

Agilent TS-5000 Functional Test System

System Software User's Guide

Notices

© Agilent Technologies, Inc. 1999 - 2011

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

Documentation History

All Editions and Updates of this manual and their creation date are listed below. The first Edition of the manual is Edition 1. The Edition number increments by 1 whenever the manual is revised. Updates, which are issued between Editions, contain replacement pages to correct or add additional information to the current Edition of the manual. Whenever a new Edition is created, it will contain all of the Update information for the previous Edition. Each new Edition or Update also includes a revised copy of this documentation history page.

Edition 1 (E8770-90000) (System Software User's Guide) July 1999

Edition 2 (E8770-90023) September 2000

Edition 3 (E8770-90033) January 2004

Edition 4 (E8770-90033) January 2006

Edition 4 (E8770-90033) April 2010

Edition 5 (E8770-90033) May 2011

Manual Part Number

E8770-90033

Printed in Malaysia

Agilent Technologies Microwave Products (Malaysia) Sdn. Bhd.
Bayan Lepas Free Industrial Zone
11900 Penang, Malaysia

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or sub-contract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use,

duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

Caution

A **Caution** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **Caution** notice until the indicated conditions are fully understood and met.



WARNING

A **WARNING** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a **WARNING** notice until the indicated conditions are fully understood and met.

Safety Summary

The following general safety precautions must be observed during all phases of operation of this system. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the system. Agilent Technologies, Inc. assumes no liability for the customer's failure to comply with these requirements.

General

This product is provided with a protective earth terminal. The protective features of this product may be impaired if it is used in a manner not specified in the operation instructions.



DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE

Do not operate the system in the presence of flammable gases or flames.

If the equipment in this system is used in a manner not specified by Agilent Technologies, the protection provided by the equipment may be impaired.

Cleaning Instructions

Clean the system cabinet using a soft cloth dampened in water.



DO NOT REMOVE ANY SYSTEM COVER

Operating personnel must not remove system covers. Component replacement and internal adjustments must be made only by qualified service personnel. Equipment that appears damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel.

Environmental Conditions

Unless otherwise noted in the specifications, this system is intended for indoor use in an installation category II, pollution degree 2 environment. It is designed to operate at a maximum relative humidity of 80% and at altitudes of up to 2000 meters. Refer to the specifications tables for the ac mains voltage requirements and ambient operating temperature range.

Before applying power

Verify that all safety precautions are taken. Note the external markings described in [“Safety Symbols and Regulatory Markings”](#) on page 4.

Ground the System

To minimize shock hazard, the instrument chassis and cover must be connected to an electrical protective earth ground. The instrument must be connected to the ac power mains through a grounded power cable, with the ground wire firmly connected to an electrical ground (safety ground) at the power outlet. Any interruption of the protective (grounding) conductor or disconnection of the protective earth terminal will cause a potential shock hazard that could result in personal injury.



WARNING

The power cable ground wire must be connected to an electrical ground (safety ground) at the power outlet. Any interruption of the protective grounding will cause a potential shock hazard that could result in personal injury.

Fuses

Use only fuses with the required rated current, voltage, and specified type (normal blow, time delay). Do not use repaired fuses or short-circuited fuse holders. To do so could cause a shock or fire hazard.

Operator Safety Information

MODULE CONNECTORS AND TEST SIGNAL CABLES CONNECTED TO THEM

CANNOT BE OPERATOR ACCESSIBLE: Cables and connectors are considered inaccessible if a tool (e.g., screwdriver, wrench, socket, etc.) or a key (equipment in a locked cabinet) is required to gain access to them. Additionally, the operator cannot have access to a conductive surface connected to any cable conductor (High, Low or Guard).

ASSURE THE EQUIPMENT UNDER TEST HAS ADEQUATE INSULATION BETWEEN THE CABLE CONNECTIONS AND ANY OPERATOR-ACCESSIBLE PARTS (DOORS, COVERS, PANELS, SHIELDS, CASES, CABINETS, ETC.): Verify

there are multiple and sufficient protective means (rated for the voltages you are applying) to assure the operator will NOT come into contact with any energized conductor even if one of the protective means fails to work as intended. For example, the inner side of a case, cabinet, door, cover or panel can be covered with an insulating material as well as routing the test cables to the module's front panel connectors through non-conductive, flexible conduit such as that used in electrical power distribution.

Safety Symbols and Regulatory Markings

Symbols and markings on the system, in manuals and on instruments alert you to

potential risks, provide information about conditions, and comply with international

regulations. [Table 1](#) and [Table 2](#) define the symbols and markings you may encounter.

Table 1 Safety Symbols and Markings











	Warning: risk of electric shock.
	Caution: refer to accompanying documents.
	Alternating current.
	Both direct and alternating current.
	Earth (ground) terminal
	Protective earth (ground) terminal
	Frame or chassis terminal
	Terminal is at earth potential. Used for measurement and control circuits designed to be operated with one terminal at earth potential.
	Switch setting indicator. O = Off, = On.
	Standby (supply); units with this symbol are not completely disconnected from ac mains when this switch is off. To completely disconnect the unit from ac mains, either disconnect the power cord, or have a qualified electrician install an external switch.

Table 2 Regulatory Symbols Markings

	The CE mark is a registered trademark of the European Community.
	The CSA mark is a registered trademark of the Canadian Standards Association.
	The C-tick mark is a registered trademark of the Spectrum Management Agency of Australia. This signifies compliance with the Australian EMC Framework regulations under the terms of the Radio Communications Act of 1992.
ISM 1-A	This text indicates that the product is an Industrial Scientific and Medical Group 1 Class A product (CISPR 11, Clause 4).

Table 2 Regulatory Symbols Markings



This product complies with the WEEE Directive (2002/96/EC) marking requirement. The affixed product label (see above) indicates that you must not discard this electrical/electronic product in domestic household waste.

Product Category: With reference to the equipment types in the WEEE directive Annex 1, this product is classified as a “Monitoring and Control instrumentation” product.

Do not dispose in domestic household waste

To return unwanted products, contact your local Agilent office, or see <http://www.agilent.com/environment/product> for more information.

Service and Support

Any adjustment, maintenance, or repair of this product must be performed by qualified personnel. Contact your customer engineer through your local Agilent Technologies Service Center.

<http://www.agilent.com>

Click the link to **Test & Measurement**. Select your country from the drop-down menus. The Web page that appears next has contact information specific for your country.

Agilent on the Web

You can find information about technical and professional services, product support, and equipment repair and service on the Web:

Agilent by Phone

If you do not have access to the Internet, call one of the numbers in [Table 3](#).

Table 3 Agilent Call Centers and Regional Headquarters

United States:	Test and Measurement Call Center (800) 829 4444(toll-free in US)
Canada:	(877) 894 4414
Europe:	(41 22) 780 8111
Japan:	Measurement Assistance Center 0120 (421) 345
Latin America:	305 269 7500
Asia-Pacific:	(85 22) 599 7777

Chapter 1

System Software Overview	9
Agilent TestExec SL	9
Testplans	10
Testgroup	10
List of Statements	10
Test	10
Switching Action	10
Action	10
Actions	11
System.ust File	11
Aliases	12
Wires	12
Modules	12
System Configuration Editor	14

Chapter 2

How to Use the System Software	15
Chapter Contents	15
Required Computer Hardware and Software	15
System Software Description	16
Selecting Agilent TestExec SL	17
Loading a Testplan	17
Creating a Testplan	18
Using TS-5000 Supplied Actions	19
Standard Action Types	19
Example Testplans	20
Adding an Action to a Testplan	20
Using Software Debug Features	23
Looping	23
Single Stepping	24
Action Debug Messages	25
Watch Window	25
Instrument Debug Front Panels	28
Fast Connection Selection	40
Viewing Waveforms	45
Instrument Handlers	48
Using the Action Wizard To Develop	49
Multithreading	50
How Multithreading is working on a Testplan?	50

Chapter 3	
Using the DMM and ARB	57
Chapter Contents.....	57
Triggered Voltage Measurement	58
How to Trigger the DMM	58
Using the Agilent E6173A Arbitrary Waveform Generator.....	61
Arbitrary Waveform Generator Description	61
How the Arb Generates Waveforms	61
Generating Arb Waveforms	62
Memory Requirements for Downloading Waveforms	82
Agilent E6173A Arbitrary Waveform Generator Error Messages	82
 Chapter 4	
Action Summary	83
 Index	111

Chapter 1

System Software Overview

This chapter contains a detailed overview of the system software. Chapter contents are:

- Agilent TestExec SL page 9
- Testplans page 10
- Actions page 11
- System.ust File page 11
- System Configuration Editor page 14

Agilent TestExec SL

The Agilent TS-5000 System uses Agilent TestExec SL and system specific software to test the Unit Under Test (UUT). Agilent TestExec SL is a test executive designed for high-volume, high-throughput functional test applications. The other system specific software provides the communications between the test executive and system instruments.

Agilent TestExec SL uses testplans (see “Testplans”) and actions (see “Actions”) to perform the tests. The actions are the building blocks from which the tests are created. The actions are called from a test which are executed in a testplan. The switching actions makes the connections from the system instruments, and/or loads on the load cards to the Unit Under Test (UUT). The switching actions are built into Agilent TestExec SL.

Both actions and switching actions are used in a testplan to run the tests. The testplan automatically closes the appropriate pin matrix and load card switches to make the connections, setup and execute the appropriate sources and detectors, and return any test results.

The switching information and the instrument types used in the system are located in the system.ust file (see “System.ust File” on page 11 for more information). This file is generated at the factory and is custom for each system. It has the necessary information for the switching actions to close the appropriate switches and for the other actions to communicate with the system instruments. Instrument data in the system.ust file can be generated and/or modified using the System Configuration Editor (see “System Configuration Editor” on page 14).

Testplans

Figure 1-1 shows a testplan and some of its components; an explanation follows the figure (see Loading a Testplan for an explanation on how to load a testplan).

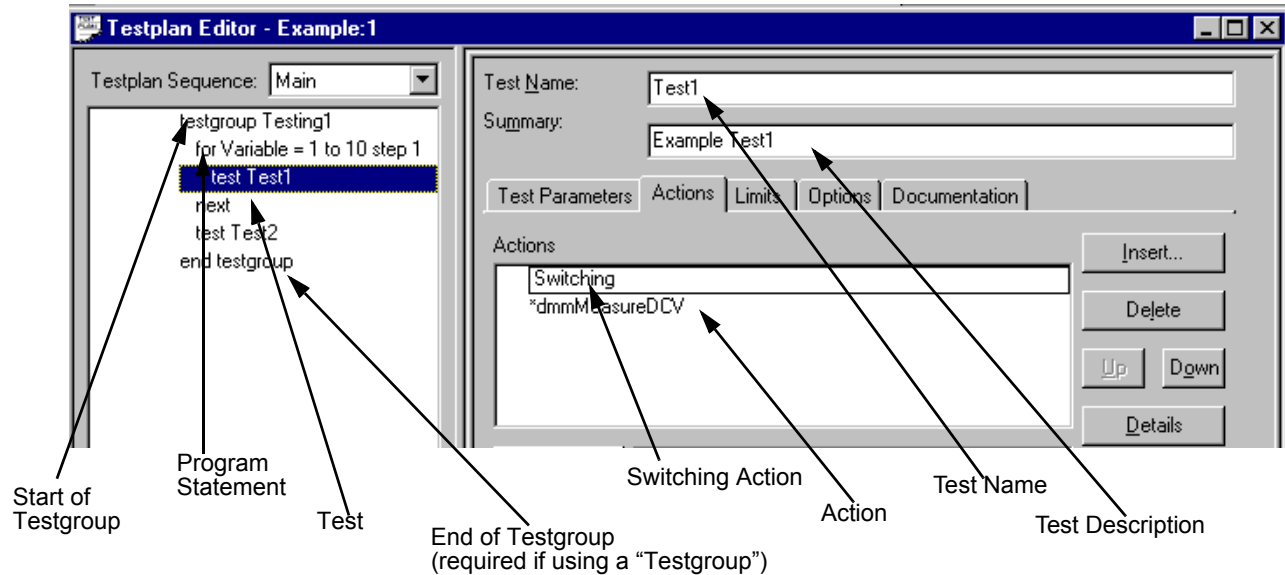


Figure 1-1. Typical Testplan Components

Testgroup

A named block of tests that can be executed in a predefined order. Each testgroup in a testplan must have a unique name; no duplicate names are allowed.

List of Statements

Test or flow control statements executed in the order shown.

Test

A named series of actions that can be executed as a group. A test can contain actions and switching actions. A test can have limit checking capabilities to determine if a test passes or fails. A test must have a unique name; no duplicate names are allowed.

Switching Action

Actions that make connections from the instrument and loads to the Unit Under Test. These actions are internal to Agilent TestExec SL. The switching information in these actions are determined by the data in the 'system.ust' file and the user-generated fixture.ust and uut.ust files.

Action

The smallest component in a test or testgroup that can be called to perform functions such as setting up an instrument, making measurements and prompting the user.

Actions

The system comes with a set of supplied actions. These actions are used for such things as configuring instruments, making measurements and prompting users for inputs. You can also generate custom actions using the Action Wizard and an application program such as Visual C/C++.

The action directory path is:

```
C:\Program Files\Agilent\TS-5000 System Software\actions
```

Actions are located in the following sub-directories:

Sub-Directory	Action Type
arb	Arbitrary Waveform Generator Actions
counter	Counter Actions
dac	D/A Converter Actions
dgn	Diagnostics Actions
digitizer	Digitizer Actions
dio	Digital I/O Actions
dmm	Digital Multimeter Actions
dso	Digital Storage Oscilloscope
event	Event Detector Actions
generic	Miscellaneous Actions
mcm	Measurement Control Module Actions
power	Power Supply Actions
serial	Serial Interface Actions
SerialProtocol	Automotive Serial Protocol Actions
switch	SLU and Switching Actions
vi	Voltage / Current Source Actions
daq	Data Acquisition Actions

System.ust File

The 'system.ust' file is created at the factory with the appropriate module/instrument and wiring data according to the system option. The file can be edited using the System Configuration Editor (see "System Configuration Editor" on page 14) or the Topology Editor in Agilent TestExec SL.

The data in the 'system.ust' file consists of modules/instruments and instrument nodes (i.e., connections at the 32-Pin Matrix and Instrument Multiplexer Module), and other nodes (connections at the 32-Pin Matrix Modules and load cards) using the Aliases, Wires, and Modules designations. The Module designations are used to determine the modules/instruments installed in the system, and the Aliases and Wires are used to generate switch paths.

A typical 'system.ust' file contains Aliases, Wires and Modules.

Aliases

These are alternate names for reference nodes. The names are descriptive in nature to easily identify the node. For example, the node name for the high current output of the Agilent 34401 or E1411 Digital Multimeter is called "IsrcHi". Figure 1-2 shows typical Aliases in the 'system.ust' file.

Wires

These are names for wires that connect to or between nodes. In some cases, these names are the same aliases used for nodes. For example, the alias called "DVMIsrcHi" is often used for the wire name that connects to that node. Figure 1-3 shows typical wires in the 'system.ust' file.

Modules

These are the names of the instruments in the system. For example, the name "MCM" indicates that there is an Agilent E6171B Measurement control module in the system. The Figure 1-4 shows a few typical modules in the 'system.ust' file.

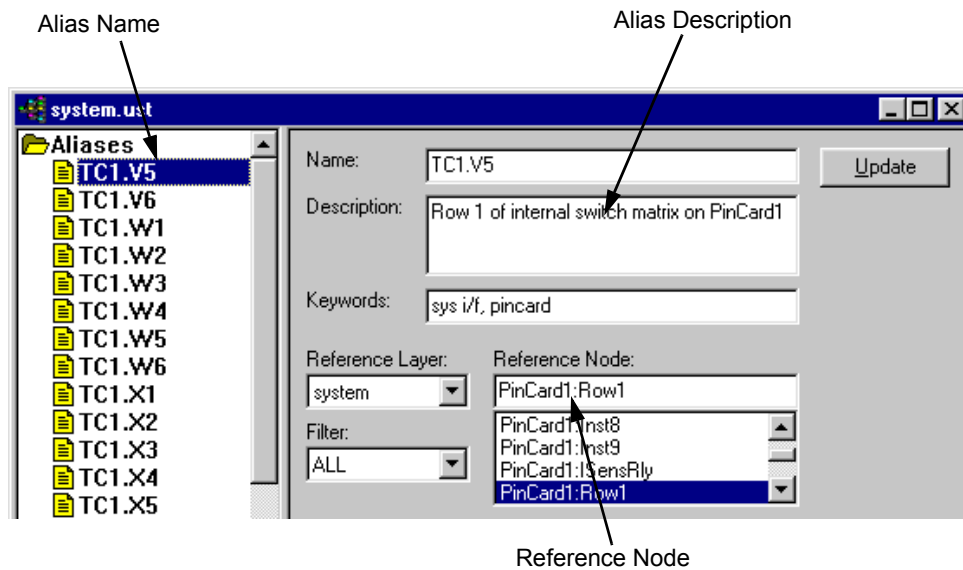


Figure 1-2. Typical Aliases

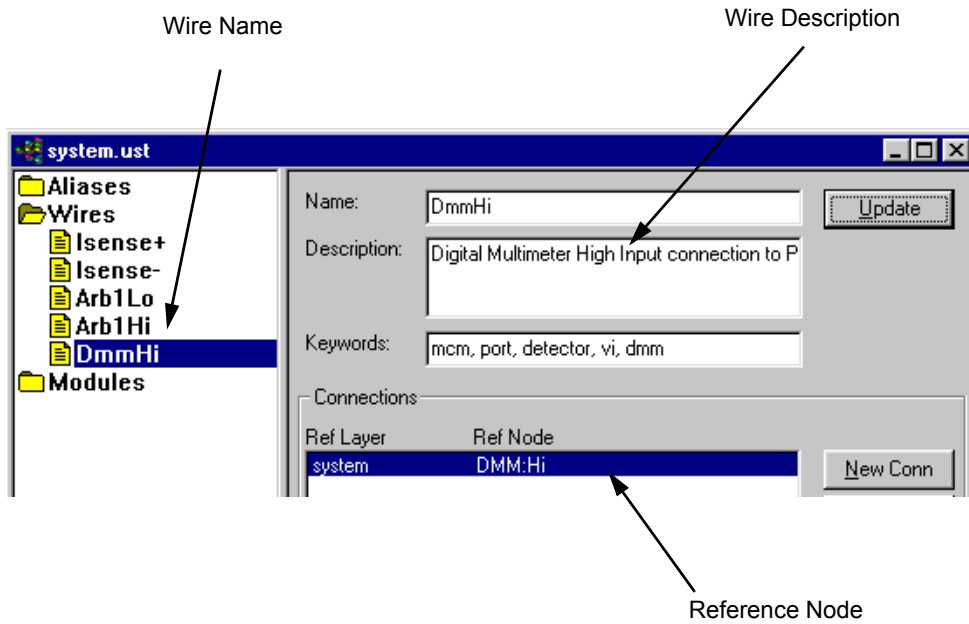


Figure 1-3. Typical Wires

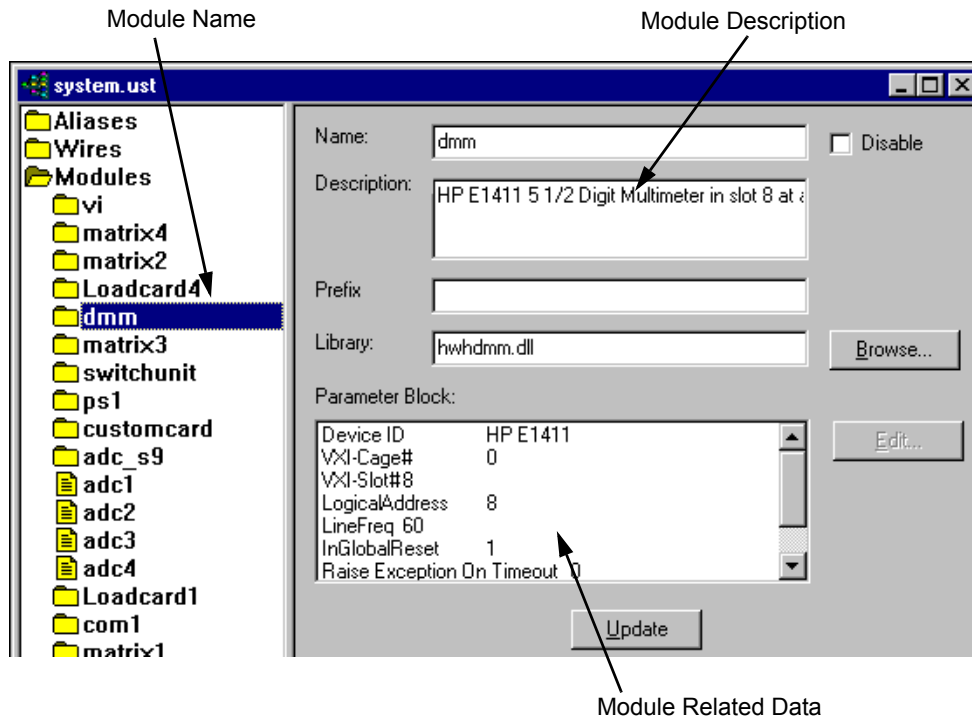


Figure 1-4. Typical Modules/Instruments

System Configuration Editor

This editor can be used to add new modules/instruments or delete old modules/instruments to/from the system.ust file (see “System.ust File” on page 11 for information). It can add/delete any supported GPIB or VXI instruments, Pin Matrix Modules, and/or load cards.

The editor has the following functions:

- Shows all supported modules/instruments.
- Automatically detects all modules/instruments currently in the system
- Able to add new custom modules/instruments to the system.ust file
- Able to edit module/instrument parameters

Any of the supported and custom modules/instruments can be added to the system.ust file. Dependent on the system type and option (e.g., Test System Interface vs. Mass Interconnect), the System Configuration Editor automatically generates the appropriate wires and aliases (see “System.ust File” on page 11 for information). The wiring, etc. data is located in a spreadsheet that is specifically generated for your system type and option.

The System Configuration Editor is available from the TestExec SL Toolbar or from a shortcut in the Desktop on the system PC controller. Refer to the System Configuration Editor’s online help for more details.

Chapter 2

How to Use the System Software

Chapter Contents

This chapter lists the needed software to run the system and shows some system specific software operation. The chapter is separated as follows:

- Required Computer Hardware and Software page 15
- System Software Description page 16
- Selecting Agilent TestExec SL page 17
- Loading a Testplan page 17
- Creating a Testplan page 18
- Using TS-5000 Supplied Actions page 19
- Using Software Debug Features page 23
- Instrument Handlers page 48
- Using the Action Wizard To Develop page 49
- Multithreading. page 50

Required Computer Hardware and Software

The following is a list of the computer hardware and software needed to run the Agilent TS-5000 Test System. All necessary software was factory installed on your system:

- IBM-compatible PC (at least a Pentium) with 256 MB of RAM, 1024x768 graphics, 100 MB of free disk space (20 MB for Agilent TestExec SL software).
- Microsoft® Windows XP®.
- Agilent TestExec SL software, Version 5.1 or later.
- Agilent TS-5000 Software version 5.1 or later.

Note For more detailed information about using the Agilent TestExec SL software, refer to the software documentation

System Software Description

The Agilent TS-5000 System uses Agilent TestExec SL and system specific software to test the Unit Under Test. Agilent TestExec SL is a test executive designed for high-volume, high-throughput functional test applications. The other system specific software provides the communications between the test executive and system instruments.

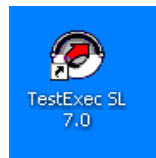
Agilent TestExec SL uses testplans and actions to perform the tests. The actions are the building blocks from which the tests are created. The actions are called from a test which are executed in a testplan. The switching actions makes the connections from the system instruments, and/or loads on the load cards to the Unit Under Test. The switching actions are built into Agilent TestExec SL.

Both actions and switching actions are used in a testplan to run the tests. The testplan automatically closes the appropriate pin matrix and load card switches to make the connections, setup and execute the appropriate sources and detectors, and return any test results.

The switching information and the instrument types used in the system are located in the `system.ust` file. This file is generated at the factory and is custom for each system. It has the necessary information for the switching actions to close the appropriate switches and for the other actions to communicate with the system instruments. Instrument data in the `system.ust` file can be generated and/or modified using the System Configuration Editor (see “Using the System Configuration Editor” in the “TS-5000 System Integrator’s Manual”).

Selecting Agilent TestExec SL

The Agilent TestExec SL software is pre-installed on your PC controller's hard drive. Start TestExec SL from this icon in the PC desktop:



You can also run TestExec SL by clicking:

Start | Programs | Agilent TestExec SL 7.0 | TestExec SL 7.0

Loading a Testplan

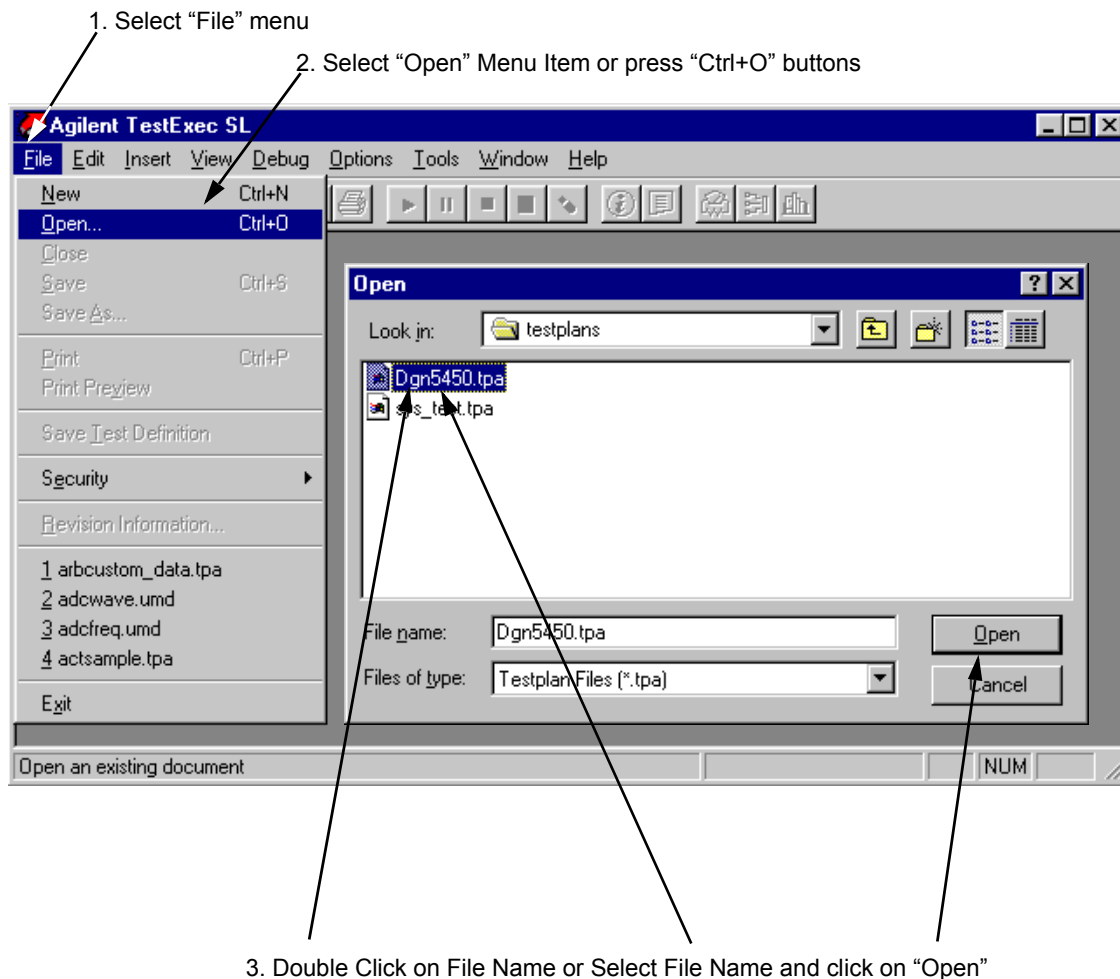


Figure 2-1. Agilent TestExec SL Main Screen and File Open Box

Creating a Testplan

Figure 2-2 shows how to create a testplan.

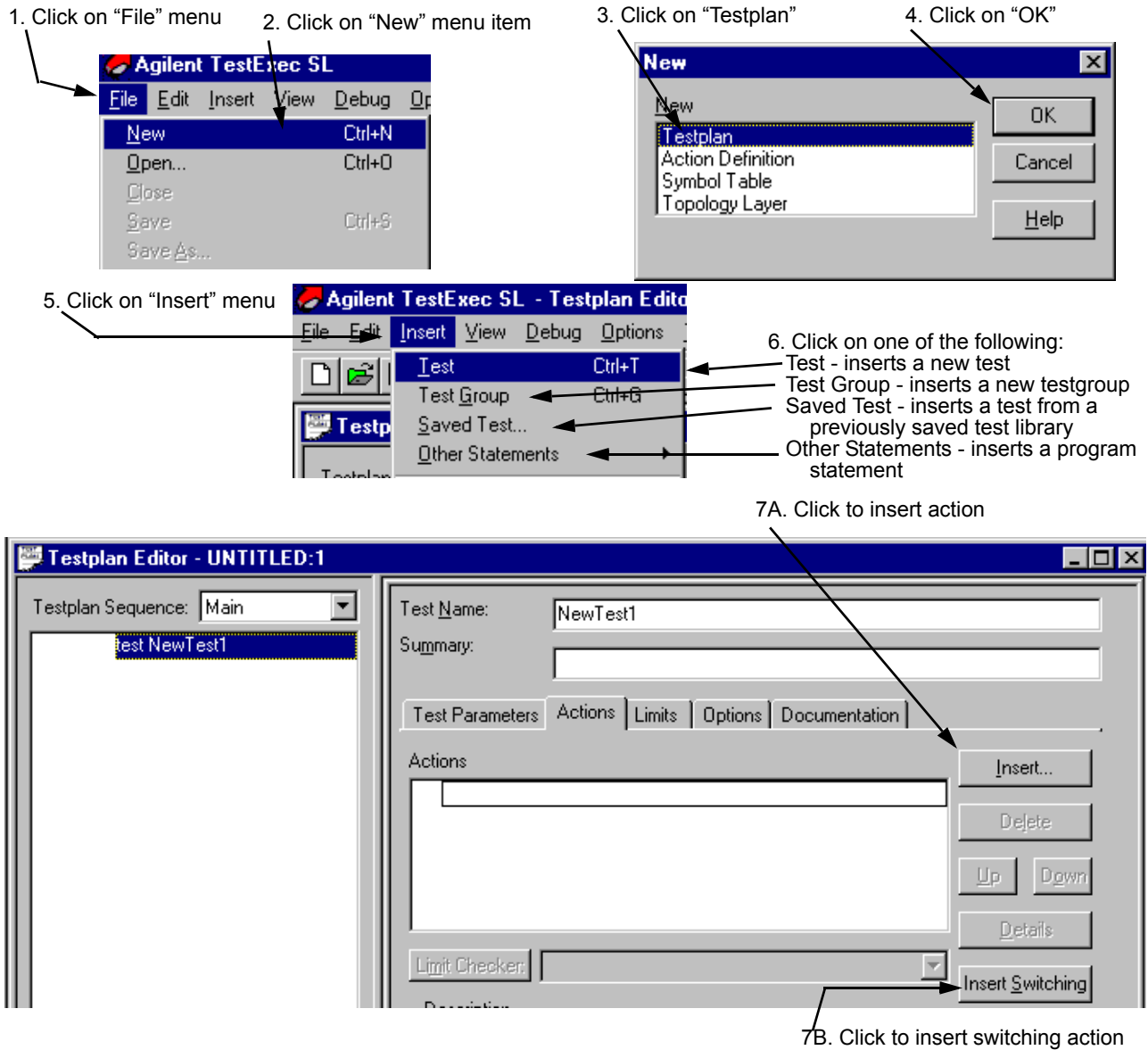


Figure 2-2. Creating a Testplan

Using TS-5000 Supplied Actions

The Agilent TS-5000 System with a set of standard actions supplied with the system. The action directory path is:

C:\Program Files\Agilent\TS-5000 System Software\actions

Actions are located in the following sub-directories:

Sub-Directory	Action Type
arb	Arbitrary Waveform Generator Actions
counter	Counter Actions
dac	D/A Converter Actions
dgn	Diagnostics Actions
digitizer	Digitizer Actions
dio	Digital I/O Actions
dmm	Digital Multimeter Actions
dso	Digital Storage Oscilloscope Actions
event	Event Detector Actions
generic	Miscellaneous Actions
mcm	Measurement Control Module Actions
power	Power Supply Actions
serial	Serial Interface Actions
SerialProtocol	Automotive Serial Protocol Actions
switch	SLU and Switching Actions

Note The actions are summarized in Chapter 4 of this manual. The actions are documented in detail in the TS-5000 online help which is available from the TestExec SL Help menu.

Standard Action Types

The standard actions are generally organized around a specific instrument or module in the system. The actions are also separated into the types discussed below.

High-Level Actions

These are actions that usually perform complete tests, such as setting up a source and then making a measurement. These actions normally, but not always, use one or more of the Low-Level Actions.

Low-Level Actions

These actions perform specific configuration or measurement function on a particular instrument. The low-level actions give you more flexibility in configuring instruments and making the measurements than do the high-level actions. Low-Level actions are organized by instrument type and function. For example, an action that is to set up triggering for a voltmeter is a low-level action.

