in case of syntax errors.

Example:
```
Command:   PROG:TEST:DIAG:FPIN:COUNt? "IC4"
Response:  2
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST[:SELected]:DIAG: FAIL:PIN? / FPIN?

Syntax **PROGram:TEST[:SELected]:DIAG:FAIL:PIN?**
PROGram:TEST[:SELected]:DIAG:FPIN?

context	HostRead	
form	query only	
parameter	**"**<ic_name>**",** <failnr>**,** { **KEEP** | **DELETE** }	<ic_name>: STRING PROGRAM DATA; <failnr>: NR1 NUMERIC DATA
response	**"**<fic_name>**", "**<pin_name>**",** <fail>	"<fic_name>", "<pin_name>": STRING PROGRAM DATA, <fail>: NR1 NUMERIC DATA

Description: Use **PROGram:TEST[:SELected]:DIAG:FAIL:PIN?** and **PROGram:TEST[:SELected]:DIAG:FPIN?** commands to query the results of the equivalent CASLAN query function **GetFailPin** via remote control in HostRead. Both types are identical and serve the query of pin errors. Query the number of pin errors first. Response value <fail> might range from 0 to 3 (0 in case of syntax errors; for bits refer to the description of **GetFailPin** function).

- <ic_name>: IC name or "*" for all ICs or name of a CASLAN string variable if it has been assigned in the program
- <failnr>: variable or constant (group or register might be used as well), has to have value 0 .. <failcount>-1; <failcount> is provided via analogue call of **GetFailPinCount**.
- **KEEP** or **DELETE**: indicates whether the error has to be removed from the list. If the entry is deleted, this will effect the following calls of **GetFailPinCount** (response error number is reduced by 1) and **GetFailPin** (given error number has to be reduced by 1 if it is higher than the deleted error number).

Example:
```
Command:   PROG:TEST:DIAG:FPIN? "*", 0, KEEP
Response:  "IC4", "OE(#14)", 2
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST[:SELected]:DIAG: FAIL:GROup:COUNt? / FGCount?



PROGram:TEST[:SELected]:DIAG:FAIL:GROup:COUNt?
PROGram:TEST[:SELected]:DIAG:FGCount?

context	HostRead	
form	query only	
parameter	**"**<group_name>**"**	Name of a group or "*" for all groups (STRING PROGRAM DATA)
response	<group_failcount>**,** <pin_failcount>	Number of groups with errors and number of faulty pins (NR1 NUMERIC RESPONSE DATA)

Description: Use **PROGram:TEST[:SELected]:DIAG:FAIL:GROup:COUNt?** und **PROGram:TEST[:SELected]:DIAG:FGCount?** commands to query the results of the equivalent CASLAN function **GetFailGroupCount** via remote control in HostRead state. Both types are identical and query the number of faulty groups.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST[:SELected]:DIAG: FAIL:GROup? / FGRoup?

Syntax **PROGram:TEST[:SELected]:DIAG:FAIL:GROup?**
PROGram:TEST[:SELected]:DIAG:FGRoup?

context	HostRead	
form	query only	
parameter	<group_failnr>, { **KEEP** | **DELETE** }	<groupfailnr>: name of variable, group or register (STRING PROGRAM DATA) or constant; has to have a value within range 0..group_failcount-1 (NR1 NUMERIC DATA);
response	"<fgroup_name>", <fgroup_failpins>, <low_fail>, <high_fail>	Name of faulty group (STRING PROGRAM DATA), number of faulty pins (NR1 NUMERIC RESPONSE DATA), fail bits with low or high errors (HEXADECIMAL NUMERIC RESPONSE DATA)

Description: Use **PROGram:TEST[:SELected]:DIAG:FAIL:GROup?** and **PROGram:TEST[:SELected]:DIAG:FGRoup?** commands to query the results of the equivalent CASLAN query function **GetFailGroup** via remote control in HostRead. Both types are identical and serve the query of group errors . First query the number of group errors.

In case of syntax errors, 0 or empty strings o no responses are sent back for all elements. If there is no UUT error, 0 or empty strings are sent back for all elements.

Parameter <group_failnr> sets the error number for which the message should be sent. Use **KEEP** or **DELETE** to determine whether the error is to be removed from the list. If the entry is deleted, this will effect the following calls of **GetFailGroupCount** (response error number is reduced by 1) and **GetFailGroup** (given error number has to be reduced by 1 if it is higher than the deleted error number).

Response are the name of the faulty group <fgroup_name>, the number of faulty pins (**<**fgroup_failpins**>)** and faulty bits. Bits that have driven low instead of high at least once in the group are set to 1 in variable <low_fail>. The others are set to 0. Bits that have driven high instead of low at least once in the group are set to 1 in variable <high_fail>. The others are set to 0.

Example:
```
Command:   PROG:TEST:DIAG:FGR? 1, DELETE
Response:  "ADDRESS", 3, #H0007, #H0000
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST[:SELected]:DIAG: FAIL:GROup:INDex? / FGINdex?

Syntax **PROGram:TEST[:SELected]:DIAG:FAIL:GROup:INDex?**
PROGram:TEST[:SELected]:DIAG:FGINdex?

context	HostRead	
form	query only	
parameter	**"**<group_name>**"** <pin_failnr>**,** { **KEEP** | **DELETE** }	<group_name>: STRING PROGRAM DATA <pin_failnr>: STRING PROGRAM DATA (variable name) or NR1 NUMERIC DATA (constant)
response	**"**<fgroup_name>**",** <index>**,** <fail>	Name of faulty group (STRING PROGRAM DATA), group index of faulty entry (NR1 NUMERIC RESPONSE DATA), Fail value (0..3, type NR1 NUMERIC RESPONSE DATA)

Description: Use **PROGram:TEST[:SELected]:DIAG:FAIL:GROup:INDex?** and **PROGram:TEST[:SELected]:DIAG:FGINdex?** commands to query the results of the equivalent CASLAN function **GetFailGroupIndex** via remote control in HostRead state. Both types are identical and serve the query of a faulty element within a group. In case of syntax errors, 0 or empty strings o no responses are sent back for all elements.

Parameter <groupname> is a string constant or variable. Its value represents the name of group or "*" for all groups. Parameter <pin_failnr> is the name of a variable, group or register (type STRING PROGRAM DATA) or constant (type NR1 NUMERIC DATA). It sets the number of the error in the group. Use constants **KEEP** or **DELETE** to determine whether the error is to be removed from the list.

Example:
```
Command:   PROG:TEST:DIAG:FGIN? "ADDRESS", 0, KEEP
Response:  "ADDRESS", 3, 1
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST[:SELected]:DIAG:FAIL:GROup:PIN? / FGPin?

Syntax **PROGram:TEST[:SELected]:DIAG:FAIL:GROup:PIN?**
PROGram:TEST[:SELected]:DIAG:FGPin?

context	HostRead	
form	query only	
parameter	**"**<group_name>**"** <pin_failnr>**,** { **KEEP** | **DELETE** }	<group_name>: STRING PROGRAM DATA; <pin_failnr>: STRING PROGRAM DATA (variable name) or NR1 NUMERIC DATA (constant)
response	**"**<fic_name>**",** **"**<pin_name>**",** <fail>	**"**<fic_name>**", "**<pin_name>**"**: of type STRING PROGRAM DATA, <fail>: fail value (0..3, type NR1 NUMERIC RESPONSE DATA))

Description: Use **PROGram:TEST[:SELected]:DIAG:FAIL:GROup:PIN?** and **PROGram:TEST[:SELected]:DIAG:FGPin?** commands to query the results of the CASLAN query function **GetFailGroupPin** via remote control in HostRead state. Both types are identical and serve the query of a faulty pin within a faulty group.

Parameter <groupname> is a string constant or variable. Its value represents the name of group or "*" for all groups. Parameter <pin_failnr> is the name of a variable, group or register (type STRING PROGRAM DATA) or constant (type NR1 NUMERIC DATA) It sets the number of the error in the group. Use constants **KEEP** or **DELETE** to determine whether the error is to be removed from the list.

In case of syntax errors, 0, empty strings or no responses are returned for all elements. If there is no UUT error, 0 or empty strings are returned for all elements.

