
Keysight DigitalTestApps

Notices

© Keysight Technologies, Inc. 2008-2023

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 Keysight Technologies, Inc. as governed by United States and international copyright laws.

Revision

Version 7.20.0

Edition

September 1, 2023

Available in electronic format only

Published by:

Keysight 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, Keysight 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. Keysight 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 Keysight 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 License

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.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement ("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), 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 an introduction to the Keysight DigitalTestApps Remote Programming Interface. For more information, please see the *Keysight DigitalTestApps Programming Guide*.

Contents

In This Book / 3

1 Introduction to Programming

How to Use This Toolkit / 9

Remote Development Toolkit / 9

Usage Scenarios / 9

System Requirements / 14

On the Server / 14

On the Client / 15

Programming Languages / 17

TAP / 17

ARSL / 17

.NET Framework / 18

Python / 19

LabVIEW / 19

Index

1 Introduction to Programming

How to Use This Toolkit / 9

System Requirements / 14

Programming Languages / 17

This document introduces the basics for remote programming of an automated test application. The programming commands provide the means of remote control. Basic operations that you can do remotely with a computer and an automated test application running on an oscilloscope or PC 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.

Computers communicate to an automated test application running on an oscilloscope by sending and receiving messages over a Microsoft .NET Remoting interface. This may be accomplished in any of these ways:

- Using the Keysight KS8400A Test Automation for PathWave version 9 along with the DigitalTest Apps API plugin.
- Using one of the remote client executables provided by Keysight Technologies in the N5452A Automated Test Application Remote Development Toolkit.
- Create a custom remote client containing commands written in a .NET Framework 2.0 (or newer) programming language such as C# 2.0. Freely available samples are provided by Keysight Technologies in the Automated Test Application Remote Development Toolkit.
- Create a custom remote client containing commands written in Python 2.7. Freely available samples are provided by Keysight Technologies in the Automated Test Application Remote Development Toolkit.

1 Introduction to Programming

- Perform graphical programming using National Instrument's LabVIEW 8.5 environment.

How to Use This Toolkit

Remote Development Toolkit

If you haven't done this already, download and install on your development computer the N5452A Automated Test Application Remote Development Toolkit from the Keysight web site at www.keysight.com/find/scope-apps. This toolkit contains helpful guides, executable sample clients (including source code), and projects to help you get started developing your own remote client. It contains sample Keysight KS8400A Test Automation for PathWave version 9 test plans as well as programming examples written in Microsoft C# 3.0 (which may be compiled using a .NET Framework 3.5-compatible compiler such as Microsoft Visual Studio 2013), Python 2.7, and in National Instruments LabVIEW (which requires the LabVIEW 8.5 environment).

NOTE

After installing the toolkit on your development computer, you can access its contents by using either the "Keysight Automated DigitalTest App Remote Toolkit" shortcut on your desktop or the Start menu path: **All Programs > Keysight DigitalTest Applications > Remote Development Toolkit**.

Usage Scenarios

If one of the following scenarios describes what you need to do, read the topics listed in the section to learn more about it.

- **"I want to write commands for the 'Automation' tab of my automated test application"** on page 9
- **"I am using the Keysight KS8400A Test Automation for PathWave"** on page 10
- **"I want to try a sample remote client provided by the toolkit"** on page 10
- **"I want to use the Application Sequencer to execute saved projects for multiple applications"** on page 11
- **"I am using a test harness that can execute .NET Framework assemblies"** on page 11
- **"I am using a test harness that can execute Python"** on page 11
- **"I want to develop my own remote client using a Microsoft .NET Framework programming language"** on page 12
- **"I am using LabVIEW"** on page 12

I want to write commands for the 'Automation' tab of my automated test application

In this *Getting Started Guide*, see:

- Chapter 1:

- Programming Languages: **"ARSL"** on page 17

In the *Programming Guide*, see:

- Chapter 1:
 - Opening paragraphs
 - Remote Interface Documentation
- Chapter 3:
 - The Automated Test Remote Scripting Language (ARSL)
- Chapter 4:
 - All

I am using the Keysight KS8400A Test Automation for PathWave

In this *Getting Started Guide*, see:

- Chapter 1:
 - **"System Requirements"** on page 14
 - Programming Languages: **"TAP"** on page 17

In the *Programming Guide*, see:

- Chapter 1:
 - Opening paragraphs
 - Remote Interface Documentation
 - TAP Interface
- Chapter 3:
 - The Keysight KS8400A Test Automation for PathWave
- Chapter 4:
 - All
- Chapter 5:
 - Plugin for Keysight KS8400A Test Automation for PathWave
- Troubleshooting:
 - As needed.