Example Testplans

Example testplans to show how to use the actions are in the following directory:

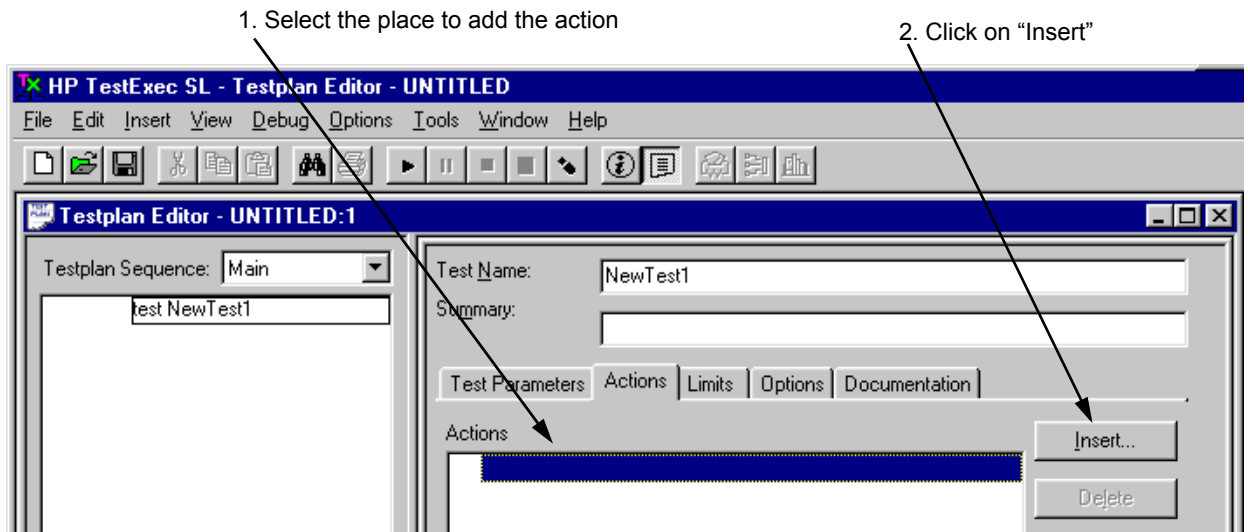
```
C:\Program Files\Agilent\TS-5000 System  
Software\testplan\examples
```

Adding an Action to a Testplan

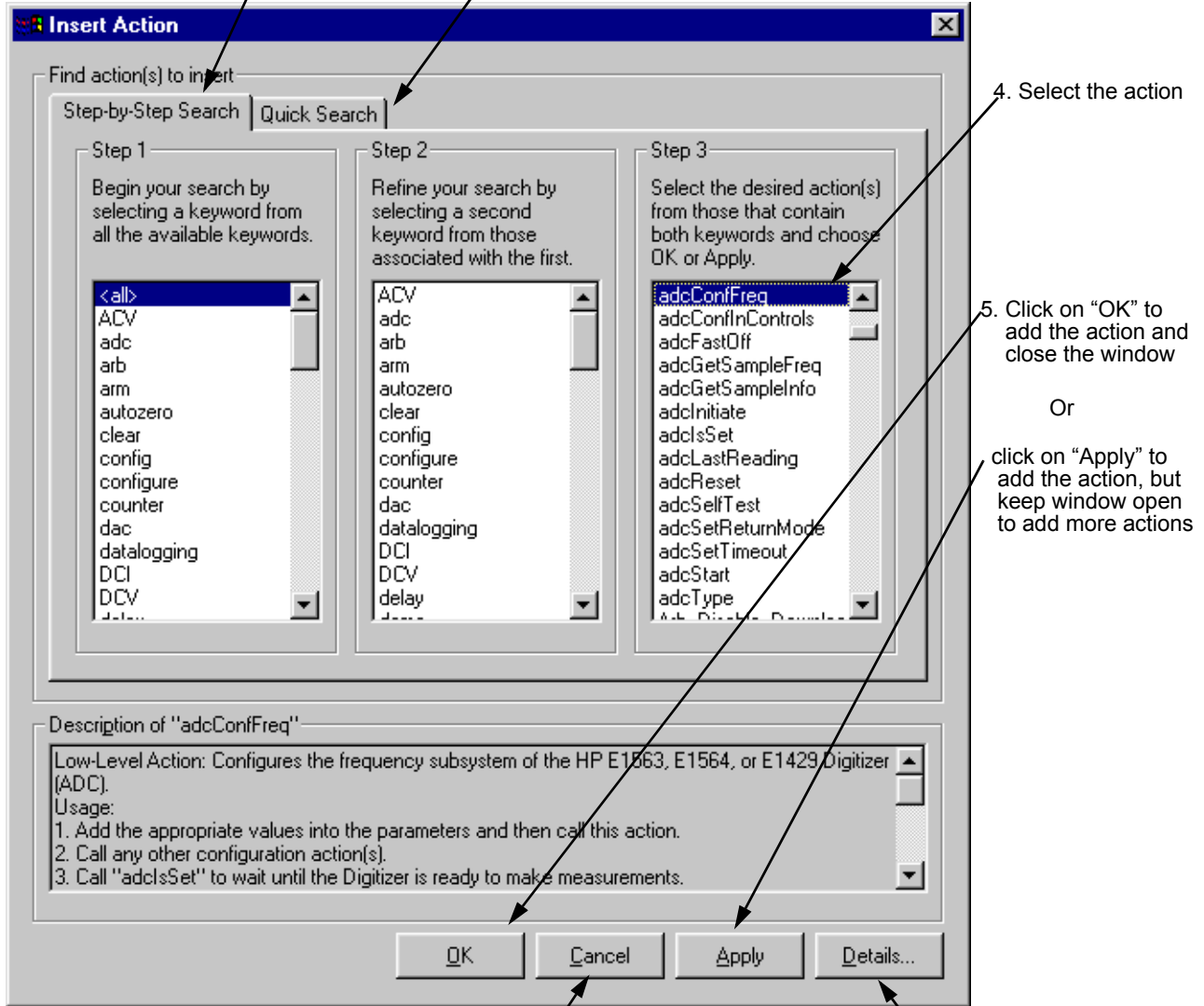
Do the following:

1. Open Agilent TestExec SL using the procedure in “Selecting Agilent TestExec SL” on page 17.
2. Either create a new or open an old testplan, using the procedure in “Loading a Testplan” on page 17 or “Creating a Testplan” on page 18, respectively.
3. If using a new testplan, do the procedure in Figure 2-2 on page 18 to add a test. If using an old testplan, select the test into which you wish to add an action and continue with step 4.
4. Add an action to the test, as shown in Figure 2-3 and Figure 2-4.

There are two different actions that can be added, a regular action and a switching action. Figure 2-3 shows how to add a regular action and Figure 2-4 shows how to add a switching action.



3. Either click on "Step-by-Step Search" or "Quick Search" ("Quick Search" shows all actions in a column)



Click on "Cancel" to close window without adding action

Click on "Detail" to open Action Definition Editor

Figure 2-3. Adding an Action

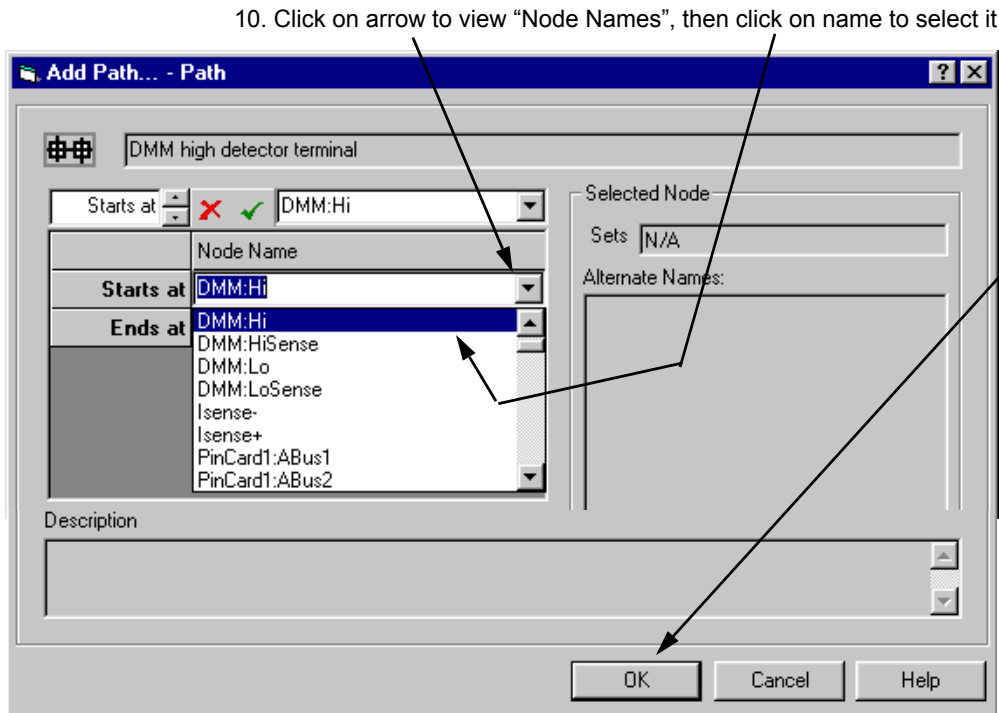
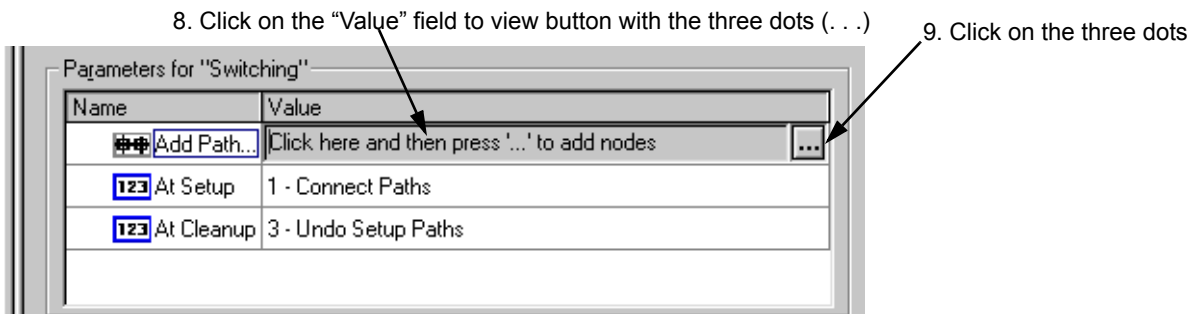
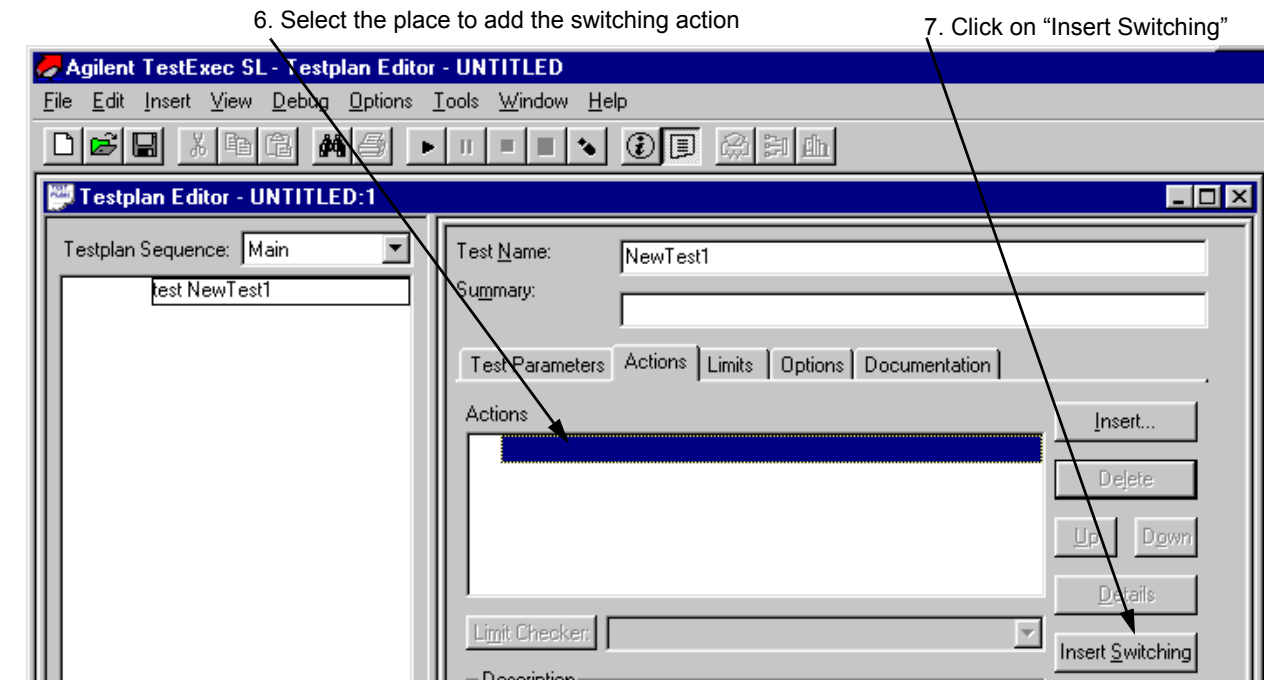


Figure 2-4. Adding a Switching Action

Using Software Debug Features

This section describes specific software features, unique to the Agilent TestExec SL version 2.0 and later, that you will find helpful in creating and debugging your tests and testplans. To enable Agilent TestExec SL, use the procedure in “Selecting Agilent TestExec SL” on page 17, if not enabled.

Looping

The loop constructs are FOR / NEXT sequencer statements. For example, a testplan developer would stop a testplan (if one is running), add FOR and NEXT lines to the testplan and then restart the testplan from the beginning. Refer to the Agilent TestExec SL documentation for additional information.

Figure 2-5 shows how to select and insert the looping statements into a testplan.

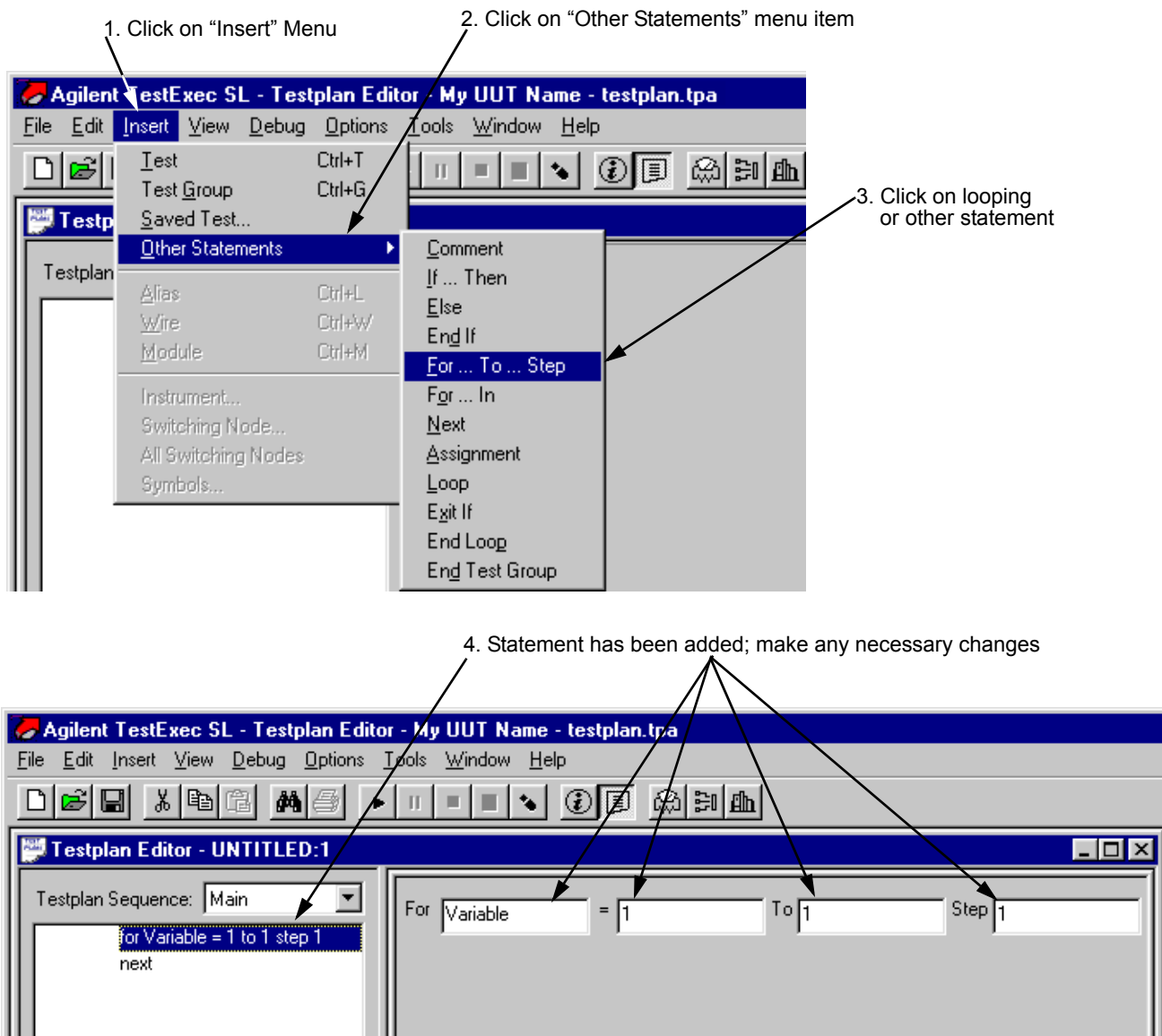


Figure 2-5. Selecting Looping and Other Statements

Single Stepping

Agilent TestExec SL adds the ability to single step the execution of the operations that make up a test. When stepping through the actions of a test, the system pauses at entry point calls for each action routine. Action stepping can be performed any time testplan execution is paused (as a result of operator pause or test breakpoint). The action step which is currently paused is listed in the trace window. Refer to the Agilent TestExec SL documentation for additional information.

Figure 2-6 shows how to select the single stepping

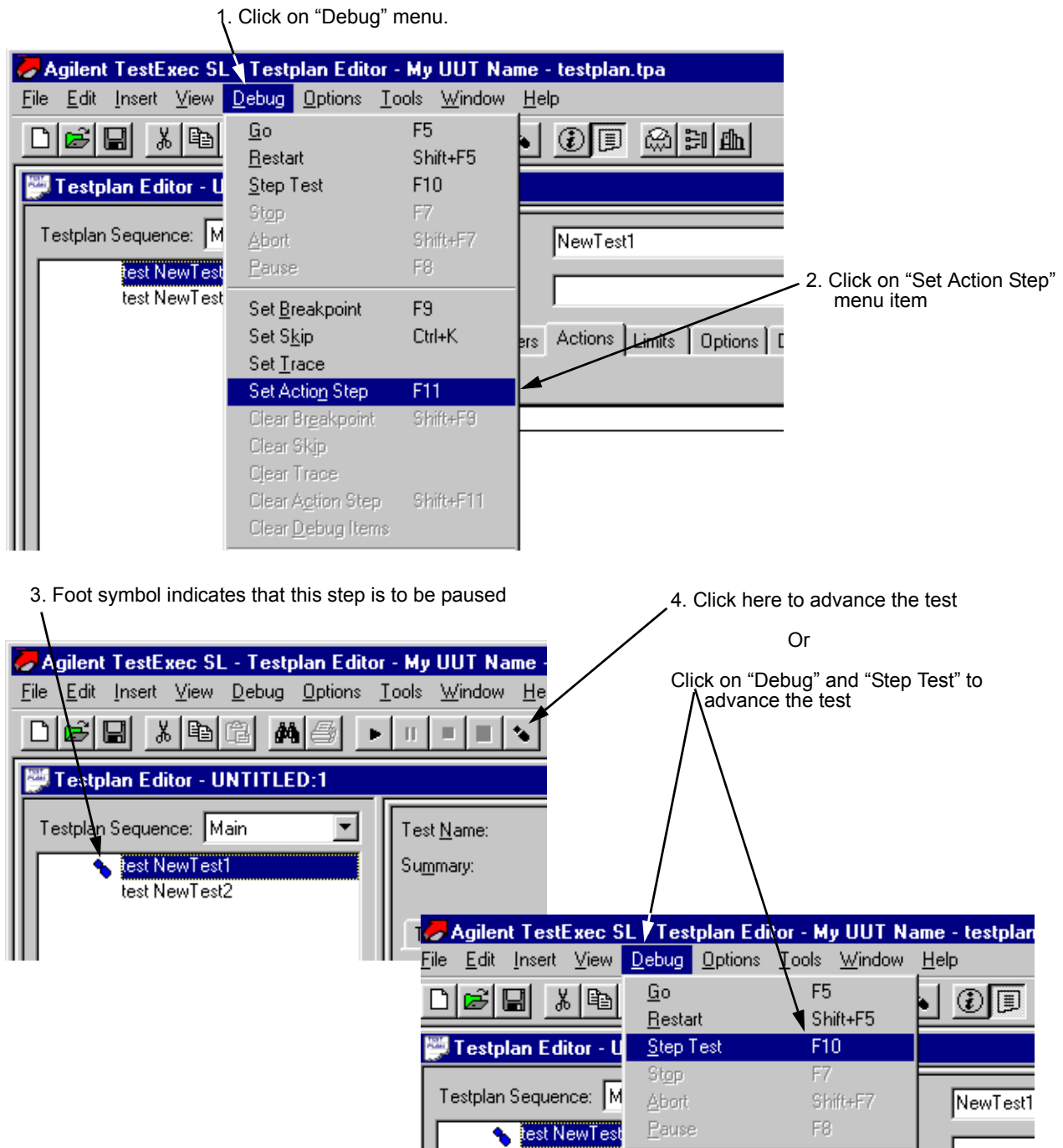


Figure 2-6. Selecting Action Stepping

Action Debug Messages

Agilent TestExec SL allows action routines to send strings to the same debug window that the system uses for the test trace output. Use the following API:

- UtaTrace (MessageString, MessageID)
 - MessageString is a string that will be sent to the trace window.
 - MessageID is an optional string that allows the messages to be group (as identified by MessageID). Note: currently, this parameter does not provide any functionality.

Watch Window

Watch windows provide the ability to select ‘objects’ to dynamically monitor as a testplan executes. Select an item to watch from a list, the item is added to the watch window list and is updated each time the testplan pauses. Items can be added or deleted at any time (as long as the testplan is not running).

The watch window can monitor 4 types of objects:

- Instruments
- Switching Node
- All Switching Nodes
- Symbols

The following shows some typical example of the window.

Watch Window Menu Selections

Watch Window Menu Selections shows how to select one of the four Watch Window objects. The object to be selected, as shown in the figure, is the ‘instrument’ object. (The selection of the other objects is the same.)

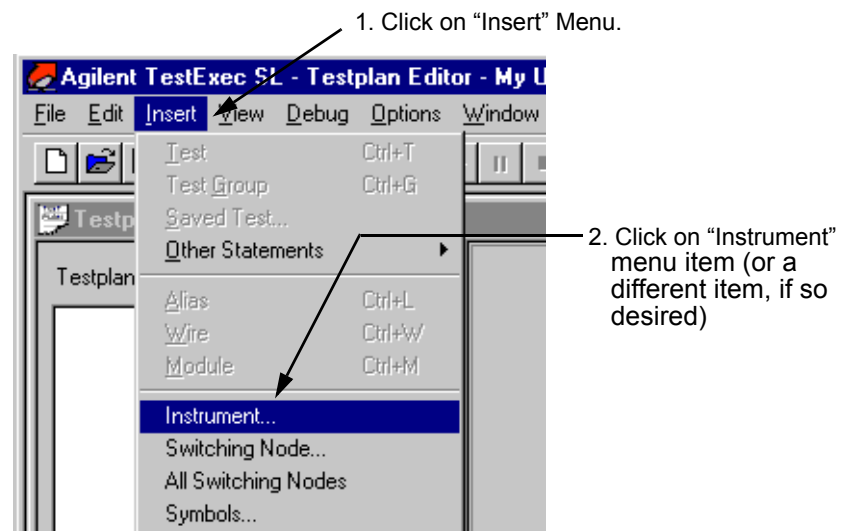


Figure 2-7. Watch Window Menu Selections

Watch Window Instrument Status

Watch windows are available for many of the instruments in an Agilent TS-5000 system. The instruments that can be monitored are:

- Agilent E1328A or E1418A Digital to Analog Converter Module (DAC)
- Agilent E1333A Counter Module
- Agilent E1411 Digit Multimeter Module (DMM)
- VXI Technology E1563A Analog to Digital Converter Module (ADC)
- Agilent E6171B Measurement Control Module (MCM)
- Agilent E6174A 32-Channel Event Detector Module
- Agilent 6642A, 6643A, 6652A, 6653A, 6673A Power Supplies

The displayed instrument states match the parameters of the instrument handler routines. For example the Agilent E1411 Digital Multimeter has the following routines which affect the state of the module:

- dmmConfFunction (dmm, Func, Range, Aperture);
- dmmConfCal (dmm, Linefreq, Autozero);
- dmmConfTrigIn (dmm, Trigselect, Trigcount, Delay);
- dmmConfSample (dmm, Sampsrc, Count, Period);

Figure 2-8 shows the menu box to select the instrument to be watched. The figure shows how to select the Digital Multimeter (dmm).

To add an instrument to the Watch Window, select the instrument type and press “OK”, as shown in Figure 2-8. The Watch Window is then displayed, as shown in Figure 2-9.

Figure 2-9 also shows the expanded tree view that shows the status of the ‘dmm’ parameters. The instruments display their state in a tree view. This allows a logical grouping of states and reduces the amount of data displayed by placing less important states in lower levels of the tree.

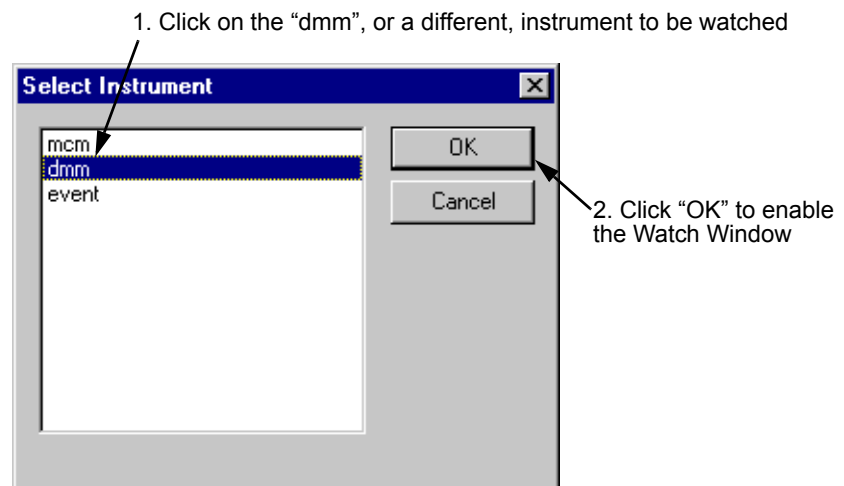


Figure 2-8. Selecting an Instrument to be Watched

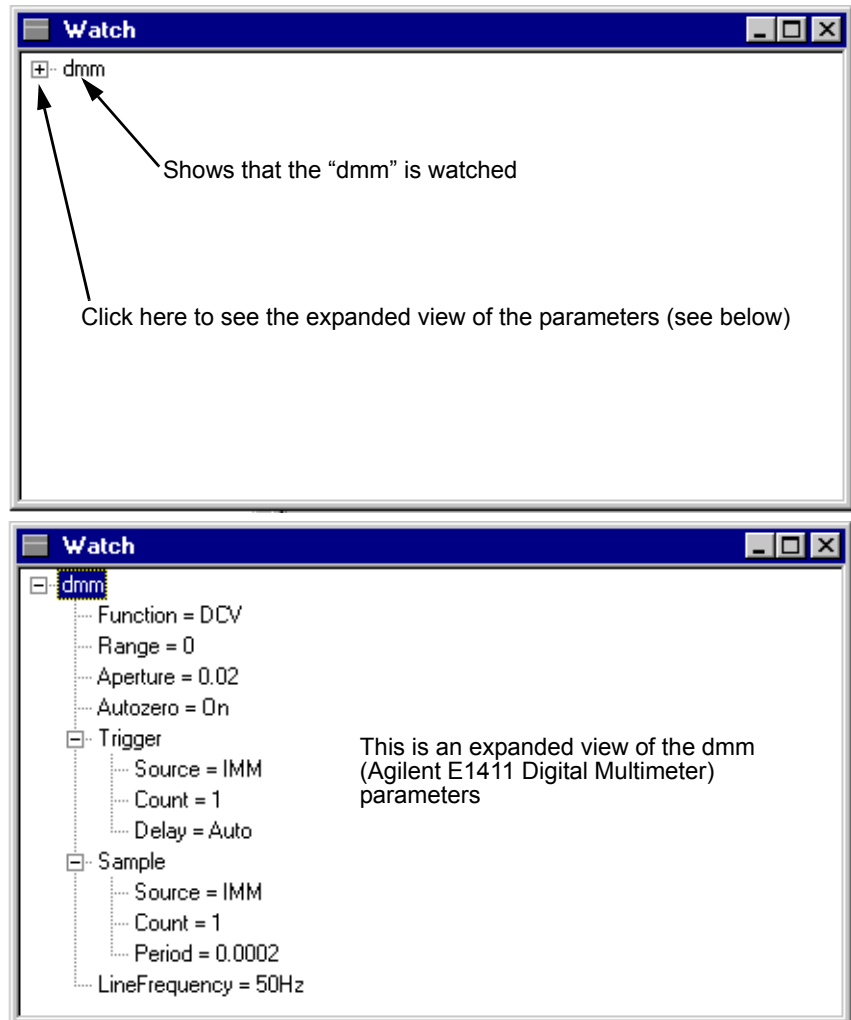


Figure 2-9. Agilent E1411B Digital Multimeter Watch Window with Parameters

Instrument Debug Front Panels

Agilent TestExec SL provides a series of debug instrument front panels for the Agilent TS-5000 system instruments. The debug panels provide two main features:

- The ability to interactively view the current Unit Under test (UUT) state
- The ability to interactively view the current state of points within the system.

Debug Panel Types

The following debug panels are supplied with the system, dependent on the modules/instrument currently installed in the system.

- Agilent E1411B Digital Multimeter page 30
- Agilent E1328A & E1418A Digital to Analog Converter page 32
- Agilent E1333A Counter page 33
- VXI Technology E1563A Digitizer page 35
- Agilent E6171B Measurement Control Module page 36
- Agilent E6174A Event Detector page 37
- Agilent E6198A Switch/Load Unit. page 38
- Agilent 6642A, 6643A, 6652A, 6653A, 6673A Power Supplies page 39

Debug Panel Features

All of the debug panels provide the following features:

- Changes on the instrument front panel are not made until either:
 - you initiate a measurement (for detectors) or
 - you press the “Apply” button (for sources)

The instrument state memory is modified in the instrument handler at this time (since this is when the instrument handler is called; See Instrument Handlers on page 48 for information about handlers).

- If the instrument is connected in the Agilent TS-5000 system through switching, the front panel provides a section which allows you to define the switching path. Either select a UUT pin to connect to directly or define a custom switch path.

Connecting to the UUT pin is the easiest way to select a switch path; you select an ABus pin and the UUT pin. If the instrument has both the high and low terminals connected to the switch matrix, then both pins need to be defined. The low terminal is connected to UUTCommon by default and does not need to be selected, if you use the default. The UUT option is available if you have defined a UUT switching configuration file. In addition to the UUT pins, the system automatically adds the option to measure an ABus pin only. This will allow you to inject a signal on the front of the system and measure it using the instrumentation within the system.

- The state of the system is saved upon entering a pause state and is restored when entering the run state. This allows the panels to be used

without affecting the testplan.

Selecting the Debug Panel

Debug Front Panel Selection and Select an Instrument from the Debug Front Panel Selection show how to enable the debug instrument front panels.

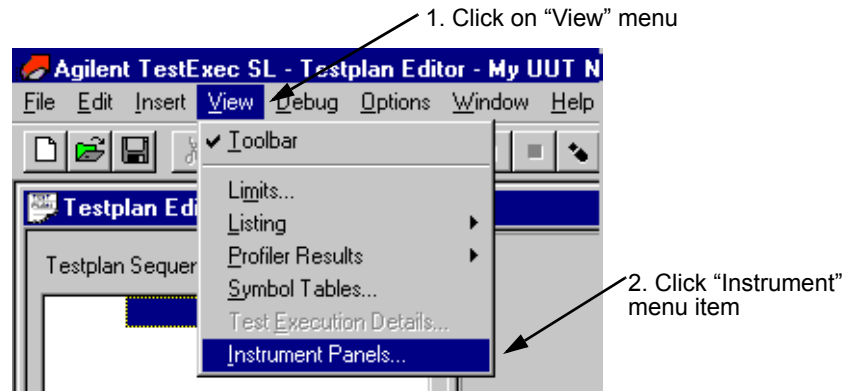


Figure 2-10. Debug Front Panel Selection

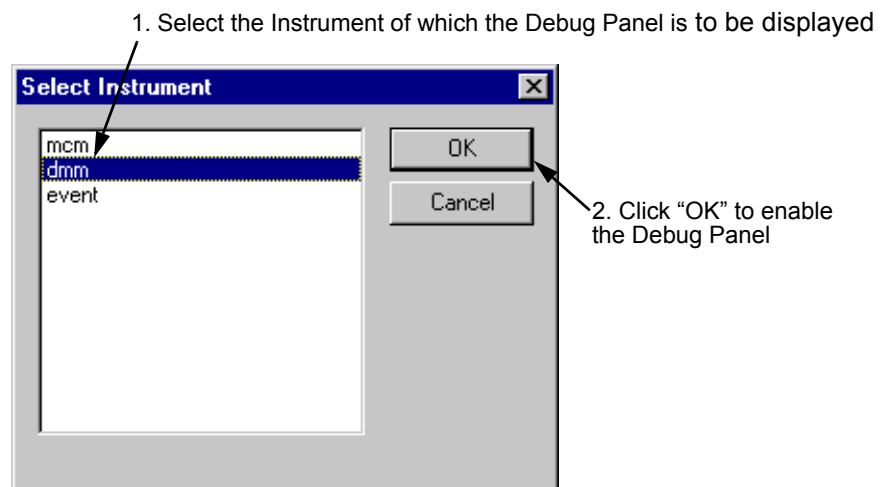


Figure 2-11. Select an Instrument from the Debug Front Panel Selection

The following sections show and explain the different debug instrument panels.

Agilent E1411B Digital Multimeter

An instrument box which connects to the Agilent E6171 Measurement Control Module, such as the Agilent E1411 DMM, usually has two different dialog boxes which depend on the type of path selected (as shown in Agilent E1411 DMM Debug Front Panel (showing Path Selection)A and Agilent E1411 DMM Debug Front Panel (showing Path Selection)B). This allows you to either connect to the pin on the UUT (Agilent E1411 DMM Debug Front Panel (showing Path Selection)A) or select any point within the system by defining a custom switching path (Agilent E1411 DMM Debug Front Panel (showing Path Selection)B). The custom switching is applied on top of the current switching state, so to undo a switching path you need to define a ‘disconnect’ path.

