

# **Agilent eMMC Test Application**

**Programmer's Reference**



**Agilent Technologies**

# Notices

© Agilent Technologies, Inc. 2013

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.

## Trademarks

Microsoft®, MS-DOS®, Windows®, Windows 2000®, and Windows XP® are U.S. registered trademarks of Microsoft Corporation.

Adobe®, Acrobat®, and the Acrobat Logo® are trademarks of Adobe Systems Incorporated.

## Manual Part Number

Version 01.00.0000

## Edition

January 16, 2013

Available in electronic format only

Agilent Technologies, Inc.  
1900 Garden of the Gods Road  
Colorado Springs, CO 80907 USA

## 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.

## In This Book

This book is your guide to programming the Agilent Technologies eMMC Test Application.

- [Chapter 1](#), “Introduction to Programming,” starting on page 7 describes compliance application programming basics.
- [Chapter 2](#), “Configuration Variables and Values,” starting on page 11, [Chapter 3](#), “Test Names and IDs,” starting on page 15, and [Chapter 4](#), “Instruments,” starting on page 21, provide information specific to programming the eMMC Test Application.

### How to Use This Book

Programmers who are new to compliance application programming should read all of the chapters in order. Programmers who are already familiar with this may review chapters 2, 3, and 4 for changes.



# Contents

In This Book 3

## 1 Introduction to Programming

Remote Programming Toolkit 8

Licensing 9

## 2 Configuration Variables and Values

## 3 Test Names and IDs

## 4 Instruments

Index





# 1 Introduction to Programming

Remote Programming Toolkit 8

Licensing 9

This chapter introduces the basics for remote programming a compliance application. The programming commands provide the means of remote control. Basic operations that you can do remotely with a computer and a compliance app running on an oscilloscope include:

- Launching and closing the application.
- Configuring the options.
- Running tests.
- Getting results.
- Controlling when and where dialogs get displayed
- Saving and loading projects.

You can accomplish other tasks by combining these functions.



## Remote Programming Toolkit

The majority of remote interface features are common across all the Agilent Technologies, Inc. family of compliance applications. Information on those features is provided in the N5452A Compliance Application Remote Programming Toolkit available for download from Agilent here: "[www.agilent.com/find/scope-apps-sw](http://www.agilent.com/find/scope-apps-sw)". The eMMC Test Application uses Remote Interface Revision 2.20. The help files provided with the toolkit indicate which features are supported in this version.

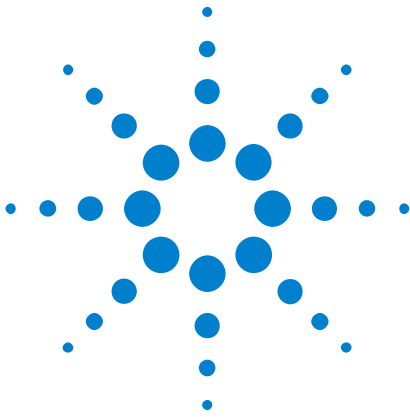
In the toolkit, various documents refer to "application-specific configuration variables, test information, and instrument information". These are provided in Chapters 2, 3, and 4 of this document, and are also available directly from the application's user interface when the remote interface is enabled (View>Preferences::Remote tab::Show remote interface hints). See the toolkit for more information.



## Licensing

To enable programming of compliance applications on your oscilloscope, please visit "[www.agilent.com/find/scope-apps](http://www.agilent.com/find/scope-apps)" to purchase an N5452A remote programming option license.

## **1 Introduction to Programming**



## 2 Configuration Variables and Values

The following table contains a description of each of the eMMC Test Application options that you may query or set remotely using the appropriate remote interface method. The columns contain this information:

- GUI Location – Describes which graphical user interface tab contains the control used to change the value.
- Label – Describes which graphical user interface control is used to change the value.
- Variable – The name to use with the SetConfig method.
- Values – The values to use with the SetConfig method.
- Description – The purpose or function of the variable.