I want to try a sample remote client provided by the toolkit

In this *Getting Started Guide*, see:

- Chapter 1:
 - **"System Requirements"** on page 14

In the *Programming Guide*, see:

- Chapter 1:

- Opening paragraphs
- Remote Interface Documentation
- Chapter 2:
 - All
- Troubleshooting:
 - As needed.

I want to use the Application Sequencer to execute saved projects for multiple applications

See this folder for more information:

```
C:\ProgramData\Keysight\DigitalTestApps\Remote Toolkit\Version
x\Tools\DigitalTest Application Sequencer\
```

I am using a test harness that can execute .NET Framework assemblies

In this *Getting Started Guide*, see:

- Chapter 1:
 - **"System Requirements"** on page 14
 - Programming Languages: **".NET Framework"** on page 18

In the *Programming Guide*, see:

- Chapter 1:
 - Opening paragraphs
 - Remote Interface Documentation
 - .NET Framework Interface
- Chapter 3:
 - Microsoft .NET Framework
- Chapter 4:
 - All

I am using a test harness that can execute Python

In this *Getting Started Guide*, see:

- Chapter 1:
 - **"System Requirements"** on page 14
 - Programming Languages: **"Python"** on page 19

In the *Programming Guide*, see:

- Chapter 1:
 - Opening paragraphs

- Remote Interface Documentation
- .NET Framework Interface
- Chapter 3:
 - Python
- Chapter 4:
 - All

I want to develop my own remote client using a Microsoft .NET Framework programming language

In this *Getting Started Guide*, see:

- Chapter 1:
 - **"System Requirements"** on page 14
 - Programming Languages: **".NET Framework"** on page 18

In the *Programming Guide*, see:

- Chapter 1:
 - Opening paragraphs
 - Remote Interface Documentation
 - .NET Framework Interface
- Chapter 3:
 - Microsoft .NET Framework
- Chapter 4:
 - All
- Chapter 5:
 - Simple Remote Client
 - ARSL Command Line Utility Implementation
 - Simple GUI Remote Client Implementation
 - Message Handling Remote Client Implementation
- Troubleshooting:
 - As needed.

I am using LabVIEW

In this *Getting Started Guide*, see:

- Chapter 1:
 - **"System Requirements"** on page 14
 - Programming Languages: **"LabVIEW"** on page 19

In the *Programming Guide*, see:

- Chapter 1:
 - Opening paragraphs
 - Remote Interface Documentation
 - LabVIEW Interface
- Chapter 4:
 - All
- Chapter 5:
 - LabVIEW Simple Remote Client
 - LabVIEW Demo Client Construction
- Troubleshooting:
 - As needed.

System Requirements

This section describes how to configure the computers on which the automated test application and the remote client program are running to enable remote control of the automated test application.

On the Server

For automated test applications that run directly on a Windows-based oscilloscope, "machine" below refers to the oscilloscope. For automated test apps that run on a separate computer, "machine" below means refers to that computer.

- 1 Your client computer must be able to access the machine via your network.
- 2 (Real-time 9xxx and 9xxx Series Infiniium versions 3.19 and older only, Remote Interface versions 1.90 and older only) An automated test application remote control license must be installed on the oscilloscope.

NOTE

A single application remote control license enables control of all remote-capable automated test applications installed on the oscilloscope.

To purchase an N5452A remote programming option license, please visit www.keysight.com/find/scope-apps.

- 3 A remote-capable automated test application must be installed on the machine and fully licensed on the oscilloscope.
- 4 The Windows Firewall on the machine must have these port exceptions defined (this is automatically taken care of during installation of a remote-capable automated test application):

Legacy (Agilent) apps require ports 9944 and 49944

Keysight apps require ports 49944 and 59944

Various names may exist for these exceptions (differences are unimportant).

- 5 The automated test application must be running on the machine. Launch it in one of these ways:
 - a To launch applications that run directly on the oscilloscope:
 - Use the TAP DigitalTest Apps API plugin step, LaunchAppAndConnectStep.
 - or -
 - (Infiniium real-time only) Use the Infiniium menu system: **Analyze > Automated Test Apps > (application name)**
 - or -
 - (FlexDCA only) Use the FlexDCA menu system: **Apps > Automated Test Apps > (application name)**

- or -

- Use the Keysight IO Libraries utility and the GPIB command:

```
:SYSTEM:LAUNCH "application name"
```

where application name is the name as it appears in the menu (case sensitive).