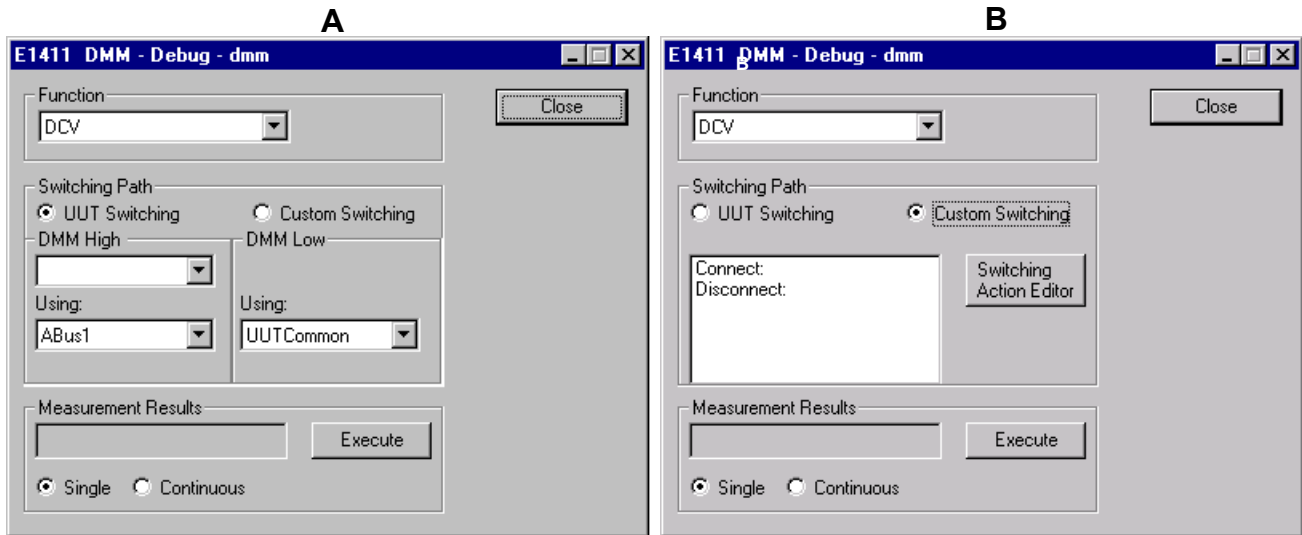


Figure 2-12. Agilent E1411 DMM Debug Front Panel (showing Path Selection)

The debug panel boxes and buttons do the following:

Function:

Choose ACV, DCV, Ohms, Ohms - Offset Compensated
The measurement is made in auto range and medium resolution.

Switching Path:

The debug panel supports the selection of the switching path. The panel switches both high and low connections of voltage sense and the high and low connection of the current terminals. The sense connections are connected in parallel with the high and low terminals. You specify the high or low terminal and the system automatically connects the associated sense terminal.

Measurement Results:

Press the “Single” button to initiate a single measurement; press the “Continuous” button to initiate a series of measurements. The continuous button is modified to a “Stop” button once pressed. When the dialog box is brought up, the measurement results box is blank.

Switching Action Editor:

This button enables the editor to select the switching paths for the

Digital Multimeter. See “Fast Connection Selection” on page 40 for a description on how to use the editor.

Execute:

This button starts a measurement.

Close:

This button closes the debug panel.

System Interface

The setting of the Digital Multimeter function uses the following actions:
dmmMeasureDCV, dmmMeasureACV, dmmMeasureOhms

The above actions also return the results of a measurement.

Agilent E1328A & E1418A Digital to Analog Converter

Agilent E1328A and E1418A DAC Debug Front Panel shows the dialog box for controlling the Agilent E1328/E1418 Digital-to-Analog Converter (DAC) Module.

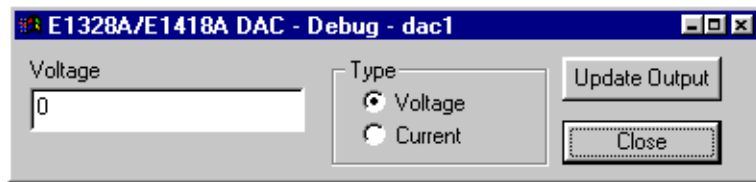


Figure 2-13. Agilent E1328A and E1418A DAC Debug Front Panel

The debug panel boxes and buttons do the following:

Voltage or Current:

You have the option to specify the output voltage or current for each channel of the DAC. The 'Type' selection sets the voltage or current mode and defines contents of the input field.

Dependent on the type selected, enter the voltage value/current value into the Voltage/Current field.

(Note the E1328A cannot programmatically select the voltage or current option.)

Update Output:

This button changes the output setting.

Close:

This button closes the debug panel.

System Interface

The setting of the DAC uses the following action routine:

```
dac16i_setup
```


Agilent E1333A Counter

Agilent E1333 Counter Debug Front Panel shows the dialog box for Agilent E1333A Counter. Only the UUT connection screen is shown. The custom path selection is the same used by the Digital Multimeter (see Figure 2-12 on page 30).

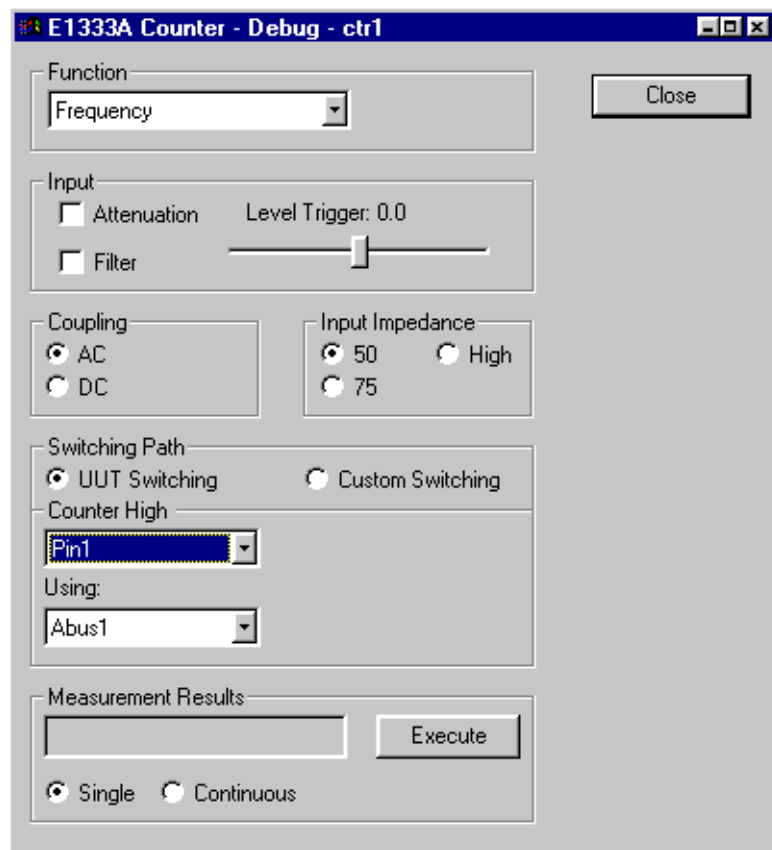


Figure 2-14. Agilent E1333 Counter Debug Front Panel

The debug panel boxes and buttons do the following:

Function:

Choose Frequency, Period, Positive Pulse Width, Negative Pulse Width, Totalize

Input, Coupling, Input Impedance:

Allows you to set input functions. The setting of each function is sent before a measurement is made.

Switching Path:

The counter panel supports the selection of a switching path. The counter switches only the high connections of the output. The low connections are automatically connected to system ground.

Measurement Results:

Select the “Single” button to initiate a single measurement or the “Continuous” button to initiate a series of measurements. The continuous button is changed to a “Stop” button, once pressed.

Note that when the dialog box is brought up, the measurement results box is blank.

Execute:

This button starts a measurement.

Close:

This button closes the debug panel.

System Interface

The setting of the counter function will use the following action routines:
ctrMeasureFrequency, ctrMeasurePeriod, ctrMeasurePulseWidth,
ctrMeasureTimeInterval, ctrMeasureTotalize

The measurement uses the level trigger specified in the dialog box (using the slider control), a default range and default resolution.

The input section uses the following action routine:
ctrMeasureInControls

VXI Technology E1563A Digitizer

Figure 2-15 shows the debug front panel for the VXI Technology E1563A Digitizer. Only the UUT connection screen is shown. The custom path selection is the same as for the Agilent E1411B Digital Multimeter (see “Agilent E1411B Digital Multimeter” on page 30).

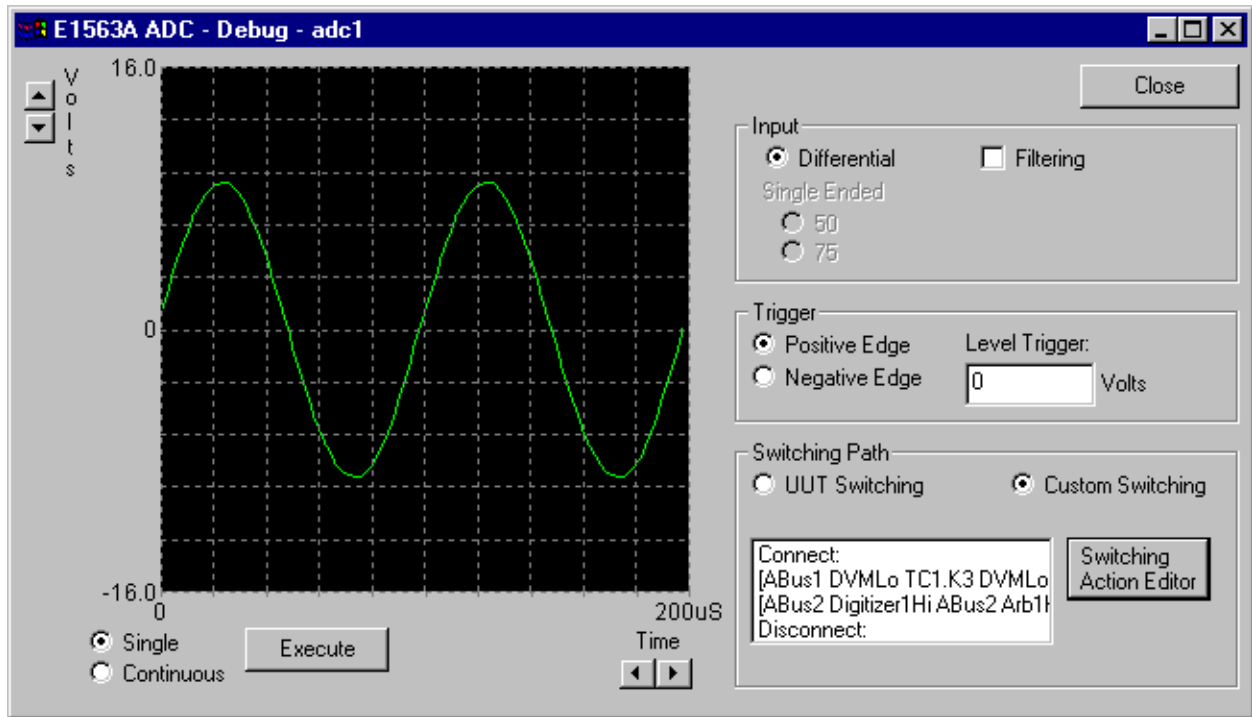


Figure 2-15. VXI Technology E1563A Digitizer Debug Front Panel

The debug panel boxes and buttons do the following:

Time Base and Range:

The range and timebase can be modified by selecting the “arrow” keys on the side and bottom of the waveform. To update the display, take a new measurement.

The time base allows the following range of selections:

50 nS to 20 Sec in 1, 2, 5 steps

the range has the following selections:

0.1 to 100 volts in 1, 2, 5 steps.

When the dialog box is brought up, the combo box contains the current time base and range. All of the setting for the digitizer are saved until an actual sweep is initiated.

Input and Triggering:

These two sections allow you to set a variety of input and trigger functions. The setting of each function is sent when a measurement is made.

Switching Path:

The debug panel supports the selection of a switching path. The digitizer switches both the high and low connections of the output.

Waveform Display:

Select the “Single” button to generate a single waveform display or the “Continuous” button to generate a continuous update of waveforms. The display updates at a rate of >10 update/sec.

Close:

This button closes the debug panel.

System Interface

The setting of the digitizer function will use the following action routines: adcConfArm, dcConfFreq, adcConfInControls, adIsSet, adcGetResults

**Agilent E6171B
Measurement Control
Module**

Agilent E6171B Measurement Control Module Debug Front Panel shows the dialog box for controlling the Agilent E6171B Measurement Control Module.

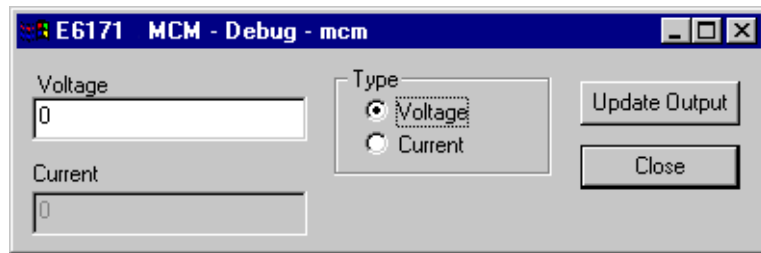


Figure 2-16. Agilent E6171B Measurement Control Module Debug Front Panel

The debug panel boxes and buttons do the following:

Voltage or Current:

You have the option to specify the output voltage or current for each channel of the MCM. The ‘Type’ selection sets the voltage or current mode.

Dependent on the type selected, enter the voltage value or current value into the Voltage or Current field, respectively.

Expected Current (when enabled by the Type-Current):

Defines the expected current of the MCM (or VI, i.e., Voltage/Current) source.

Update Output:

This button changes the output source of the MCM.

Close:

This button closes the debug panel.

System Interface

The setting of the counter function will use the following action routines: viSetSourceDCI, viSetSourceDCV, viSet

Agilent E6174A Event Detector

Agilent E6174A Event Detector Debug Front Panel shows the dialog box for controlling the Event Detector Module.

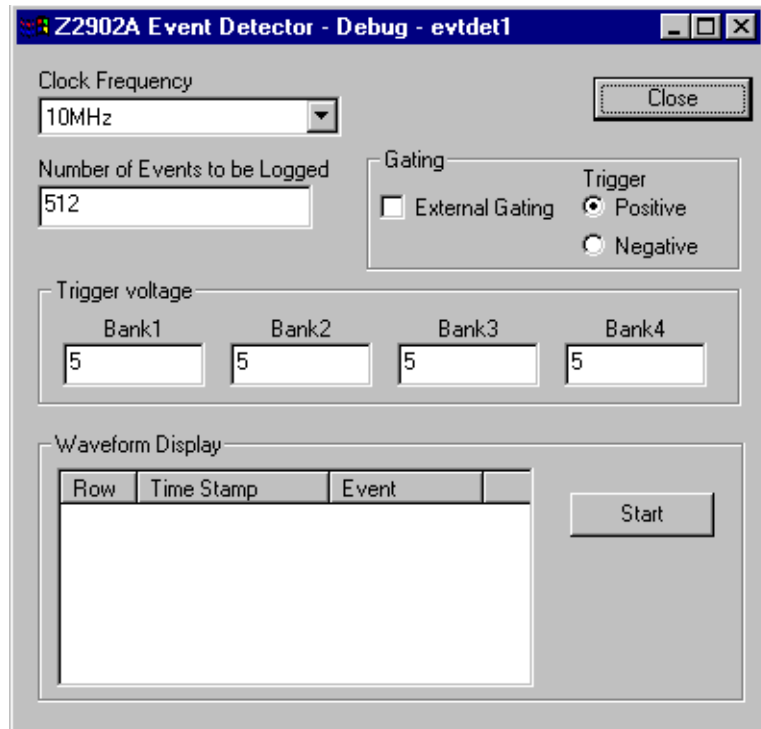


Figure 2-17. Agilent E6174A Event Detector Debug Front Panel

The debug panel boxes and buttons do the following:

Clock Frequency:

Select the clock rate of the event detector module.

Edge Trigger, External Gating and Number of Events:

Specifies the type of edge triggering to be used and external gating. The text box defines the number of events to be logged before a measurement is complete.

Event Display:

The event detector can generate either a single display or a continuous update of events. The table provides a simple list view of the event and its time stamp. The column width can be adjusted by dragging the line which separates the titles.

Execute:

This button starts a measurement.

Close:

This button closes the debug panel.

System Interface

The setting of the event detector will use the following action routines:
eventMeasure

Agilent E6198A Switch/Load Unit

Agilent E6198A Switch/Load Unit Debug Front Panel shows the dialog box for controlling the Switch/Load Unit.

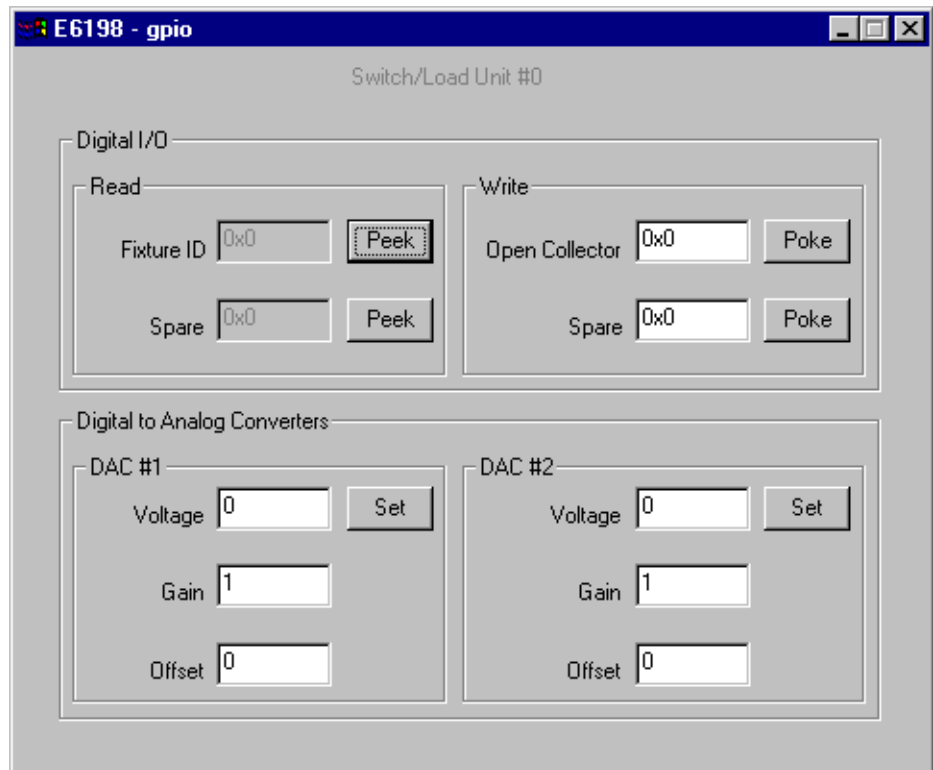


Figure 2-18. Agilent E6198A Switch/Load Unit Debug Front Panel

The debug panel boxes and buttons do the following:

Digital I/O Read:

Reads a value from the Agilent E6198 switch/load unit digital input ports. Fixture ID is value from Fix_ID(0..7) of access connector J104 and Spare is value from Spare_DigIn(0..7) of access connector J104.

Digital I/O Write:

Write a value to the Agilent E6198 switch/load unit digital output ports. Open Collector writes to output port on the switch/load unit backplane. Spare writes to output Spare_DigOut(0..7) on backplane connector J104.

Digital to Analog Converter DAC #1/DAC #2:

Change the voltage, gain, and offset values used to set the DAC voltage in the switch/load unit for both DAC #1 and DAC #2.

System Interface

The setting of the switch/load unit will use the following action routines: digitalWriteSU, digitalReadSU, dacSetDCVUSU, dacSetGainOffsetSU

**Agilent 6642A, 6643A,
6652A, 6653A, 6673A
Power Supplies**

Power Supply Debug Front Panel shows the dialog box for controlling the Agilent 6642A, 6643A, 6653A, 6673A Power Supplies. You can specify the output voltage or current for each channel of the Power Supply. You must specify a protection voltage greater than the voltage setting.

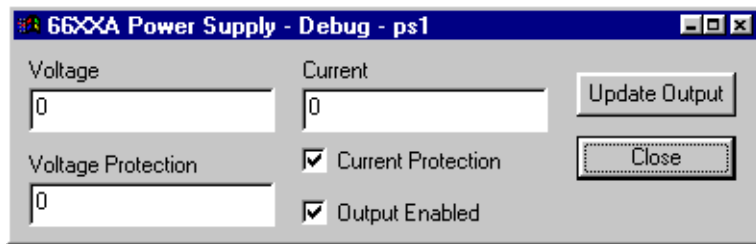


Figure 2-19. Power Supply Debug Front Panel

The debug panel boxes and buttons do the following:

Voltage/Current:

Specifies the output voltage or current for each channel of the power supplies. The 'Type' selection sets the voltage or current mode.

Dependent on the type selected, enter the voltage value or current value into the Voltage or Current field, respectively.

Update Output:

This button changes the power supply output.

Close:

This button closes the debug panel.

System Interface

The setting of the counter function uses the following action routines:
psConfVI, psSet, psIsSet, psConnect, psDisconnect

Fast Connection Selection

You can select a switching path to a module/instrument either by selecting pins on the Unit Under Test (UUT) (see “Switching Path” box in Agilent E1411 DMM Debug Front Panel (showing Path Selection)A on page 30) or by creating a custom path using the Switching Action Editor (see “Switching Action Editor” button in Agilent E1411 DMM Debug Front Panel (showing Path Selection)B on page 30).

Specifying Unit Under Test Pins

When selecting UUT pins, you normally specify from 1 to 4 connections, depending on the module/instrument. For example, for the Digital Multimeter (DMM), you define these four connections: high, low, high current sense, and low current sense. The Counter only has only a high connection.

The software works backwards though the switching configuration and generates a switching path which connects to ABusx, depending on the option selected. For example, a typical connection from ABus1 to the high input of the Digital Multimeter would be: [DVMHi ABus1 VISrcHi]

Creating a User Defined Switching Path

User defined switching paths are created using the “Select Switching Path” screen which is enabled by the “Switching Action Editor” button (see Agilent E1411 DMM Debug Front Panel (showing Path Selection)B on page 30). The “Select Switching Path” screen shows two different fields, the ‘Connect’ and ‘Disconnect’ fields. All paths added or listed in the ‘Connect’ field are always connected. All paths added or listed in the ‘Disconnect’ field are always disconnected. Note that the fields are blank, if no paths have previously been added.

To add/edit/delete a path, first enable the “Select Switching Path” screen using an appropriate instrument panel, as shown in Figure 2-20. The figure uses the Agilent E1411B Digital Multimeter instrument panel.

The following shows how to add/edit/delete a path.

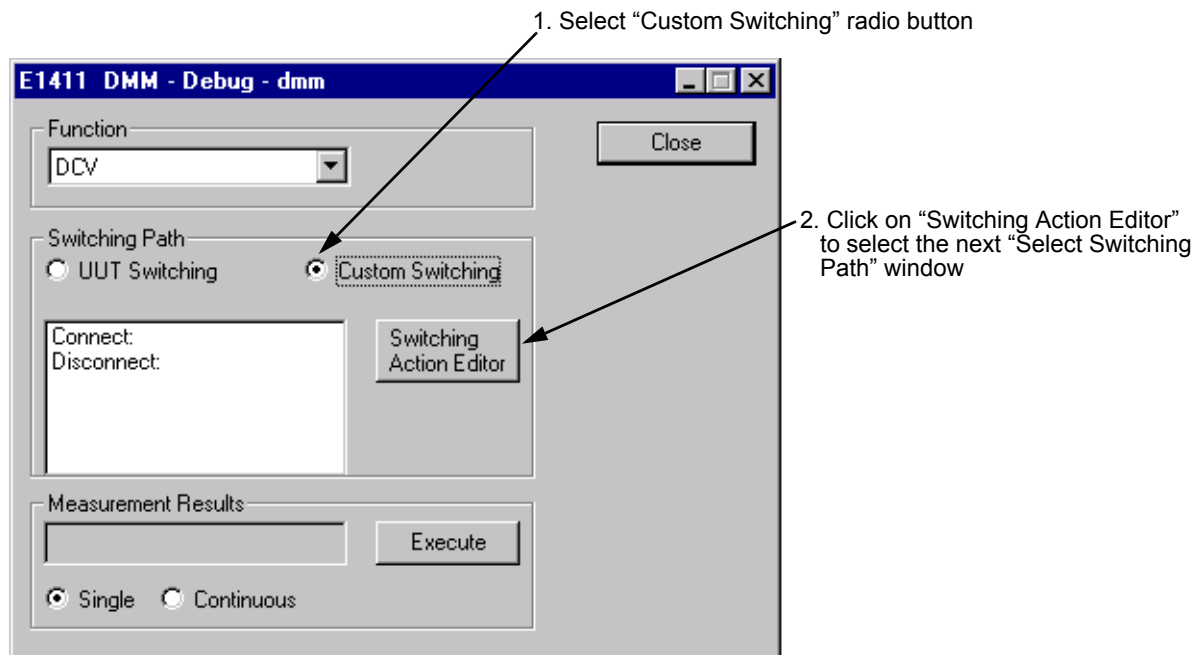


Figure 2-20. Enabling the “Select Switching Path” Screen

Adding a New Path

To add a new path, first determine the end nodes or points in the path. Then use the “Switching Action Editor” to find a path using the intermediate nodes between the end notes or points.

For example, to connect the Hi input of the Agilent E1411B Digital Multimeter (DMM) to the Hi output of the Agilent E6171 Measurement Control Module (MCM), you must know the node names for these connections. Also, since the DMM has no direct connections to the VI source or MCM, it must connect using ABusx. Thus, the nodes for this path would be: [DVMHi ABus1 VISrcHi].

Before adding a new path, first use the procedure in Figure 2-20 to enable the “Select Switching Path” screen. Then use the procedure in Figure 2-21 to add the path.

Editing a Switching Path

To edit a path, first use the procedure in Figure 2-20 to enable the “Select Switching Path” screen, if the screen is not enabled. Then use the procedure in Figure 2-22 to edit the path.

Deleting a Path

To delete a path, first use the procedure in Figure 2-20 to enable the “Select Switching Path” screen, if the screen is not enabled. Then use the procedure in Figure 2-23 to delete the path.

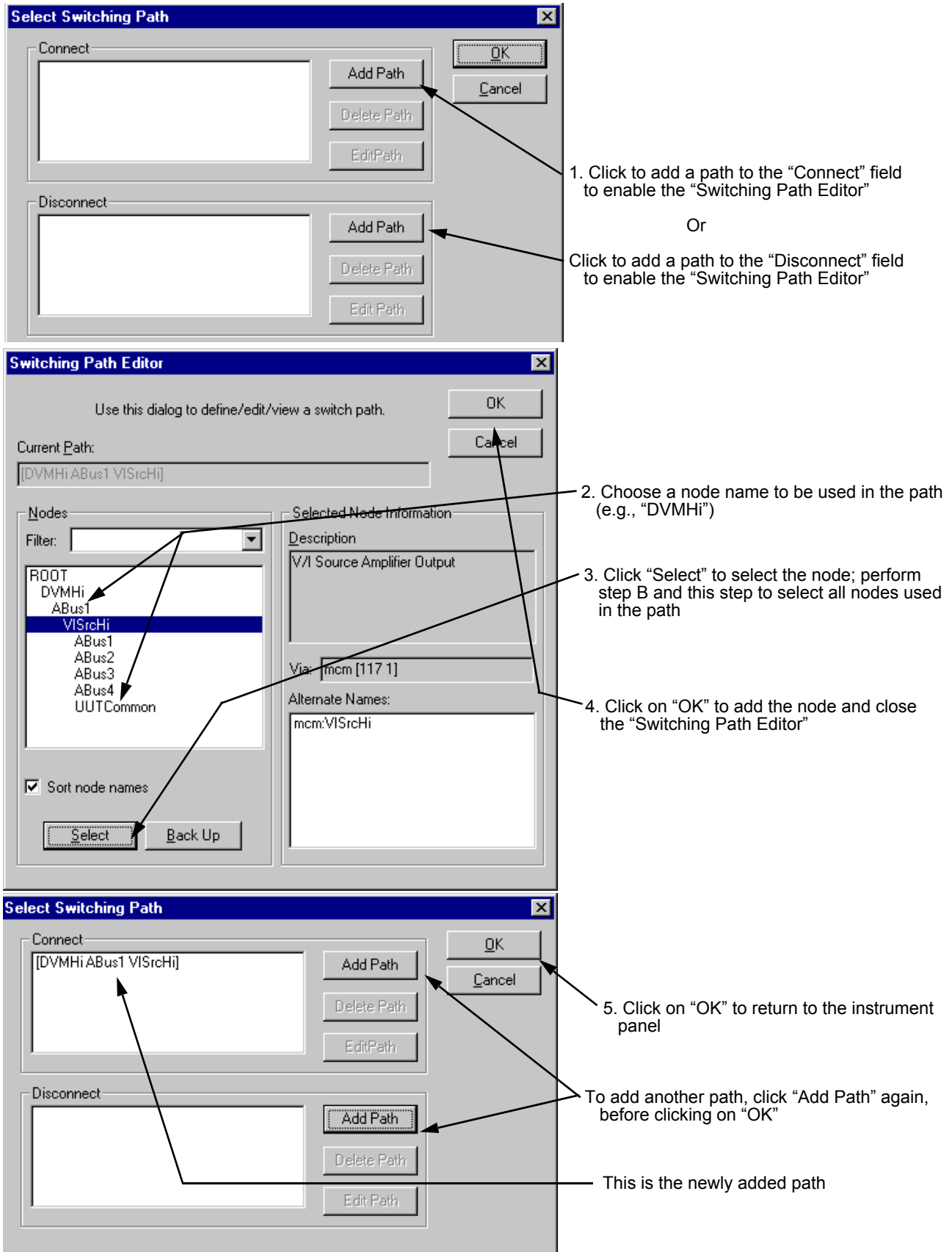


Figure 2-21. Adding a New Switching Path

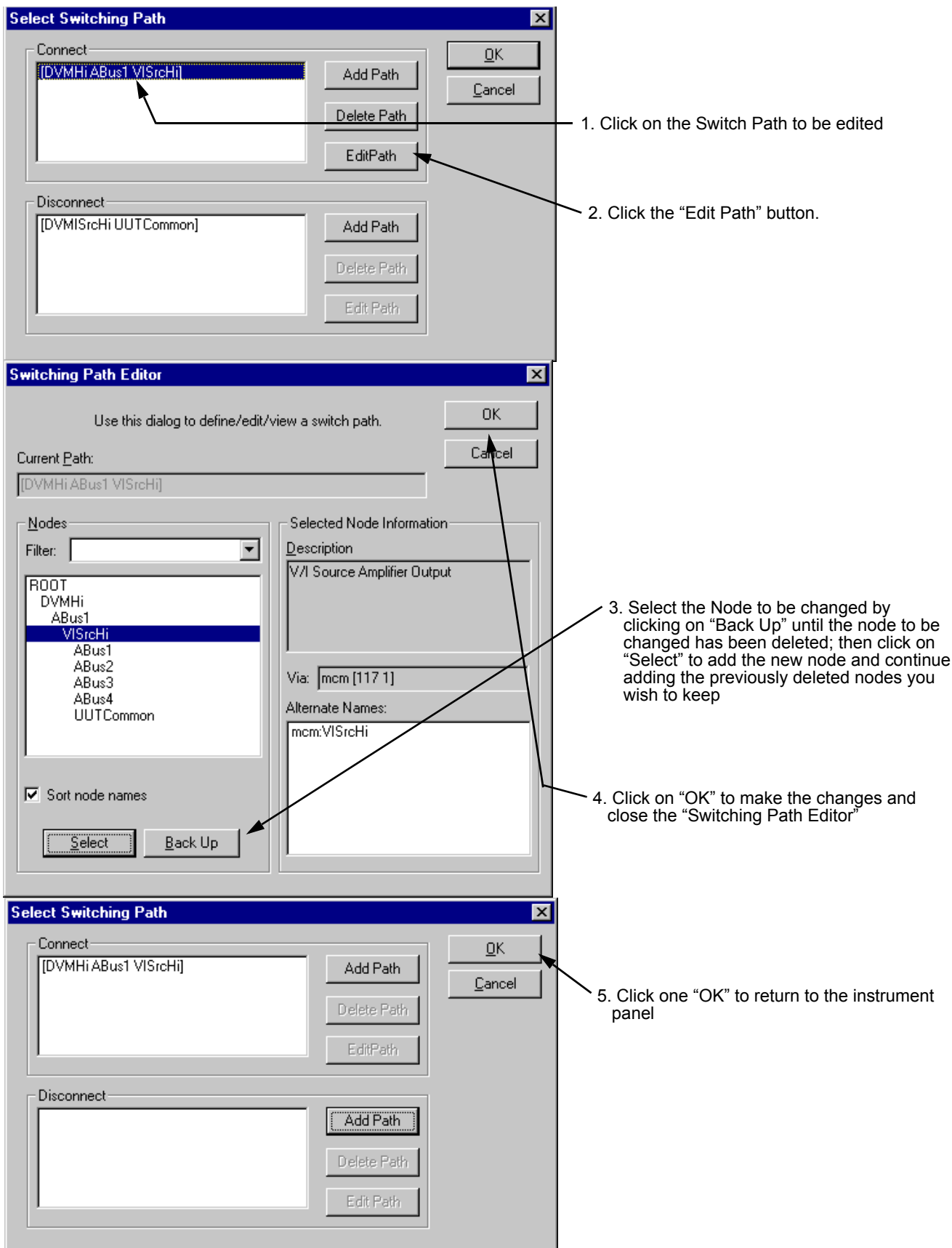


Figure 2-22. Editing a Switching Path

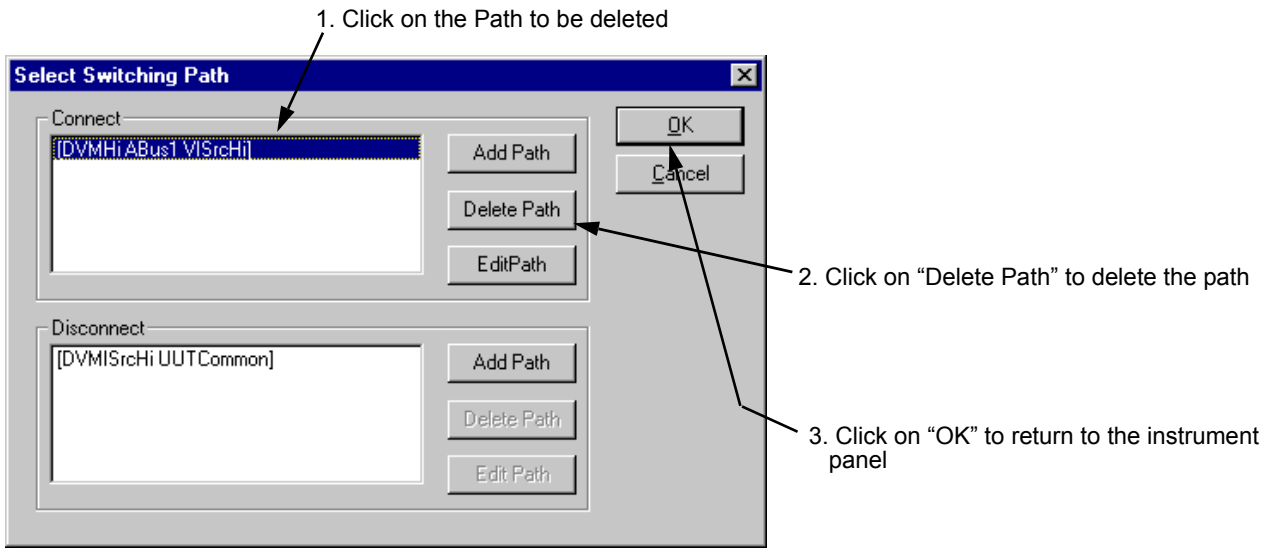
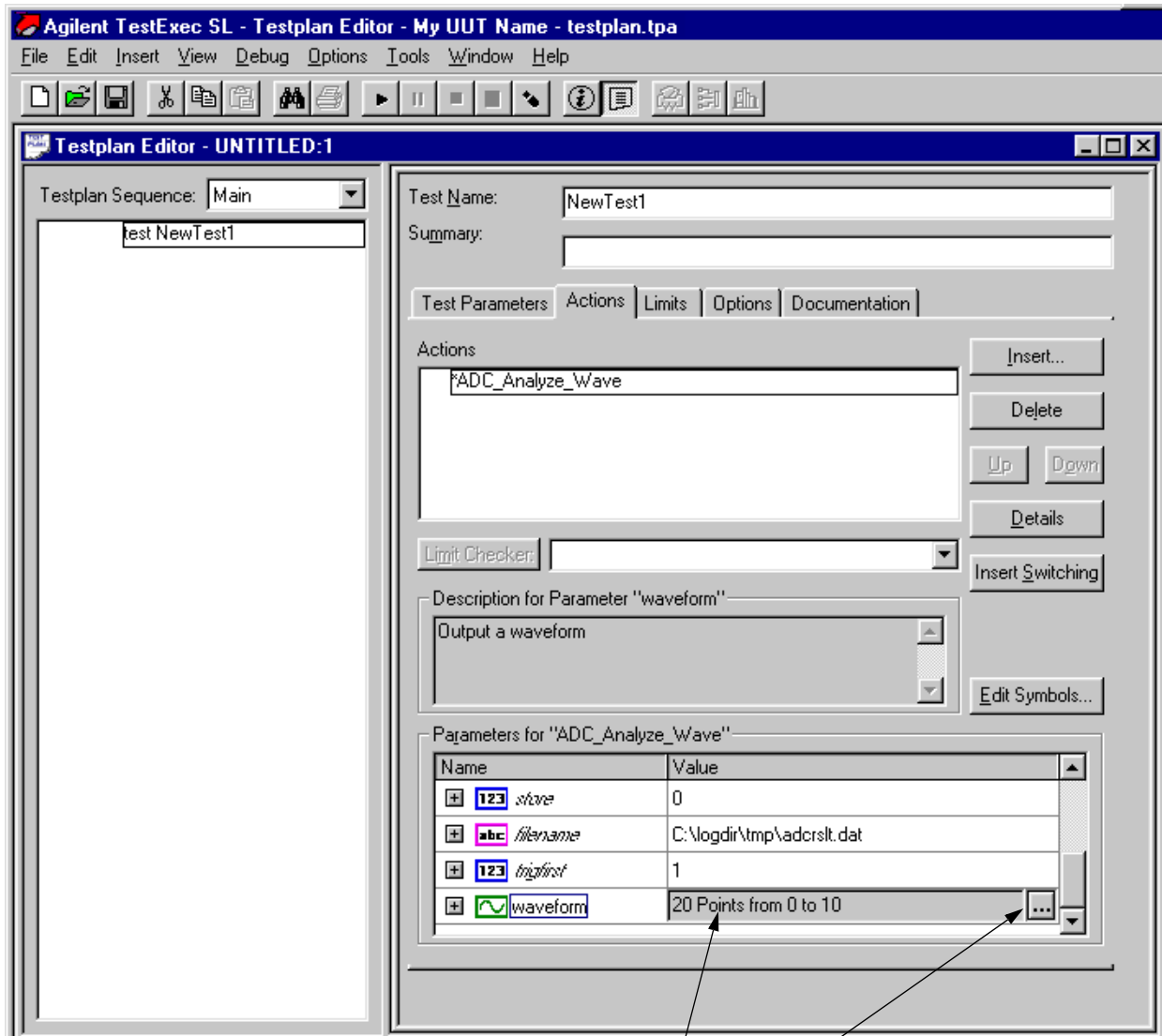


Figure 2-23. Deleting a Switching Path

Viewing Waveforms

A 'Waveform' data type is available in the "ADC_Analyze_Wave" High-Level Action. This data type includes a graphical viewer for data. Sample Testplan Showing Waveform Data Type shows a sample testplan using the "ADC_Analyze_Wave" Action with the Waveform parameter. Range Data on the Waveform Display Graphical Editor and Sample Waveform Display show the waveform editor, and Figure 2-27 shows the waveform display.