Example:
```
Command:   PROG:TEST:DIAG:FGP? "ADDRESS",0,KEEP
Response:  "IC4", "OE (#14)", 2
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST[:SELected]:DIAG: GROup:SIZe? / GSIZe?

Syntax **PROGram:TEST[:SELected]:DIAG:GROup:SIZe?**
PROGram:TEST[:SELected]:DIAG:GSIZe?

context	HostRead	
form	query only	
parameter	**"**<group_name>**"**	STRING PROGRAM DATA
response	<size>	NR1 NUMERIC RESPONSE DATA

Description: Use **PROGram:TEST[:SELected]:DIAG:GROup:SIZe?** and **PROGram:TEST[:SELected]:DIAG:GSIZe?** commands to query the results of CASLAN function **GetGroupSize** via remote control in HostRead. They detect the size of groups and do not depend on error lists.

Parameter is the group name as string parameter or CASLAN string variable if a string variable with the group name was assigned in the program.

Response is the number of elements in the selected group. Value 0 is sent back in case of faults (group does not exist).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PROGram:TEST[:SELected]:DIAG: GROup:PIN? / GPIN?

Syntax **PROGram:TEST[:SELected]:DIAG:GROup:PIN?**
PROGram:TEST[:SELected]:DIAG:GPIN?

context	HostRead	
form	query only	
parameter	**"**<group_name>**",** <goup_index>**,** <celltype>	<group_name>: STRING PROGRAM DATA <group_index>: STRING PROGRAM DATA or NR1 NUMERIC DATA***;*** <celltype>: **IN** | **OUT** | **CONTROL** | **ANY** (CHARACTER PROGRAM DATA)
response	**"**<ic_name >**",** **"**<pin_name>**"**	all STRING PROGRAM DATA

Description: Use **PROGram:TEST[:SELected]:DIAG:GROup:PIN?** and **PROGram:TEST[:SELected]:DIAG:GPIN?** commands to query the results of CASLAN function **GetGroupPin** via remote control in HostRead. They detect a group element's pin name and do not depend on error lists.

The following values are transferred:

- Group name as string parameter or CASLAN string variable if a string variable with the group name was assigned in the program.
- Index of the group element the pin name is sent for; name of a CASLAN variable containing the index (type STRING PROGRAM DATA) or constant (type NR1 NUMERIC DATA)
- A control parameter for the cell type:
  { **IN** | **OUT** | **CONTROL** | **ANY** }
  As group elements are cells and not pins, this serves the control whether the group element is to be input cell, output cell, control cell or any cell in the pin. Thus it is possible to determine what pin is sent back. (For instance, a cell can be input cell of a CE pin and control cell of a data bus pin at the same time).

Response is IC name and the pin (pin name followed by pin number in a string). Empty strings are sent back in case of syntax errors.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.5 OUTPut, TRIGger and CBUS Subsystems

These subsystems serve the connection of trigger lines. To realize needed trigger connections command sequences often have to be used in Scan Commander.

	parameter form	notes
OUTPut		
:TTLTrg	<n>	trigger line; (part of the header!)
:STATe	{ **ON** | **OFF** }	
:SOURce	{ **EXTernal** | **INTernal** }	
:EXTernal		
:SOURce	**INTernal**	
:STATe	{ **ON** | **OFF** }	
:RESet		no query

TRIGger		
:SOURce	{ **IMMediate | EXTernal** | **TTLTrg**<n> }	
:RESet		no query

	parameter form	notes
CBUS		
:SOURce	<line>, <sync_state>	Parameter: all CHARACTER PROGRAM DATA
:DESTination		
:TTLTrigger	<ttl_line>, <ttl_sync_state>	Parameter: all BINARY NUMERIC DATA, HEXADECIMAL NUMERIC DATA or NR1 NUMERIC DATA
:LOCL	<local_line>, <sync_state>	<local_line>: NR1 NUMERIC DATA and CHARACTER PROGRAM DATA <sync_state>: CHARACTER PROGRAM DATA
:LOCR	<local_line>, <sync_state>	see above

The BSTRG (input) C-bus signal can have exactly one source, either a TTL trigger signal, a signal of one of the two local busses or the input signal on the connecter at the Scan Commander's front panel.

The BSRDY C-bus signal can be distributed to several signals: to each TTL trigger signal on the backplane and to one signal each of LBL or LBR bus. Moreover it is fixed to the output signal BSRDY on the connecter at the Scan Commander's front panel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

OUTPut:TTLTrg<n>:SOURce

Syntax **OUTPut:TTLTrg<n>:SOURce**

form	command and query	
parameter	{ **INTernal** | **EXTernal** } none	CHARACTER PROGRAM DATA (command only) (query)
response	{ **INTernal** | **EXTernal** | **NONE** }	CHARACTER PROGRAM DATA

Description Use **OUTPut:TTLTrg<n>:SOURce** command to define signal sources for TTL trigger lines on VXI backplane. Select either internal signal source (signal BSRDY from C-bus of ASC 16) or an external trigger input (TRIG IN) at the front side. No connections is made here.

Query signal sources for VXI trigger line on the backplane by means of **OUTPut:TTLTrg<n>:SOURce?** command. Response **NONE** indicates that no source has been assigned (state after **OUTPut:RESet**).

OUTPut:TTLTrg<n>:STATe

Syntax **OUTPut:TTLTrg<n>:STATe**

form	command and query		
parameter	{ **ON** | **OFF** } none		CHARACTER PROGRAM DATA (command only) (query)
response	{ **ON** | **OFF** }		CHARACTER PROGRAM DATA
errors	-200	"Execution error; source not selected"	No trigger source has been selected.
	-200	"Execution error; state is ON"	New connection is possible only after disconnection of existing source.
	-200	"Execution error; state is OFF"	VXI backplane is already deactivated.
	-200	"Execution error; map = <error_code>"	Signal connections caused error <error_code>
	-200	"Execution error; unmap = <error_code>"	Signal disconnection caused error <error_code>

Description: Use **OUTPut:TTLTrg<n>:STATe** command to connect (parameter **ON**) or disconnect (parameter **OFF**) the trigger line selected with **OUTPut:TTLTrg<n>:SOURce**. Deactivating the trigger connection will make selected trigger source invalid as well.

Query the connection state of the selected trigger line in the trigger connection with **OUTPut:TTLTrg<n>:STATe?** command (**ON** or **OFF**).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

OUTPut:EXTernal:SOURce

Syntax **OUTPut:EXTernal:SOURce**

form	command and query	
parameter	**INTernal** none	CHARACTER PROGRAM DATA (command only) (query)
response	{ **INTernal** | **NONE** }	CHARACTER PROGRAM DATA

Description: Use **OUTPut:EXTernal:SOURce** command to define the signal source for the external trigger output. Signal BSRDY from C-Bus of ASC 16 (parameter **INTernal**) is the only possible signal source.

Use **OUTPut:EXTernal:SOURce?** command to query whether the external output is connected to a signal source (**INTernal**) or not (**NONE**).

OUTPut:EXTernal:STATe

Syntax **OUTPut:EXTernal:STATe**

form	command and query		
parameter	{ **ON** | **OFF** } none		CHARACTER PROGRAM DATA (command only) (query)
response	{ **ON** | **OF**F }		CHARACTER PROGRAM DATA
errors	-200	"Execution error; source not selected"	No trigger source has been selected.
	-200	"Execution error; state is ON"	New connection is possible only after disconnection of existing source.
	-200	"Execution error; state is OFF"	VXI backplane is already deactivated.
	-200	"Execution error; map = <error_code>"	Signal connections caused error <error_code>
	-200	"Execution error; unmap = <error_code>"	Signal disconnection caused error <error_code>

Description: Use **OUTPut:EXTernal:STATe** command to connect (parameter **ON**) the external trigger output to, or to disconnect it (parameter **OFF**) from, the selected trigger source. Deactivating the trigger connection will make selected trigger source invalid as well.

Query the connection state of the selected external trigger output with **OUTPut:EXTernal:STATe?** command (**ON** or **OFF**).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

OUTPut:RESet

Syntax **OUTPut:RESet**

form	command only	
parameter	none	

Description: Use **OUTPut:RESet** command to reset entire trigger connections. Thus, it is not necessary to query and deactivate every single input and output.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TRIGger:SOURce

Syntax **TRIGger:SOURce**

form	command and query	
parameter	{ **IMMediate** | **EXTernal** | **TTLTrg**<n> } none	CHARACTER PROGRAM DATA (command only) <n> stands for 0..7 (query)
response	{ **IMMediate** | **EXTernal** | **TTLTrg**<n> }	CHARACTER PROGRAM DATA

Description: Use **TRIGger:SOURce** command to define the Scan Commander's trigger source. Both trigger source selection and its connection as well as connection to signal BSTRG (internal input from C-bus of ASC 16) are realised by this command.

Parameter **EXTernal** stands for trigger input at the Scan Commander's front side(**TRG_IN**) and **TTLTrg<n>** stands for the trigger line on the VXI backplane selected as trigger source.

Parameter **IMMediate** indicates that the Scan Commander does not wait for a trigger pulse and effects the disconnection of BSTRG input from the trigger source.

Query the current trigger source for the Scan Commander with **TRIGger:SOURce?** command.

TRIGger:RESet

Syntax **TRIGger:RESet**

form	command only	
parameter	none	

Description: Use **TRIGger:RESet** command to reset entire trigger connections. Thus, it is not necessary to query and deactivate every single input and output. The command has the same effect as **OUTPut:RESet** command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CBUS:SOURce

Syntax **CBUS:SOURce**

form	command and query	
parameter1	**TRGIN**	signal **TRGIN** on the front panel connector
	TTLTx	<x> = 0..7; TTL trigger line on the backplane
	LBLx	<x> = 0....11; LocalBus left / right
	LBRx	One of these parameters permitted only!
parameter2	**SYNC | ASYNC**	indicates whether the signal is synchronized with the clock

Description: Use the **CBUS:SOURce** command to select the BSTRG source. All parameters are of type CHARACTER PROGRAM DATA. Indicate the synchronization type.

Query both parameters with **CBUS:SOURce?**

CBUS:DESTination:TTLTrigger

Syntax **CBUS:DESTination:TTLTrigger**

form	command and query	
parameter1	<ttl_line>	TTL trigger line BSRDY is connected to (8 bit number; one line per bit, bit 0 corresponds to TTLT0); Data can be given in BINARY NUMERIC DATA, HEXADECIMAL NUMERIC DATA or NR1 NUMERIC DATA format.
parameter2	<ttl_sync_stat>	Synchronization state (8 bit number, one trigger line per bit as above; bit = 1 stands for synchronous); Data can be given in BINARY NUMERIC DATA, HEXADECIMAL NUMERIC DATA or NR1 NUMERIC DATA format.

Description: Use the **CBUS:SOURce** command to select the BSTRG source. All parameters are of type ***Character Program Data***. Indicate the synchronization type.

Query both parameters with **CBUS:SOURce?**

Use **CBUS:DESTination:TTLTrigger** command to select the BSRDY output target. The signal can have several output targets: any TTL trigger line with independent synchronization (this command) and one signal each on LBL and LBR bus with own synchronization each.

Parameters usually are of type BINARY NUMERIC DATA, yet they can also be given in decimal or hexadecimal format. Using a target that already is used as source does not make sense but is allowed.

Query both parameters as BINARY NUMERIC RESPONSE DATA ***a*** with **CBUS:DESTination:TTLTrigger?**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CBUS:DESTination:LOCL / LOCR

Syntax **CBUS:DESTination:LOCL**
CBUS:DESTination:LOCR

form	command and query	
parameter1	**OFF** **0 .. 11**	localBusLine (LBL / LBR), BSRDY is connected to (line 0..11 or tristate);
parameter2	**SYNC | ASYNC**	indicates whether the signal is synchronized with the clock

Description: Use the **CBUS:DESTination:LOCL** or **...LOCR** command to select a target for BSRDY. The signal can have several output targets (see **CBUS:DEST:TTLT**), but only one each on the two local busses.

Unless **parameter1** is **OFF**, it is of type NR1 NUMERIC DATA.

Query the two parameters with **CBUS:DESTination:LOCL?** or **...LOCR?**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.6 SYSTem Subsystem

	parameter form	notes
SYSTem		
:ERRor?		query only
:CTYPe?		query only

SYSTem:ERRor?



SYSTem:ERRor?

form	query only	
parameter	none	
response	<err_nr>, "<err_text>"	<err_nr>: NR1 NUMERIC RESPONSE DATA, value 0, if there is no error); <err_text>: STRING PROGRAM DATA

Description: **SYSTem:ERRor?** command serves query purpose only. It indicates the next entry in the device's error message list. The error message begins with an error number. Negative error numbers are reserved for general SCPI errors. Positive ones indicate device specific errors. Then, the error message follows as separated by commas.

SYSTem:CTYPe?



SYSTem:CTYPe?

form	query only	
parameter	none	
response	<ident_string>	Four fields separated by comma (manufacturer, type, [serial no., version]); last two fields are optional; type ARBITRARY ASCII RESPONSE DATA

Description: **SYSTem:CTYPe?** command serves the query purpose only. It provides the identification of the firmware run on BSE interface card (not of CASCON version!) in case of BSE 1149.1 and the identification of CASREMOT program in case of Scan Commander VXI. The command is identical to query ***idn?**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7 STATus Subsystem

Subsystem **STATus** serves the control of the VXI interrupt/ event polling.

	parameter form	notes
STATus		
:QUESTionable		
:TEST		
[:EVENt]?		query only
:ENABle	<nr1>	

STATus:QUEStionable:TEST[:EVENt]?

Syntax **STATus:QUEStionable:TEST[:EVENt]?**

form	query only	
parameter	none	
response	<event_reg>	Contents of event register (type NR1 NUMERIC RESPONSE DATA)

Description: Use **STATus:QUEStionable:TEST[:EVENt]?** command to query the VXI interrupt status (VSC 1149.1 with MXI or embedded PC). The respective event can be triggered at end of test (bit 0 of the register) and on a request for HostRead (bit 1 of the register). Reading will clear the event register.

STATus:QUEStionable:TEST:ENABLE

Syntax **STATus:QUEStionable:TEST:ENABLE** <nr1>

form	command and query	
parameter	<int_mask> none	NR1 NUMERIC DATA (command only) (query only)
response	<int_mask>	Contents of event enable register (type NR1 NUMERIC RESPONSE DATA)

Description: Use **STATus:QUEStionable:TEST:ENABle** command to activate a VXI interrupt in case of respective events (see above for events and bits). Activate it before running the test program. It remains activated afterwards. To use this command bit 3 of standard state enable register has to be activated as well with ***SRE** command (instruction ***SRE 8**). Set the respective bits to 0 (activation state) to avoid triggering a VXI interrupt at test end or getting to HostRead.

Query activation with **STATus:QUEStionable:TEST:ENABle?** command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.1 Commands *SRE and *STB?

Despite commands in the subsystems, common GPIB standard instruction are used ("*" instructions). Instructions for the control of VXI interrupt/ event polling will be described here.



***SRE**

form	command and query	
parameter	<stat_ev_en_reg> none	NR1 NUMERIC DATA (command only) (query only)
response	<stat_ev_en_reg>	Contents of state event enable register (type NR1 NUMERIC RESPONSE DATA)

Description: Use ***SRE** instruction to activate SRQ in the standard state event register. Bit 3 is reserved for CASCON (sub-register TEST). All other bits are not used (even errors will not trigger a SRQ; use a respective command to query them).

After executing ***SRE** activate the respective bits in test enable register (**STATus:QUESTionable:TEST:ENABle**) as well so that a test program can trigger SRQ.

Query SRQ activation in standard state event register without clearing it with ***SRE?** instruction.



***STB?**

form	query only	
parameter	none	
response	<stand_stat_reg>	Contents of standard state register (type NR1 NUMERIC RESPONSE DATA)

Description: Use ***STB?** instruction for BSE 1149.1 to query registers that triggered SRQ (standard state registers). The query will clear the registers first. Prior to canceling a SRQ, read and thus delete the causing register (here register 3) with the respective instruction. Then read the standard state register by means of serial poll until value 0 is responded.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6 Programming Examples

The following programming examples are given:

- write file,
- read file,
- test execution.

6.1 Write File

This example describes the transmission of a short ASCII file from host system to BSE 1149.1/ Scan Commander.

name	*TEST_TR.TXT*
contents	`This is a short <CR><LF> test file in ASCII format.`

- Send file name and open given file
- Read file from host system memory
- Detect length of file: 47
- Detect length of length: 2
- Send file in DEFINITE LENGTH ARBITRARY BLOCK PROGRAM DATA format, send last character with EOI
- Close file

Example Write:

```
MMEM:NAME "TEST_TR.TXT";OPEN
MMEM:FEED #247 This is a short <CR><LF> test file in
               ASCII format.
MMEM:CLOSE
```

6.2 Read File

This example describes the transmission of a short ASCII file from to BSE 1149.1/ Scan Commander to host system.

name	*TEST_REC.TXT*
contents	`This is a short <CR><LF> test file in ASCII format.`

- Sending of reading request (file name selection, Open, reading request and file closing)

Example Read:

```
MMEMory:NAME "TEST.TXT";OPEN;FEED?;CLOSe
```

- Read interface of 2 byte ('#' and length of length)
  Response = #2
- Read interface of bytes indicated by length of length (2)
  Response = 47
- Read interface of length 47 indicated (file contents)
  Response = `This is a short<CR><LF>test file in ASCII format`
- Write read file contents into a host system file

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.3 Boundary Scan Test Execution

This example describes a possible test execution. First select the UUT and the test program. This is not necessary for every test start, as UUT and test programming settings remain valid until they are modified. Then define a protocol name and activate protocolling. Finally run the test and read test result and test protocol.

UUT name	OCTALS
test program	OCT_ATG

1. Select the desired UUT (send instruction):

   PROG:UUT:NAME OCTALS

   Alternatively, use the **SNAMe** command. It permits UUT names beginning with numbers:

   PROG:UUT:SNAM "OCTALS**"**

2. Select the desired test program (send instruction). Here the test name may begin with a letter only.

   PROG:TEST:NAME OCT_ATG

3. Send instruction for test run:

   PROG:TEST:STAT RUN

4. Wait for the test end event by the installed event handler (VXI interrupt).

5. Query the event type.

   STAT:QUES:TEST:EVENT?

   Response 1: End of test end (continue with 6)

   Response 2: Program is in HostRead (diagnosis and dialog commands can be used; then continue with 5)

6. Continue program, then proceed as after program start:

   PROG:TEST:STAT CONT

7. Send instruction to read test results

   PROG:SEL:STAT?

   This query can be send after break in HostRead as well, however this is not useful. Possible results are:

   PASS
   FAIL
   NOT_ STARTED
   PAUSED

8. Get test protocol (useful only in case of **FAIL**)

   PROG:RES:DEF?

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7 Remote Support in CASLAN

There is an instructions within CASCLAN provided especially for remote communication. The following requirement has to be met so that SYSTEM CASCON can communicate with a host system during test execution:

A running test can communicate with the host only if it is in HostRead state. The special CASLAN instruction **HOSTREAD** is provided for this. It is equivalent to the instruction **READNUMBERS**.

For detailed description of the instructions refer to the CASCON Reference Manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.1 HOSTREAD Instruction

Remote management of SYSTEM CASCON allows communication with the user application before test execution (selection and creation of tests and UUTs, test start) and after test execution (get test result, read test protocols, etc.). **HOSTREAD** instruction extents these feasibilities substantially as it interrupts the CASLAN program and provides communication to the host during the test. In this connection, the following actions are possible:

- query of **HOSTREAD** parameters
- setting of **HOSTREAD** parameters
- execution of CASLAN instructions as in CASCON debugger
- continuation of the CASLAN program

Thus the user application is able to:

- synchronize CASLAN tests with other tests
- cause another CASLAN test execution (if **HOSTREAD** parameters are used for program branching in CASLAN)
- set other test vectors (if **HOSTREAD** parameters are used to set vectors in CASLAN)

If **HOSTREAD** instruction is executed on a CASCON system without Remote DLL, then it is processed the same way as a **READNUMBERS** instructions.



HOSTREAD ('<comment>**'** [**,'**<labelN>**',** <cas_var>]... **);**

Description: **HOSTREAD** generates a CASLAN program interruption. Here, up to 16 pairs [<labelN>**'**, <cas_var>], representing one parameter each, can be specified.

<cas_var>	CASLAN dialog variable; use **PROGram:DIALog:CATalog?** to query all variables used in the instruction, use **PROGram:DIALog:VALue** for specific setting/ querying.
<comment>	comment (string); use **PROGram:DIALog:NAME?** command to query it from host side;
<labelN>	symbolic name for the respective variable

Example:

```
ld vAdr, 1245;
ld vDat, 245;
HOSTREAD ('Values for address and data bus',
          'Address', vAdr, 'Data', vDat);
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2 Manual Diagnosis Support

Manual diagnosis support is available for pins and groups. These commands effect that an internal list of actual deviations on pin or group level is created/ continued after each following DRSHIFT instruction. Query this list with special CASLAN instructions (see below) or by means of described SCPI commands from host side in HostRead state. It remains valid until entries are deleted by respective instructions. Access to diagnosis information requires the following:

- Activate diagnosis recording in the CASLAN program.
- Program steps have to be executed already. Response to a query of the diagnosis information within a HostRead instruction at program start will always be 0 or empty strings.