(Infiniium 3.00 or newer) If the same name appears in another Infiniium menu (for example, Demos), you must prefix the name with as many backslash-separated segments as needed to make a unique path, for example "Automated Test Apps\USB Test").

- b** To launch applications that run on a separate computer:

- Use the TAP DigitalTest Apps API plugin step, LaunchAppAndConnectStep.

- or -

- Use the Windows Start Menu.

- or -

- (Infiniium real-time only) Use the Infiniium offline menu system: **Analyze > Automated Test Apps > (application name)**

- or -

- (FlexDCA only) Use the FlexDCA menu system: **Apps > Automated Test Apps > (application name)**

On the Client

The "computer" below refers to the PC on which you are running your remote client program.

- 1** The computer must be able to access the server (defined above) via your network.
- 2** The computer must have Microsoft .NET Framework 3.5 installed.
- 3** (TAP only) The computer must have Keysight KS8400A Test Automation for PathWave version 9 or newer installed, and the DigitalTest Apps API plugin (contained in this toolkit) must be installed. This covers all of the remaining requirements below.
- 4** Your remote client must be able to access the .NET Framework assembly that defines the Automated Test Engine interface:
 - Legacy (Agilent) apps: Agilent.Infiniium.AppFW.Remote.dll
 - Keysight apps: Keysight.DigitalTestApps.Framework.Remote.dll

(Please see the readme.txt file in the "Agilent-Keysight Transition" or "Keysight Apps Only" subdirectories for more information on the Agilent-to-Keysight transition.)

1 Introduction to Programming

- 5 (LabVIEW clients only) The computer must have LabVIEW 8.5 installed.

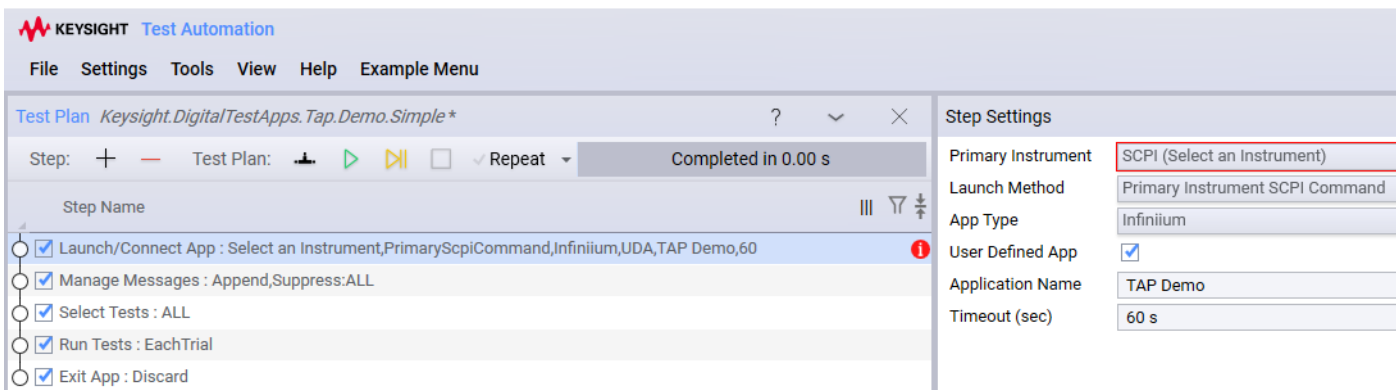
Programming Languages

You may program a Keysight DigitalTest app using different platforms and languages, according to your preferences and requirements:

- Using the Keysight KS8400A Test Automation for PathWave, or TAP.
- Using the Automated Test Engine Remote Scripting Language, or ARSL.
- Using programming languages including C# and Python.
- Using the LabVIEW graphical programming environment.