Click on the parameter value to select it then click on the dots (...) to enable the waveform editor

Figure 2-24. Sample Testplan Showing Waveform Data Type

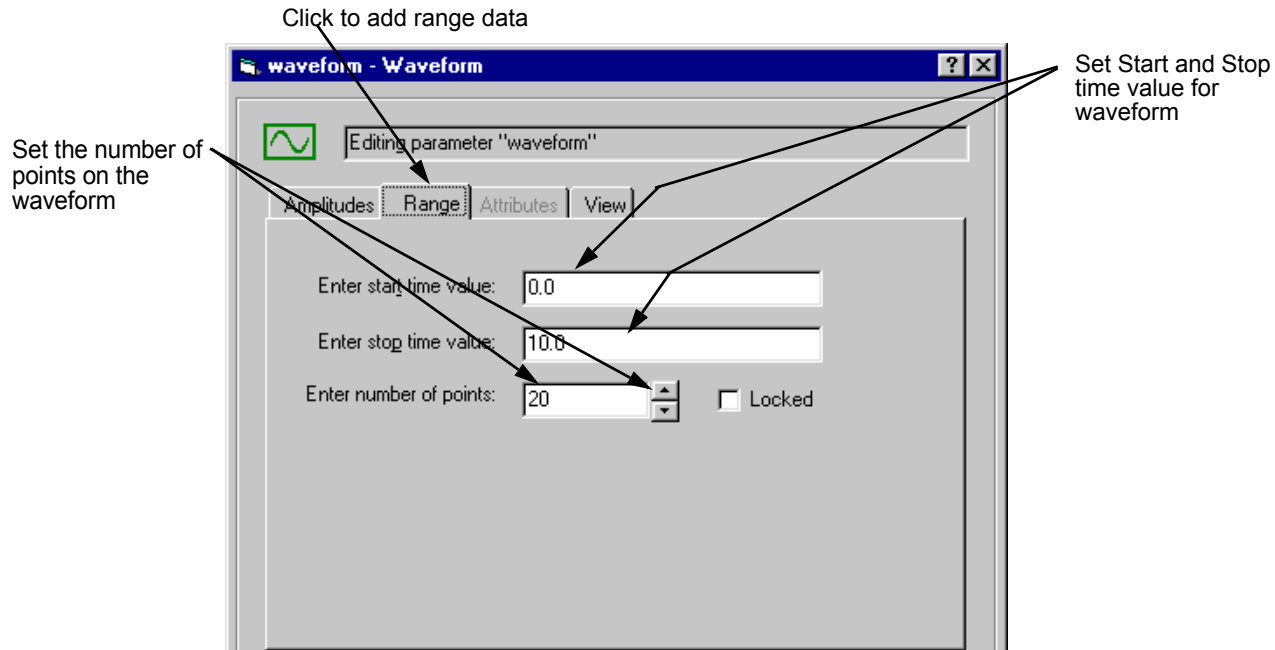


Figure 2-25. Range Data on the Waveform Display Graphical Editor

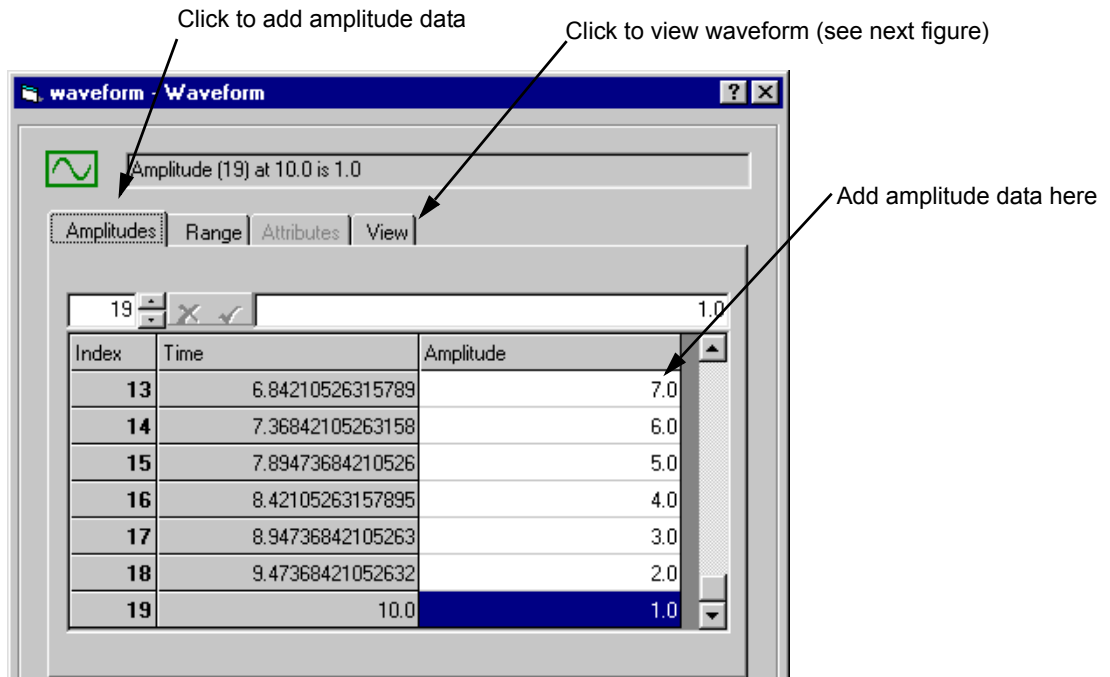
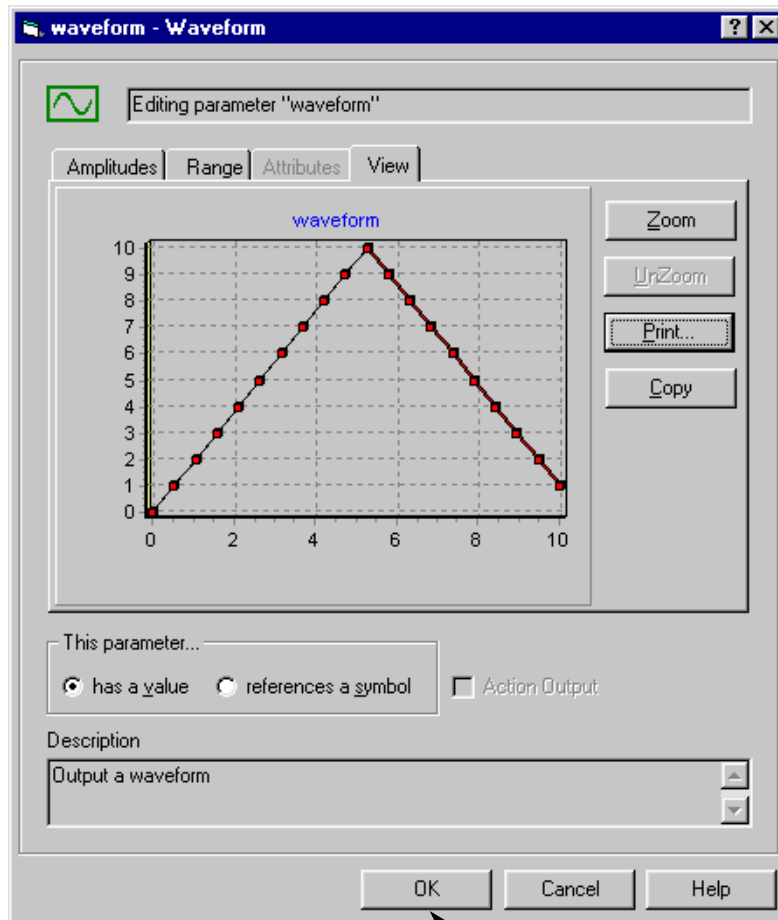


Figure 2-26. Amplitude Data on the Waveform Display Graphical Editor



Click on "OK" to store waveform

Figure 2-27. Sample Waveform Display

Instrument Handlers

Instrument Handlers are a layer of software between Agilent TestExec SL and standard instrument drivers (see Figure 2-28). In general, Instrument Handlers are designed to be called from C/C++ code action.

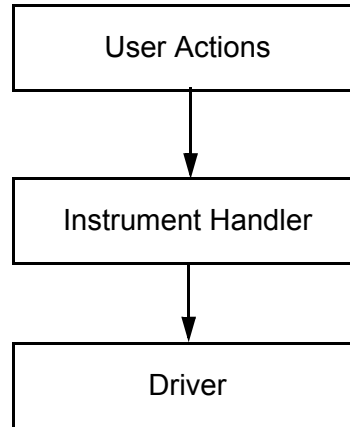


Figure 2-28. Software Layers

Instrument handlers contain functions written in C code that are organized by instrument type and function, and require parameters relevant to the function. For example, the call to set up triggering for a voltmeter is:

```
dmmConfTrigIn (dmm, trigselect, count, delay);
```

Function `dmmConfTrigIn` accepts four parameters: `dmm`, `trigselect`, `count`, and `delay`. The generic name of the function is `ConfTrigIn`, and the name of the instrument, in this case `dmm`, is added as a prefix to form the full, specific name of the function.

Note Instrument handlers are maintained constant in different TS-5000 software releases, which is not the case with the drivers. Thus, use instrument handlers for actions whenever possible.

Using the Action Wizard To Develop

The Agilent TestExec SL application used in the TS-5000 System comes with a program called Action Wizard to develop actions. The Action Wizard automatically runs through the steps necessary to develop an action. The wizard also includes a help file with more detailed instructions and information on how to use the wizard.

To run the “Action Wizard”, use the procedure in Figure 2-29.

Note Actions are the smallest components of a test and are used to setup and execute instruments, perform cleanup functions, and to make measurements.

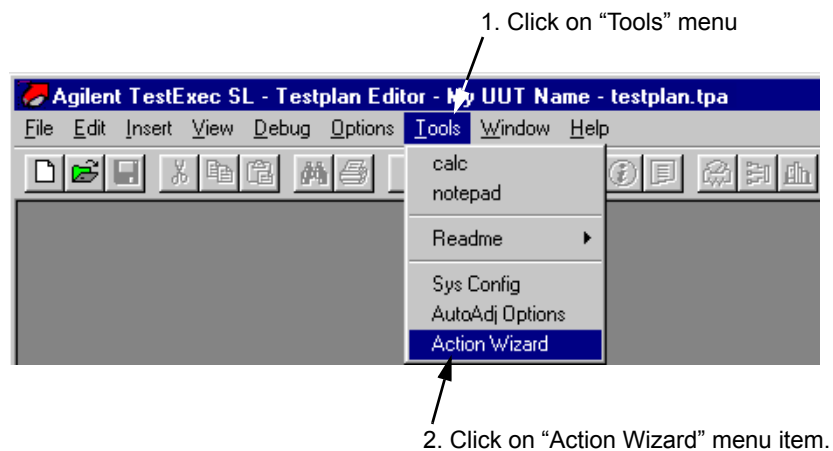


Figure 2-29. Selecting and Running the Action Wizard

Multithreading

Multithreading is an execution model which allows multiple threads to exist within a single process. It contains more than one unit of execution in a single process. In the context of TestExec SL, multithreading is the capability of running more than one thread. These threads share the process's resources but are able to execute independently.

A thread is a basic unit of execution. A single thread executes a series of application instructions, following a single path of logic through the application. All applications have at least one thread, but you can design your applications to use multiple threads, with each thread executing separate logic.

This capability becomes useful when a process thread stalls due to necessary data that are not yet available or when switching to another thread will yield a better result. In a single threaded program, once the main execution thread is blocked the whole application comes to a standstill. However, with multithreading, blocked programs can be moved from the main execution to a separate thread and can be concurrently executed with the main execution. This allows the application to stay responsive while executing tasks in the background. Multithreading can lead to improved performance as many threads are executed concurrently.

However, there are always disadvantages to complex application like this. Executing a multithreaded test can be more difficult than a typical test as there can be time related defects. In order for the operating system to track a large number of threads, it is going to consume processor time. If there are too many threads, then each thread may not be given enough time to execute during its time slice. In addition, each thread is scheduled for execution less frequently due to the volume and time slice committed to each thread.

How Multithreading is working on a Testplan?

a. Creating a Thread Safe Action

Thread safe is a concept that can be applied in multithreading. A piece of code is thread-safe, if it functions correctly during simultaneous execution in multiple threads. By using thread-safe routines, the risk that one thread will interfere and modify data elements of another thread is eliminated by circumventing potential data race situations with coordinated access to shared data.

For a thread to be thread-safe, it first must behave correctly in a single-threaded environment. Furthermore, for a thread to be thread-safe, it must continue to behave correctly when accessed from multiple threads, regardless of the scheduling or interleaving of the execution threads by the runtime environment and without any additional synchronization on the part of the calling code. The effect is that operations on a thread-safe object will appear in a fixed, globally consistent order to all threads.

The thread safe declaration is introduced as an indicator. By default, an action is not thread safe and is indicated by a visual cue.

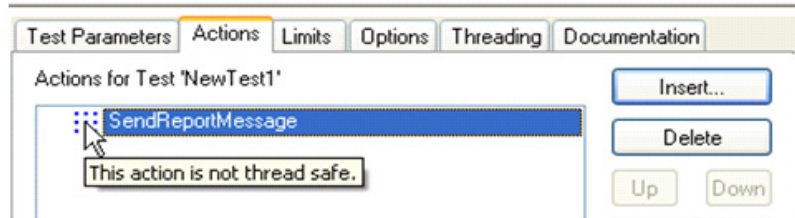


Figure 2-30. Symbol of thread safe declaration

If you do not wish to see a visual cue next to non-thread safe action, click Options | Testplan Options and in the Threading tab, uncheck the Mark non-thread-safe action checkbox.

Note: If check “Force testplan to run in sequential mode” checkbox, all testplans will run in sequential mode. For multithreading mode, the checkbox must be unchecked.

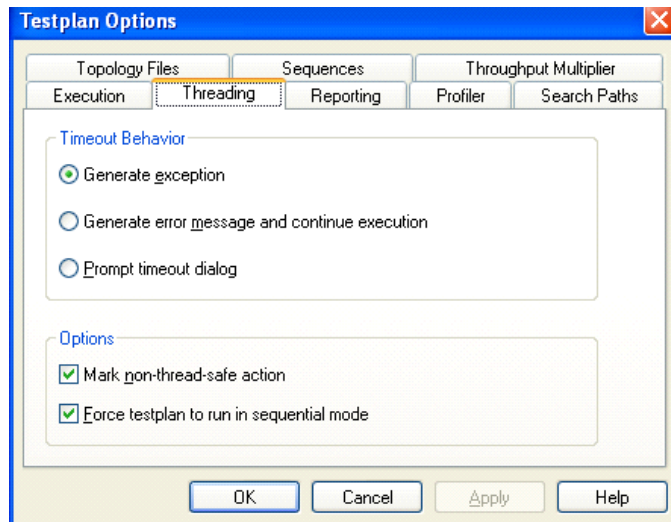


Figure 2-31. Testplan Option for Mark non-thread-safe action

To create a thread safe action,

1. Double-click an action and the Action Definition Editor window will appear.
2. Enter "thread_safe" under Keywords.
3. Click Add.

You should now be able to see thread_safe in the right column. Take note of the highlighted areas.

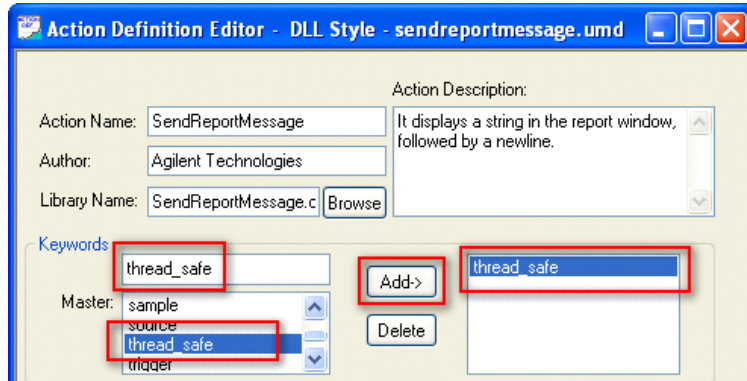


Figure 2-32. Action Definition Editor

4. The visual cue will disappear once the action has been declared as thread safe.

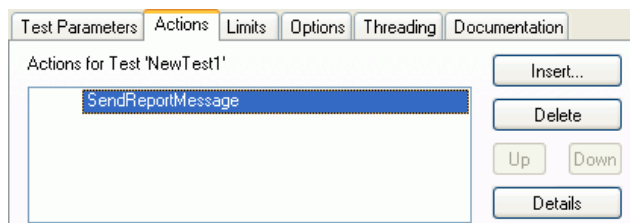


Figure 2-33. Disappear Thread Safe's symbol

b. Settings for a Threaded Test

To thread a test, select the Threading tab and check the Thread This Test checkbox. A threaded visual cue will appear next to the test once the test has been threaded, as shown below.

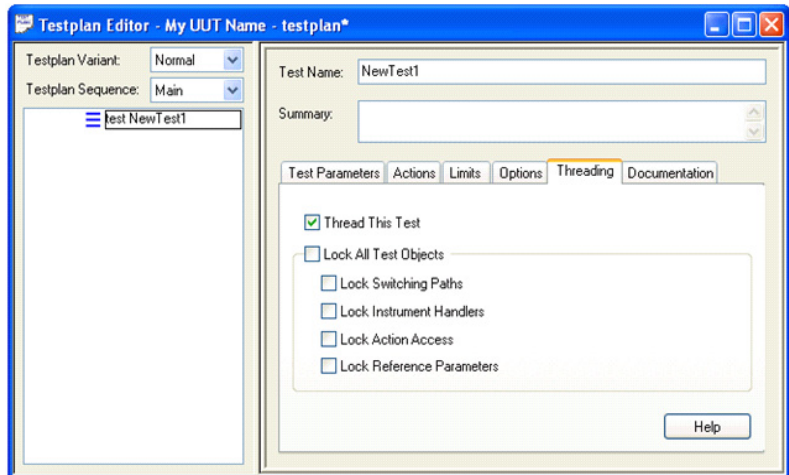


Figure 2-34. Thread This Test

c. Using Resource Locking

Resource locking is the process of protecting data in a multiprocessing environment. It means at the same point of time only one thread can access to this method of created object. Unless the lock is released after completion of the code, the next routine or iteration cannot enter the block. In the context of TestExec SL, resources that can be locked are switching paths, instrument handlers, action access, and reference parameters.

TestExec SL supports two types of resource locking which are full locking and partial locking. By implementing multithreaded tests and choosing the correct resource locking that fits your test, you can easily control the sharing of instruments and synchronize multiple UUTs. In full locking mode, TestExec SL will automatically perform locking on all objects underneath the test objects which are switching paths, instrument handlers, action access, and reference parameters. In partial locking mode, TestExec SL allows you to manually apply one or more of the above mentioned locks to the object(s).

Resource locking can be applied to each of the containing objects under the test object. Tests can be manually locked by selecting the Threading tab. Click the Lock All Test Objects checkbox for a full lock or click an individual lock to choose your preferred lock. The visual cues for full locking and partial locking slightly differ in color.

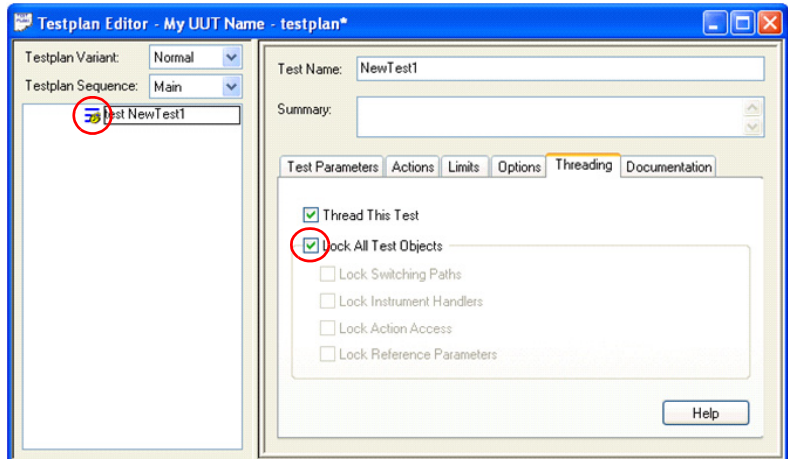


Figure 2-35. Lock All Test Objects

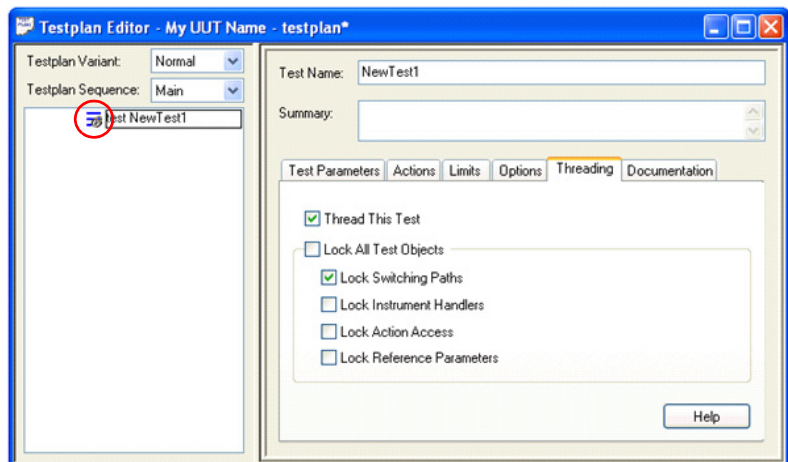


Figure 2-36. Lock Test Objects partially

d. Using the Wait Thread Statement

One way to think of the wait thread statement is to imagine an item of data such as an integer variable as if it were a field in a database. If you do not have some locking mechanism in the database you stand a chance of corruption to the data.

Thus, one user might retrieve the data and perform a calculation and write back the data. If in the meantime someone else has retrieved the data, performed the calculation and written it back, the second user's calculations will be lost when the first person writes back to the database. In the way that a database has to handle updates at unpredictable times, so a multithreaded program has to cater for this possibility. TestExec SL has a wait thread statement especially to resolve matters like this.

The Wait Thread statement allows you to block remaining statements from execution prior to completion of threaded tests statement in the Wait For column.

To insert a Wait Thread statement, click Insert | Wait Thread in TestExec SLs menu bar. The maximum timeout value is used to specify the maximum duration a Wait Thread statement would need to wait for the threads. A timeout of 0 ms means the program will wait forever until all the waited threaded tests have completed execution.

The below testplan shows that NewWaitthread needs to wait for NewTest1 to complete execution.

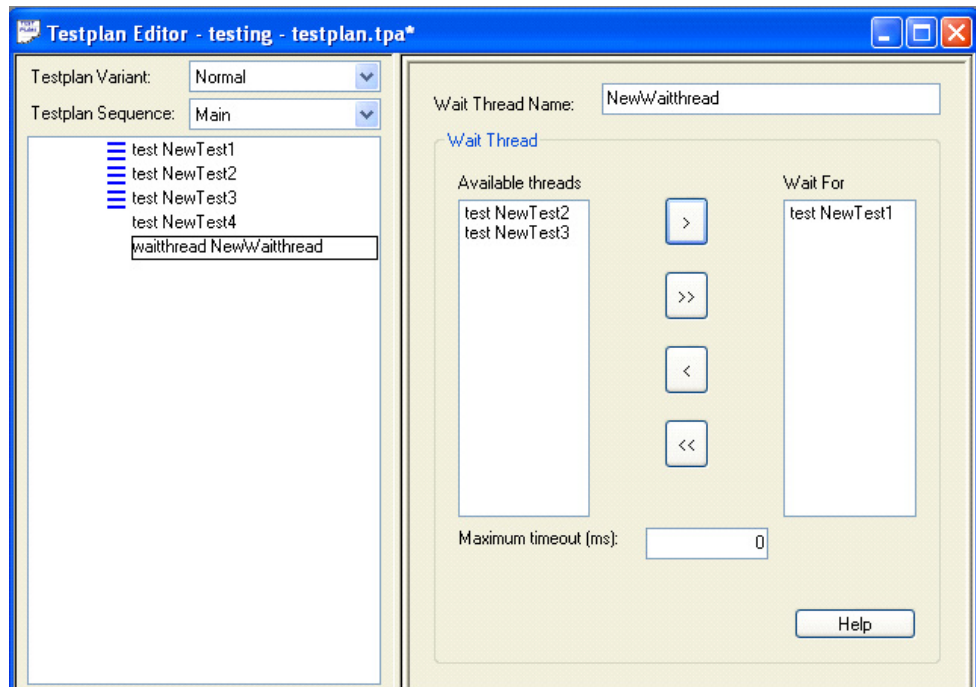


Figure 2-37. Wait Thread statement

Using Wait Thread Statement in Loop, For...In and For...To...Step Statements

It is recommended to insert Wait Thread statement right before Next statement in Loop, For...In and For...To...Step statements if threaded tests exists within these statements. The Wait Thread statement needs to wait for all threaded tests within the Loop and For Loop statements, otherwise, the results might be unexpected.

Using Wait Thread Statement for Throughput Multiplier Threaded Test Statement

It is also recommended to insert Wait Thread statement after a throughput multiplier threaded test statement to wait for all the threads running for all the UUTs to complete execution before proceeding on with next statement.

Using Wait Thread Statement in Testgroup Statement

A Wait Thread statement should be inserted right before end testgroup statement if threaded tests exist within the testgroup statement. The Wait Thread statement needs to wait for all threaded tests within the testgroup, otherwise, the results might be unexpected.

Using Wait Thread Statement in Sequence Library

A Wait Thread statement should be inserted as the last statement of a sequence library (local or external) if threaded tests exist in the sequence library. The Wait Thread statement is to ensure all the threaded tests in the sequence library have completed their execution before returning to Main sequence. If the Wait Thread statement is not inserted, threads in the sequence library might be still running upon returning to Main sequence.

e. Using Timeout

A specified period of time that will be allowed to elapse in a system before a the next specified event is to take place. TestExec SL allows you to specify a timeout value as well as the behavior of the timeout. The timeout value is set in the Wait Thread statement while it's behavior is set in Testplan Options.

In Options | Testplan Options, a new Threading tab has been added to allow users to specify the behavior of the timeout. There are three options you can choose from:

- Generate exception
- Generate error message on report window and continue execute
- Prompt timeout dialog

Threaded objects are killed when timeout is exceeded. This timeout feature is the simplest method of preventing of deadlocks.

Chapter 3

Using the DMM and ARB

Chapter Contents

This chapter describes how to trigger the DMM and how to use the ARB. The chapter is separated as follows:

- Triggered Voltage Measurement page 58
- Using the Agilent E6173A Arbitrary Waveform Generator . page 61

Triggered Voltage Measurement

The Agilent E1411B Multimeter (DMM) can be triggered by a signal on the VXI Mainframe's backplane. This signal uses one of the TTL (i.e., TTL0 to TTL7) trigger lines.

How to Trigger the DMM

Testplan 'dmmtrig.tpa' shows how the DMM is triggered using the Agilent E6174A Arbitrary Waveform Generator (Arb). The Arb is also used to generate the signal measured by the DMM. The testplan also demonstrates how to use the Agilent E6171B Measurement Control Module's (MCM) analog comparator. The testplan is in the following directory:

```
C:\Program Files\Agilent\TS-5000 System  
Software\testplan\examples
```

The Arb generates a $\pm 5V$ 10 Hz square wave as the trigger signal. This method uses the analog comparator of the MCM to send the trigger to the DMM using the VXI Mainframe's backplane (see connections in Figure 3-1 and Figure 3-2).

The testplan configures the MCM analog comparator to trigger at 0 volts. Setting the "trigedge" parameter in the "viConfCompare" action to "0" (positive edge detection), causes negative going backplane trigger when the input signal makes a low to high (positive going) transition through the trigger threshold. Setting the "trigedge" parameter in the "viConfCompare" action set to "1" (negative edge detection), causes triggers when the input signal makes a high to low (negative going) transition.

The following is a review of the testplan

<u>Test/Action Name</u>	<u>Description</u>
test: Dmm Init	Initializes the Digital Multimeter.
dmmConfCal	Disables autozero and set the power line frequency.
dmmIsSet	Waits for setup to complete.
test: Arb 1 pgm	Configures Arb and outputs a 10 Hz +5V Square wave.
arbReset	Reset the Arbitrary Waveform Generator (Arb) to its turn-on state.
arbConfOutControls	Configures the Arb's output circuitry.
Arb_DI_Std_Waveform	Download a 10 Hz square wave into the Arb.
arbInitiate	Start outputting the waveform.
Arb_Select_Wave_By_Name	Select waveform downloaded above.
test: VI Config0 (+5)	Configures MCM's analog comparator to trigger at 0V and a low to high (positive transition) trigger.
viConfCompare	Configures the comparator. The "trigedge" parameter value (0) causes a backplane trigger when the input signal makes a low to high (positive going) transition trough trigger threshold.
vilsSet	Waits for setup to complete.
Test: Meas V - Trigger	Trigger DMM and measure trigger voltage.

Test/Action Name	Description
Switching	Make the necessary switching to setup the Arb to DMM measurements and trigger signal paths.
dmmMeasureTrigVoltage	Configures DMM for a measurement using the backplane TTL trigger line 0.
dmmGetResults	Make the measurement and return the reading.
test: VI Config1 (-5)	Configures MCM's analog comparator to trigger at 0V and a high to low (negative transition) trigger.
viConfCompare	Configures the comparator. The "trigedge" parameter value (0) causes a negative going backplane trigger when the input signal makes a high to low (negative going) transition trough trigger threshold.
viIsSet	Waits for setup to complete.
test Meas V - Expanded	Trigger DMM and measure trigger voltage.
Switching	Make the necessary switching to setup the Arb to DMM measurements and trigger signal paths.
DelayMillisecond	Wait for switching to complete.
dmmInitiate	Initiates a reading cycle.
dmmGetResults	Make the measurement and return the reading.