Use the following CASLAN instruction to activate or deactivate the diagnosis. Switching is possible at any place within the program and as often as needed.



MODE DIAGNOSTICS <diag_parameter>**;**

<diag_parameter>	**PINS** | **GROUPS** | { **PINS, GROUPS** }	activate
	NOPINS | **NOGROUPS** | { **NOPINS, NOGROUPS** }	deactivate

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8 Interface for the User Application

An user application can take remote control for SYSTEM CASCON by the VISA-compatible driver *SCC4HDRV.DLL*. This driver is available as LabWindows CVI instrument driver in form of a DLL and a function panel.

In contrast to normal VISA drivers, this driver itself does not use VISA functions and only provides a subset of the required VISA functionality in its entirity. A series of further functions, required because of formal compatibility to VISA and historical compatibility to CASCON 3.x, is implemented with empty contents and should not be used anymore owing to intricateness.

Thus, the driver can be used in the same way as a VISA driver, but it is subject to some restrictions (therefore the term VISA-*compatible* driver and not VISA driver):

- Functions of the VISA / IVI library, not defined in SCC4HDRV.H, must not be used together with the instrument handle of the SCC4HDRV.DLL init function.
- Many general VISA attributes do not exist anymore and they are not used by the VISA-compatible driver respectively.
- Other attributes are supplied together with the data of the Scan Commander because of compatibility, although the VXI Scan Commander is not accessed.
  DEVTYP_CAS32_SM (7) is provided as the attribute **CRG_ATTR_CAS_REMOTE_TYPE**.
- Trouble shooting is made analogous to VISA via the corresponding attributes. However, VISA error codes do not occur. The applied codes **SCC4_ERROR_xxx** are defined in *SCC4HDRV.H*.
- It is not possible to access to SYSTEM CASCON by means of the tools T&M Explorer, VIC etc. There is only access to the Scan Commander itself.
- General file transfers are made in the corresponding functions CRG_ReceiveAnyFile and CRG_TransmitAnyFile by simple copying without access to SYSTEM CASCON because of speed. Contrary, the archive transfer CRG_TransmitArchive and reading of protocol CRG_ReceiveProtocolFile are made via SCPI interpreter in CASCON by means of the MMEMory subsystem.

The interface of most important functions is explained in the following chapters by means of the declarations in "C". For detailed information refer to the help available on the single function panels within the driver.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.1 Function Tree

The function tree gives an overview of function available with the instrument. Each function is displayed in a separate panel where the function as well as its input and output values are described. Functions are listed in standardized groups. Some functions are available with every VISA driver and have general parameters.

All functions have an error code (Int32) as function value. Codes >= 0 indicate correct execution.

CASCON Remote VXI

Init / Close

- Initialize
- Initialize with Options
- Close

Application Functions

- Run Test
- Launch Test
- Check Test End
- Receive Protocol File
- Receive Any File
- Transmit Any File
- Transmit Archive
- Transmit Update

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Configuration Functions

Set / Get / Check Attribute

Set Attribute

Set Attribute ViInt32
Set Attribute ViReal
Set Attribute ViString
Set Attribute ViBoolean
Set Attribute ViSession

Get Attribute

Get Attribute ViInt32
Get Attribute ViReal
Get Attribute ViString
Get Attribute ViBoolean
Get Attribute ViSession

Check Attribute

Check Attribute ViInt32
Check Attribute ViReal
Check Attribute ViString
Check Attribute ViBoolean
Check Attribute ViSession

Data Functions

Write Instrument Data
Read Instrument Data
Query Instrument Data
Transmit File Data Block
Receive File Data Block

Utility Functions

Reset
Self Test
Revision Query
Error Query
Error Message

Error Info

Get Error Info
Clear Error Info

Locking

Lock Session
Unlock Session

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.2 Init / Close Functions

In this section, the following open/ close functions are described:

- Init
- Init With Options
- Close

The **CRG_init** or **CRG_InitWithOptions** function respectively and the **CRG_close** function form a frame, where all the other functions have to be embedded in.

8.2.1 Init

Use function **CRG_init** for initialization. It provides the instrument handle needed for all other driver functions. The initialization string depends on the type of remote control. Basically, all VISA drivers are initialized this way.

```
ViStatus CRG_init (ViRsrc ResourceName,
                   ViBoolean IDQuery,
                   ViBoolean ResetDevice,
                   ViSession *Inst_Handle);
```

For <ResourceName>, the following systax is valid (depending on the interface):

interface	syntax
GPIB (BSE 1149.1)	**GPIB**[board]::<primary address>
VXI - MXI (SCC)	**VXI**[board]::<logical address>::**INSTR**
VXI with GPIB Slot-0 Controller (SCC)	**GPIB-VXI**[board]::<logical address>::**INSTR**

<ResourceName>, <IDQuery> and <Reset Device> are ignored by the VISA-compatible driver.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.2.2 Init With Options

```
ViStatus CRG_InitWithOptions (
                    ViRsrc ResourceName,
                    ViBoolean IDQuery,
                    ViBoolean ResetDevice,
                    ViString  OptionString,
                    ViSession *Inst_Handle);
```

<ResourceName>	Input	resource name as in **CRG_Init**
<IDQuery>	Input	with or without ***IDN?** query
<ResetDevice>	Input	with or without ***RST**
<OptionString>	Input	timeout for test execution given in seconds
<pInst_Handle>	Input/ Output	pointer for provided instrument handle

The **CRG_InitWithOptions** function is called by **CRG_Init**. The options string then contains only the item "Simulate = ...".

The function executes the following activities:

- creates a new instrument driver and initializes driver attributes
- opens the session with the device set in the resource name
- if **IDQuery** is set, identification is queried.
- if <ResetDevice> is set, the ***RST** command is executed.
- sends the initialization command with options in the options string
- provides the session handle on the resource

8.2.3 Close

```
ViStatus CRG_close (ViSession Inst_Handle);
```

Use the **CRG_close** function to quit the instrument driver. The handle on the driver then is invalid.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.3 Application Functions

The following application functions are provided:

- Run Test
- Launch Test
- Check Test End
- Receive Protocol File
- Receive Any File
- Transmit Any File
- Transmit Archive
- Transmit Update

The application functions can be considered as high-level functions. They combine some typical low-level actions and facilitate programming in this way.

8.3.1 Run Test

The **CRG_RunTest** function combines some processes belonging to test execution. The test is started, and the event is waited for (event triggering via VXI interrupt for VXI Scan Commander). The test result is queried also in case of event end of test.



Due to different mechanisms of detecting the test end used by the different instruments, you should use the application functions described here instead of the example shown in the SCPI description for test start.

```
ViStatus CRG_RunTest (ViSession Inst_Handle,
                      ViInt32 RunType,
                      ViChar* pResult,
                      ViInt32 TimeOut,
                      ViInt32* pStopReason);
```

<Inst_Handle>	Input	instrument handle provided by init
<RunType>	Input	**0**: Run test **1**: Continue test (after **HOSTREAD**) **2**: Run batch
<pResult>	Input/ Output	pointer to the test result (provided only in case of <pStopReason> test end)
<TimeOut>	Input	timeout for test execution in seconds (leaving the function, the timeout of **CRG_ReadInstrData** is reset to 5 secs)
<pStopReason>	Input/ Output	reason for stop (bit 0 set: test end, bit 1 set: HostRead state)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.3.2 Launch Test

The **CRG_LaunchTest** function and the following **CRG_CheckTestEnd** function divide test execution into two steps: starting the test and checking if the test is ready.

CRG_LaunchTest function runs test execution, but does not wait for test end. Thus, other activities are possible while the test is executed.



Note that no other communication functions can be called during test execution, as SYSTEM CASCON does not react to SCPI commands in this state.

```
ViStatus CRG_LaunchTest (ViSession Inst_Handle,
                         ViInt32 RunType);
```

<Inst_Handle>	Input	instrument handle provided by init
<RunType>	Input	**0**: Run test **1**: Continue test (after **HOSTREAD**) **2**: Run batch

8.3.3 Check Test End

Use the **CRG_CheckTestEnd** function to check whether a test was finished. It then provides test result and stop reason.