For example, if the graphical user interface contains this control on the **Set Up** tab:

- Enable Advanced Features

then you would expect to see something like this in the table below:

**Table 1** Example Configuration Variables and Values

GUI Location	Label	Variable	Values	Description
Set Up	Enable Advanced Features	EnableAdvanced	True, False	Enables a set of optional features.

and you would set the variable remotely using:

```
ARSL syntax  
-----  
arsl -a ipaddress -c "SetConfig 'EnableAdvanced' 'True'"
```

```
C# syntax  
-----  
remoteAte.SetConfig("EnableAdvanced", "True");
```

## 2 Configuration Variables and Values

Here are the actual configuration variables and values used by this application:

### NOTE

Some of the values presented in the table below may not be available in certain configurations. Always perform a "test run" of your remote script using the application's graphical user interface to ensure the combinations of values in your program are valid.

### NOTE

The file, ""ConfigInfo.txt"", which may be found in the same directory as this help file, contains all of the information found in the table below in a format suitable for parsing.

**Table 2** Configuration Variables and Values

GUI Location	Label	Variable	Values	Description
Configure	CMD Channel	CMDCHAN	CHANnel1, CHANnel2, CHANnel3, CHANnel4	The channel to get CMD from.
Configure	CMD Memory	CMDCHANWFM	WMEMory1, WMEMory2, WMEMory3, WMEMory4	The waveform memory to get CMD from.
Configure	Clock Channel	CKCHAN	CHANnel1, CHANnel2, CHANnel3, CHANnel4	The channel to get clock from.
Configure	Clock Memory	CKCHANWFM	WMEMory1, WMEMory2, WMEMory3, WMEMory4	The waveform memory to get clock from.
Configure	DAT Channel	DATCHAN	CHANnel1, CHANnel2, CHANnel3, CHANnel4	The channel to get DAT from.
Configure	DAT Memory	DATCHANWFM	WMEMory1, WMEMory2, WMEMory3, WMEMory4	The waveform memory to get DAT from.
Configure	DAT Pin Under Test	DATPin	(Accepts user-defined text), DAT0, DAT1, DAT2, DAT3, DAT4, DAT5, DAT6, DAT7	Enter the DAT pin DAT0-7 that is being tested

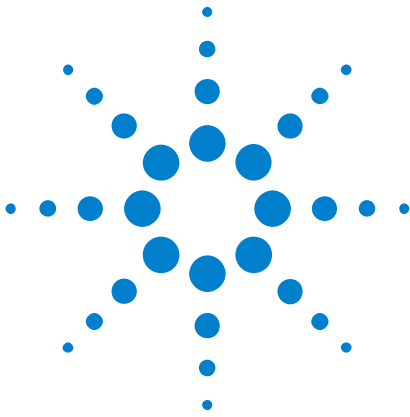
**Table 2** Configuration Variables and Values (continued)

GUI Location	Label	Variable	Values	Description
Configure	Limited Waveform Files	WFMFilesLim	Y, N	Select "Yes" if the waveform files were created knowing they are the beginning of a command cycle for CMD tests or knows read or write data for DAT tests. Select "No" if waveform files are uncertain. If "Yes" is selected, the basic logic of the test will be ignored. (i.e. DAT test logic uses all three channels to guarantee a data burst is read or write, only Clock and DAT would be required if "No" selected.) In short, if "Yes" is selected, the user is guaranteeing the data presented with the .wfm is the correct data to be measured.
Configure	Number of Clock Edge Measurements	NumMeas	(Accepts user-defined text), 200	Enter the number of clock edges to Measure
Configure	Use Waveform Files	WFMFiles	Y, N	Select Yes to use loaded .wfm files, No to use real time signals
Configure	Vcc	VccVal	(Accepts user-defined text), 1.7, 1.825, 1.95, 2.7, 3.25, 3.6	Enter the Vcc Value being tested
Configure	Vccq	VccqVal	(Accepts user-defined text), 1.1, 1.2, 1.3, 1.7, 1.825, 1.95, 2.7, 3.25, 3.6	Enter the Vccq Value being tested
Run Tests	Event	RunEvent	(None), Fail, Margin < N, Pass	Names of events that can be used with the StoreMode=Event or RunUntil RunEventAction options
Run Tests	RunEvent=Margin < N: Minimum required margin %	RunEvent_Margin < N_MinPercent	Any integer in range: 0 <= value <= 100	Specify N using the 'Minimum required margin %' control.
Set Up	Device ID	pcboOverallDeviceID	(Accepts user-defined text)	This option allow user to key in related test details.