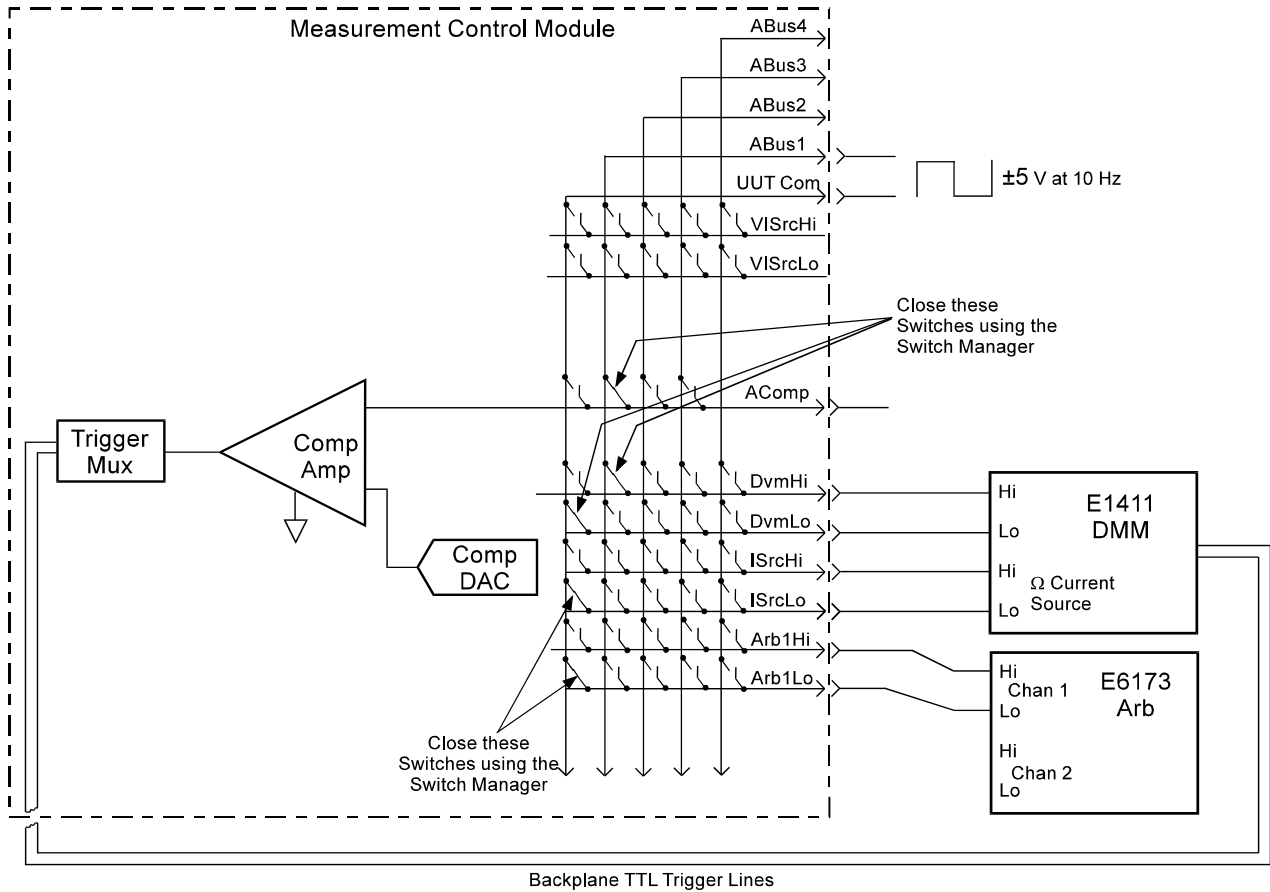


Figure 3-1. Trigger Connections

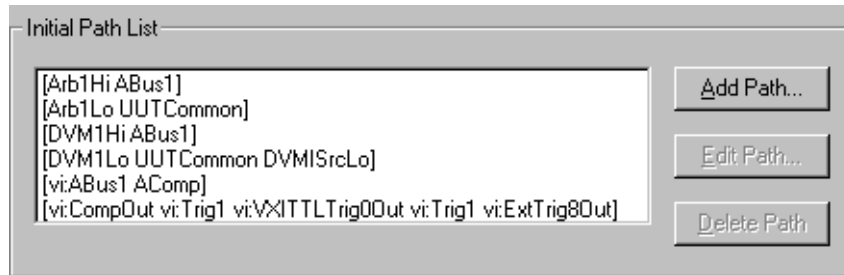


Figure 3-2. Switch Paths Listing

Using the Agilent E6173A Arbitrary Waveform Generator

The following describes the Agilent E6173A Arbitrary Waveform generator and how to use it.

Arbitrary Waveform Generator Description

The Agilent E6173A Arbitrary Waveform Generator (Arb), formerly the Agilent Z2471A, is a register-based, two channel signal generator whose channels are isolated from each other and from ground. The factory default configuration connects first channel to the Agilent E6171 Measurement Control Module (MCM).

You can amplify channel one's output using the MCM's V/I amplifier/current source. You can connect channel two directly to your Unit Under Test (UUT) or use one of the unassigned MCM inputs. The Arb can also switch to any UUT pin by switching to Aux ports, if using the default system pins.

The Arb can output standard waveforms, such as sine, square, triangle, knock, or crank of a specified frequency, amplitude, and offset, and custom arbitrary waveforms defined by the user.

How the Arb Generates Waveforms

The Arb generates waveforms using pre-defined or user defined segment and sequence data. Figure 3-3 shows a typical example of a waveform.

The segments of a waveform are the voltage levels of each point on the waveform. The sequence consists of a group of pre-defined or user defined number of segments. To generate a waveform, the Arb outputs the sequence, consisting of the defined number of segments, using a pre-defined time.

The number of segments output (i.e., how often they occur) and the time of the Arb's internal clock (i.e., segment duration time) determines the time or frequency of the waveform to be output. The Arb's internal clock time is 2 μ S. Either single or multiple clock cycles can be used to set the appropriate frequency. Selecting the clock cycles is usually known as selecting the "dwell" count.

For example, a waveform consisting of 250 segments using two 2 μ S clock cycles (or 2 dwell counts), outputs a 1 kHz waveform, shown as follows:

$$250 * (0.000002 * 2) = 250 * 0.000004 = 0.001 \text{ S} = 1 \text{ kHz}$$

In addition to the 2 μ S clock, the Arb also has a 100 nS counter that can be used to add time in 100 nS steps to the 2 μ S clock cycles. This is used to generate frequency accurate waveforms. The counter is normally set using the "clock adjust" function. Note: Segment time = (Dwell + 1) * (2 μ S + (clock adjust * 0.1 μ S)).

Both the segment data and sequence data is *downloaded* into the Arb's memory using the standard actions supplied with the system. These actions can be used to download both standard (pre-defined) waveforms and user defined waveforms.

Note For more information about the waveform timing, refer to the *Agilent E6173A Arbitrary Waveform Generator User's Manual*.

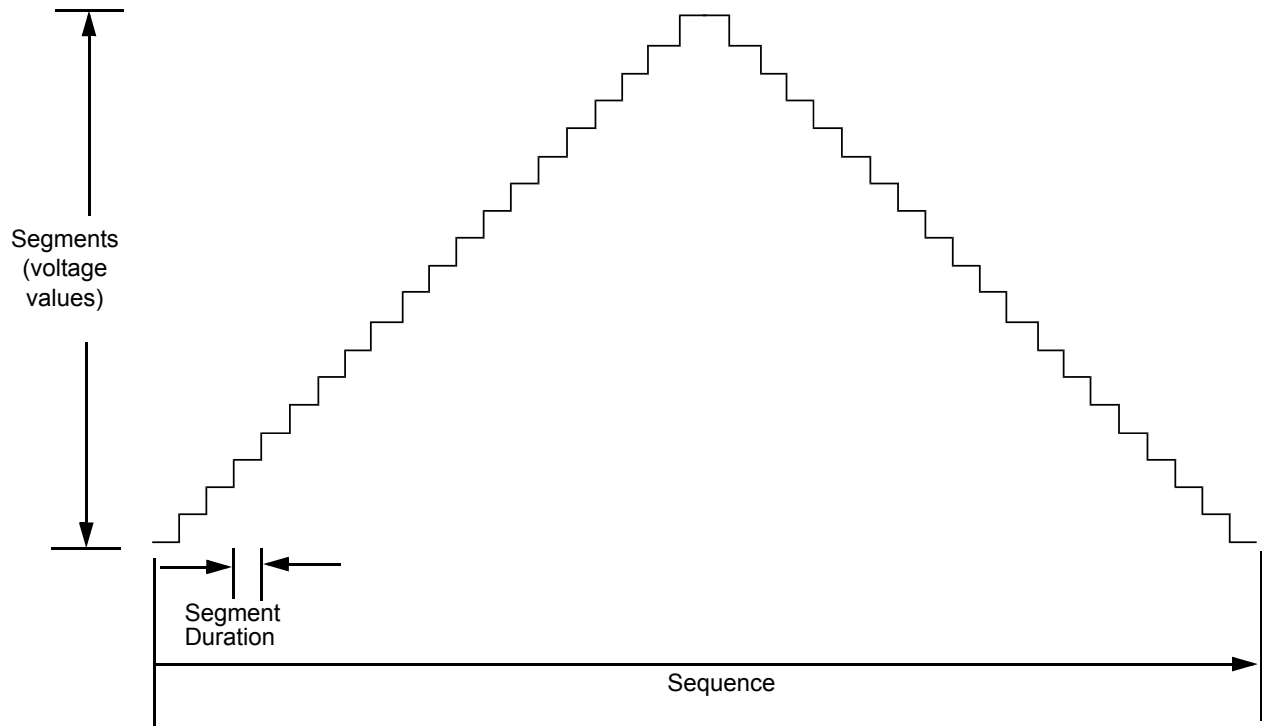


Figure 3-3. Typical Waveform using Segments and Sequence

Generating Arb Waveforms

There are several methods to generate Arb waveforms using the standard actions. You can generate standard waveforms, such as sine, pulse, and triangle waveforms, or you can create custom waveforms.

Many actions have been developed for high throughput operations. These actions are generally named “Arb_DI_XXX” where the “DI” indicates the actions with the high throughput capability (see “In all testplans, the Arb connects to the Analog Bus ABus1. To view the output, connect an oscilloscope to the ABus 1 connector at the front of the system. Be sure to make the necessary connection to the Mass Interconnect/Express Connect so the interlock is shorted in order to use the system (see the appropriate documentation). Instead of an oscilloscope, you can also use an VXI Technology E1563A Digitizer to read the waveforms and display them using the appropriate actions.” below).

Note DO NOT use the “Arb_DI_XXX” actions with the older “ArbSetSegment” and/or “ArbSetSequence” actions on the same Arb channel.

Note When selecting different downloaded waveforms, usually when there are

major frequency differences, the first few cycles of a newly selected waveform may appear to be distorted. The reason is that the previous waveform must complete, but the new “clock adjust” is already in effect, which may change the timing of the last cycles of the previous waveform.

The following explains the methods used. For each method, there is a testplan available it. The testplans are located in the following directory:

```
C:\Program Files\Agilent\TS-5000 System  
Software\testplan\examples
```

In all testplans, the Arb connects to the Analog Bus ABus1. To view the output, connect an oscilloscope to the ABus 1 connector at the front of the system. Be sure to make the necessary connection to the Mass Interconnect/Express Connect so the interlock is shorted in order to use the system (see the appropriate documentation). Instead of an oscilloscope, you can also use an VXI Technology E1563A Digitizer to read the waveforms and display them using the appropriate actions.

Downloading and Executing Waveforms Using the “Arb_DI_xxx” Actions

Using the “Arb_DI_xxx” actions to download standard or user defined waveforms gives you the highest throughput capability. Using these actions, the waveforms are downloaded the first time the testplan is run. The waveforms remain in memory and can then be selected any time during testplan execution. The only time new waveforms are downloaded is when a testplan is modified.

Each downloaded waveform needs a unique name to keep track of the waveform. This name can then be used to call a particular waveform in any part of the testplan as many times as needed. Since both waveform data and other parameters, such as output voltage are also included with a name, the waveform will be output the same way every time it is selected.

Note When initializing the Arb for output using the “arbInitiate” action, the Arb outputs the last downloaded waveform, if “Arb_Select_Wave_By_Name” is not called before “arbInitiate”. Use the waveform names in the ‘waveform_name’ parameters of the of “Arb_DI_xxx” actions to select them.

A typical sequence to generate a waveform is as follows:

1. Setup Arb using the “arbConfOutControls” action.
2. Send the setup data to the Arb using the “arbSet” action.
3. Download waveform data using the “Arb_DI_xxx” action; use a unique waveform name in the ‘waveform_name’ parameter.
4. Initialize the Arb to output the waveform using the “arbInitiate” action.
5. Select the downloaded waveform using the “Arb_Select_Wave_By_Name” action using the waveform name in the “Arb_DI_xxx” action.

Generating Standard Waveforms

The following shows how to download standard waveforms for output using the standard actions. For high throughput capability, use the “Arb_DI_XXX” actions.

Downloading a Standard Waveform

Use the “Arb_DI_Std_Waveform” action to download the waveform and the “Arb_Select_Wave_By_Name” action to select the waveform. In this case, the same waveform name is referenced by both actions to select the waveform for output.

Using this method, a waveform can be selected on-the-fly to allow fast waveform access. This is because the waveform has previously been generated and stored in memory. All that is needed is to select the waveform to output it.

An example to download and then select a waveform is as follows. In the example, a waveform is generated and stored using a particular name (i.e., “Arbwave1”). Once this waveform is generated, it can then be selected in any part of the testplan.

Generate the waveform using Arb_DI_Std_Waveform with the waveform name = “Arbwave1” (parameter: ‘waveform_name’ = Arbwave1).

to select the waveform, set:

waveform name of an Arb_Select_Wave_By_Name action = “Arbwave1” (parameter: ‘waveform_name’ = Arbwave1) and then execute that action.

Note The waveform names are NOT case sensitive.

Using the above actions allow you to download and store several different waveforms into the Arb that can be selected in different parts of the testplan. One way to do this is to download several different waveforms with different names at the beginning of the testplan and then select the different waveforms throughout the testplan.

An example to use different downloaded waveforms in different parts of a testplan is as follows:

```
Test1:  Arb_DI_Std_Waveform: 'waveform_name' = sine
        Arb_DI_Std_Waveform: 'waveform_name' = square
        Arb_DI_Std_Waveform: 'waveform_name' = triangle
        •
        •
Test2:  Arb_Select_Wave_By_Name: 'waveform_name' = triangle
        •
        •
Test3:  Arb_Select_Wave_By_Name: 'waveform_name' = sine
        •
```


Test4: Arb_Select_Wave_By_Name: 'waveform_name' = square

Testplan 'arbdnwn_name.tpa' illustrates how to download a 1 kHz sine wave, 2 kHz square wave, and 4 kHz triangle wave. A review of the testplan is below.

<u>Test Group/Test/Action Name</u>	<u>Description</u>
testgroup Arb download waveforms	Downloads standard waveforms.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb's output circuitry.
arbSet	Sends setup data to Arb.
test download sine wave	Downloads the standard sine wave to the Arb.
Arb_DI_Std_Waveform	Downloads a sine wave into the Arb. The 'function' parameter is set to "0" to download a sine wave. The 'waveform_name' parameter is set to "sine" to give the downloaded waveform that name. The 'frequency' parameter is set for a 1 kHz waveform. The 'Vpeak' parameter is set for a 5 V peak waveform. The 'waveform_name' parameter is set to
test download square wave	Downloads the standard pulse wave to the Arb.
Arb_DI_Std_Waveform	Downloads a square wave into the Arb. The 'function' parameter is set to "1" to download a pulse. The 'waveform_name' parameter is set to "square" to give the downloaded waveform that name. The 'duty_cycle' parameter is set to "0.5" so the pulse function downloads a square wave. The 'frequency' parameter is set for a 2 kHz waveform. The 'Vpeak' parameter is set for a 5 V peak waveform.
test download triangle wave	Downloads the standard triangle wave to the Arb.
Arb_DI_Std_Waveform	Downloads a triangle wave into the Arb. The 'function' parameter is set to "2" to download a triangle wave. The 'waveform_name' parameter is set to "triangle" to give the downloaded waveform that name. The 'frequency' parameter is set for a 4 kHz waveform. The 'Vpeak' parameter is set for a 5 V peak waveform.
test output waveforms	This test executes the downloaded waveforms.
arbInitiate	Initialize the Arb to output the waveform. Start outputting the waveform. The Arb at this time outputs the last downloaded waveform (i.e., "triangle").
DialogOkay	This optional action is used here to view the current waveform (i.e., triangle wave).
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the 'waveform_name' parameter (i.e., "sine"). Since this is the name of the 'waveform_name' parameter of the "Arb_DI_Std_Waveform" action in "test download sine wave", a sine wave is output by the Arb.
DialogOkay	This optional action is used here to view the current waveform (i.e., sine wave).

<u>Test Group/Test/Action Name</u>	<u>Description</u>
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the 'waveform_name' parameter (i.e., "square"). Since this is the name of the 'waveform_name' parameter of the "Arb_DI_Std_Waveform" action in "test download sine wave", a square wave is output by the Arb.
DialogOkay	This optional action is used here to view the current waveform (i.e., square wave).
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the 'waveform_name' parameter (i.e., "triangle"). Since this is the name of the 'waveform_name' parameter of the "Arb_DI_Std_Waveform" action in "test download sine wave", a triangle wave is output by the Arb.
DialogOkay	This optional action is used here to view the current waveform (i.e., triangle wave).

Outputting a Standard Waveform Immediately

The "Arb_Output_Std_Waveform" action can also be used to generate and output standard waveforms. This action enables the Arb to output a sine, pulse, or triangle wave. Unlike the "Arb_DI_Waveform" action, this action does not require "arbIntiate" to enable it to output a waveform, however, it also has the high throughput capability of the "Arb_DI_Waveform" action. "Arb_Output_Std_Waveform" can be used in any part of the testplan.

Testplan 'arboutputstd.tpa' illustrates how to output a 1 kHz sine wave, 2 kHz square wave, and a 4 kHz triangle wave. A review of the testplan is below.

<u>Test Group/Test/Action Name</u>	<u>Description</u>
testgroup Arb output std waveforms	Outputs standard waveforms.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test output sine	Output sine wave.
Arb_Output_Std_Waveform	Immediately outputs a sine wave; 'function' parameter is set to "0".
DialogOkay	This optional action is used here to view the current waveform (i.e., sine wave).
test output square	Output square wave.
Arb_Output_Std_Waveform	Immediately outputs a square wave; 'function' parameter is set to "1" (pule) and 'duty_cycle' parameter is set to "0.5" for the "pulse" function to output the square wave.
DialogOkay	This optional action is used here to view the current waveform (i.e., square wave).
test output triangle	Output triangle wave.
Arb_Output_Std_Waveform	Immediately outputs a triangle wave; 'function' parameter is set to "2".
DialogOkay	This optional action is used here to view the current waveform (i.e., triangle wave).

Downloading a Standard Phase Shifted or Burst Modulated Waveform

Use the “Arb_DI_Ext_Waveform” action for phase shifted and burst modulated waveforms. This action has the following functionality:

1. Square wave generation
2. Positive and negative ramp generation
3. Positive and negative pulse generation
4. Burst modulation
5. Phase shift
6. Pulse width for positive and negative pulses, and burst pulses.

To correctly generate a waveform, it is important to understand how to set some of this action’s different parameters. These are explained as follows:

frequency - Specifies the output frequency if the waveform is NOT a “burst” waveform (‘F_burst’ parameter set to “0”). If “burst” is used, this is the frequency of the burst wave, as shown in Figure 3-4. (Note: “burst” can not be used in the positive or negative pulse functions; i.e., functions 5 or 6, respectively).

tpulse_width - Sets the pulse width of the positive or negative pulses function (for functions 5 and 6, respectively) or sets the width, in seconds, of the burst wave, as shown in Figure 3-4. If used with “burst”, the pulse width must be less than 100 mS.

F_burst - If set to “0”, no “burst” is generated. Otherwise, this sets the waveform frequency that is to be modulated by the burst wave (the burst wave frequency is set by the ‘frequency’ parameter), as shown in Figure 3-4.

phase_shift - Specifies the amount of phase shift in a waveform in degrees, relative to a non-phase shifted version of itself.

force_seg_count - If set to “0”, the frequency accuracy is generated using the 2 μS clock cycles plus any number of 100 nS clock cycles needed to make the frequency accurate. This is illustrated in the following formula:

$$\text{segments} = (1 / \text{frequency}) / (2 \mu\text{S} + (100 \text{ nS} * \text{clockadj}))$$

where ‘clockadj’ is used to add 1 or more 100 nS ticks to the 2 μS clock.

If this parameter is set to “1” (or “-1”), the number of segments closest to the frequency value will be used, without using ‘clockadj’. Only the 2 μS clock is used without the 100 nS clock. This is illustrated in the following formula:

$$\text{segments} = 1 / (\text{frequency} * 2 \mu\text{S})$$

Normally, 'force_seg_count' is used (set to "1" or "-1") if you wish to have the segment counts of two waveforms be a multiple of each other to guarantee phase lock. However, this method uses more segment memory since the number of segments determines the frequency.

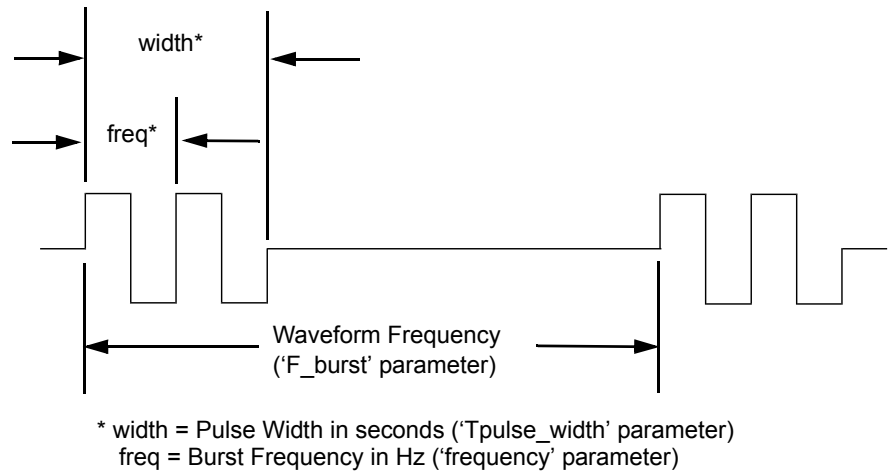


Figure 3-4. Typical Burst Waveform

Testplan 'arbburst.tpa' illustrates how to generate a burst modulated waveform, with waveform frequency set to 30 kHz, burst frequency set to 1 kHz, and burst pulse width set to 0.0001 seconds. A review of the testplan is below.

Test Group/Test/Action Name	Description
testgroup Arb burst waveform	Outputs a burst waveform.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb's output circuitry.
arbSet	Sends setup data to Arb.
test download burst	Setup arb to download a burst waveform.
Arb_DI_Ext_Waveform	Setup a burst waveform with waveform frequency set to 30 kHz ('F_burst' parameter), burst frequency of 1 kHz ('frequency' parameter), and a burst pulse width of 100 μS ('Tpulse_width' parameter).
test output waveform	Selects and outputs a burst waveform by name.
arbInitiate	Initialize the Arb to output the waveform. Start outputting the waveform. The Arb at this time outputs the last downloaded waveform.
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the 'waveform_name' parameter (i.e., "wave1").
DialogOkay	This optional action is used here to view the current waveform (i.e., forward and reverse sweep).

Downloading a Swept Sine Waveform

The “Arb_DI_Swept_Sine” and “Arb_DI_Swept_Sine_Ex” actions perform frequency sweeps of sine waves. The “Arb_DI_Swept_Sine” action only sweeps in the forward direction. The “Arb_DI_Swept_Sine_Ex” action can sweep in both forward and reverse directions with a selectable delay between sweeps (i.e., last frequency remain for the selected time).

Testplan ‘arbsweep.tpa’ illustrates how to output a sine wave sweep using both actions. In one test, the sine wave sweeps from 1 kHz to 2 kHz with a sweep time of 0.5 seconds. In another test, the sine wave sweeps from 1 kHz to 2 kHz in both forward and backwards directions with a sweep time of 0.2 seconds. In the latter test, the last frequency is held for 0.5 seconds before the next sweeps is initiated. A review of the testplan is below.

Note The number of segments needed to generate the swept sine waves depend on the frequency span between the start and end frequencies, and the sweep time. Large spans and/or sweep times could exceed the maximum allowable segments.

Test Group/Test/Action Name	Description
testgroup Arb sine sweeps	Sweeps Arb sine waves.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb’s output circuitry.
arbSet	Sends setup data to Arb.
test setup forward sweep	Sweeps a sine wave in the forward direction.
Arb_DI_Swept_Sine	Downloads a sine wave for a forward sweep from 1 kHz to 2 kHz for a time of 0.5 seconds.
test setup forward & reverse sweeps	Sweeps a sine wave in both forward and reverse direction.
Arb_DI_Swept_Sine_Ex	Downloads a sine wave for a forward and reverse sweep from 1 kHz to 2 kHz for a time of 0.2 seconds with a delay of 0.5 seconds between sweeps.
test execute sweeps	Execute both sweeps.
arbInitiate	Initialize the Arb to output the waveform. Start outputting the waveform. The Arb at this time outputs the last downloaded waveform (i.e., forward and reverse sweep).
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the ‘waveform_name’ parameter (i.e., “wave1”).
DialogOkay	This optional action is used here to view the current waveform (i.e., forward sweep).
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the ‘waveform_name’ parameter (i.e., “wave2”).
DialogOkay	This optional action is used here to view the current waveform (i.e., forward and reverse sweep).

Note Use the Watch Window to determine the amount of segment memory used by the swept sine waveforms.

Generating Custom Waveforms

Custom waveforms can be generated using the “Arb_DI_Waveform_Data” action and or the “Arb_DI_Custom_Waveform” actions.

Downloading a Data Type Custom Waveform

The “Arb_DI_Waveform_Data” action uses a ‘Waveform’ type array variable called “Waveform” to generate a custom waveform. The variable stores both the segments of the waveform (i.e., amplitude) and the time all segments are to be executed (i.e., waveform frequency).

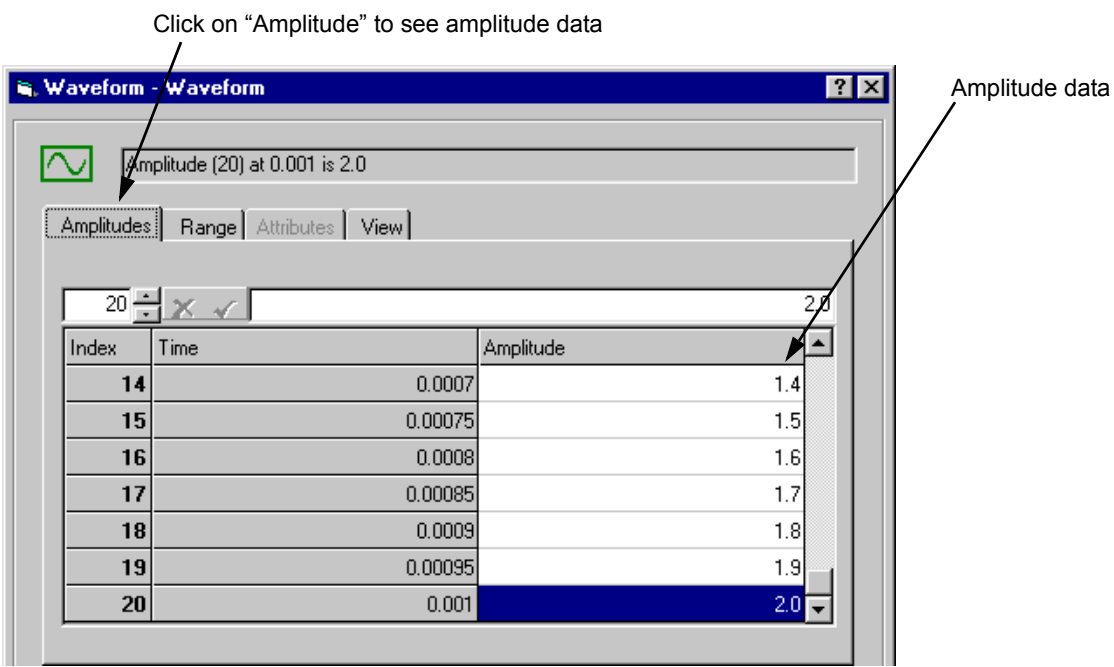
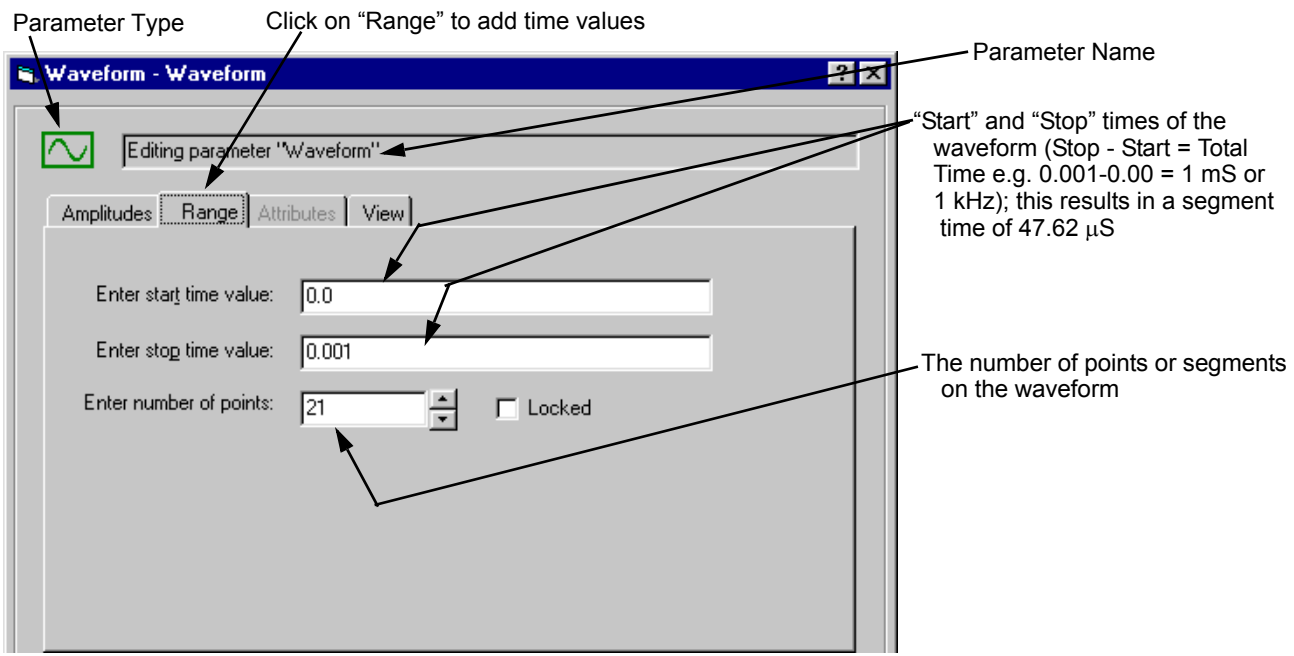
Figure 3-5 shows the data used to generate the waveform shown in the bottom part of the figure. The segment data of the waveform is stored in the array part of the “Waveform” parameter. (You can view the resultant waveform by clicking on the “View Waveform” button, as shown in Figure 3-5.)

How to download custom data type waveforms is shown in testplan ‘arbcustom_data.tpa’. The waveform is a 1 kHz (1 mS), 2 V peak ramp, consisting of 21 segments. A review of the testplan is below.

Test Group/Test/Action Name	Description
testgroup download custom waveform	Downloads a user generated data type custom waveform.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb's output circuitry.
arbSet	Sends setup data to Arb.
test download custom waveforms	Downloads custom waveforms to Arb.
Arb_DI_Waveform_Data	Downloads data type waveform into Arb. The waveform data is the data stored into the ‘Waveform’ variable (see Figure 3-5 for typical waveform data). The ‘waveform_name’ parameter is set to “wave1” to give the downloaded waveform that name. The maximum value in the ‘Waveform’ array variable is “2.0000000000” which makes the peak Waveform voltage 2.0 V. The Star and /Stop values in the ‘Waveform’ array are set to “0.0000000000” and “.0010000000”, respectively. This outputs a 1 kHz waveform (Stop - Start = .0010000000 - 0.0000000000 = 1 mS or 1 kHz).
test output custom waveform	This test executes the downloaded waveform. Although the Arb outputs the waveform from the previous test (after ‘arbinitiate’ in “test download custom waveform”), this test is to show that the waveform can be selected in any part of the testplan. The Arb normally outputs the last downloaded waveform, whether a custom waveform or a standard waveform, but can output a different waveform in any part of the testplan using the action in this test.
arbInitiate	Initialize the Arb to output the waveform. Start outputting the waveform.

Test Group/Test/Action Name	Description
-----------------------------	-------------

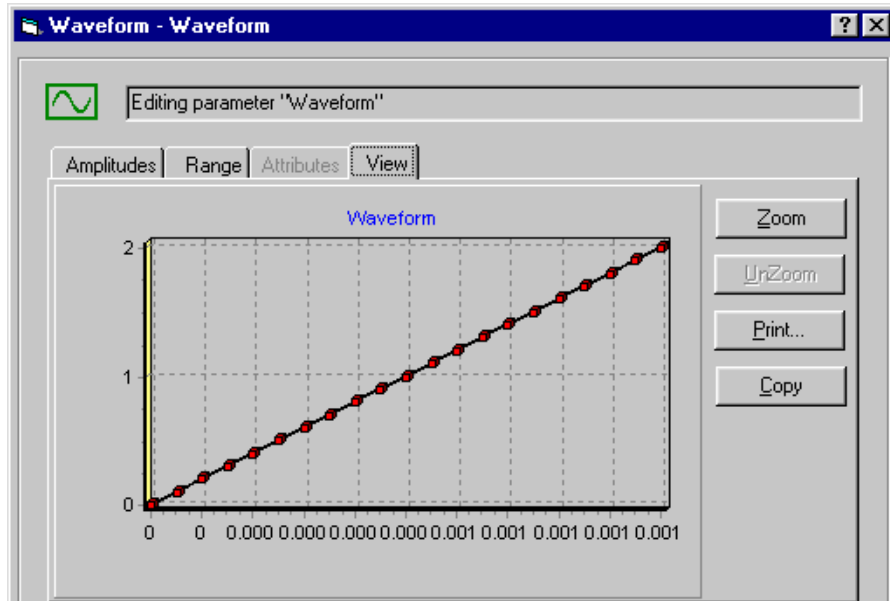
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the 'waveform_name' parameter (i.e., "wave1") This is the same name as the name in the 'waveform_name' parameter of the "Arb_DI_Waveform_Data" action.
DialogOkay	This optional action is used here to view waveform.