SYSTEM CASCON only reacts to further SCPI commands if <pStopReason> is unequal to zero.

```
ViStatus CRG_CheckTestEnd (ViSession Inst_Handle,
                           ViChar* pResult,
                           ViInt32* pStopReason);
```

<Inst_Handle>	Input	instrument handle provided by init
<pResult>	Input/ Output	pointer to the test result (provided only in case of <pStopReason> test end)
<pStopReason>	Input/ Output	stop reason (bit 0=1: test end, bit 1=1: HostRead, 0: test not finished)

8.3.4 Receive Protocol File

Use the **CRG_ReceiveProtocolFile** function to transfer the protocol file after test execution. The protocol is provided only after the test was finished (not in HostRead state!).

```
ViStatus CRG_ReceiveProtocolFile (
              ViSession InstHandle,
              ViChar* pName,
              void pCallback (ViInt32, ViInt32),
              ViInt32* pReceived);
```

<Inst_Handle>	Input	instrument handle provided by init
<pName>	Input	name of the file the protocol is stored in on the hard disk
<pCallback>	Input	Pointer to a function called at the beginning and after every block. Current and maximum number of blocks is provided. Thus a progress bar can be created. If it is ZERO, no function is called. The first call is before transfer of the first block, i.e. the total number of blocks is known then.
<pReceived>	Input/ Output	protocol length

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.3.5 Receive Any File

The function **CRG_ReceiveAnyFile** copies the file <pSrcName> to the file <pDstName>. This function is obsolete. It is contained because of compatibility (task of the function: copy of a file from the computer running CASCON to the computer running the user application – it is the same computer in a system with CASCON 4.x).

```
ViStatus CRG_ReceiveAnyFile (
              ViSession Inst_Handle,
              ViChar* pDstName,
              ViChar* pSrcName,
              void pCallback (ViInt32, ViInt32),
              ViInt32* pReceived);
```

<Inst_Handle>	Input	instrument handle provided by init
<pDstName>	Input	name of the destination file
<pSrcName>	Input	name of the source file
<pCallback>	Input	is ignored, but contained because of compatibility
<pReceived>	Input/ Output	size of the received file

8.3.6 Transmit Any File

The function **CRG_ TransmitAnyFile** copies the file <pSrcName> to the file <pDstName>. This function is obsolete. It is contained because of compatibility (task of the function: copy of a file from the computer running the user application to the computer running CASCON – it is the same computer in a system with CASCON 4.x).

```
ViStatus CRG_TransmitAnyFile (
              ViSession Inst_Handle,
              ViChar* pDstName,
              ViChar* pSrcName,
              ViBoolean OneBlock,
              void pCallback (ViInt32, ViInt32),
              ViInt32* pTransmitted);
```

<Inst_Handle>	Input	instrument handle provided by init
<pDstName>	Input	name of the destination file
<pSrcName>	Input	name of the source file
<OneBlock>	Input	is ignored, but contained because of compatibility
<pCallback>	Input	is ignored, but contained because of compatibility
<pTransmitted>	Input/ Output	size of the transferred file

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.3.7 Transmit Archive

Use the **CRG_TransmitArchive** function to transfer a UUT or test archive or a CASCON batch file to SYSTEM CASCON. Option Test/ Debug enabled is not checked, i.e. a UUT transferred this way might not be selectable. SCPI commands are used for archive transfer. It is then transferred similar to a any other file (exception: **PROG:[UUT/TEST/BATC]:EXPL:ARCH** is sent instead of **MMEM:NAME**). Then the archive is unpacked with **PROG:[UUT/TEST]:EXPL:UNP**.

```
ViStatus CRG_TransmitArchive (
              ViSession Inst_Handle,
              ViChar* pDstName,
              ViChar* pSrcName,
              ViBoolean OneBlock,
              void pCallback (ViInt32, ViInt32),
              ViInt32* pTransmitted);
```

<Inst_Handle>	Input	instrument handle provided by init
<pDstName>	Input	name of the file is created under Scan Commander/ BSE 1149.1 (without path and extension)
<pSrcName>	Input	name of the file on the host (without path and extension)
<OneBlock>	Input	TRUE: transfer as for file, every block is transferred with **FEED** command, then unpacking is called FALSE: archive is ARBITRARY BLOCK DATA parameter of SCPI command **PROG:[TEST/UUT]:EXPL:DEF**
<pCallback>	Input	pointer to a function called at the beginning and after every block. Current and maximum number of blocks is provided. Thus a progress bar can be created. If it is ZERO, no function is called. The first call is before transfer of the first block, i.e. the total number of blocks is known then.
<pTransmitted>	Input/ Output	size of the transferred file

8.3.8 Transmit Update

The function **CRG_TransmitUpdate** was used to transfer a CASCON update to the Scan Commander in SYSTEM CASCON 3.x



This function must not be used anymore as no corresponding CASCON update files <pSrcName> are provided for CASCON 4.x!

```
ViStatus CRG_TransmitUpdate (
              ViSession Inst_Handle,
              ViChar* pSrcName,
              void pCallback (ViInt32, ViInt32),
              ViInt32* pTransmitted);
```

<Inst_Handle>	Input	instrument handle provided by init
<pSrcName>	Input	name of the file on the host
<pCallback>	Input	pointer to a function called at the beginning and after every block.
<pTransmitted>	Input/ Output	size of the transferred file

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.4 Data Functions

The following data functions are provided:

- Write Instrument Data
- Read Instrument Data
- Query Instrument Data
- Transmit File Data
- Receive File Data

The data functions can be considered as low-level functions which allow to send commands and receive responses according to SCPI guidelines.

8.4.1 Write Instrument Data

Use the **CRG_WriteInstrData** function to write any SCPI command to Scan Commander/ BSE 1149.1. Thus, any defined SCPI command can be executed. An end bit is send per default. Reset **CRG_ATTR_SEND_END_EN** in case no end bit should be send. This is valid until next modification. It does not influence the execution of complex functions.

```
ViStatus CRG_WriteInstrData (
                ViSession Inst_Handle,
                ViConstStr WriteBuffer);
```

<Inst_Handle>	Input	instrument handle provided by init
<WriteBuffer>	Input	command to be written (without 0x0A)

8.4.2 Read Instrument Data

Use the **CRG_ReadInstrData** function to read the response from Scan Commander/ BSE 1149.1. The set number of bytes is read, maximally until EOI. Make sure the buffer is large enough.

```
ViStatus CRG_ReadInstrData (
                ViSession Inst_Handle,
                ViInt32 NumBytes,
                ViChar* pRdBuffer,
                ViInt32* pBytesRead);
```

<Inst_Handle>	Input	instrument handle provided by init
<NumBytes>	Input	maximum number of bytes to be read
<pRdBuffer>	Input/ Output	pointer to read responses
<pBytesRead>	Input/ Output	number of bytes read

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.4.3 Query Instrument Data

The **CRG_QueryInstrData** function combines **CRG_WriteInstrData** and **CRG_ReadInstrData** functions. Make sure that the sent string requires a response, otherwise an timeout error occurs.

```
ViStatus CRG_QueryInstrData (
                ViSession Inst_Handle,
                ViChar* pQuestion,
                ViInt32 NumBytes,
                ViChar* pRdBuffer,
                ViInt32* pBytesRead);
```

<Inst_Handle>	Input	instrument handle provided by init
<pQuestion>	Input	SCPI string to be sent
<NumBytes>	Input	maximum number of bytes to be read
<pRdBuffer>	Input/ Output	pointer to read responses
<pBytesRead>	Input/ Output	number of bytes read

8.4.4 Transmit File Data

Use the **CRG_TransmitFileData** function to send any data to the respective instrument. Attribute **CRG_ATTR_SEND_END_EN** determines whether EOI is sent or not. It has to be set before by the function **CRG_SetAttribute**. All introducing commands have to be sent before. The ARBITRARY BLOCK DATA header has to be generated by the user.

```
ViStatus CRG_TransmitFileData (
    ViSession Int_Handle,
    ViChar *pBuffer,
    ViInt32 BufSize,
    ViInt32* pTransmitted);
```

<Inst_Handle>	Input	instrument handle provided by init
<NumBytes>	Input	maximum number of bytes to be read
<pBuffer>	Input	pointer to data
<pBytesRead>	Input/ Output	number of bytes read

8.4.5 Receive File Data

Use the **CRG_ReceiveFileData** function to read any data from the respective instrument. All introducing commands have to be sent before. The ARBITRARY BLOCK DATA header has to be interpreted by the user. The set number of bytes is read, maximally until the end EOI.

```
ViStatus CRG_ReceiveFileData (
    ViSession Int_Handle,
    ViChar *pBuffer,
    ViInt32 BufSize,
    ViInt32* pTransmitted);
```

<Inst_Handle>	Input	instrument handle provided by init
<NumBytes>	Input	maximum number of bytes to be read
<pBuffer>	Input	pointer to data
<pBytesRead>	Input/ Output	number of bytes read

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.5 Utility Functions

Main utility functions are described in the following sections:

- Error Message
- Error Query
- Revision Query

Functions not mentioned here exist but they are irrelevant.

8.5.1 Error Message

The **CRG_error_message** function does not take an action on the VXI/ GPIB interface. It translates the VISA error code into a string that can be read by the user. A buffer of 256 bytes has to be available for this.

```
ViStatus CRG_error_message (
              ViSession Inst_Handle,
              ViStatus  ErrorCode,
              ViChar ErrorText[256],
```

<Inst_Handle>	Input	instrument handle provided by init
<ErrorCode>	Input	error code as output by all driver functions
<ErrorText>	Input/ Output	error string

8.5.2 Error Query

The **CRG_error_query** function is a special version of **CRG_QueryInstrData** and queries errors occurred. Every call deletes an entry in the error queue of SYSTEM CASCON. The error is stated with number and text.

```
ViStatus CRG_error_query (
              ViSession Inst_Handle,
              ViInt32* pErrCode,
              ViChar* pErrText,
```

<Inst_Handle>	Input	instrument handle provided by init
<pErrCode>	Input/ Output	pointer to error code
<pRdBuffer>	Input/ Output	pointer to read error text

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.5.3 Revision Query

The **CRG_revision_query** function is a special version of **CRG_QueryInstrData** and queries the identification. It is separated into driver revision and firmware revision. Make sure buffers with 256 bytes each are available.

```
ViStatus CRG_revision_query (
    ViSession Inst_Handle,
    ViChar Driver_Revision[],
    ViChar Firmware_Revision[]);
```

<Inst_Handle>	Input	instrument handle provided by init
<Driver_Revision>	Input/ Output	pointer to driver revision
<Firmware_Revision>	Input/ Output	pointer to firmware revision

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.6 Configuration Functions

Configuration functions to check, set and read attributes are available. Each function exists for the following data types:

- ViSession:
- ViBoolean:
- ViString
- ViInt32
- ViReal64

Enter the respective data type for '...' in the following declarations. Default channel name is **VI_NULL**.