TAP

The Keysight KS8400A Test Automation for PathWave (TAP) version 9 provides powerful, flexible and extensible test sequence and test plan creation with additional capabilities that optimize your test software development and overall performance. The N5452A Remote Toolkit includes a "DigitalTest Apps API" plugin for TAP that enables you to include control of automated test apps in your TAP test plans using a set of predefined steps enabling connections, configuration, test execution and results export. Here is a sample script:



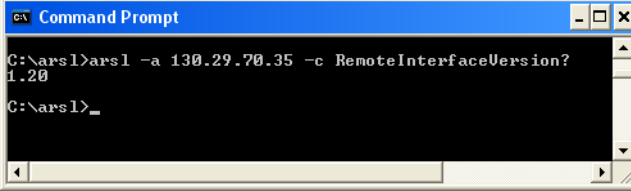
ARSL

ARSL commands are SCPI-like text that enable you to automate an application with a minimum of programming expertise. They may be sent to an automated test application using either of the following methods:

- Using the ARSL Command Line Utility.

This tool is provided in the Remote Development Toolkit in the "Tools" folder and is executed in a Command Prompt window.

Here is an example of an ARSL command being executed in the utility:



```

C:\arsl>arsl -a 130.29.70.35 -c RemoteInterfaceVersion?
1.20
C:\arsl>_

```

In the example, the utility is running on a PC, the automated test application is running on an oscilloscope at IP address 130.29.70.35, and the command being sent is a property query to discover the version of the app's remote interface (1.20).

ARSL syntax has a simple mapping to .NET Framework languages and can be used to query and set properties and execute methods, even those with custom type parameters.

You may execute files containing scripts of ARSL commands, too.

ARSL is appropriate to use only when you do not require conditional or looping algorithms or callback constructs, such as event subscriptions. If you need these additional capabilities, use .NET Framework programming.

In addition to providing the ARSL Command Line Utility itself, the Toolkit also provides the Visual Studio project and source code used to build the tool.

- Using the TAP step, ExecuteArsl.

The ExecuteArslStep enables you to send any ARSL command to a compliance app, opening up the entire remote interface for use within your TAP test plan.

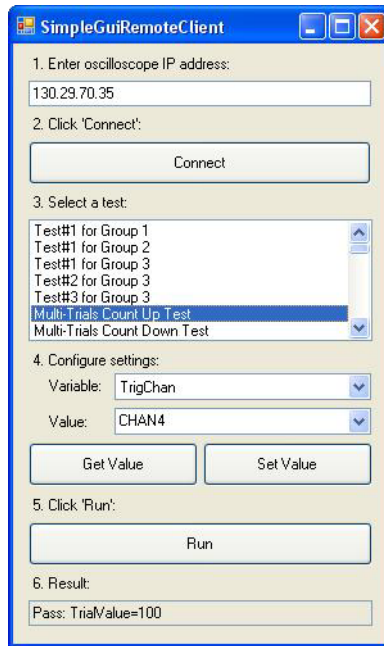
- Using the .NET Framework API method, ExecuteArsl().

ExecuteArsl() may be called by any .NET Framework program or from scripting languages that provide support for .NET Framework interfaces, such as Python.

.NET Framework

Use of a .NET Framework language enables access to the full spectrum of automated test application programming capabilities, although it does require software programming skills.

Here is an example of a C#-based remote client, the "Simple GUI Remote Client":



Python

You may use the Python programming language to automate Keysight oscilloscopes as well. By installing the Pythonnet 2.0.0 package, which is compatible with versions up to 2.7, you can develop scripts using both "method call" and SCPI-like "text based" paradigms.

LabVIEW

Remote Clients constructed using National Instrument's LabVIEW graphical programming environment are also possible. Please see the Toolkit for details.

1 Introduction to Programming

Index

Symbols

.NET Framework programming languages, 18

A

application name, 15
Application Sequencer, 11
ARSL programming language, 17
Automation tab, commands for, 9

C

client requirements, 15
copyright, 2

I

in this book, 3
introduction to programming, 7

K

Keysight KS8400A Test Automation for PathWave, using, 10
Keysight.DigitalTestApps.Framework.Remote.dll, 15

L

LabVIEW 8.5, 15
LabVIEW programming language, 19
LabVIEW, using, 12
languages, programming, 17
licensing requirements, 14

M

Microsoft .NET Framework 2.0, 15

N

notices, 2

P

programming languages, 17
programming, introduction to, 7
Python programming language, 19

R

remote client, Microsoft .NET Framework, 12
remote client, sample, 10
remote development toolkit, 9
requirements, 14
requirements, client, 15
requirements, server, 14

S

sample remote client, 10
scenarios, usage, 9
server requirements, 14
system requirements, 14

T

TAP programming language, 17
test harness, .NET Framework assemblies, 11
test harness, Python, 11
toolkit, how to use, 9

U

usage scenarios, 9

W

warranty, 2
Windows Firewall exceptions, 14