Click on "View Waveform" to view resultant waveform (see next figure)

Figure 3-5. Generating a Data Type Waveform

The values in this example produces a 2 Vpeak, 1 kHz ramp, as shown below; the waveform was generated using the following values



0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2.0
and Stop / Start values of
0.000 / 0.001

Note: The Arb output looks more “staircased” on an oscilloscope due to the 47.62 μ S/segment time.

Figure 3-6. Arb Waveform

Downloading User Defined Waveforms

This “Arb_DI_Custom_Waveform” action uses several double and single dimensioned array type parameters to set the voltage segments, sequences, markers, dwell, next sequences, and other values to download and generate user defined custom waveforms.

Figure 3-3 shows a typical custom waveform. The figure shows a triangle waveform that shows the relationship between the segments, sequence, and segment execution time.

To generate a waveform, it is important to understand how to set the different parameters of the action. This is explained as follows:

voltage - Specifies the voltage value of the waveform segments for one or more sequences (voltage range is ± 16.4 V). This is a double dimensioned array where the column is the sequence number and the rows are the segment numbers. Figure 3-7 shows a typical example.

marker - Specifies a marker on the selected segment in a valid sequence. The marker pulse will only be output if the sequence is non zero. The marker signal always appears on the Arb’s front panel BNC connector. It can also be routed to the VXI TTL backplane. Figure 3-8 shows a typical example on how to specify markers.

nr_sequence - Specifies the number of sequences to be downloaded. This value must be “1” or above, or no waveform will be downloaded. If more than one sequence is available and the parameter is set to “1”, only the waveform using the first sequence is downloaded.

next_sequence - Specifies which sequence is to be downloaded next. An example is in Figure 3-9.

sequence_repeat - Specifies if the sequence is to be repeated (i.e., value is set to “1” or more), as shown in Figure 3-10.

nr_segments - Specifies the number of segments for each sequence. If the number of specified segments is lower than the number of segments entered in the corresponding sequence of the ‘voltage’ parameter, only the number of segments specified here will be downloaded. Figure 3-11 shows a typical example.

segment_dwell - Specifies for how many 2 μS clock cycles each segment of a sequence is to be output. Normally, each segment is output in a single 2 μS clock cycle (if the parameter is set to “0”), which is a dwell count of “0”. Setting the ‘segment_dwell’ to a number above zero, the clock cycles will increase by that new number plus a single clock cycle.

Since the number of segments and the duration time of the segments determines the waveform frequency, each additional number added to ‘segment_dwell’ lowers the waveform frequency.

For example, if the dwell count is 0, for a 20 segment waveform, the total time the segments are output is:

$$20 * 2 \mu\text{S} = 40 \mu\text{S} = \text{a frequency of } 25 \text{ kHz}$$

Changing the dwell count to 49, changes the time and resultant waveform frequency to:

$$20 * ((49 * 2 \mu\text{S}) + 2 \mu\text{S}) = 20 * ((98 \mu\text{S}) + 2 \mu\text{S}) = 20 * 100 \mu\text{S} = 0.002 \text{ S} = \text{a frequency of } 500 \text{ Hz.}$$

An example of using the ‘segment_dwell’ parameter is in Figure 3-12. In the figure, the dwell count is 49 for two 20 segment sequences which gives a total waveform frequency of 250 Hz.

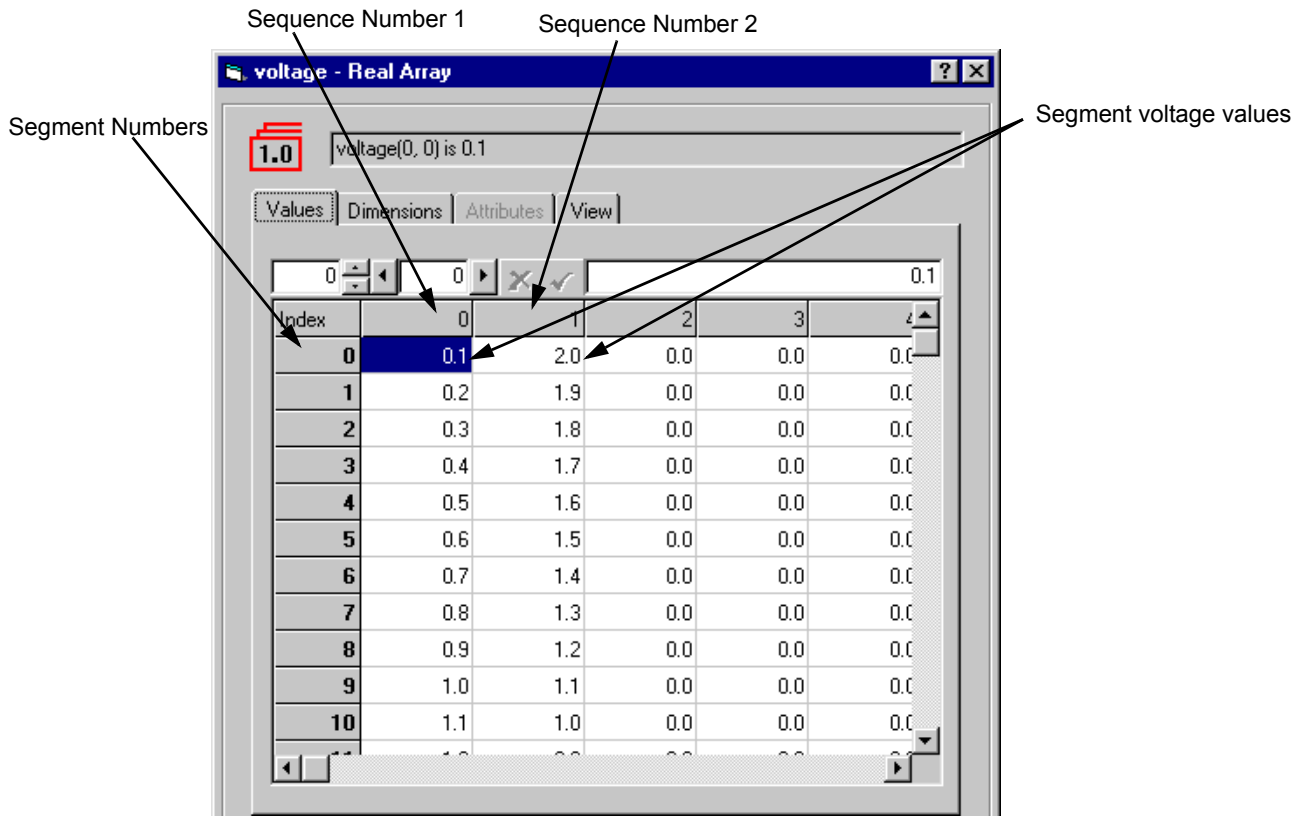


Figure 3-7. Segment/Sequence Array Values

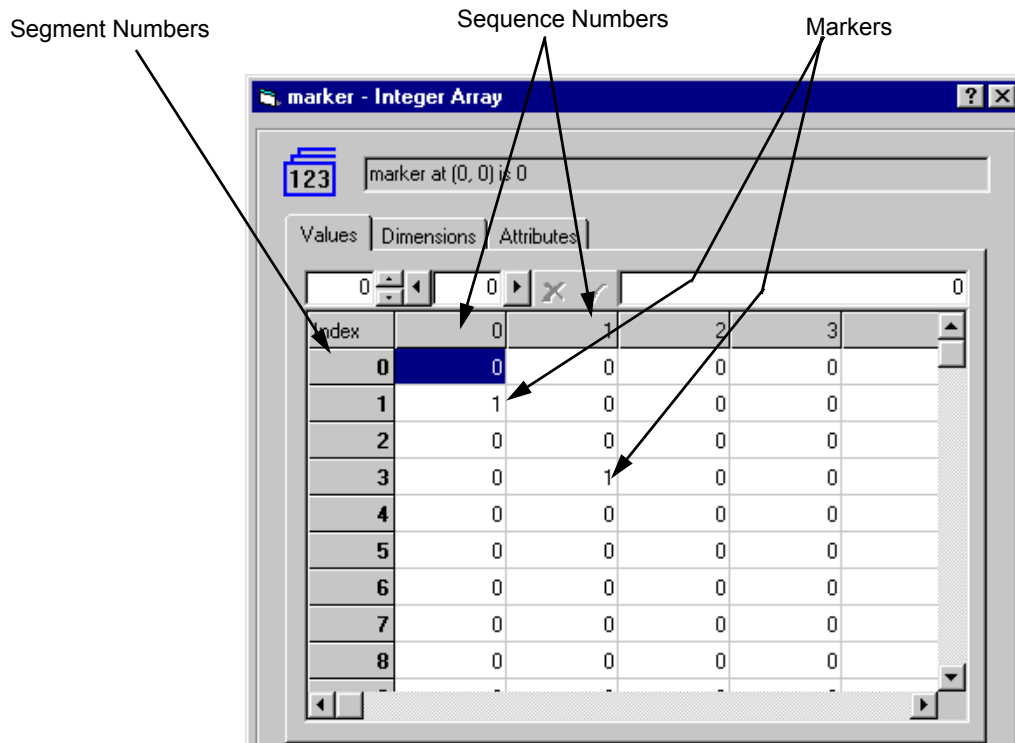


Figure 3-8. Typical Marker Values

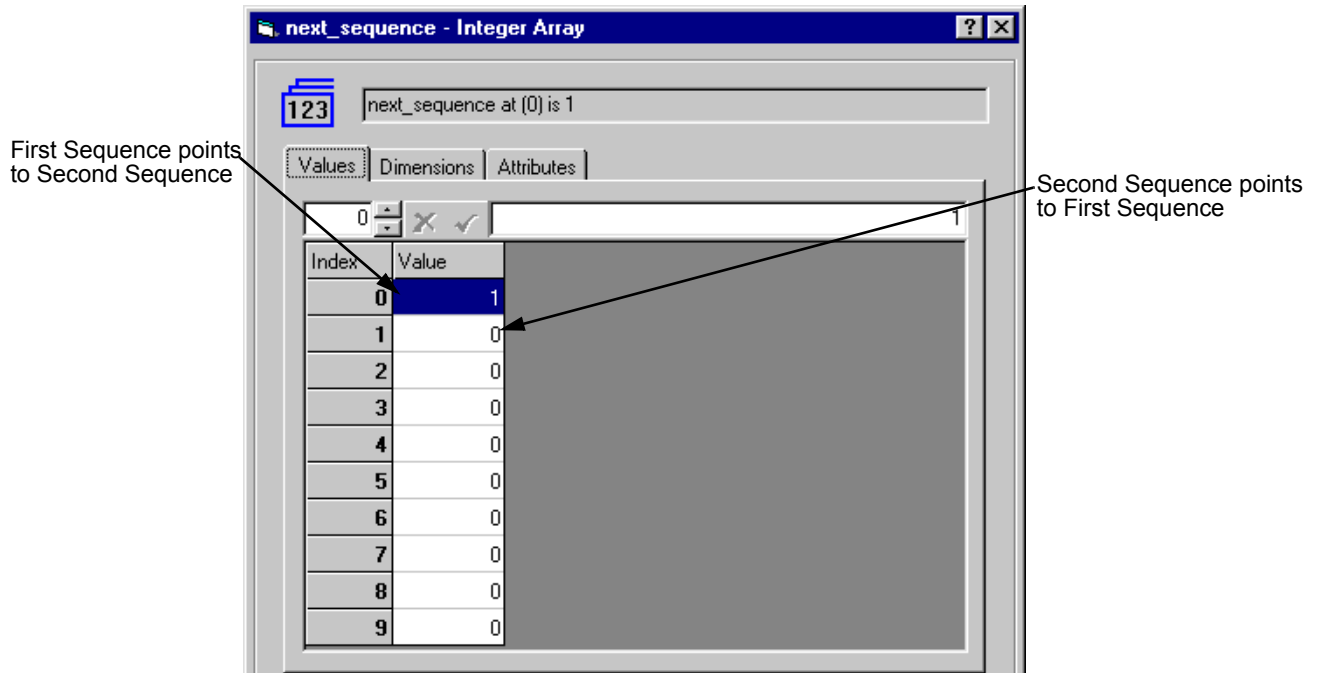


Figure 3-9. Typical Next Sequence Values

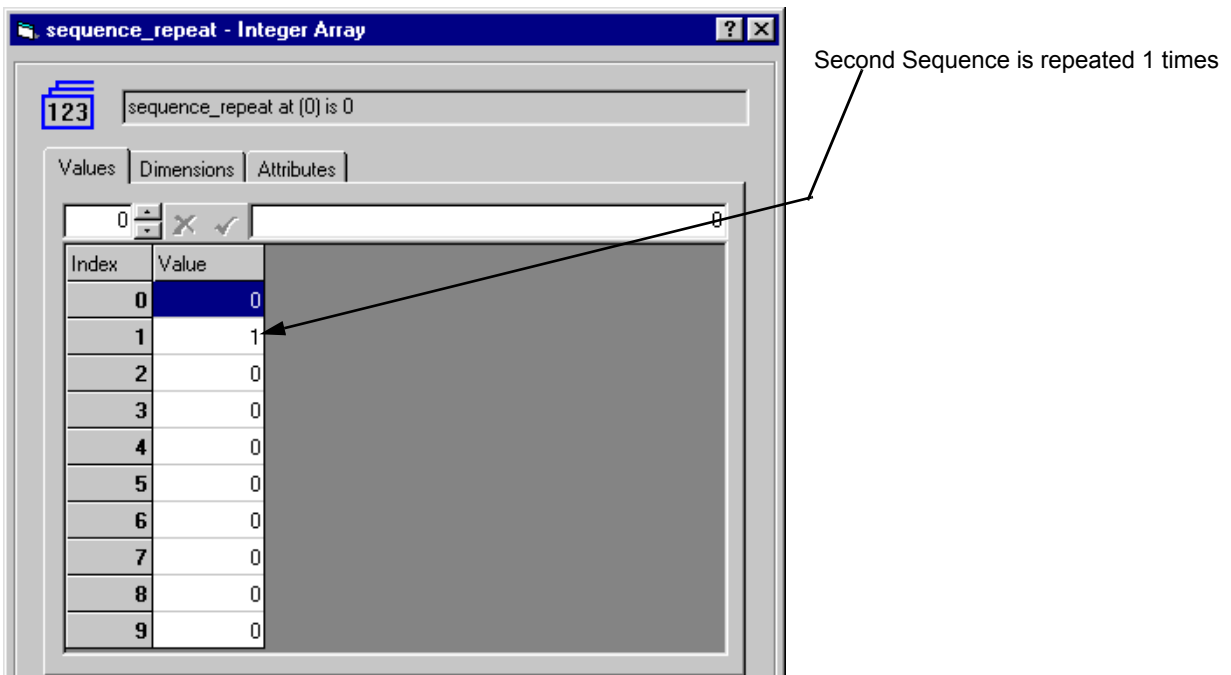


Figure 3-10. Typical Repeat Sequence Value

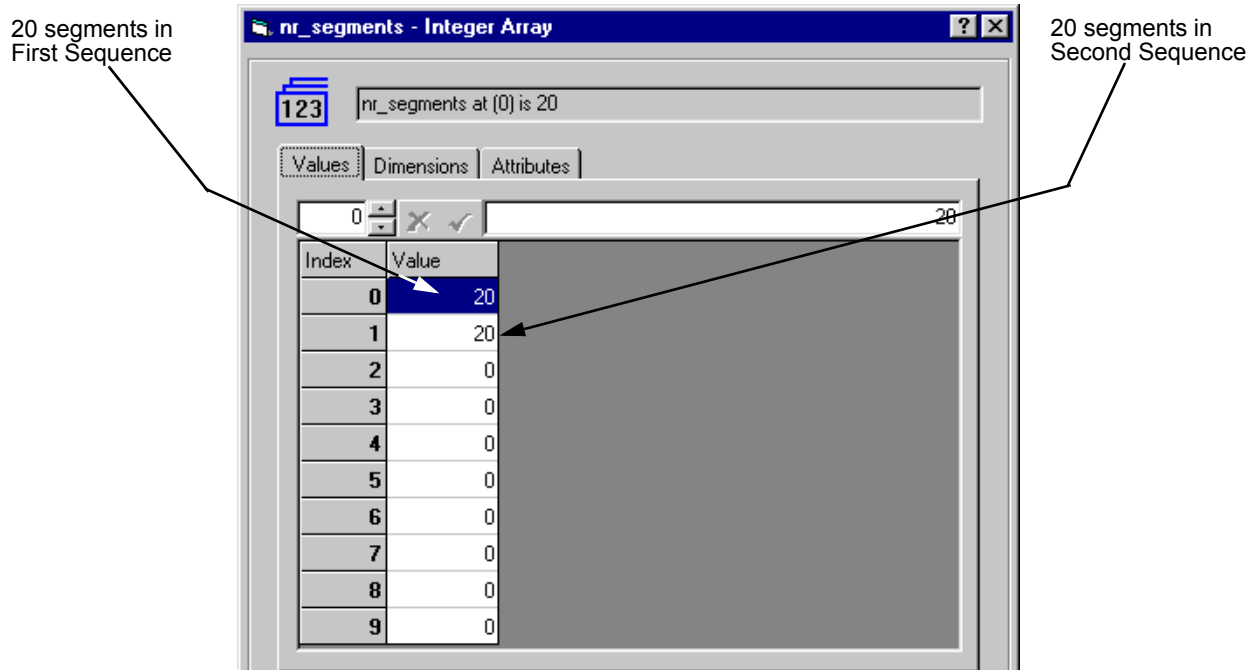
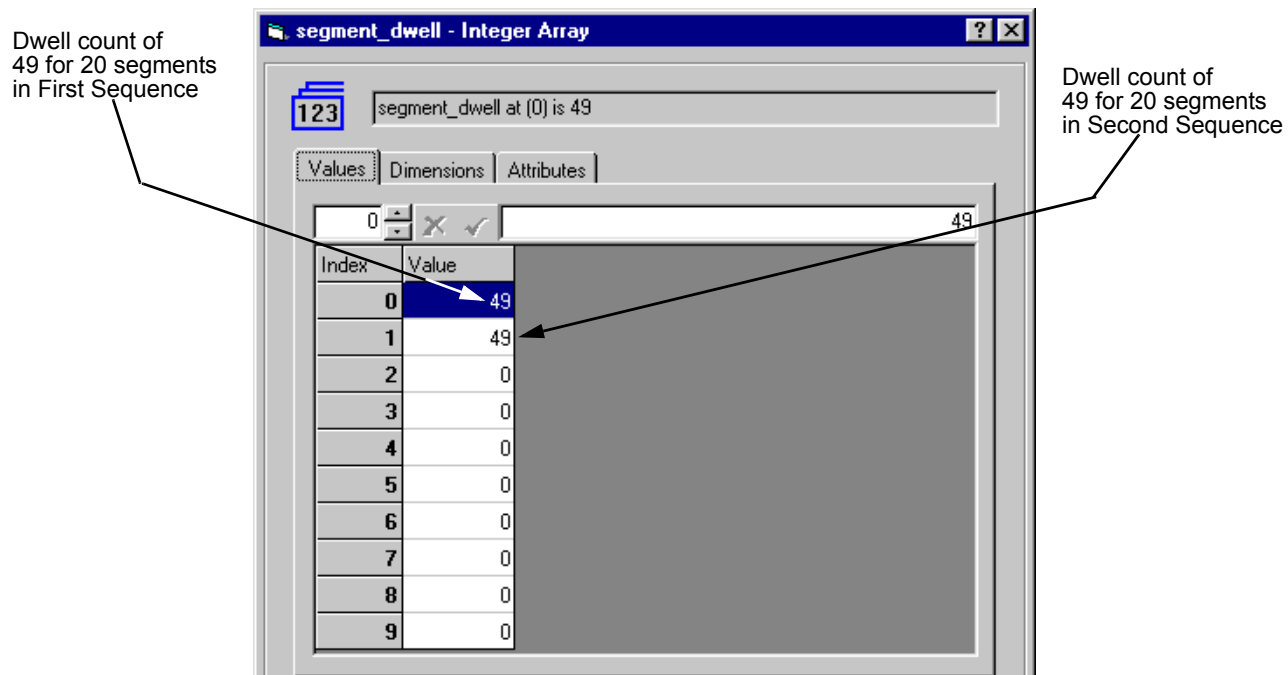


Figure 3-11. Typical Segment Number Value



Waveform time for both sequences is: total of (segments * clock time for each sequence) =
 $20 * ((49 * 0.000002) + 0.000002) = 20 * (0.000098 + 0.000002) = 20 * 0.0001 = 0.002$ per sequence
 Since both sequence time is the same, the total time = $2 * 0.002 = 0.004 = 250$ Hz

Figure 3-12. Typical Segment Number Value

How to download custom waveforms is shown in testplan 'arbcustom_user.tpa'. A review of the testplan is as follows:

<u>Test Group/Test/Action Name</u>	<u>Description</u>
testgroup download custom waveform	Downloads user generated custom waveforms.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb's output circuitry.
arbSet	Sends setup data to Arb.
test download custom waveforms	Downloads the custom waveforms to Arb.
Arb_DI_Custom_Waveform	Downloads two ramps (one positive the other negative going) to the Arb to output a triangle waveform. The number of segments is 20 for each waveform. The waveforms require 2 sequences. The 'segment_dwell' is set to 49 for both waveforms to output at 250 Hz.
test execute waveforms	Executes custom waveforms.
arbInitiate	Initialize the Arb to output the waveforms. Start outputting the waveform.
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the 'waveform_name' parameter (i.e., "wave1") This is the same name as the name in the 'waveform_name' parameter of the "Arb_DI_Custom_Waveform" action.
DialogOkay	This optional action is used here to view waveform.

Generating Custom Waveforms From a Data File

This is similar to generating a data type custom waveform using the “Arb_DI_Waveform_Data” action, except in this case the waveform data comes from a data file using the “WaveformDataRead” action. The following shows how to download custom waveform data from a file and other pertinent information.

Downloading from a File

The following shows how the actions are used:

1. The “WaveformDataRead” action reads the data from a file and stores it into its ‘Waveform’ variable (at first run of the testplan only).
2. The data is then stored into the ‘Waveform’ variable of the “Arb_DI_Waveform_Data” action.
3. The data is downloaded into the Arb and output whenever the “Arb_Select_Wave_By_Name” action selects the appropriate waveform name.

To perform the correct operation, the following must take place:

- The “WaveformDataRead” action must be referenced to the data file using the file name. Figure 3-13 shows how the file is referenced.

Normally, “WaveformDataRead” action reads the data file when the testplan is first run (i.e., the System RunCount = 0). After that, the data file will not be read.

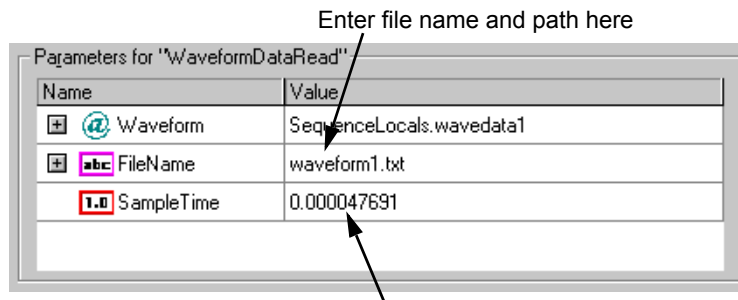
- The waveform time/frequency must be set by the ‘SampleTime’ parameter in the “WaveformDataRead” action. Figure 3-13 shows how to set the waveform time/frequency using the ‘SampleTime’ parameter.
- The ‘Waveform’ variable in the “Arb_DI_Waveform_Data” action must be referenced to the ‘Waveform’ variable in the “WaveformDataRead” action using symbols. Figure 3-14 shows how to create a symbol in the “WaveformDataRead” action and Figure 3-15 shows how to reference to the symbol in the “Arb-DI_Waveform_Data” action.
- The data in the file must be in such a format that each line is a data point or segment of the waveforms. The following is a typical example on how data is stored into the file:

```
0.0000000000
0.1000000000
0.2000000000
•
•
•
0.5000000000
```

Note A data file using this format can be generated by an E1563A Digitizer using an appropriate action or the using the “WaveformDataWrite” or other supplied standard action.

How to download a custom data type waveform is shown in testplan ‘arbcustom_file.tpa’. The downloaded waveform is a 1 kHz (1.0 mS), 2 V peak ramp, consisting of 21 segments. A review of the testplan is as follows:.

Test Group/Test/Action Name	Description
testgroup Arb waveform from file	Downloads a waveform from a file.
Switching	Connect Arb channel 1 output to ABus1 and UUT Common.
test setup Arb	Setup arb output circuitry.
arbConfOutControls	Configures Arb's output circuitry.
arbSet	Sends setup data to Arb.
test get and download data	Gets download waveform data from file and downloads the data to the Arb.
WaveformDataRead	Reads data from file and store it in an array variable.
Arb_DI_Waveform_Data	Download the data into the Arb that was read from the file to generate a data type waveform. The ‘waveform_name’ parameter is set to “wave1” to give the downloaded waveform that name. The waveform time or frequency is set to 1 kHz using the ‘SampleTime’ parameter value in the “WaveformDataRead” action where frequency = 1 / (sample time * number of samples) = 1 / (0.000047619 * 21 = 0.001 S = 1 kHz
test output custom waveform	Executes the downloaded waveform.
arbInitiate	Initialize the Arb to output the waveforms. Start outputting the waveform.
Arb_Select_Wave_By_Name	Selects and outputs the waveform using the name entered in the ‘waveform_name’ parameter (i.e., “wave1”) This is the same name as the name in the ‘waveform_name’ parameter of the “Arb_DI_Waveform_Data” action.
DialogOkay	This optional action is used here to view waveform.



Enter waveform time/frequency here (e.g., “SampleTime’ of 0.000047619 = frequency of 1 kHz (0.000047619 * 21) = 0.001 S; 1 / 0.001 = 1 kHz)

Figure 3-13. Entering File Name and Sample Time

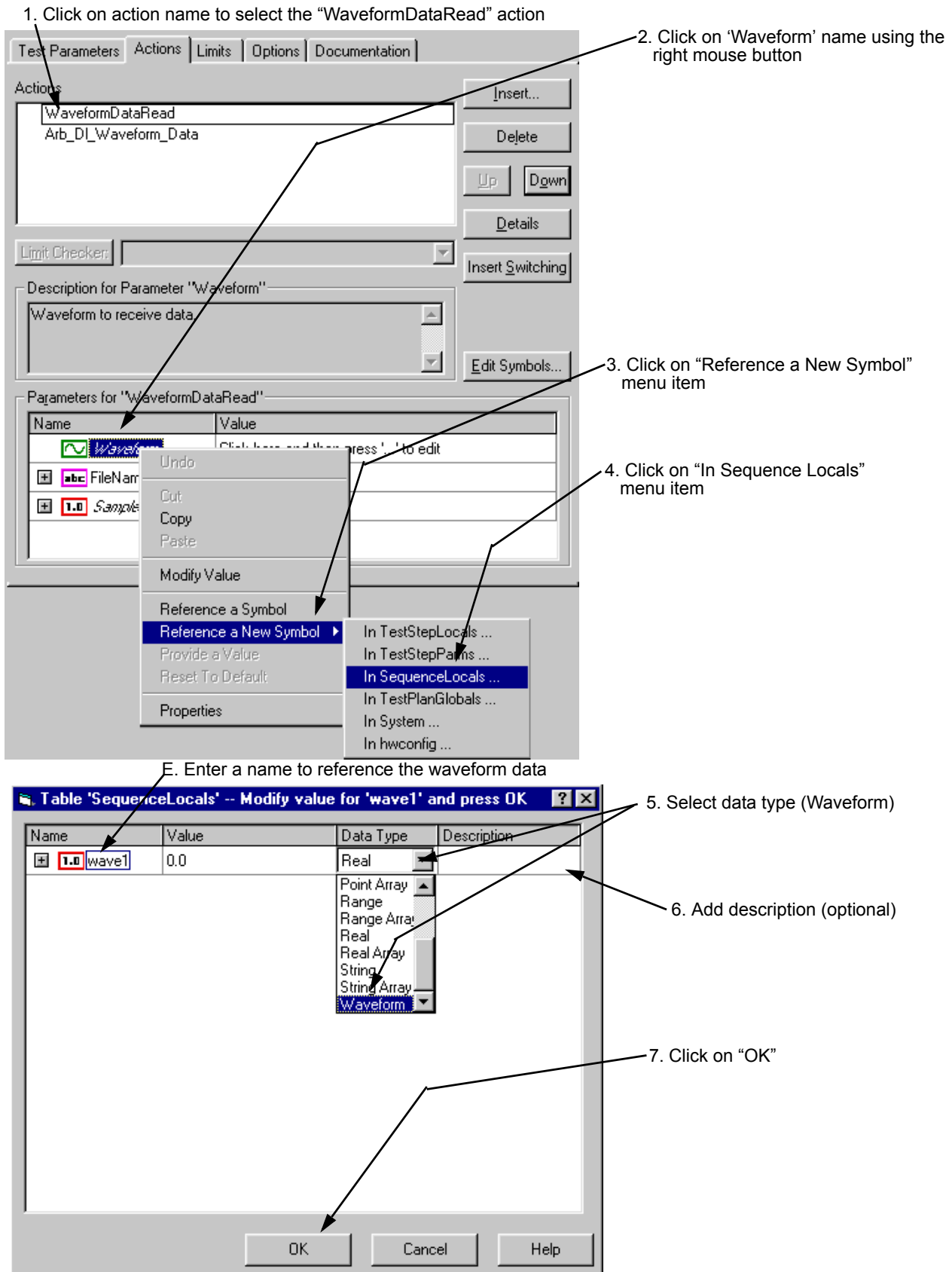
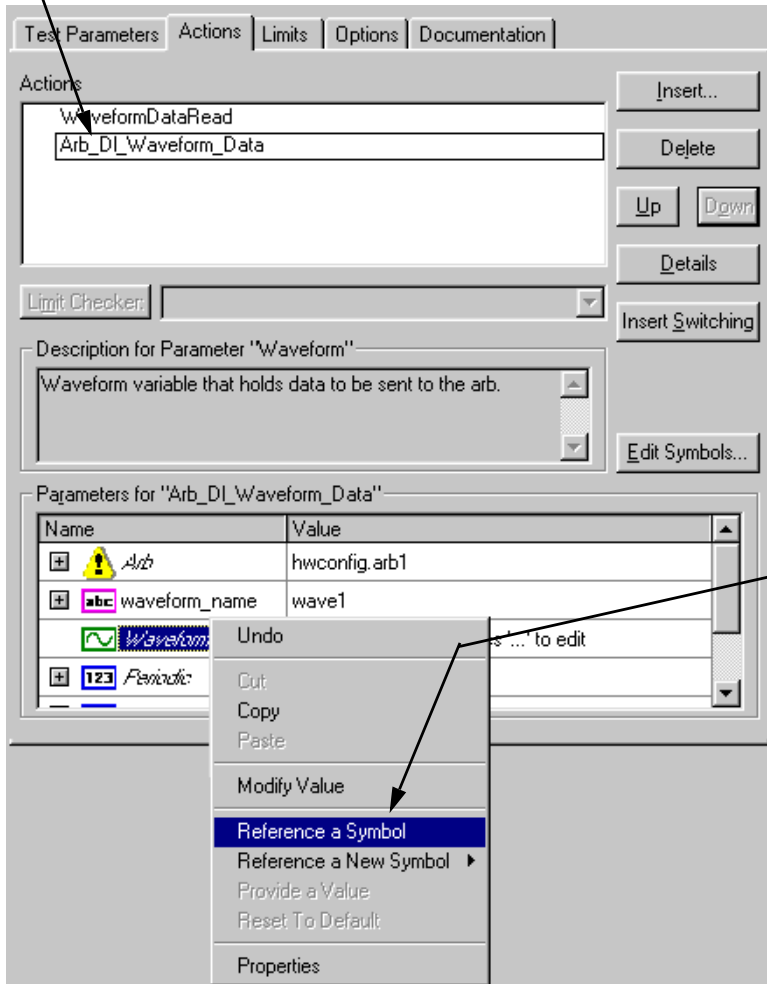


Figure 3-14. Create a Symbol for the "WaveformDataRead" Action

1. Click on action name to select the “Arb_DI_Waveform_Data” action

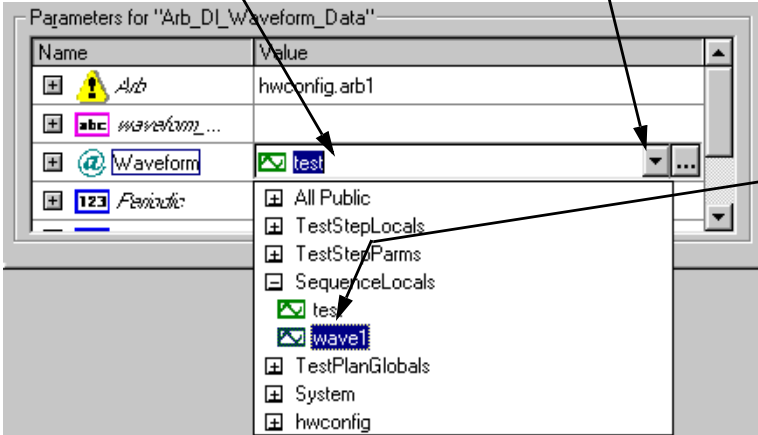


2. Click on “Reference a Symbol” menu item

3. The program automatically selects a symbol. If only one symbol is stored, it selects that symbol. If more than one symbol is stores, you must select the symbol, as shown in steps D to F

4. Click on parameter value to select the parameter

5. Click on the arrow to select the symbol (you can also click on the three dots to select it, but a different procedure is used)



6. Click on the appropriate symbol name to select it

Figure 3-15. Reference a Symbol in the “Arb_DI_Waveform_Data” Action

Creating a Data File

The data file is an ASCII file that contains waveform data. You can add data manually where each segment data point is a line in the file. You can also use the “WaveformDataWrite” action to store the data into the file. For example, you can use the action to store a waveform read by the E1563A A/D Converter into a file, then download the file to the Arb and output.

Memory Requirements for Downloading Waveforms

For the Arb to achieve high throughput, it downloads into memory all waveforms to be used in the testplan. A waveform can then be output in any part of the testplan.

The Arb is limited on how many waveforms can be stored, depending on the size of the waveforms. Normally, the Arb has sufficient memory for all the waveforms in a testplan unless the testplans use a very large number of waveforms or several long waveforms. The Arb driver then generates an error; normally an “Segment count too big during download” error (see “Agilent E6173A Arbitrary Waveform Generator Error Messages” below).