## 2 Configuration Variables and Values

**Table 2** Configuration Variables and Values (continued)

<b>GUI Location</b>	<b>Label</b>	<b>Variable</b>	<b>Values</b>	<b>Description</b>
Set Up	Speed Grade	DeviceType	Backwards, High-speed, Dual-rate, HS200	This option allow user to select specific speed grade.
Set Up	Test Mode	TestMode	Compliance, Custom	This option allow user to select test mode.
Set Up	User Comment	txtOverallUserComment	(Accepts user-defined text)	This option allow user to key in related test detail.
Set Up	User Comment	txtOverallUserComment	(Accepts user-defined text)	This option allow user to key in related test detail.
Set Up	User Description	pcboOverallDeviceDescripti on	(Accepts user-defined text)	This option allow user to key in test detail.



## 3 Test Names and IDs

The following table shows the mapping between each test's numeric ID and name. The numeric ID is required by various remote interface methods.

- Name – The name of the test as it appears on the user interface **Select Tests** tab.
- Test ID – The number to use with the RunTests method.
- Description – The description of the test as it appears on the user interface **Select Tests** tab.

For example, if the graphical user interface displays this tree in the **Select Tests** tab:

- All Tests
  - Rise Time
  - Fall Time

then you would expect to see something like this in the table below:

**Table 3** Example Test Names and IDs

Name	Test ID	Description
Fall Time	110	Measures clock fall time.
Rise Time	100	Measures clock rise time.

and you would run these tests remotely using:

```
ARSL syntax
-----
arsl -a ipaddress -c "SelectedTests '100,110'"
arsl -a ipaddress -c "Run"
```

```
C# syntax
-----
remoteAte.SelectedTests = new int [] {100,110};
remoteAte.Run();
```

Here are the actual Test names and IDs used by this application:

**NOTE**

The file, ""TestInfo.txt"", which may be found in the same directory as this help file, contains all of the information found in the table below in a format suitable for parsing.

**Table 4** Test IDs and Names

Name	TestID	Description
Duty Cycle: Dual Rate	300	Duty Cycle for Dual Rate Device interface timing
Duty Cycle: HS200	403	Duty Cycle for HS200 Device interface timing
Vih(CMD)	3	Vih for Bus Signal Levels (CMD)
Vih(Clock)	1	Vih for Bus Signal Levels (Clock)
Vih(DAT)	9	Vih for Bus Signal Levels (CMD)
Vil(CMD)	4	Vil for Bus Signal Levels (CMD)
Vil(Clock)	2	Vil for Bus Signal Levels (Clock)
Vil(DAT)	10	Vil for Bus Signal Levels (DAT)
Voh(CMD): Open Drain	7	Voh for Bus Signal Levels (CMD)
Voh(CMD): Push-Pull	5	Voh for Bus Signal Levels (CMD)
Voh(DAT)	11	Voh for Bus Signal Levels (DAT)
Vol(CMD): Open Drain	8	Vol for Bus Signal Levels (CMD)
Vol(CMD): Push-Pull	6	Vol for Bus Signal Levels (CMD)
Vol(DAT)	12	Vol for Bus Signal Levels (DAT)
fOD Clock frequency Identification Mode: Backward-compatible	201	fOD for Backward Compatible Device interface timing
fOD Clock frequency Identification Mode: High-speed	101	fOD for High-speed Device interface timing
fpp Clock frequency Data Transfer Mode: Backward-compatible	200	fpp for Backward Compatible Device interface timing
fpp Clock frequency Data Transfer Mode: High-speed	100	fpp for High-speed Device interface timing
tFALL(CMD) Output Fall Time: Dual Rate	308	tFALL(CMD) for Dual Rate Device interface timing
tFALL(CMD) Output Fall Time: High-speed	111	tRISE(CMD) for High-speed Device interface timing
tFALL(DAT) Output Fall Time: Dual Rat	313	tFALL(DAT) for Dual Rate Device interface timing