The Attribute_ID can be found in the table of section 8.7: "Attributes".

Setting Attributes



```
ViStatus CRG_SetAttribute ( ViSession Instrument_Handle,
                            ViConstString Channel_Name,
                            ViAttr Attribute_ID,
                            ... Attribute_Value );
```

Reading Attributes



```
ViStatus CRG_GetAttribute ( ViSession Instrument_Handle,
                            ViConstString Channel_Name,
                            ViAttr Attribute_ID,
                            ... *Attribute_Value);
```

For string attributes, add the array size between Attribute_ID and *Attribute_Value.

Checking Attributes



```
ViStatus CRG_CheckAttribute ( ViSession Instrument_Handle,
                              ViConstString Channel_Name,
                              ViAttr Attribute_ID,
                              ... Attribute_Value[] );
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.7 Attributes

The driver has a few attributes the user should know. Some are used internally only.

Most important, used attributes are described here only. All other attributes are also contained for compatibility reasons, however there is no use for them. Range Tables do not exist.



ViBoolean **CRG_ATTR_SEND_END_EN**

This attribute determines whether EOI is set for the last byte for **CRG_WriteInstrData** and **CRG_TransmitFileData** functions.

All more complex functions set EOI as needed and afterwards restore the former setting. For instance, the sending with all functions based on **CRG_QueryInstrData** is done with EOI no matter how this attribute is set.



ViInt32 **CRG_ATTR_OPC_TIMEOUT**

This attribute contains the timeout time for CASCON test execution. As CASLAN tests or batch programs might take longer, a separate configurable timeout is available for them. Before running the test, the timeout of the **CRG_ReadInstrData** function is set to this value. After finishing the test it is reset to the default 5 seconds.



ViInt32 **CRG_ATTR_CAS_REMOTE_TYPE**

This attribute defines the instrument type. It is set automatically by init, and it cannot be modified by the user. Depending on the instrument type, communication applies different mechanisms. For instance, all SCPI-compliant accesses are done via memory access on VSC 1149.1. All other instruments use functions such as **viPrintf**/ **viScanf**.

All driver functions query this attribute and, thus, consider the instrument type.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following attributes are provided:

Attribute_ID	R/W	type	used by SCC4HDRV.DLL
User Options			
CRG_ATTR_RANGE_CHECK	RW	ViBoolean	no
CRG_ATTR_QUERY_INSTR_STATUS	RW	ViBoolean	no
CRG_ATTR_CACHE	RW	ViBoolean	no
CRG_ATTR_SIMULATE	RW	ViBoolean	no
CRG_ATTR_RECORD_COERCIONS	RW	ViBoolean	no
Instrument Capabilities	RO		
CRG_ATTR_NUM_CHANNELS	RO	ViInt32	no
CRG_ATTR_SPECIFIC_PREFIX	RO	ViString	yes
Version Info			
CRG_ATTR_DRIVER_MAJOR_VERSION	RO	ViInt32	yes
CRG_ATTR_DRIVER_MINOR_VERSION	RO	ViInt32	yes
CRG_ATTR_DRIVER_REVISION	RO	ViString	yes
CRG_ATTR_ENGINE_MAJOR_VERSION	RO	ViInt32	no
CRG_ATTR_ENGINE_MINOR_VERSION	RO	ViInt32	no
CRG_ATTR_ENGINE_REVISION	RO	ViString	no
Error Info			
CRG_ATTR_PRIMARY_ERROR	RW	ViInt32	yes
CRG_ATTR_SECONDARY_ERROR	RW	ViInt32	yes
CRG_ATTR_ERROR_ELABORATION	RW	ViString	yes
Advanced Session I/O			
CRG_ATTR_VISA_RM_SESSION	RO	ViSession	no
CRG_ATTR_IO_SESSION	RO	ViSession	no
CRG_ATTR_DEFER_UPDATE	RW	ViBoolean	no
CRG_ATTR_RETURN_DEFERRED_VALUES	RW	ViBoolean	no
Instrument-specific Attributes			
CRG_ATTR_ID_QUERY_RESPONSE	RO	ViString	yes
CRG_ATTR_VXI_MANF_ID	RO	ViInt32	no
CRG_ATTR_VXI_MODEL_CODE	RO	ViInt32	no
CRG_ATTR_SEND_END_EN	RW	ViInt32	yes
CRG_ATTR_CAS_REMOTE_TYPE	RW	ViInt32	yes
CRG_ATTR_DELAY_MS	RW	ViInt32	no (always 0)

The not used attributes are still contained because of compatibility.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9 Comparison to Scan Commander 3

The software architecture of the VXI Scan Commander VSC 1149.1, version 4, is changed essentially in contrast to Scan Commander version 3. The main difference is, that SYSTEM CASCON runs on the Scan Commander itself in Scan Commander 3, whereas SYSTEM CASCON runs on the host PC in Scan Commander 4.

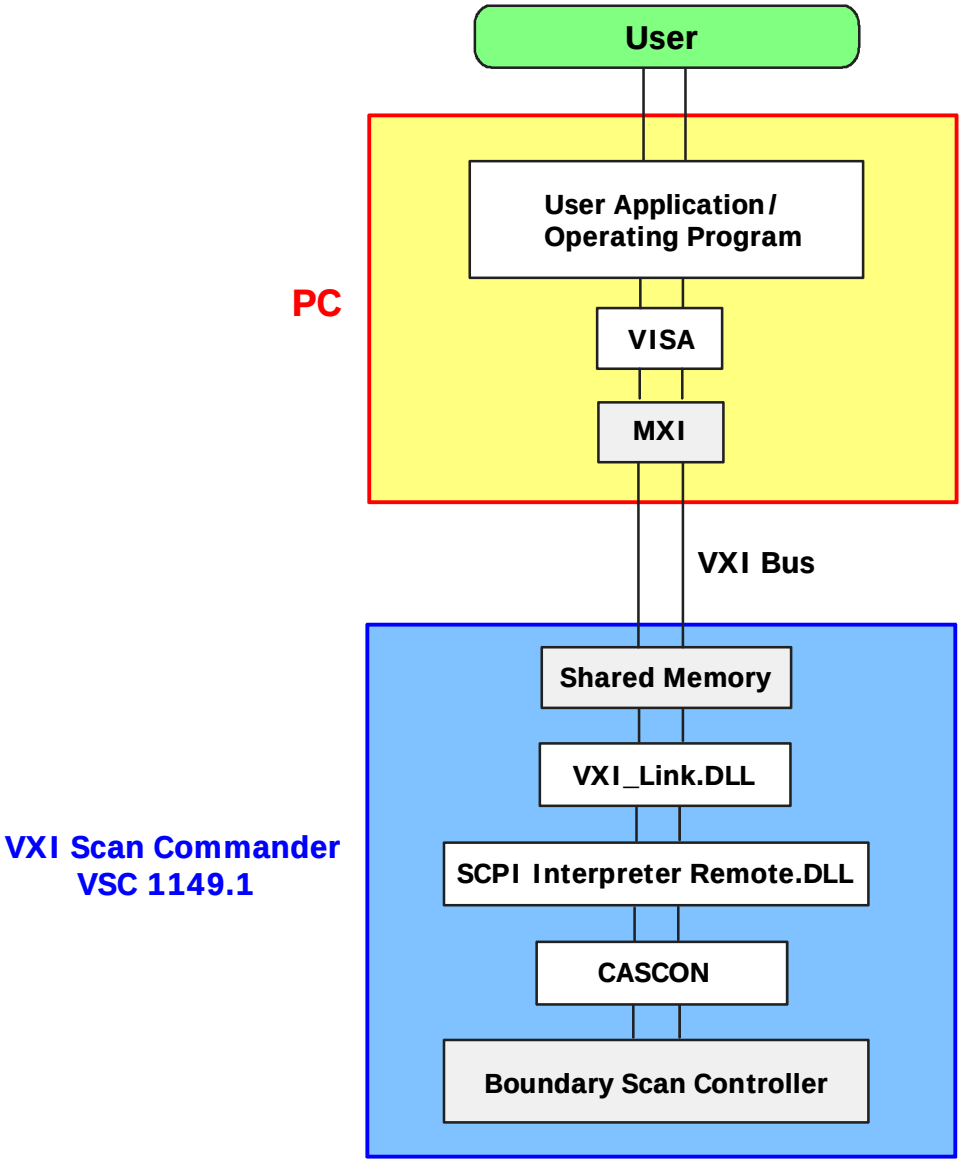


Figure 9-1: Position of Scan Commander 3 in the Entire System

Nevertheless, a maximum compatibility to the user application interface was achieved. However, some differences result, mainly influencing effectivity.

- VSC 1149.1 and SYSTEM CASCON
- Compatibility of the Interface to the User Application
- Modifications in the Operating Program

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.1 VSC 1149.1 and SYSTEM CASCON

Since SYSTEM CASCON runs on the host PC in Scan Commander 4, many tasks can be solved simplier and more directly.

- There is no necessity of an user application or the operating program *SCC4HOST.EXE*. The terminal emulation is not available anymore.
- The Scan Commander can be selected and set as controller in SYSTEM CASCON.
- Tests, generated during the test program development, can be executed immediately by SYSTEM CASCON, an archive transfer is obsolete.
- UUT- / test archives from other PCs can be unpacked directly by SYSTEM CASCON.
- A file transfer is obsolete, since all files exist on the host PC.