In case of a memory overflow, the only way for the Arb to receive new waveforms is to remove the current waveforms from memory. To do this, use the “arbReleaseWaveforms” action. This action will then remove all waveforms downloaded for the selected channel.

Note

It becomes necessary to use the “arbReleaseWaveforms” action, place the action both at the beginning of the testplan and at the point where more waveforms need to be added.

Agilent E6173A Arbitrary Waveform Generator Error Messages

The following lists the error numbers in hex and corresponding error messages the Arb can generate. Use the messages to troubleshoot the testplan or Arb, if needed.

Table 3-1. Agilent E6173A Error Codes and Descriptions

Error Number	Description
21	Unimplemented function call
22	Timeout waiting for arbStop
23	Illegal segment voltage
24	Segment count too big during download
25	Segment range error (setSegment)
26	Sequence range error (setSequence)
27	Illegal trigger input
28	Not channel 1 or 2
29	Sequence function without a segment fn
2A	Segment funct. without a sequence fn
2B	Couldn't find specified download ID
2C	Sequence count too big during download

Chapter 4

Action Summary

This chapter summarizes the TestExec SL actions supplied with the Agilent TS-5000 System. The action directory path is:

C:\Program Files\Agilent\TS-5000 System Software\actions

Actions are located in sub-directories and summarized in the following tables:

Sub-Directory	Table and Action Type
arb	Table 4-1, "arb (Arbitrary Waveform Generator) Actions"
counter	Table 4-2, "Counter Actions"
dac	Table 4-3, "dac (D/A Converter) Actions"
dgn	Table 4-4, "dgn (Diagnostic) Actions"
digitizer	Table 4-5, "digitizer Actions"
dio	Table 4-6, "dio (Digital I/O) Actions"
dmm	Table 4-7, "DMM Actions"
dso	Table 4-8, "DSO Actions"
event	Table 4-9, "Event Detector Actions"
generic	Table 4-10, "Generic Actions"
mcm	Table 4-11, "Measurement Control Module Actions"
power	Table 4-12, "Power Supply Actions"
serial	Table 4-13, "Serial Interface Actions"
SerialProtocol	Table 4-14, "Automotive Serial Protocol Actions"
switch	Switch and load card actions: Table 4-15, "Switch Actions" Table 4-16, "Load Card Actions"
vi	Table 4-17, "Voltage / Current Source Actions"
daq	Table 4-18, "Data Acquisition Actions"

Note This chapter contains action summaries. The actions are documented in detail in the TS5000 online help. This help is available from the Help menu in TestExec SL.

Table 4-1. arb (Arbitrary Waveform Generator) Actions

Agt33220GetBurstProperties	Returns the values of the Agilent33220Burst interface properties.
Agt33220GetOutputProperties	Returns the values of the Agilent33220Output interface properties.
Agt33220MsgReceive	Receives SCPI commands via the Agilent33220System interface.
Agt33220MsgSend	Sends SCPI commands via the Agilent33220System interface.
Agt33220MsgSendReceive	Receives SCPI commands via the Agilent33220System interface.
Agt33220OutputArbWaveform	Downloads an arbitrary waveform and enables the output.
Agt33220OutputStdWaveform	High level action to generate a basic waveform.
Agt33220PutBurstProperties	Sets the values of the Agilent33220Burst interface properties.
Agt33220PutOutputProperties	Sets the values of the Agilent33220Output interface properties.
Agt3352xGetBurstProperties	Returns the values of the Agilent3352xBurst interface properties.
Agt3352xGetOutputProperties	Returns the values of the Agilent3352xOutput interface properties.
Agt3352xMsgReceive	Receives SCPI commands via the Agilent3352xSystem interface.
Agt3352xMsgSend	Sends SCPI commands via the Agilent3352xSystem interface.
Agt332xMsgSendReceive	Receives SCPI commands via the Agilent3352xSystem interface.
Agt332xOutputArbWaveform	Downloads an arbitrary waveform and enables the output.
Agt3352xOutputStdWaveform	High level action to generate a basic waveform.
Agt3352xPutBurstProperties	Sets the values of the Agilent3352xBurst interface properties.
Agt3352xPutOutputProperties	Sets the values of the Agilent3352xOutput interface properties.
CreateArbWaveform	Uses the arbitrary waveform function to create an array that defines a square wave pulse with a calculated rise time and fall time. The waveform consists of 4000 points.
awgGetBurstProperties	This action is use to get the burst properties in the 33220A/33250A arbitrary waveform generator.
awgGetOutputProperties	This action is use to get the Output properties in the 33220A/33250A arbitrary waveform generator.
awgOutputArbWaveform	This action is use to downloads an arbitrary waveform an enables the output for the 33220A/33250A.
awgOutputStdWaveform	This action is use to generate a basic waveform from 33220A/33250A.
awgPreset	This action is to preset the 33220A/33250A parameter.
awgQueryCmd	This action is use to send a SCPI query command.
awgSendCmd	This action is use to send a SCPI command.
awgSendSoftwareTrigger	This action is use to send a software trigger to the 33220A/33250A.
awgSetBurstProperties	This action is use to set the burst properties in the 33220A/33250A arbitrary waveform generator.
awgSetOutputProperties	This action is use to set the burst properties in the 33220A/33250A arbitrary waveform generator.
awgSetTrigger	This action is use to configure the trigger source and slope in the 33220A/33250A arbitrary waveform generator.
Arb_Disable_Download	Not recommended for use in new testplans.

Table 4-1. arb (Arbitrary Waveform Generator) Actions

Arb_DI_Custom_Waveform	Downloads a custom, user defined waveform consisting of up to 10 sequences with 50 segments per sequence to the Agilent E6173 Arb.
Arb_DL_Datafile	Creates a waveform from a data file.
Arb_DI_Ext_Waveform	Downloads a sine, square, or triangle waveform of specified frequency, amplitude, offset and phase (with optional burst modulation) to the Agilent E6173 Arb.
Arb_DI_Std_Waveform	Downloads a sine, square/pulse or triangle waveform of specified frequency, amplitude, and offset to the Agilent E6173 Arbitrary Waveform Generator (Arb).
Arb_DI_Swept_Sine	Downloads a basic swept sine waveform of specified frequency, amplitude, and offset to the Agilent E6173 Arbitrary Waveform Generator (Arb).
Arb_DI_Swept_Sine_Ex	Downloads a swept sine waveform of specified frequency, amplitude, and offset to the Agilent E6173 Arbitrary Waveform Generator (Arb).
Arb_DI_Waveform_Data	Downloads a waveform data type to the Agilent E6173 Arbitrary Waveform Generator (Arb).
Arb_Download	Not recommended for use in new testplans (use "Arb_DI_Std_Waveform" or "Arb_DI_Ext_Waveform").
Arb_Download_Deluxe	Not recommended for use in new testplans (Use "Arb_DI_Ext_Waveform" or "Arb_DI_Custom_Waveform").
Arb_Download_Waveform	Not recommended for use in new testplans (use "Arb_DI_Waveform_Data").
Arb_Enable_Download	Not recommended for use in new testplans.
Arb_Output_Std_Waveform	Configures Agilent E6173 Arb to generate a sine, pulse, or triangle waveform of specified frequency, amplitude, and offset.
Arb_Select_Synced_Waveforms	Not recommended for use in new testplans (Use "Arb_Select_Synced_Waves_By_Name").
Arb_Select_Synced_Waves_By_Name	Allows selection of Agilent E6173 Arb waveforms (on the fly) of both channels referenced by name assigned in the "Arb_DI_xxx" actions.
Arb_Select_Wave_By_Name	Allows the selection of Agilent E6173 Arb waveforms (on the fly) referenced by name assigned in the "Arb_DI_xxx" actions.
Arb_Select_Waveform	Not recommended for use in new testplans (Use "Arb_Select_Wave_By_Name").
Arb_Source_MDA	High-Level Action: Programs the arb to act as the MDA source.
arbConfOutControls	Configures the Agilent E6173 Arb's output circuitry.
arbConfSync	Enable/disable synchronous mode on a given arb channel pair.
arbConfTimebase	Select internal 10MHz clock or an external clock supplied through the front panel BNC.

Table 4-1. arb (Arbitrary Waveform Generator) Actions

arbConfTimeout	Sets the I/O timeout value (in milliseconds) for communication with the Agilent E6173 Arb.
arbConfTrigIn	Set input triggering parameters.
arbConfTrigOut	Set trigger parameters for output.
arbInitiate	Starts the arb sequencer to output the waveform.
arblsSet	Program arb - transfers all arb settings specified by the arbConf.
arbReleaseWaveforms	Causes the Agilent E6173 handler and driver to "forget" about all previously downloaded waveforms.
arbReset	Causes a hard reset of the arb hardware.
arbSet	Program arb - transfers all arb settings specified by the arbConf.
arbSetClkAdjust	Change a channel's clock timing on the fly.
arbSetSegment	Not recommended for use in new testplans (use Arb_DI_Custom_Waveform).
arbSetSequence	Not recommended for use in new testplans (use Arb_DI_Custom_Waveform).
arbSetWaveform	Not recommended for use in new testplans.
arbStop	Stops waveform output, at end of current sequence, on a given channel.
CreateArbWaveform	This program uses the arbitrary waveform function to create an array that defines a square wave pulse with a calculated rise time and fall time. The waveform consists of 4000 points.
HP33120ApplyWaveform	To output a waveform from the Agilent 33120 ARB
HP33120ClearWaveforms	To clear the currently downloaded waveforms from the Agilent 33120 ARB.
HP33120DownloadWaveform	To download a waveform to the Agilent 33120 ARB.
HP33120ErrorStatus	To receive the error status from the Agilent 33120 ARB.
HP33120Reset	To cause a hard reset to the Agilent 33120 ARB.
HP33120SelectWaveform	To select a downloaded waveform by name and output it immediately. (Must call HP33120DownloadWaveform first).
HP33120SweepWaveform	To output a swept waveform from the Agilent 33120 ARB.

Table 4-2. Counter Actions

Low-Level Actions	
Ag53220AConflnControls	Using Agilent 53220A Universal Counter to configure the counter inputs.
Ag53220AErrorStatus	Get error status from Agilent 53220A Universal Counter.
Ag53220AMeasTimeInterval	To send a query command to an instrument and receives its response.
Ag53220AMeasureAgain	Repeat the last measurement (use for speed) on Agilent 53220A Universal Counter.
Ag53220AMeasureFreq	Measure frequency using Agilent 53220A Universal Counter.
Ag53220AMeasureNegPulse	Measure negative pulse width using Agilent 53220A Universal Counter.
Ag53220AMeasurePeriod	Measure period using Agilent 53220A Universal Counter.
Ag53220AMeasurePosPulse	Measure positive pulse width using Agilent 53220A Universal Counter.
Ag53220AMeasureVpp	Measure Peak to Peak Voltage using Agilent 53220A Universal Counter.
Ag53220AReset	This action causes a hard reset of the Agilent 53220A Universal Counter.
Ag53220ASetTimeOut	This action can be used to set operation time out value (in milliseconds).
ctrConfFunction	Selects function, range, and resolution of the Agilent E1333 Counter.
ctrConflnControls	Configures the input circuit of the Agilent E1333 Counter.
ctrConfTrigIn	Sets the input trigger of the Agilent E1333 Counter.
ctrGetResults	Reads the Agilent E1333 Counter results.
ctrInitiate	Starts the previously configured Agilent E1333 Counter measurement.
ctrlsSet	Waits until the Agilent E1333 Counter is ready for measurement.
ctrReset	Resets all of the Agilent E1333 Counter's channels to the power-on state.
CtrSelfTest	Runs a selftest on the Agilent E1333 Universal Counter that is provided by its Plug and Play driver.
ctrSetReturnMode	This action allows time-outs on the specified Agilent E1333 Universal Counter to raise an exception when encountered.
ctrSetTimeout	Sets the Agilent E1333 Counter operation time out value (in milliseconds).
High-Level Actions	
ctrMeasureFrequency	An Agilent E1333 Counter action that measures frequency.
ctrMeasurePeriod	An Agilent E1333 Counter action that measures period.
ctrMeasurePulseWidth	An Agilent E1333 Counter action that measures pulse widths.
ctrMeasureRatio	High level Agilent E1333 Counter Action to measure Ratio using the counter.
ctrMeasureTimeInterval	High level Agilent E1333 Counter action that measures time interval.
ctrMeasureTotalize	High level Agilent E1333 Counter action that starts a totalize measurement.
HP53131ConflnControls	Using Agilent 53131A Universal Counter to configure the counter inputs.
HP53131ErrorStatus	This action is received error status from Agilent 53131A Universal Counter
HP53131MeasTimeInterval	To send a query command to an instrument and receives its response.
HP53131MeasureAgain	This action only for repeats the last measurement (use for speed) on Agilent 53131A Universal Counter.

Table 4-2. Counter Actions

HP53131MeasureFreq	Measure frequency using Agilent 53131A Universal Counter.
HP53131MeasureNegPulse	Measure negative pulse width using Agilent 53131A Universal Counter.
HP53131MeasurePeriod	Measure period using Agilent 43131A Universal Counter.
HP53131MeasurePosPulse	Measure positive pulse width using Agilent 53131A Universal Counter.
HP53131MeasureVpp	Measure Peak to Peak Voltage using Agilent 53131A Universal Counter.
HP53131Reset	This action causes a hard reset of the Agilent 53131A Universal Counter.
HP53131SetTimeOut	This action can be used to set operation time out value (in milliseconds). Only for Agilent 53131A Universal Counter.

Table 4-3. dac (D/A Converter) Actions

Agt34951ConfigureExternalPacing	Configures the trigger type and external clock divisor for the specified channels. After using this action to set up the external clock and trigger type, call the following functions: Agt34951ConfigureWaveform and Agt34951StartWaveform.
Agt34951ConfigureInternalPacing	Configures the trigger type and internal clock period for the specified channels. After using this action to set up the internal clock and trigger, call the following actions: Agt34951ConfigureWaveform and Agt34951StartWaveform.
Agt34951ConfigureTrigOut	Enable or disable the external trigger output of the DAC. This will cause the DAC to output a 3.5V trigger signal when a waveform starts. NOTE: This line is also the Trigger Input for external triggers. Thus this function can not be used at the same time an external trigger input is expected. See ConfigureExternalPacing and ConfigureInternalPacing.
Agt34951ConfigureWaveform	Configures the specified channels for waveform output. This action does not start the waveform or configure the clock or trigger for the channel. (Use the following sequence of actions: Agt34951ConfigureTrigger, Agt34951ConfigureClock, Agt34951ConfigureWaveform, Agt34951StartWaveform. Two shortcut actions, Agt34951SetVoltWaveform and Agt34951SetCurrentWaveform, perform these same tasks but do not allow control of the clock, trigger, or number of cycles.)
Agt34951CreateTrace	Load trace data from an array of real values. Each point should be between -1.0 and +1.0. IMPORTANT: The number of points in the array MUST be specified by the NumPoints parameter.
Agt34951DebugInstErrs	Sets the debug flag to On, which causes every action to query the instrument for errors upon completion of the action and to raise an exception an instrument occurs.
Agt34951DeleteAll	Deletes all traces. NOTE: Traces which are currently running are not deleted! To halt all traces, call Agt34951HaltWaveform with all channels set to 1 and Disable set to 1 before calling this action.

Table 4-3. dac (D/A Converter) Actions

Agt34951DeleteTrace	Deletes the trace specified. NOTE: Traces which are currently running are not deleted! To halt all traces, call Agt34951HaltWaveform with all channels set to 1 and Disable set to 1 before calling this action.
Agt34951Enable	Enables, disables, or leaves unchanged multiple DAC channels. This opens or closes the output relay for the specified channels.
Agt34951GetChanState	Returns the output state and trace state for a given channel.
Agt34951HaltWaveform	Halts waveforms running on the specified channels.
Agt34951LoadTrace	Loads trace data from a text file. The file should contain one point per line, each point between -1.0 and +1.0 with a minimum of 8 points per file. Total trace data storage for the DAC is 512,000 points for all traces.
Agt34951MsgQuery	Sends an instrument query and get the results.
Agt34951MsgReceive	Reads the 34980 results buffer.
Agt34951MsgSend	Sends a SCPI command to the 34980. NOTE: This action is not slot specific and may affect other cards in the 34980 besides the 34951. If slot information is required for the command, it must be supplied (unlike most 34951 actions).
Agt34951SetCurrentDC	Sets up a DAC channel to output DC current.
Agt34951SetCurrentWaveform	Applies the specified trace to the output channels selected and enables the channels. Defaults: Clock: Internal; Trigger: Immediate; # of Cycles: Infinite. To change these defaults or to avoid enabling the channels use the ConfigureWaveform action.
Agt34951SetStandardWaveform	Creates trace data from a standard waveform: Ramp, Sine, Square or Triangle. Points are generated between +1.0 and -1.0. Minimum of 8 points per trace. Total trace data storage for the DAC is 512,000 points for all traces.
Agt34951SetVoltDC	Sets up a DAC channel to output DC voltage.
Agt34951SetVoltWaveform	This action will apply the specified trace to the output channels selected and enable the channels. Defaults: Clock: Internal; Trigger: Immediate; # Cycles: Infinite. To change these defaults or to avoid enabling the channels use the ConfigureWaveform action.
Agt34951StandardTrace	Creates trace data from a standard waveform: Ramp, Sine, Square or Triangle. Points are generated between +1.0 and -1.0. Minimum 8 points per trace. Total trace data storage for the DAC is 512,000 points for all traces.
Agt34951StartWaveform	Starts the waveforms on the specified channels. Use the Agt34951ConfigWaveform action to configure the waveforms before using this action.
dacConfSourceDCI	Configures the Agilent E1418 DAC to output the specified current (Agilent E1418 only).
dacConfSourceDCV	Configures the Agilent E1418 DAC to output the specified voltage (Agilent E1418 only).

Table 4-3. dac (D/A Converter) Actions

dacConfTrigIn	Selects the Agilent E1418 DAC's trigger source (Agilent E1418 only) for all channels.
dacConnect	Immediately connects the output relays on the selected Agilent E1418 DAC (Agilent E1418 only).
dacDisconnect	Immediately disconnects the output relays on the selected Agilent E1418 DAC (Agilent E1418 only).
dacIsSet	Waits until the Agilent E1418/E1328 DAC is ready for output.
dacReset	Resets all channels of the Agilent E1418/E1328 DAC to the power-on state (0V output) and disconnects the output relay.
dacSelftest	Runs selftest on the Agilent E1418/E1328 DAC; returns a pass/fail indication.
dacSet	Sends current setup information to the Agilent E1418 DAC (Agilent E1418 only).
dacSetDCI	Sets the Agilent E1418/E1328 DAC to output the specified current; causes an immediate output.
dacSetDCV	Sets the Agilent E1418/E1328 DAC to output the specified voltage; causes an immediate output.
dacSetDCVSU	Program a DAC channel on the Agilent E6198 switch/load unit to a desired voltage.
dacSetGainOffsetSU	Change the gain/offset values used to set the DAC voltage in the switch/load unit.
dacSetGainSU	Change the gain value used to set the DAC voltage in the switch/load unit.
dacSetOffsetSU	Change the offset value used to set the DAC voltage in the switch/load unit.
dacSetPSVSU	Program a DAC channel voltage on the Agilent E6198 switch unit using the gain value in the module parameter block.
dacSetTimeout	Sets the Agilent E1328 or E1418 Digital to Analog Converter (DAC) operation time out value (in milliseconds).
dacAdvanceConfigureExternalPacing	Configure the trigger type and external clock divisor for the specified channels.
dacAdvanceConfigureInternalPacing	Configure the trigger type and internal clock period for the specified channels.
dacAdvanceConfigureWaveform	Configure waveform output for the specified channels. It will not start the waveform, nor will it configure the clock or trigger for the channels.
dacAdvanceCreateTrace	Load trace data from an array of real values.
dacAdvanceDebugInstErrs	Setting the debug flag to On will cause every action to query the instrument for errors upon completion of the action and raise an exception if there is an instrument error.
dacAdvanceDeleteAll	Delete all traces. NOTE: Traces which are currently running will not be deleted!

Table 4-3. dac (D/A Converter) Actions

dacAdvanceDeleteTrace	Delete the trace specified. NOTE: Traces which are currently running will not be deleted!
dacAdvanceEnable	Enable, disable or leave unchanged multiple DAC channels. This will open or close the output relay for the specified channels.
dacAdvanceGetChanState	Return the output state and trace state for a given channel.
dacAdvanceHaltWaveform	This action will halt waveforms running on the specified channels.
dacAdvanceLoadTrace	Load trace data from a text file.
dacAdvanceMsgQuery	Send an instrument query and get the results.
dacAdvanceMsgReceive	Read the L4451A results buffer.
dacAdvanceMsgSend	Send a SCPI command to the L44xx.
dacAdvanceReset	This action is used for L4451A reset.
dacAdvanceSetCurrentDC	Setup a DAC channel to output DC current.
dacAdvanceSetCurrentWaveform	Apply current waveform of the specified trace to the output channels selected and enable the channels.
dacAdvanceSetStandardWaveform	Create trace data from a standard waveform: Ramp, Sine, Square or Triangle.
dacAdvanceSetVoltDC	Setup a DAC channel to output DC voltage.
dacAdvanceSetVoltWaveform	Apply voltage waveform of the specified trace to the output channels selected and enable the channels.
dacAdvanceStandardTrace	Create trace data from a standard waveform: Ramp, Sine, Square or Triangle.
dacAdvanceStartWaveform	Start the waveform on the selected channel. The waveform must have already been configured with the <code>dacAdvanceConfigWaveform</code> action.
m9185dacGetCardInfo	This action retrieves the M9185A card information.
m9185dacSetTrig	Set the trigger configuration of the M9185A card.
m9185dacGetTrig	Retrieve the trigger configuration of the M9185A card.
m9185dacSetRelay1Ch	Turn on/off the output relay for a single channel of M9185A.
m9185dacSetRelay8Ch	Turn on/off the output relay for 8 channels simultaneously of M9185A.
m9185dacSetRelay16Ch	Turn on/off the output relay for 16 channels simultaneously of M9185A.
m9185dacGetRelay1Ch	Return the output relay state for a single channel of M9185A.
m9185dacGetRelay8Ch	Return the output relay state for 8 channels of M9185A.
m9185dacGetRelay16Ch	Return the output relay state for 16 channels simultaneously of M9185A.
m9185dacSetOut1Ch	Set the voltage/current output for a single channel of M9185A.
m9185dacSetOut8Ch	Set the voltage/current output for 8 channels of M9185A.
m9185dacSetOut16Ch	Set the voltage/current output for 16 channels of M9185A.

Table 4-3. dac (D/A Converter) Actions

m9185dacGetOut1Ch	Return the voltage/current output level for a single channel of M9185A.
m9185dacGetOut8Ch	Return the voltage/current output level for a 8 channel of M9185A.
m9185dacGetOut16Ch	Return the voltage/current output level for 16 channels of M9185A .
m9185dacReset	Reset the M9185A to its power-on state.
High-Level Actions	
DAC: 16 Chan (Individual)	This action programs each channel of the 16 channel Agilent E1418 DAC either as a voltage or current source.
DAC: 16 Channels	This action programs all channels of the 16 channel Agilent E1418 DAC either as a voltage or current source.

Table 4-4. dgn (Diagnostic) Actions

Ag3499SlotCheck	Verifies that the 3499 contains the N2261, N2262, and N2264 cards in one of the 5 slots.
ArbAdjust	Measures Voltage/Current (V/I) gains using the MCM and DMM on external Arb connections to the system.
AutoAdjust	Measures all Voltage/Current (V/I) gains and offsets using the MCM and DMM.
dgnConnectString	Connects the switch path referenced by the parameter 'pathstring'. This action is helpful when the path you wish to connect may be a variable or a symbol table entry.
dgnDisConnect	Disconnects the switch path referenced by the parameter 'path'. This action is helpful when the path you wish to disconnect may be a variable or a symbol table entry.
EFTMuxTestDebugPorts	This action tests the Series 2 debug ports by prompting the user to put a 50 ohm resistor on each port one-at-a-time.
EFTMuxTestPath12	This action test the Series 2 instrument mux relays for ABus1 and ABus2.
EFTMuxTestPath34	This action test the Series 2 instrument mux relays for ABus3 and ABus4.
fetchDeviceByName	Diagnostics Action: This action returns the module handle of the module referenced by string name.
getICA	This action returns: 0 - no ICA defined 1 - Mass Interconnect (MAC panel) ICA 2 - Test System Interface (Express connect) ICA -1 - error condition
getModByName	This action returns the module handle of the module referenced by string name.
getModuleName	This action returns the string name of a module.
isThereA33120	This action returns a value of 1 if there is a 33120 Arbitrary Waveform Generator in the system.
isThereA34401	This action returns a value of 1 if a 34401 DMM is in the system.
isThereA53131	This action returns a value of 1 if a 53131 Counter is in the system.
PinAuxRelayTests	Tests the "Aux" relays for opens and shorts on the Pin Matrix Module.

Table 4-4. dgn (Diagnostic) Actions

PinAuxRelayTestsAll	Tests the "Aux" relays for opens and shorts on all of the Pin Matrix Modules in the system.
PinProtBypass	Closes relays to short the 200 ohm column protection resistors on the Pin Matrix Module.
PinRelayOpenTest	Tests the Pin Matrix Module for shorted relays.
PinRelayOpenTestAll	Tests all of the Pin Matrix Modules for stuck open relays.
PinRelayPathTest	Tests the Pin Matrix Module for stuck open relays.
PinRelayPathTestAll	Tests all the Pin Matrix Modules for stuck open relays.
PinReset	Reset Pin Matrix Module relays.
PinShortsTest	Test for shorts between pins on the Pin Matrix Module.
PinShortsTestAll	Test for shorts between pins on all of the Pin Matrix Modules in the system.
VI Debug Port Test	Tests debug ports 1-4 of the MCM using the DMM.
VI HV DC Accy	Tests HV DC source accuracy of the MCM using the DMM.
VI LV DC Accy	Tests LV DC source accuracy of the MCM using the DMM.
VI RelayTest1	Tests bus 1, 2 and UUTCOM matrix relays for opens and shorts in the MCM.
VI RelayTest2	Tests bus 3 and 4 matrix relays for opens and shorts in the MCM.
whichDmm	This action returns the system DMM as follows: 0 - No system DMM found 1 - E1411 with MCM 2 - 34401 with E8792 mux

Table 4-5. digitizer Actions

L53xA_AcqWave	Acquire Digitizer Waveform Trace.
L453xA_ConfAcq	Config Acquisition (Low Level).
L453xA_ConfArmTimer	Config Arm Source Time(Low Level).
L453xA_ConfChannel	Config L453xA Channel number and Vertical Range.
L453xA_ConfExTrigInOut	Configures the slope of the EXternal trigger BNC signal (Trig In/Out) for either ARM or TRIGger. You can configure the slope to be either a POSitive (rising) or NEGative (falling) edge.
L453xA_ConfTrigChanOff	This command configures which sources can cause a trigger event. This is the only way to set a trigger source.
L453xA_ConfTrigChanWin	This command configures window trigger attributes for the specified channel list. A window trigger specifies a voltage region within the channel's voltage range where a trigger is generated when the signal enters or leaves the window boundaries.
L453xA_ConfTrigTimestamp	This command specifies whether the digitizer's timestamp counter runs continuously or is cleared at each INITiate command.
L453xA_Init	This sets the command changes the state of the triggering system from the "idle" state to the "wait-for-arm" state.
L453xA_MeasDutyCycle	This query returns the duty cycle, starting at the midpoint of a falling or rising edge.

Table 4-5. digitizer Actions

L453xA_MeasFreq	This query returns the frequency (in Hz) of the given cycle, starting at the midpoint of a falling or rising edge. The frequency of a signal is defined as the reciprocal of the period (1 / period). For the MEASure:FALL:FRE-Quency[VERBose]? query, the period is defined as the time from the midpoint of the specified falling edge to the midpoint of the next falling edge.
L453xA_MeasOvershoot	This query returns the falling or rising overshoot as a percentage.
L453xA_MeasPeriod	This query returns the period of the given cycle (in seconds).
L453xA_MeasPreShoot	This query returns the falling or rising preshoot overshoot as a percentage.
L453xA_MeasPulseWidth	This query returns the pulse width (in seconds) of the given cycle, starting at the midpoint of a rising/falling edge.
L453xA_MeasTime	This query measures and returns the fall time value of the falling or rising edge closest to the trigger reference. Fall time is defined as the time in seconds from the upper threshold to the lower threshold of a falling edge. Rise time is defined as the time in seconds from the lower threshold to the upper threshold of a rising edge.
L453xA_MeasVamplitude	This query returns the amplitude for the specified <channel>, as assessed by the histogram method or the absolute method.
L453xA_MeasVaverage	This query returns the base voltage used for cycle based measurements. This value forms the lower voltage of an amplitude measurement. Base may be evaluated using the STANdard (histogram) method or the ABSolute method.
L453xA_MeasVmax	This query returns the maximum voltage for the specified record and channel.
L453xA_MeasVmin	This query returns the minimum voltage on the specified channel and record number.
L453xA_MeasVpp	This query returns the peak-to-peak voltage for the specified record on the specified channel.
L453xA_MeasVrms	This query returns the root mean square (RMS) average voltage for either a cycle or a measurement window.
L453xA_MeasVthreshold	This query returns the base voltage used for cycle based measurements. This value forms the lower voltage of an amplitude measurement. Base may be evaluated using the STANdard (histogram) method or the ABSolute method.
L453xA_MeasVtop	This query returns the top voltage used for cycle based measurements. This value forms the higher voltage of an amplitude measurement. Top may be evaluated using the STANdard (histogram) method or the ABSolute method.
L453xA_QueryCmd	Send a query command to the ADC.
L453xA_SendCmd	Send command to L453xA.
L453xA_SetMeasThreshold Abs	Configures the absolute threshold level for the given channels.
L453xA_SetMeasThreshold Percent	Configures the percentage threshold level for the given channels.

Table 4-5. digitizer Actions

L453xA_SetMeasThreshold TopBase	Configures the Top-Base method for the given channels. If <Method> ABSOLUTE is selected, vTop and Base value will be used to configure the absolute Top and Base threshold levels for the specified channels.
L453xA_SetMeasWin	Sets the measurement window in which on-board measurements are performed.
digitalReadRegCC	Low-Level Action: Reads a value from the specified E8794A custom card digital input register.
digitalWriteRegCC	Low-Level Action: Write a value to the specified Agilent E8794A Custom Card digital output register.
digitalWriteRegQueryCC	Low-Level Action: Returns the last value written to the specified register on the Agilent Technologies E8794A custom card.
Low-Level Actions	
ADCClear	Clears output buffer on digitizer (ADC) and aborts reading, if in progress.
ADCConfArm	Configures the arm subsystem of the E1563, E1564, or E1429 Digitizers.
ADCConfFreq	Configures the frequency subsystem of the E1563, E1564, or E1429 Digitizer (ADC).
ADCConfInControls	Configures the input circuit of the E1563, E1564, or E1429 Digitizer (ADC).
ADCFastOff	This action shuts off the Agilent1429 Digitizer's fast fetch mode until it is re-opened.
ADCGetSampleFreq	Returns the sample rate values from the most recent setup of the Agilent E1429 Digitizer.
ADCGetSampleInfo	Returns sample rate and data point count values from the most recent setup of the Agilent E1429 Digitizer (ADC).
ADCInitiate	Arms the E1563, E1564, or E1429 Digitizer (ADC) and retrieves the data after triggering occurs.
ADCIsSet	Waits until the E1563, E1564, or E1429 Digitizer (ADC) is ready for measurement.
ADCLastReading	Transfers the current digitizer data from the E1563, E1564, or E1429 from memory into a waveform data type to allow easy viewing.
ADCReset	Resets the digitizer (ADC) to its power-on state for all channels.
ADCSelfTest	Runs a selftest on the Agilent E1429 Digitizer (ADC) that is provided by its Plug and Play driver.
ADCSetReturnMode	This action allows time-outs on the specified E1563, E1564 or E1429 Digitizer (ADC) to raise an exception when encountered.
ADCSetTimeout	Sets the E1563, E1564 or E1429 Digitizer (ADC) operation time out value (in milliseconds).
ADCSetTimeoutException	This action allows enabling/disabling exception response to a timeout on the specified E1563, E1564 or E1429 digitizer (ADC).
ADCStart	Arms the Agilent E1429 Digitizer (ADC) for the configured trigger event and returns.
ADCType	Returns the type of digitizer (ADC) installed: either E1429, E1563, or E1564.

Table 4-5. digitizer Actions

digitalReadCC	Reads a value from the custom card digital input used in the Agilent E6198 switch/load unit.
digitalReadRegCC	Reads a value from the specified E8794A custom card digital input register.
digitalReadSU	Reads a value from the Agilent E6198 switch/load unit digital input ports.
digitalResetCC	Perform a soft reset on the Agilent E8794 custom card.
digitalWriteCC	Write a value to the Custom Card open collector digital output port used in the Agilent E6198 switch/load unit.
DigitalWriteQuery	Returns the current output state of the TS-5430 digital output ports.
digitalWriteQueryCC	Returns the current output state of the custom card open collector digital output port used in the Agilent E6198 switch/load unit.
digitalWriteQuerySU	Returns the current output state of the Agilent E6198 switch/load unit digital output ports.
digitalWriteRegCC	Write a value to the specified Agilent E8794A Custom Card digital output register.
digitalWriteRegQueryCC	Returns the last value written to the specified register on the Agilent Technologies E8794A custom card.
digitalWriteSU	Write a value to the Agilent E6198 switch/load unit digital output ports.
High-Level Actions	
ADC_Analyze_Wave	Measures a waveform using the E1563, E1564, or 1429 Digitizer to analyze it for Vmin, Vmax, High pulsewidth, Low pulsewidth and period values.
ADC_Config_1_Chan	Configures E1563, E1564, or E1429 for capturing a waveform for later analysis on Channel 1.
ADC_DCV	Returns instantaneous DC voltage from the Agilent Technologies E1563, E1564, or E1429 Digitizer using the "offset" parameter time from the trigger point.
ADC_DCV-Avg	Returns average DC voltage from the Agilent E1429 Digitizer in a window defined by the "start" parameter and "stop" parameter time values from the trigger point.
ADC_Min_Max	Measures a waveform by the E1563, E1564, or E1429 Digitizer and directly analyzes it for Vmin & Vmax values.
ADC_Min_Max_(Win)	Returns Vmin or Vmax DC voltage values from the E1563, E1564 or E1429 Digitizer in a window defined by the "start" parameter and "stop" parameter times from the trigger point.
ADC_Transform	Transform E1563, E1564, or E1429 Digitizer returned data with Measurement Control Module attenuator gain and offset terms.