**Table 4** Test IDs and Names (continued)

Name	TestID	Description
tFALL(DAT) Output Fall Time: High-speed	117	tFALL(DAT) for High-speed Device interface timing
tIH(CMD) Input Hold Time: Backward-compatible	207	tIH(CMD) for Backward Compatible Device interface timing
tIH(CMD) Input Hold Time: HS200	405	tIHddr(CMD) for HS200 Device interface timing
tIH(CMD) Input Hold Time: High-speed	107	tIH(CMD) for High-speed Device interface timing
tIH(DAT) Input Hold Time: HS200	409	tIH(DAT) for HS200 Device interface timing
tIH(DAT) Input Hold Time: High-speed	113	tIH(DAT) for High-speed Device interface timing
tIH(DAT) Input Setup Time: Backward-compatible	211	tIH(DAT) for Backward Compatible Device interface timing
tIHddr(CMD) Input Hold Time: Dual Rate	304	tIHddr(CMD) for Dual Rate Device interface timing
tIHddr(DAT) Input Hold Time: High-speed	310	tIHddr(DAT) for Dual Rate Device interface timing
tISU(CMD) Input Setup Time: Backward-compatible	206	tISU(CMD) for Backward Compatible Device interface timing
tISU(CMD) Input Setup Time: HS200	404	tISU(CMD) for HS200 Device interface timing
tISU(CMD) Input Setup Time: High-speed	106	tISU(CMD) for High-speed Device interface timing
tISU(DAT) Input Setup Time: Backward-compatible	210	tISU(DAT) for Backward Compatible Device interface timing
tISU(DAT) Input Setup Time: HS200	408	tISU(DAT) for HS200 Device interface timing
tISU(DAT) Input Setup Time: High-speed	112	tISU(DAT) for High-speed Device interface timing
tISUddr(CMD) Input Setup Time: Dual Rate	303	tISUddr(CMD) for Dual Rate Device interface timing
tISUddr(DAT) Input Setup Time: Dual Rate	309	tISUddr(DAT) for Dual Rate Device interface timing
tODLY(CMD) Output Delay Time: Dual Rate	305	tODLY(CMD) for Dual Rate Device interface timing
tODLY(CMD) Output Delay Time: High-speed	108	tODLY(CMD) for High-speed Device interface timing

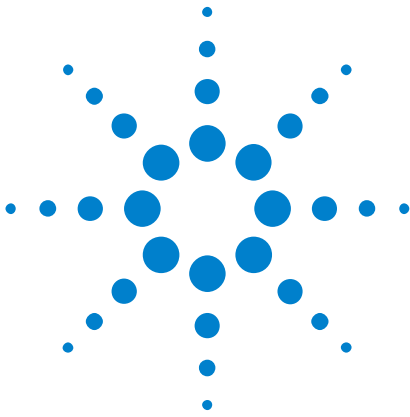
**Table 4** Test IDs and Names (continued)

Name	TestID	Description
tODLY(DAT) Output Delay Time: High-speed	114	tODLY(DAT) for High-speed Device interface timing
tODLYddr(DAT) Output Delay Time: Dual Rate	311	tODLYddr(DAT) for Dual Rate Device interface timing
tOH(CMD) Output Hold Time: Backward-compatible	209	tOH(CMD) for Backward Compatible Device interface timing
tOH(CMD) Output Hold Time: Dual Rate	306	tOH(CMD) for Dual Rate Device interface timing
tOH(CMD) Output Hold Time: High-speed	109	tOH(CMD) for High-speed Device interface timing
tOH(DAT) Output Hold Time: High-speed	115	tOH(DAT) for High-speed Device interface timing
tOH(DAT) Output Setup Time: Backward-compatible	213	tOH(DAT) for Backward Compatible Device interface timing
tOSU(CMD) Output Setup Time: Backward-compatible	208	tOSU(CMD) for Backward Compatible Device interface timing
tOSU(DAT) Output Setup Time: Backward-compatible	212	tOSU(DAT) for Backward Compatible Device interface timing
tPERIOD Clock frequency: HS200	400	tPERIOD for HS200 Device interface timing
tRISE(CMD) Output Rise Time: Dual Rate	307	tRISE(CMD) for Dual Rate Device interface timing
tRISE(CMD) Output Rise Time: High-speed	110	tRISE(CMD) for High-speed Device interface timing
tRISE(DAT) Output Rise Time: Dual Rate	312	tRISE(DAT) for Dual Rate Device interface timing
tRISE(DAT) Output Rise Time: High-speed	116	tRISE(DAT) for High-speed Device interface timing
tTHL Clock Fall Time: Backward-compatible	205	tTHL for Backward Compatible Device interface timing
tTHL Clock Fall Time: Dual Rate	302	tTHL for Dual Rate Device interface timing
tTHL Clock Fall Time: HS200	402	tTHL for HS200 Device interface timing
tTHL Clock Fall Time: High-speed	105	tTHL for High-speed Device interface timing
tTLH Clock Rise Time: Backward-compatible	204	tTLH for Backward Compatible Device interface timing