- An user application and the operating program *SCC4HOST.EXE* do not control the Scan Commander, but SYSTEM CASCON on the host PC. In this way, the remote control can also be used together with other BScan controllers.
- Only the controller driver in CASCON can communicate to the Scan Commander, since merely the protocol translator together with the driver DLL run on the Scan Commander.
- As alternative to the VISA-compatible driver, SYSTEM CASCON can also be controlled via DDE. However, less functions are provided.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.2 Compatibility of the Interface to the User Application

An user application, made for Scan Commander 3, can be adapted to Scan Commander 4 without much effort, since the interface of *SCC4HDRV.DLL* is identical to the VISA-DLL *SCC3HDRV.DLL*. However, it is not accessed to VISA.

If only functions defined/ applied in the header file *SCC3HDRV.H* were used by the VISA / IVI functions in the user application, it is sufficient to translate them again while using the new *SCC4HRRV.LIB* and *SCC4HDRV.H*. All used definitions and functions prescribed by the "former" VISA DLL are available, but many of them are not used. The VISA data types are defined in the file *SCC4HDRV.H* too.

If other functions of the VISA / IVI library with the instrument handle of the Scan Commander as parameter were used, their calls have to be eliminated.

Modified Functions

- General file transfer is made in the corresponding functions by simple copying (because of speed). The ARBITRARY BLOCK DATA transfer was kept for archive transfer and protocol reading only. Of course, it still works in the basis functions.
- The function **CRG_TransmitUpdate** served the transfer of a CASCON update on the Scan Commander in CASCON 3.x.



The function **CRG_TransmitUpdate** must not be used anymore, since no corresponding CASCON update files <pSrcName> are provided for CASCON 4.x.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.3 Modifications in the Operating Program

The operating program *SCC4HOST.EXE* was adapted to the requirements of the new user application interface. But the user interface was kept to the greatest possible extent. The terminal emulation is not available anymore, since SYSTEM CASCON can be used directly.

A separate program *SCCUpdate.EXE* is available for Scan Commander firmware update (refer to section 2.5: Scan Commander Firmware Update").

The features and possibilities of file transfer were kept and will be described in the following section, but they are obsolete now..

9.3.1 Transferring Archives



The following proceeding is still possible in the same way as in CASCON 3.x, but should not be used anymore. Since CASCON 4.x, archives can be transferred by simple copying within SYSTEM CASCON on the host computer.

To transfer UUTs and tests to VSC 1149.1/ BSE 1149.1, respective archives are sent. This is possible from any section. Use menu point *File* for file transfer. Necessary activities are similar to those of *Create UUT* in the test section of previous operating programs.

Preparation

First of all, generate a UUT archive/ test archive for a repair or test station with the CASCON GALAXY PS software. Make sure that the UUT has been enabled for testing (option Test/ Repair enable). Locate your archive. Contrary to previous program version, archives can be selected not depending on the host system's paths.

Transfer

After opening menu *File/ UUT Archive...*, *File/ Test archive...* or *File/ Batch program...* a multi-file dialogue is opened for archive selection.

Select the respective archive not depending on CASCON directory structures, or from a network (multi-file dialogue). The extension is fixed. Double click the archive file to load it into the lower collection field. Archives can be collected from different paths and drives. Pressing ***OK*** the following activities are executed:

- Creation of an empty UUT/ test named after the archive
- Transfer of the archive
- Activating and extracting the archive by CASCON

Transfer progress is indicated by the bar in the status line. After extracting the file existing UUTs and tests are queried again and then updated in the selection boxes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

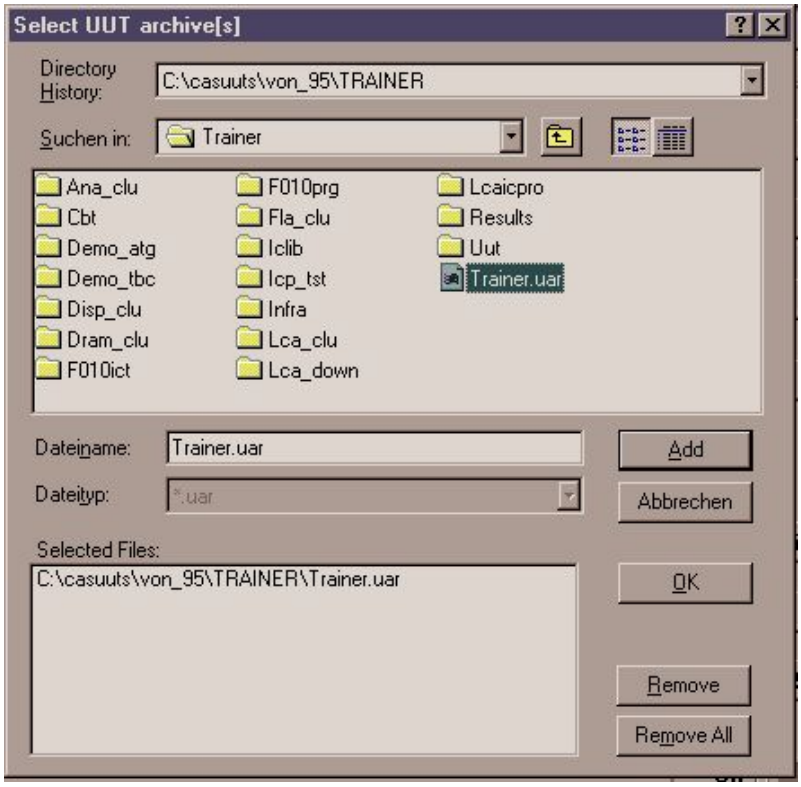


Figure 9-2: Transfer Dialog

For details about the use and creation of archives refer to the CASCON documentation (CASCON User Interface).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.3.2 File Transfer Section

The following proceeding is still possible in the same way as in CASCON 3.x, but should not be used anymore. Since CASCON 4.x, files can be transferred by simple copying within SYSTEM CASCON on the host computer.

The file transfer section allows files to be copied, deleted and moved between VSC 1149.1 and the host. All management mechanisms of CASCON are avoided. Therefore, be careful in using this section. Drive C: on VSC 1149.1 can be protected under *Options/Settings*.

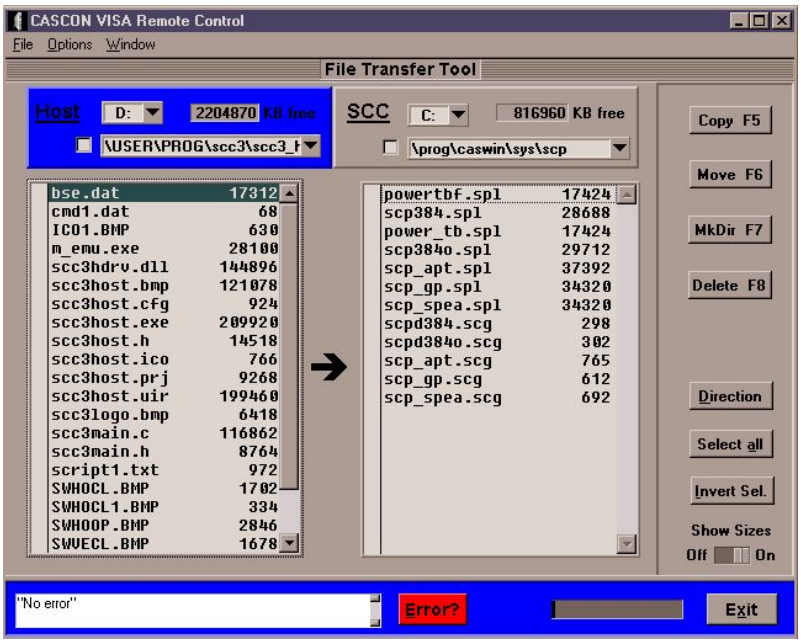


Figure 9-3: File Transfer Panel

Transfer Direction

The panel is divided into two section. The left window contains the files on the host system, the right one those of the VSC 1149.1. Switch between both sides using the tab key or change the transfer direction using the ***Direction*** button. The source side is indicated in blue in the panel's header.

Directory Change and Selection