Table 4-6. dio (Digital I/O) Actions

Low-Level Actions	
ADDigitalRead	Reads digital output data from the specified digital port on the Advantech 1750 PCI device.
ADDigitalWrite	Writes digital output data to the specified digital port on the Advantech 1750 PCI device.
DigitalRead	Reads a value from the TS-5430 digital input ports.
DigitalWrite	Write a value to the TS-5430 digital output ports.
m9187GetCardInfo	A brief description of DIO M9187A module information.
m9187InputConfigThresHold	Configures the level of both thresholds in one operation. Note that the two levels may be set to the same value if desired.
m9187InputQueryAll	Reads all 32 channels of input. Return values are as follows: 0 = State Low = below both thresholds; 1 = State High = above both thresholds; 2 = State Middle = in between the two threshold levels.
m9187InputQuerySingle	Reads the specified input channel. Return values are as follows: 0 = State Low = below both thresholds; 1 = State High = above both thresholds; 2 = State Middle = in between the two threshold levels.
m9187InputReadThresHoldLevel1	To read back Input Threshold level 1. Specifies threshold level 1. Units are volts.
m9187InputReadThresHoldLevel2	To read back Input Threshold level 2. Specifies threshold level 2. Units are volts.
m9187OutputConfigAll	Configures all 32 output channels.
m9187OutputConfigSingle	Configures the output state on a single output channel.
m9187OutputQueryAll	Returns all 32 output settings, with channel 1 in the lowest order array element. Each element contains a single channel's setting. The values returned are: 0 for a Sink (driven low), 1 for a Source (driven high), or 2 if the channel is Off (inactive).
m9187OutputQuerySingle	Returns the current setting of the specified output channel. The value returned will be either 0, 1, or 2 where: 0 = Sink (driven low), 1 = Source (driven high), and 2 = Off (inactive).
Spare Dig Read	Reads a value from the digital input port of the Agilent E1330 Digital I/O Module.
Spare Dig Write	Sends a value to the digital output port of the Agilent E1330 Digital I/O Module.

Table 4-7. DMM Actions

E1411/34401/34980A DMM Low-Level Actions	
dmmConfCal	Configures the 50/60Hz line frequency and enables/disables autozero of the Agilent E1411 DMM. (Ignores the line frequency parameter for the Agilent 34401A & 34980A DMMs).
dmmConfFunction	Set the DMM function, range and resolution of the Agilent E1411, Agilent 34401A, 34980A or AG918X DMM.
dmmConfSample	Selects the sample count is relevant for the Agilent 34401 and 34980A DMMs. Selects the sample source, sample count, and sample period of the Agilent E1411 DMM.
dmmConfSimple	Select one of three measurement types:fast measurement speed with low resolution, medium measurement speed with medium resolution, and slow measurement speed with high resolution.
dmmConfTrigIn	Selects the trigger input parameters of the Agilent E1411 or Agilent 34401 DMM.
dmmConfTrigOut	Lets you route the E1411 DMM's "voltmeter complete" signal to the VXibus TTL trigger lines.
dmmGetMultipleResults	Returns multiple readings from the DMM (waits, if necessary, for optionally triggering the DMM first).
dmmGetResults	Return one reading from the Agilent E1411, 34401, 34980A or AG918X DMM. It waits, if necessary, for optionally triggering the DMM first.
dmmInitiate	Initiate a reading cycle on the Agilent E1411, Agilent 34401 or AG918X DMM.
dmmIsSet	Wait until the Agilent E1411, Agilent 34401 or AG918X DMM is ready for measurement.
dmmPbQueryCmd	(For 34401 DMM only.) Sends a SCPI query string to the DMM and returns the result.
dmmPbSendCmd	(For 34401 DMM only.) Sends a SCPI command string.
dmmQueryCmd	(For 34401 DMM only.) Sends a SCPI query string to the 34401 and returns the result.
dmmReset	Reset the E1411 or 34401 or AG918X DMM to its power-on state.
DmmSelfTest	Runs a selftest on the E1411 or 34401 DMM that is provided by its Plug&Play driver.
dmmSendCmd	(For 34401 DMM only.) Sends a SCPI command string.
dmmSet	Send current setup information to the E1411 or 34401 DMM.
dmmSetTimeout	Sets the E1411 or 34401 DMM operation time out value (in milliseconds).
dmmVerifyTerminalSwitch	(For the 34401A DMM only.) Checks the state of the front/rear terminal switch on and compares the current switch state with the value in the parameter block.
E1411/34401/34980A DMM High-Level Actions	

Table 4-7. DMM Actions

dmmAutoZeroEX	(For the 34980A DMM only.) Sets Auto Zero to ON, OFF, or ONCE for the function specified without affecting the Auto Zero state of other functions.
dmmConfTrigInEX	(For the 34980A DMM only.) Lets you set up Bus (software) trigger mode as well as Immediate and External.
dmmGetMultipleResultsEX	(For the 34980A DMM only.) Returns the number of DMM measurements requested and generates an error if there are fewer measurements available than the number requested.
dmmGetResultEX	(For the 34980A DMM only.) Returns the most recent DMM measurement.
dmmMeas2WResEx	Perform an immediate resistance (2 wire) measurement without requiring configuration using 34980A or AG918X DMM.
dmmMeas4WResEx	Perform an immediate resistance (4 wire) measurement without requiring configuration using 34980A or AG918X DMM.
dmmMeasACCurrentEX	(For the 34980A DMM only.) Performs an immediate AC current measurement without requiring configuration.
dmmMeasACVoltsEX	(For the 34980A DMM only.) Performs an immediate AC voltage measurement without requiring configuration.
dmmMeasDCCurrentEX	(For the 34980A DMM only.) Performs an immediate DC current measurement without requiring configuration.
dmmMeasDCVoltsEX	(For the 34980A DMM only.) Performs an immediate DC voltage measurement without requiring configuration.
dmmMeasFrequencyEX	(For the 34980A DMM only.) Performs an immediate frequency measurement without requiring configuration.
dmmMeasPeriodEX	(For the 34980A DMM only.) Performs an immediate period measurement without requiring configuration.
dmmMeasureACI	Measure AC current using 34980A or AG918X DMM.
dmmMeasureACV	Measure AC voltage using the E1411, 34401 or AG918X DMM.
dmmMeasureCurrent	Measures voltage across a user supplied shunt or sense resistor to calculate current using the E1411 or 34401 DMM.
dmmMeasureDCI	Measure DC current using 34401 or AG918X DMM .
dmmMeasureDCV	Measure DC voltage using the Agilent E1411, Agilent 34401 or AG918X DMM.
dmmMeasureFrequency	(For the 34401 DMM only.) Measures frequency.
dmmMeasFrequencyPXI	(For AG918X DMM only) Measure frequency.
dmmMeasureOhms	Measure resistance using the Agilent E1411, Agilent 34401 DMM or AG918X.
dmmMeasurePeriod	(For the 34401 DMM only.) Measures period.
dmmMeasPeriodPXI	(For AG918X DMM only) Measures period.
dmmMeasCAPPXI	(For AG918X DMM only) Measures capacitance.

Table 4-7. DMM Actions

dmmMeasureTrigVoltage	Configures the Agilent E1411 or Agilent 34401 DMM to measure a DC voltage, triggered by a VXI backplane TTL signal (Agilent E1411 only) or external signal.
dmmSelfTestEX	(For the 34980A DMM only.) Resets the DMM and all cards in the chassis, sets the timeout to 20 seconds, executes the DMM Selftest, returns the result code and message, then returns the timeout to the previous value (even if the action fails).
dmmSendSWTrigEX	(For the 34980A DMM only.) Sends a Bus (software) trigger to the DMM.

Table 4-8. DSO Actions

dsoConfigureTrigger	To configure the trigger setting of the oscilloscope.
dsoGetDutyCycle	To measure and outputs the duty cycle of the signal.
dsoGetFallTime	To measure and outputs the fall time of the displayed falling (negative-going) edge closest to the trigger reference.
dsoGetFrequency	To measure and outputs the frequency of the cycle on the screen closest to the trigger reference.
dsoGetPeriod	To measure and outputs the period of the cycle on the screen closest to the trigger reference.
dsoGetPulseWidth	To measure and outputs the width of the displayed positive pulse closest to the trigger reference.
dsoGetRiseTime	To measure and outputs the rise time of the displayed rising (positive-going) edge closest to the trigger reference.
dsoGetVmaxVmin	To measure and outputs the maximum and minimum vertical value present on the selected waveform.
dsoGetVpp	To measure the maximum and minimum vertical value for the selected source, then calculates the vertical peak-to-peak value and returns that value.
dsoGetWaveform	To acquires the signal and save the data in the "Waveform" parameter array.
dsoMsgQuery	To send a SCPI query string to the oscilloscope and returns the result. THE RESULT BUFFER IS LIMITED TO 32767 CHARACTERS.
dsoMsgSend	To send a SCPI command string to the oscilloscope.

Table 4-9. Event Detector Actions

Low-Level Actions	
eventConf	Configures the Event Detector for a measurement.
eventInitiate	Starts the previously configured Event Detector measurement.
eventIsSet	Waits until the Event Detector is ready for operation.
eventReset	Resets the Event Detector to its power-on state.
eventSet	Sends current setup information to the Event Detector.
High-Level Actions	
eventGetResults	Returns stored data from the Event Detector.
eventMeasure	Makes an Event Detector measurement and stores data into memory.

Table 4-10. Generic Actions

Low-Level Actions	
DialogOkay	Display a message box with a prompt message and "Ok" button.
DialogOkayModal	Display a message box with a prompt message and "Ok" button.
DialogYesNo	Display a message in a dialog box; gives a Yes/No choice to continue in two different ways.
DialogYesNoModal	Display a message in a dialog box; gives a Yes/No choice to continue in two different ways.
GetFixtureID	Reads back the fixture I.
globalReset	Call all reset routines of instruments listed in the hardware configuration table.
msgDbIIO	This action sends a string from an Instrument and receives a double response.
msgIntIIO	This action sends a string from an Instrument and receives an integer response.
msgSend	This action sends a string to an instrument (INST).
msgStringIO	This action sends and reads strings from an instrument.
StoreLogString	This action allows the user to place any string into the log data storage area which will be used by any log record which accesses that field.
High-Level Actions	
ConcatenateStrings	The value of String2 is concatenated onto the end of String1.
DelayMicroseconds	Delays the test by the specified number of microseconds.
DelayMillisecond	Delays the test by the specified number of milliseconds.
DelaySecond	Delays the test by the specified number of seconds.
FormatString	The value of Argument is processed through the FormatString and will show up in ResultString.

Table 4-10. Generic Actions

IVIDriverDisable	Quickly places the instrument in a state where it has no, or minimal, effect on the external system to which it is connected. This state is not necessarily a known state.
IVIDriverErrorQuery	Queries the instrument and returns instrument specific error information. This function can be used when QueryInstrumentStatus is True to retrieve error details when the driver detects an instrument error.
IVIDriverLockObject	Obtains a multithread lock on the driver after waiting until all other execution threads have released their locks on the instrument session.
IVIDriverReset	Places the instrument in a known state and configures instrument options on which the IVI specific driver depends (for example, enabling/disabling headers). For an IEEE 488.2 instrument, Reset sends the command string *RST to the instrument.
IVIDriverSelfTest	Performs an instrument self test, waits for the instrument to complete the test, and queries the instrument for the results. If the instrument passes the test, TestResult is zero and TestMessage is 'Selftest passed'.
IVIDriverUnlockObject	Releases a previously obtained multithread lock.
LogDoubleToReport	This action allows the user to place any double into the log data file by producing a report record which will use the string argument.
LogIntToReport	This action allows the user to place any integer into the log data file by producing a report record which will use the string argument.
LogReport	This action allows the user to place any string into the log data file by producing a report record which will use the string argument.
LogReportImmediate.	This action allows the user to place any string into the log data file by producing a report record which will use the string argument.
NameSwitch	Switch string name.
ParameterLoad	Load (or write file if flag set) values for test parameters and skip control from[to] an ascii file.
UsecsElapsed	This action returns the time elapsed since the last "UsecsStart" Action.
UsecsStart	Starts clock for use with the "UsecsElapsed" Action.
UUTConnect	Connects to UUT pins by constructing Path names by adding a prefix (NamePrefix) to the base path name (BaseNames).
WaveformDataRead	Reads waveform sample data into waveform data type.
WaveformDataWrite	Write samples from 'hWaveform', one floating point number sample per line, to 'Filename'.

Table 4-11. Measurement Control Module Actions

Low-Level Actions	
viConfAttenuator	Configures the knee voltage of the MCM's programmable attenuator.
viConfCompare	Configures the MCM's comparator threshold voltage for triggering other instruments.
viConfSourceACV	Configures the MCM to source AC voltage.
viConfSourceDCI	Configures the MCM to source DC current.
viConfSourceDCV	Configures the MCM to source a DC voltage.
viIsSet	Waits until the MCM is ready for operation.
viReset	Resets the MCM to its power-on state.
viSet	Sends current setup information to the MCM.
High-Level Actions	
ForceIMeasV	Uses the MCM as a current source and measures the resultant voltage on the UUT using the DMM.
ForceIMeasValt	Uses the MCM as a current source to measures the resultant voltage on the UUT using the DMM.
ForceVMeasI	Uses the MCM as a voltage source and measures the resultant UUT current flow using the DMM.
viOutputACV	Configures the MCM to source AC voltages.
viOutputDCI	Configure the MCM to source DC current.
viOutputDCV	Configure the MCM to source DC voltage.
viSetCompare	Selects the comparator's trigger threshold voltage of the MCM.
viSetSourceACV	Configures the MCM to source AC voltages.
viSetSourceDCI	Configure the MCM to source DC current.
viSetSourceDCV	Configure the MCM to source DC voltage.

Table 4-12. Power Supply Actions

Low-Level Actions	
psConfTrigIn	Selects the trigger input of the Power Supplies.
psConfVI	Selects output voltage and current of the Power Supplies.
psConnect	Enables output relays of the selected Power Supply.
psDisconnect	Disconnects output relays from the selected Power Supply.
psGetCurrent	Reads and returns the output current of the Power Supply.
psGetStatus	Reads and returns Power Supply error status from last "psSet" or "psIsSet" Action.
psGetVoltage	Reads and returns the output voltage of the Power Supply.
psIsSet	Waits until the Power Supply is ready for output.
psProgVI	Programs and outputs the voltage/current of the Power Supply.

Table 4-12. Power Supply Actions

psReset	Resets the Power Supply to its power-on state.
psSelfTest	Runs a selftest on the Agilent Power Supply specified.
psSet	Sends configuration information to the Power Supply.
psSetReturnMode	Sets sequencer behavior on Power Supply timeout errors.
psSetTimeout	Sets Power Supply operation time out value (in milliseconds).
psSetVI	Selects the voltage and current parameters of the power supplies.
psTrigger	Send a software trigger to the Power Supply.
psPbMsgSend	Only send a SCPI command string to the power supply. Only valid for 6673A&N6700 Power Supply series only.
psPbMsgQuery	To send a SCPI query string to the power supply and returns the result. Only valid for 6673A&N6700 Power Supply series only.

Table 4-13. Serial Interface Actions

Low-Level Actions	
scommBreak	Sends or clears a break signal.
scommFlush	Flushes the specified port.
scommGetConfig	Returns the current configuration of the given serial port.
scommReceive	Reads the specified number of characters from the Serial Port and checks those numbers against the "compareString" parameter.
scommReceiveBytes	Reads the Serial Port.
scommReset	Resets the Serial Port.
scommSend	Sends data to the Serial Port.
scommSendBytes	Sends user specified bytes of data to the Serial Port.
scommSet	Sends current setup information to the Serial Port.
scommSpawnSendBytes	Sends user specified bytes of data to the Serial Port.
scommStop	Flushes the transmit buffer used for the Serial Port and resets the number of bytes sent and received to 0.
scommTransmit	Writes to and reads from the Serial Port the number of bytes requested.
scommTransmitBytes	Writes to and reads from the Serial Port the number of bytes requested.
scommWaitForSendBytes	Sends user specified bytes of data to the Serial Port.

Table 4-14. Automotive Serial Protocol Actions

CANGetInfo	Gets information on current state of J1939 or CAN interface including Bit Rate, Mode and Header Style (11 or 29 bit version).
CANRead	Reads CAN bus.
CANSetBitRate	Sets J1939 or CAN Interface bit rate in Bits/Sec.
CANSetIdMode	Sets Header Style (11 or 29 bit version).
CANSetReadFilter	Setup the CAN read filter.
CANSetTermination	Sets 124 ohm termination resistor across CAN bus (from CAN+ to CAN-).
CANSetUp	Performs general purpose set-up of CAN/J1939 protocol.
CANWrite	Writes to CAN bus.
CANWriteAlternate	Writes to CAN bus.
ISO9141EchoKeyword	Provides detection for 2 keywords and echos the complement keyword #2 as specified in ISO9141-2.
ISO9141FRead	Reads ISO9141 formatted data.
ISO9141FReadConfig	Constructs ISO9141 frame for the "ISO9141FRead" and "ISO9141Read" actions.
ISO9141FWrite	Writes formatted ISO9141 data.
ISO9141FWriteConfig	Configuration parameters for ISO9141 formatted write action.
ISO9141GetInfo	Gets information on the configuration and mode of operation of the ISO9141 serial interface device.
ISO9141InitSeq	Initializes the ISO9141 sequence.
ISO9141Read	Reads unformatted ISO9141 data.
ISO9141SetBitRate	Sets the bit rate of the ISO9141 serial protocol interface.
ISO9141SetByteTime	Sets the time delay between bytes in an ISO9141 message.
ISO9141SetUp	Provides general purpose set-up of ISO9141 protocol.
ISO9141Write	Writes unformatted data to the ISO9141 protocol.
J1850BlockTransfer	Reads data from a file and sends the data along with header data to the J1850 device.
J1850GetConfig	Gets information on the configuration and mode of operation of the J1850 serial interface device.
J1850Read	Reads unformatted J1850 data.
J1850SetConfig	Sets information on the configuration and mode of operation of the J1850 serial interface device.
J1850SetSourceAddr	Sets the physical ID of the sender of the message.
J1850Write	Writes unformatted J1850 data.
J1939Read	Reads J1939 bus.
J1939Write	Writes unformatted J1939 data.
mComClrRxBuff	Clears the receive buffer of the associated serial protocol interface.
mComConfigGroup	Sets information on the configuration and mode of operation of the group message feature.

Table 4-14. Automotive Serial Protocol Actions

mComGetBoardInfo	Returns results of Multicom serial interface board self test, board serial number, board version number and firmware/hardware id numbers for on-board functions & Physical Interface Module (PIM) slots.
mComMask	Converts string to real number.
mComReset	Resets the Multicom box.
mComSetBitRate	Sets the bit rate of the associated serial protocol interface.
mComStartGroup	Starts group messages.
mComStopGroup	Stops any group messages that were previously started by mComStartGroup.
softCANDefineID	(For the Softing CAN-AC2-PCI card.) Defines a Standard or Extended identifier.
softCANGetFifoLevels	(For the Softing CAN-AC2-PCI card.) Returns the number of transmit jobs in the transmit FIFO waiting to be transmitted by the interface and the number of events in the receive FIFO waiting to be read. The ReturnCode parameter indicates if any errors occurred.
softCANRead	(For the Softing CAN-AC2-PCI card.) Read data on the specified CAN channel.
softCANReadFifo	(For the Softing CAN-AC2-PCI card.) Reads a data frame from the FIFO.
softCANReadFifoAlt	(For the Softing CAN-AC2-PCI card.) Reads a data frame from the FIFO and returns integers. (This is similar to the softCANReadFifo action, which returns strings.)
softCANResetFifos	(For the Softing CAN-AC2-PCI card.) Resets the specified fifo. The Softing card shares the singular FIFO with both channels, so either channel can be passed in. The ReturnCode parameter indicates if any errors occurred.
softCANSetup	(For the Softing CAN-AC2-PCI card.) Sets up the Softing CAN-AC2-PCI card and permits changes to the settings for the two channels. The card is re-initialized, and any IDs that were previously defined are lost.
softCANStartGroupMsg	(For the Softing CAN-AC2-PCI card.) Starts group messaging (periodic or cyclic messaging) on one channel of the Softing card. This action starts group messaging for one ID. Call this action as many times as needed to send multiple IDs.
softCANStopGroupMsg	(For the Softing CAN-AC2-PCI card.) Stops group messaging for one ID. If you wish to stop group messaging for all IDs, enter "ALL" (no quotes) in the HexStringID parameter.

Table 4-14. Automotive Serial Protocol Actions

softCANWrite	(For the Softing CAN-AC2-PCI card.) Writes one ID and data to the CAN channel specified. The ID must be defined first (see softCANDefineID). Once a read or write has occurred, no more IDs can be defined until the card is reset via softCANSetup or a global reset.
softCANWriteFifo	(For the Softing CAN-AC2-PCI card.) Transmits a data frame onto the specified channel (CanChannel). The transmit request is processed through the transmit FIFO. The ReturnCode parameter indicates if the FIFO is full or any other errors occurred.
softCANWriteFifoAlt	(For the Softing CAN-AC2-PCI card.) Transmits a data frame onto the specified channel (CanChannel) using integers. (This is similar to the softCANWriteFifo action, which uses strings.) The transmit request is processed through the transmit FIFO. The ReturnCode parameter indicates if the FIFO is full or any other errors occurred.

Table 4-15. Switch Actions

Low-Level Actions	
GetLoadCardID	Reads the configuration register of the load box.
hardResetSU	Reset all Agilent E6198 switch units and all cards contained therein.
SUreset	Perform a soft reset on the Agilent Technologies E6198 switch unit.
High-Level Actions	
switching	This action connects / disconnects up to 10 switching paths.

Table 4-16. Load Card Actions

loadCardGetInfo	Returns information about the load card referenced by the instrument handle parameter.
loadCardReset	Routine to reset a given load card.
loadCardSwitchDiffAmp	(Applies only to the N9377A load card.) Controls the switching of the differential amplifier into the ISense path. (You must use a switching action to switch in the Current Sense node before you use this action.) Subsequent switching of any ISense paths on this card remove the differential amp from the path. This includes the switching cleanup of the ISense path. Default state does not have the differential amp in the ISense path.
loadCardTest	Validates that a relay is set to a given position.

Table 4-17. Voltage/Current Source Actions

m9186GetCardInfo	This action retrieves the M9186A card information.
m9186Reset	Reset the M9186A to its power-on state.
m9186SetHighCurrent	Set output current and level (high range).
m9186SetHighVoltage	Set output voltage and level (high range).
m9186SetLowCurrent	Set output current and level (low range).
m9186SetLowVoltage	Set output voltage and level (low range).

Table 4-18. DAQ M9216 Action List (IVI Version)

m9216daqConfAllInputPort	Configure input ports: Parameters setting include trigger level, hysteresis, trigger reference, trigCount, trigDelay, pre/post sample size and etc.
m9216daqConfInputPort	Configure input port: Parameters setting include trigger level, hysteresis, trigger reference, trigCount, trigDelay, pre/post sample size and etc.
m9216daqConfMeasurement	Configure daq to do single channel or multiple channel measurement and immediate or trigger mode. This configuration must be called before any measurement.
m9216daqGetCardInfo	Retrieves the card information.
m9216daqMultipleDCVAvg	Measure average DCV for every input port in multiple channel mode.
m9216daqMultipleMinMax	Measure min/max DCV for every input port in multiple channel mode.
m9216daqReset	Reset DAQ.
m9216daqSingleDCVAvg	Measure single input port average DCV.
m9216daqSingleMinMax	Measure single input port min/max DCV.
m9216daqSwitchInputPortToAux	Configure to switch selected input port to auxillary output port.
m9216daqConfAllMonitorInputPorts	Perform monitoring: Configure input ports.
m9216daqGetMeasureResult	Perform monitoring: Return measurement result of input ports selected for monitoring. This action must run parallel (multithread) with action m9216daqMultipleMonitoring.
m9216daqGetMonitorResult	Perform monitoring: Return input ports monitoring status and voltage triggered result. This action must run parallel (multithread) with action m9216daqMultipleMonitoring.

Table 4-18. DAQ M9216 Action List (IVI Version)

m9216daqMultipleMonitoring	Perform monitoring: Monitor (with the specified voltage trigger level) input ports to capture the electric spark, voltage drop and etc . It will return monitored and voltage triggered result after monitoring finished.
m9216daqSingleWave	Do single channel measurement and return raw data in waveform format.
m9216daqMultipleWave	Do multiple channel measurement and return raw data in waveform format

THIS PAGE IS INTENTIONALLY LEFT BLANK.

A

- Action Debug Messages, 25
- Action Quick Reference
 - AutoAdjust Actions, 84
 - Diagnostic Actions, 92
 - Low-Level Actions, 87
- Action Wizard
 - use to develop actions, 49
- Actions
 - adding a standard to a testplan, 19
 - adding a switching to a testplan, 20
 - high-level, 19
 - low-level, 19
 - Using the Action Wizard To Develop, 49
 - Using TS-5000 Supplied Actions, 19
- ADC. *See* E1563A Digitizer
- Addig Instruments, 58
- Adding a New Path, 41
- Adding a New Switching Path, 41
- Adding Aliases' Adding Wires, 58
- Adding/Modifying Aliases, 58
- Adding/Modifying Aliases, Wires, and Instruments, 58
- Adding/Modifying Instruments, 58
- Adding/Modifying Wires, 58
- Agilent E1328A 4-Channel Digital/Analog Converter
 - Debug Front Panel, 32
- Agilent E1333A 3-Channel Universal Counter
 - Debug Front Panel, 33
- Agilent E1333A Counter
 - Debug Front Panel, 33
- Agilent E1411B Digital Multimeter
 - Debug Front Panel, 30
- Agilent E1418A 16-Channel Digital/Analog Converter
 - Debug Front Panel, 32
- Agilent E6171 Measurement Control Module
 - Debug Front Panel, 36
- Agilent E6171B Measurement Control Module Front Panel, 36
- Agilent E6173A
 - memory requirements for downloading waveforms, 82
- Agilent E6173A Arbitrary Waveform Generator, 61
 - downloading a data type custom waveform, 70
 - Downloading and Executing Waveforms
 - Using the “Arb_DI_xxx” Actions, 63
 - generating custom waveforms, 70
 - generating custom waveforms from a file, 78
 - generating standard waveforms, 64
 - how to download a standard waveform, 64
 - how to generate waveforms
 - how the arb generates, 61
 - outputting a standard waveform
 - immediately, 66
- Agilent E6173A Arbitrary Waveform Generator
 - Description, 61
- Agilent E6173A Arbitrary Waveform Generator Error Messages, 82
- Agilent E6174A 32-Channel Event Detector
 - Debug Front Panel, 37
- Agilent E6174A Event Detector
 - Debug Front Panel, 37
- Agilent E6198A Switch/Load Unit
 - Debug Front Panel, 38
- Agilent Power Supplies
 - Debug Front Panel, 39
- Agilent TestExec SL
 - selecting, 17
- Aliases
 - adding, 58
 - modifying, 58
- Arb
 - downloading a data type custom waveform, 70
 - Downloading and Executing Waveforms
 - Using the “Arb_DI_xxx” Actions, 63
 - generating arbitrary waveforms, 70
 - generating custom waveforms from a file, 78
 - generating standard waveforms, 64
 - How to download a standard waveform, 64
 - how to generate waveforms, 61
 - memory requirements for downloading waveforms, 82
 - outputting a standard waveform
 - immediately, 66

- arb (Arbitrary Waveform Generator) Actions, 84
- Arb Description, 61
- Arb Error Messages, 82
- Arb Waveforms
 - generating, 62
- Arbitrary Waveform Generator
 - generating arbitrary waveforms, 70
 - generating standard waveforms, 64
 - using, 61
- Arbitrary Waveform generator Description, 61
- Arbitrary Waveform Generator Error Messages, 82

B

- Breakpoints, 24

C

- Computer Hardware
 - required, 15
- Counter
 - Debug Front Panel, 33
- counter Actions, 87
- Counter. *See* Agilent E1333A 3-Channel Universal Counter
- Counter. *See* Agilent E1333A Counter
- Creating a Testplan, 18
- Creating a User Defined Switching Path, 40
- Custom Waveform
 - downloading a data type, 70
- Custom Waveforms
 - generating (Arbitrary Waveform Generator), 70
 - generating from a file, 78

D

- dac (D/A Converter) Actions, 88
- DAC. *See* Agilent E1328A 4-Channel Digital/Analog Converter *or* Agilent E1418A 16-Channel Digital/Analog Converter
- Data Type
 - Waveform, 45
- Debug Features
 - using software, 23
- Debug Front Panel
 - Agilent 1411B Digital Multimeter, 30
 - Agilent E1328A 4-Channel Digital/Analog Converter, 32
 - Agilent E1333A 3-Channel Universal Counter, 33

- Agilent E1333A Counter, 33
- Agilent E1418A 16-Channel Digital/Analog Converter, 32
- Agilent E6171B Measurement Control Module, 36
- Agilent E6174A 32-Channel Event Detector, 37
- Agilent E6174A Event Detector, 37
- Agilent E6198A Switch/Load Unit, 38
- Agilent Power Supplies, 39
 - Counter, 33
 - Digital Multimeter, 30
 - Digitizer, 35
 - E1563A Digitizer, 35
 - Event Detector, 37
 - Measurement Control Module, 36
 - Power Supplies, 39
 - Switch/Load Unit, 38
- Debug Instrument Front Panels, 28
- Debug Messages, 25
- Debug Panel
 - selecting, 29
- Debug Panel Features, 28
- Debug Panel Types, 28
- Debug Panels
 - instrument, 28
- Debug Path Selection, 28, 40
- Deleting a New Path, 41
- Deleting a New Switching Path, 41
- Description
 - Agilent E6173A Arbitrary Waveform Generator, 61
- Develop actions
 - using the "Action Wizard", 49
- dgn (Diagnostic) Actions, 92
- Digital Multimeter
 - Debug Front Panel, 30
- Digitizer
 - Debug Front Panel, 35
- digitizer Actions, 93
- dio (Digital I/O) Actions, 97
- DMM
 - how to trigger, 58
- DMM Actions, 98
- DMM. *See* Agilent E1411B Digital Multimeter
- Download a Standard Waveform
 - how to, 64
- Downloading a Data Type Custom Waveform, 70
- Downloading Waveforms
 - memory requirements, 82

Downloading Waveforms Using the “Arb_DI_XXX”
Actions, 63

E

Editing a New Path, 41
Editing a New Switching Path, 41
Error Messages
 Agilent E6173A Arbitrary Waveform
 Generator, 82
event Actions, 100–101, 108
Event Detector
 Debug Front Panel, 37
EVT. *See* Agilent E6174A Event Detector
Example Testplans, 20
Executing Waveforms Using the “Arb_DI_XXX”
 Actions, 63

F

Features
 debug panel, 28
 using software debug, 23
File
 generating custom waveform from a, 78
For/Next Sequences, 23
Front Panels
 Debug, 28
 instrument, 28

G

Generating Arb Waveforms, 62
Generating Custom Waveforms, 70
Generating Custom Waveforms From a Data File, 78
Generating Standard Waveforms, 64
Generic Actions, 101

H

Handlers
 instrument
 Handlers
 module, 48
High-Level Actions, 19
How the Arb Generates Waveforms, 61
How to Download a Standard Waveform, 64
How to Trigger the DMM, 58

I

Instrument Debug Front Panel
 features, 28

 selecting, 29
 types, 28
Instrument Debug Panels, 28
Instrument Front Panels, 28
 Debug, 28
Instrument Handlers, 48
Instrument Status, 26
Instruments
 adding, 58
 modifying, 58

L

Load Card Actions, 107
Loading a Testplan, 17
Looping, 23
Low-Level Actions, 19

M

mcm Actions, 103
MCM. *See* Agilent E6171B Measurement Control
 Module
Measurement Control Module
 Debug Front Panel, 36
Memory Requirements for Downloading
 Waveforms, 82
Menu Selections
 watch window, 25
Messages
 debug, 25
Modifying Aliases, 58
Modifying Aliases, Wires, and Instruments, 58
Modifying Instruments, 58
Modifying Wires, 58
Module/Instrument Handlers, 48
Monitor Testplans, 25
Multithreading, 50
 thread safe action, 50
 threaded test settings, 52
 using resource locking, 53
 using timeout, 56
 using wait thread statement, 54

O

Outputting a Standard Waveform Immediately, 66

P

Panel Types
 debug, 28

Path
 adding a new switching, 41
 creating a user defined switching, 40
 deleting a new switching, 41
 editing a new switching, 41

Path Selection
 debug, 28, 40
power (Power Supply) Actions, 103
Power Supplies
 Debug Front Panel, 39
Product Version Capability
 using the, 48

Q

Quick Reference
 AutoAdjust Actions, 84
 Diagnostic Actions, 92
 Low-Level Actions, 87

R

Required Computer Hardware and Software, 15
Resource Locking, 53

S

Selecting Agilent TestExec SL, 17
Selecting the Debug Panel, 29
Selections
 watch window menu, 25
serial Actions, 104
SerialProtocol (Automotive Serial Protocol)
 Actions, 105
Single Stepping, 24
Software
 description, 16
 required, 15
Specifying Unit Under Test Pins, 40
Standard waveform
 how to download, 64
Standard Waveforms
 generating (Arbitrary Waveform
 Generator), 64
SU. *See* Agilent E6198A Switch/Load Unit
switch Actions, 107
Switch/Load Unit
 Debug Front Panel, 38
Switching Path
 adding a new, 41
 creating a user defined, 40

 deleting a new, 41
 editing a new, 41
System Software Description, 16
System Software Overview, 9

T

Testplan
 adding a standard action, 20
 adding a switching action, 20
 creating, 18
 how multithreading working, 50
 loading, 17
Testplans
 example, 20
 monitoring, 25
Thread Safe Action, 50
Threaded Test, 52
Timeout, 56
Triggering
 how to Trigger the DMM, 58
TS-5000 Actions
 adding a standard to a testplan, 20
 adding a switching to a Testplan, 20
 high-level, 19
 low-level, 19
 standard types, 19
 using, 19

U

Unit Under Test Pins
 specifying, 40
using, 61
Using Software Debug Features, 23
Using the "Action Wizard" to Develop Actions, 49
Using the Agilent E6173A Arbitrary Waveform
 Generator, 61
Using the Product Version Capability, 48
Using TS-5000 Supplied Actions, 19

V

Viewing Waveforms, 45
VXI Technology E1563A Digitizer
 Debug Front Panel, 35

W

Wait Thread Statement, 54
Watch Window, 25
 Instrument Status, 26

- Watch Window Menu Selections, 25
- Waveform
 - outputting a standard immediately, 66
- Waveform Data Type, 45
 - viewing, 45
- Waveform generator
 - using, 61
- Waveforms
 - generating Arb, 62
 - viewing, 45
- Wires
 - adding, 58
 - modifying, 58