**Table 4** Test IDs and Names (continued)

<b>Name</b>	<b>TestID</b>	<b>Description</b>
tTLH Clock Rise Time: Dual Rate	301	tTLH for Dual Rate Device interface timing
tTLH Clock Rise Time: HS200	401	tTLH for HS200 Device interface timing
tTLH Clock Rise Time: High-speed	104	tTLH for High-speed Device interface timing
tVW(DAT) Output Valid Window Time: HS200	410	tVW(DAT) for HS200 Device interface timing
tWH Clock High Time: Backward-compatible	202	tWH for Backward Compatible Device interface timing
tWH Clock High Time: High-speed	102	tWH for High-speed Device interface timing
tWL Clock Low Time: Backward-compatible	203	tWL for Backward Compatible Device interface timing
tWL Clock Low Time: High-speed	103	tWL for High-speed Device interface timing

### **3 Test Names and IDs**



## 4 Instruments

The following table shows the instruments used by this application. The name is required by various remote interface methods.

- Instrument Name – The name to use as a parameter in remote interface commands.
- Description – The description of the instrument.

For example, if an application uses an oscilloscope and a pulse generator, then you would expect to see something like this in the table below:

**Table 5** Example Instrument Information

Name	Description
scope	The primary oscilloscope.
Pulse	The pulse generator used for Gen 2 tests.

and you would be able to remotely control an instrument using:

ARSL syntax (replace [description] with actual parameter)

```
-----  
arsl -a ipaddress -c "SendScpiCommandCustom 'Command=[scpi  
command];Timeout=100;Instrument=pulsegen'"
```

```
arsl -a ipaddress -c "SendScpiQueryCustom 'Command=[scpi  
query];Timeout=100;Instrument=pulsegen'"
```

C# syntax (replace [description] with actual parameter)

```
-----  
SendScpiCommandOptions commandOptions = new SendScpiCommandOptions();  
commandOptions.Command = "[scpi command]";  
commandOptions.Instrument = "[instrument name]";  
commandOptions.Timeout = [timeout];  
remoteAte.SendScpiCommand(commandOptions);
```

```
SendScpiQueryOptions queryOptions = new SendScpiQueryOptions();  
queryOptions.Query = "[scpi query]";  
queryOptions.Instrument = "[instrument name]";  
queryOptions.Timeout = [timeout];  
remoteAte.SendScpiQuery(queryOptions);
```



## 4 Instruments

Here are the actual instrument names used by this application:

### NOTE

The file, ""InstrumentInfo.txt"", which may be found in the same directory as this help file, contains all of the information found in the table below in a format suitable for parsing.

---

**Table 6** Instrument Names

<b>Instrument Name</b>	<b>Description</b>
scope	The primary oscilloscope.

# Index

## C

configuration variables and values, [11](#)

## I

IDs and names of tests, [15](#)  
instrument names, [21](#)

## L

licensing, [9](#)

## N

names and IDs of tests, [15](#)  
names of instruments, [21](#)  
notices, [2](#)

## P

programming, introduction to, [7](#)

## R

Remote Programming Toolkit, [8](#)

## T

test names and IDs, [15](#)  
trademarks, [2](#)

## V

variables and values, configuration, [11](#)