The change of directories and drives is done in the respective fields in the upper part of the panel. Clicking the arrow in the respective field opens a list of available items. Directories are given relatively. Select ".." to switch to the upper level. Scrollable lists contain files only.

Drive selection is limited to local drives or mapped network drives on the host.

Files are selected by clicking in the first column or by marking the current line with the space bar. Repeated clicking undoes the action. Selected files are indicated by a check mark in the first column. Manual selection does not require that the selection side is the active side (source).

Pressing ***SelectAll*** all files on the source side are selected. Button ***Invert Sel.*** undoes the selection there.

Directories are selected for deletion by marking the checkbox on the left side of the directory field in the upper part of the panel.

Press ***ShowSizes*** to display file sizes next to the name. The empty space is indicated in the respective field in the upper part.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Activities

The following actions are available in the File Transfer Tool:

- F5 Copy (both directions)
- F6 Move (both directions)
- F7 Create directory (both sides) – After pressing the button or F7 a dialog is opened where the name is entered.
- F8 Delete files or empty directory (both sides)

Directories to be deleted have to be empty and may not contain sub-directories.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix A: Physical Specifications

dimensions	C-size VXI bus modules , 1 slot
front display elements	2-color-LED (green / red)
front connector XH3 (68-pin)	testbus IEEE 1149.1 PORT 1and 2
	parallel I/O interface
	C-bus interface
	SCP-bus interface
front socket AUX1	PS2 keyboard
front socket AUX2	monitor VGA
front socket COM	Ethernet (in preparation)

Maximum current input for various supply voltages:

+5 VDC	7A
+12 VDC	200 mA
-12 VDC	200 mA
-5.2 VDC	150 mA
-2 VDC	200 mA
-2 VDC	200 mA

emission	FFC Class A
operating conditions	
temperature	5°C to 55°C
relative humidity	0% to 95%, no condensation
storage Conditions	
temperature	-40°C to 70°C
relative humidity	0% to 100%, no condensation

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix B: Front Panel Elements

A two-color LED is provided in the upper right corner of the front panel to indicate operating modes of VSC 1149.1.

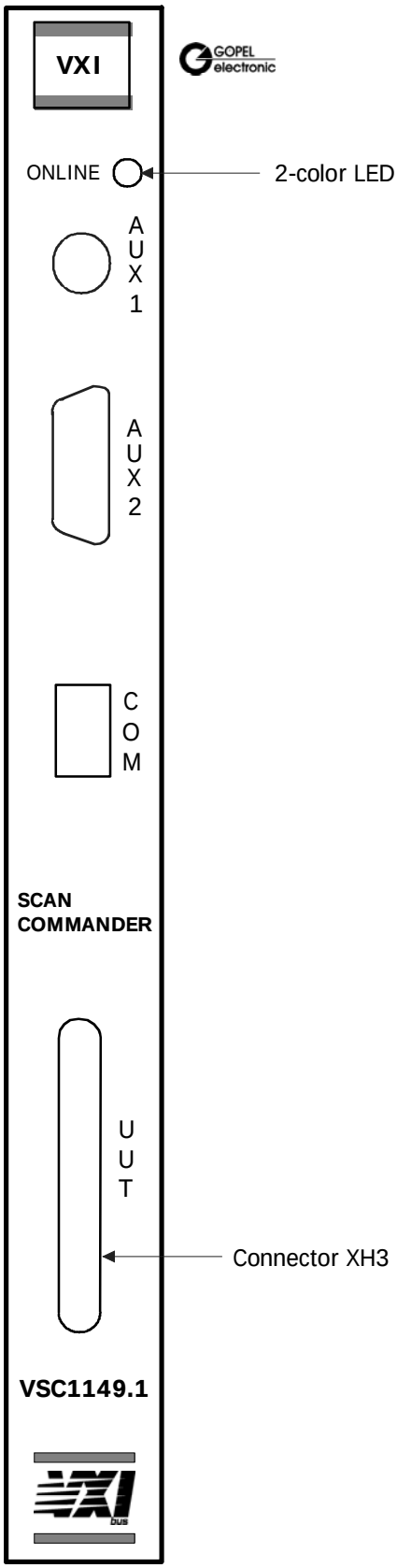
- The LED is inactive after power on or reset or during a file transfer just started.
- After successful initialization the LED lights up green.
- The operating software of VSC 1149.1 switches the LED to green during data transfer without wait states.
- The operating software of VSC 1149.1 switches the LED to red during wait states within data transfer.

LED	state	error
INACTIVE	after power on after reset at beginning of data transfer	
GREEN	after successful initialization after successful data transfer	
Switch RED/ GREEN	during data transfer (length of the red phase depends on the amount of data and processing speed)	
RED	wait state after data transfer start	error on accessing the VXI bus (static red LED)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

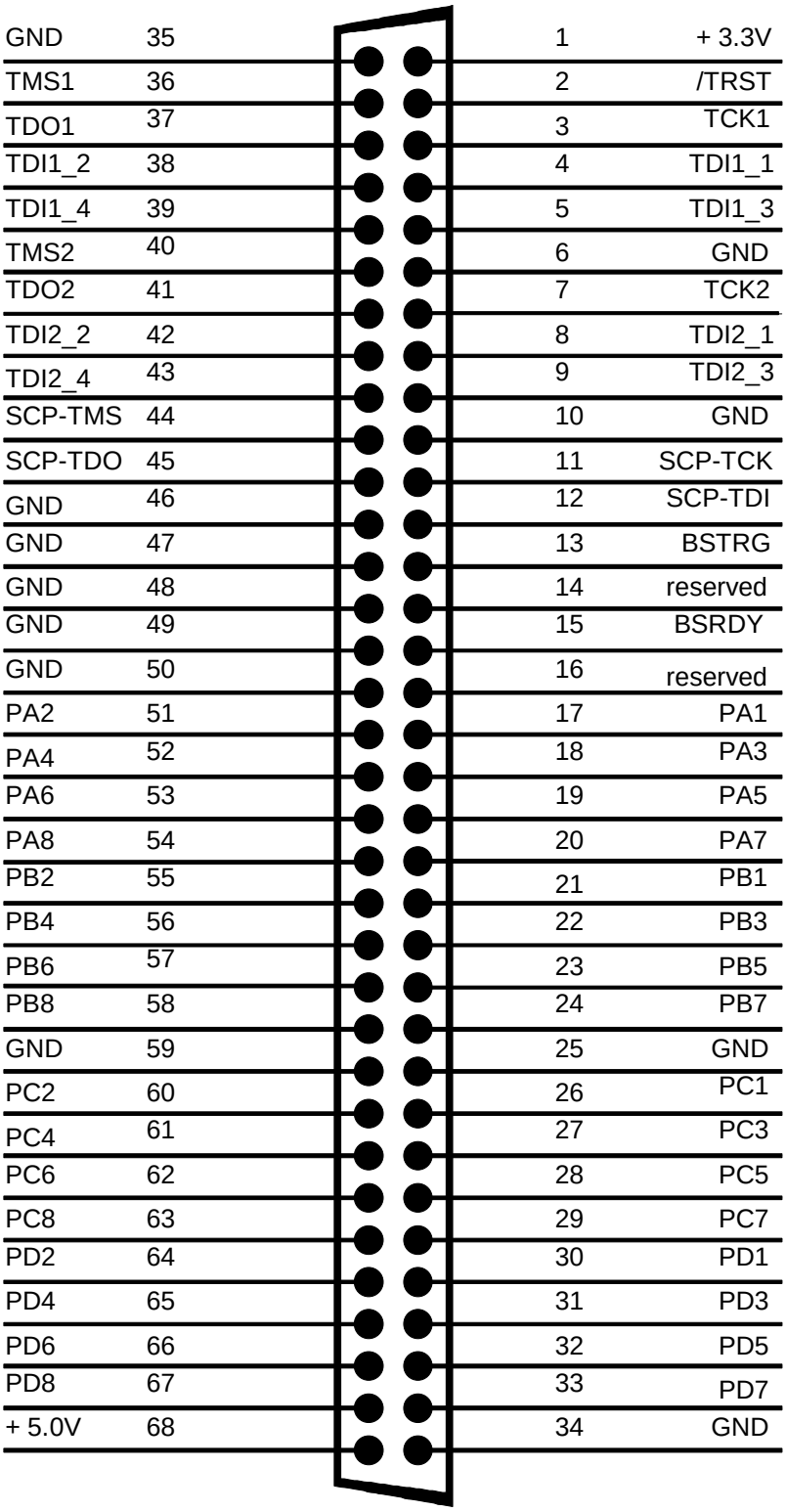
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix C: Connectors



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

XH3: User Interface



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A

application functions 8-6
ARBITRARY ASCII RESPONSE DATA 5-2
ARBITRARY BLOCK PROGRAM DATA **5-3**
archive transfer . 5-16, 5-19, 5-24
archive work 4-5
attributes 8-15
  checking 8-14
  reading 8-14
  setting 8-14

B

batch
  delete 5-22, 5-24
  download 5-22
  list 5-21
  program name 5-21
  select 5-21
  start 5-23
BINARY NUMERIC DATA ... **5-3**

C

CASCON Remote Operating Program 4-1
CASLAN variable 5-27
C-bus control 3-6
C-bus programming 3-6
compatibility to Scan Commander 3 9-3
configuration file 5-13
configuration functions 8-14
connectors 9-1

D

data functions 8-10
DEBUG command 5-27
DELETE 5-30
directory change 5-5
directory structure 5-4
download 5-15, 5-18

F

features 1-3
file
  close 5-5
  copy 5-6
  delete 5-6
  move 5-9
  open 5-8
  receive 5-8
  send 5-7
file entries 5-4
file handling 5-4
file transfer 4-5
firmware update 2-7
front panel elements 9-1
function tree 8-2
functions
  application ~ 8-6
  data ~ 8-10
  open/ close ~ 8-4
  utility ~ 8-12

G

group 5-30
  faulty 5-31, 5-32
group errors 5-32

H

HEXADECIMAL NUMERIC DATA **5-3**
HostRead 5-28
HOSTREAD instruction 7-2

I

IC name 5-30

K

KEEP 5-30

L

logic address 2-1
long form 5-1

M

model types 1-2

N

NR1 NUMERIC DATA **5-2**

O

open/ close functions 8-4

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

P

physical specifications 9-1
pin errors 5-30
programming examples 6-1

R

Resource Manager 2-2
result file 5-25

S

scope of supply 1-2
SCPI standard 5-1
serial number 5-15
short form 5-1
slot 2-2
static configuration 2-2
subsystem 5-1
system controller 2-2

T

Test
  delete 5-20
  download 5-18
  new 5-17
  select 5-17
  state 5-20
Test program 5-17
  start 5-13
tree structure 5-1

U

user interface 9-2
utility functions 8-12
UUT
  delete 5-15, 5-16
  download 5-15
  select 5-14
  serial number 5-15
UUT archive file 5-16
UUT configuration 5-14
UUT name 5-14

V

variable 5-28, 5-30
VISA-compatible driver 8-1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP