

Wireless Wi-Fi

Dual Channel Wi-Fi OSX Station Integration and Operations Guide

WR-GL-DCW-OSX-STA-V01-190513

RELEASED

Notice

This Wi-Fi document is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. You may download, copy, distribute, and reference the documents herein only for the purpose of developing products or services in accordance with such documents, and educational use. Except as granted by CableLabs® in a separate written license agreement, no license is granted to modify the documents herein (except via the Engineering Change process), or to use, copy, modify or distribute the documents for any other purpose.

This Guide document may contain references to other documents not owned or controlled by CableLabs. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in this document. To the extent this document contains or refers to documents of third parties, you agree to abide by the terms of any licenses associated with such third-party documents, including open source licenses, if any.

© Cable Television Laboratories, Inc. 2019

DISCLAIMER

This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and CableLabs and its members shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from CableLabs, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	WR-GL-DCW-OSX-STA-V01-190513			
Document Title:	Dual Channel Wi-Fi OSX Station Integration and Operations Guide			
Revision History:	D01 – Released 01/18/19 V01 – Released 05/13/19			
Date:	May 13, 2019			
Status:	Work in Progress	Draft	Released	Closed
Distribution Restrictions:	Author Only	GL/Member	GL/Member/Vendor	Public

Trademarks:

CableLabs® is a registered trademark of Cable Television Laboratories, Inc. Other CableLabs marks are listed at <http://www.cablelabs.com/certqual/trademarks>. All other marks are the property of their respective owners.

Contents

1	SCOPE	5
1.1	Introduction and Overview	5
1.2	Purpose of Document	5
2	REFERENCES	5
2.1	Informative References.....	5
3	TERMS AND DEFINITIONS	5
4	ABBREVIATIONS AND ACRONYMS	5
5	ARCHITECTURE OVERVIEW	7
5.1	Component Responsibility Breakdown	7
6	HARDWARE COMPONENTS	7
7	SOFTWARE COMPONENTS	8
7.1	Individual Components.....	8
7.1.1	<i>LIBDCWPROTO</i>	8
7.1.2	<i>LIBDCWSOCKET</i>	8
7.1.3	<i>DCSTAD</i>	8
7.1.4	<i>DCSTAD SCRIPT</i>	8
7.2	Code Repositories.....	8
7.3	Building	9
7.3.1	<i>BUILD MACHINE PREPARATION</i>	9
7.3.2	<i>BUILDING THE STATION DAEMON</i>	9
8	USER GUIDE	9
8.1	Configuration and Setup	9
8.2	Starting the Dual Channel Wi-Fi CLI Client	10
8.3	Stopping the Dual Channel Wi-Fi Client.....	11
9	TROUBLESHOOTING	11
9.1	Validating Interface Packet Counters	11
10	KNOWN ISSUES AND LIMITATIONS	11
10.1	Third-Party Wi-Fi Hardware Support.....	11
10.2	System Preferences Locks Up on Network Pane.....	12

Figures

Figure 1	– OSX DCW Station S/W Architecture Overview.....	7
Figure 2	– System Preferences Network Pane Showing Unbonded DCW	10
Figure 3	– System Preferences Network Pane Showing Bonded DCW	11
Figure 4	– System Preferences Network Pane When Locked Up.....	12

1 SCOPE

1.1 Introduction and Overview

This document describes how to build and use the Dual Channel Wi-Fi (DCW) feature on devices running the OSX platform.

1.2 Purpose of Document

The purpose of this document is to explain the software compilation process for the Dual Channel Wi-Fi feature on the OSX platform. In addition to the integration steps, a basic user manual and troubleshooting guide are included.

2 REFERENCES

2.1 Informative References

None

3 TERMS AND DEFINITIONS

This document uses the following terms.

AirPort	Apple-branded Wi-Fi card.
data channel	A downstream-only Wi-Fi connection used in Dual Channel Wi-Fi for offloading traffic from the primary channel connection.
Linux	Open-source operating system created by Linus Torvalds.
OSX	Proprietary operating system developed and owned by Apple.
POSIX	Portable Operating System Interface—a set of APIs and command-line utilities to help standardize across different flavors of UNIX and UNIX-like systems.
primary channel	The main Wi-Fi connection used in Dual Channel Wi-Fi for both DCW signaling and upstream and downstream traffic.
Wi-Fi	A technology enabling the wireless transmission and reception of LAN traffic.

4 ABBREVIATIONS AND ACRONYMS

This document uses the following abbreviations.

AP	access point
API	application programming interface
BSD	Berkeley Software Distribution
CLI	command-line interface
DCW	Dual Channel Wi-Fi

LAN	local area network
SSID	service set identifier
USB	Universal Serial Bus
WPA	Wi-Fi Protected Access

5 ARCHITECTURE OVERVIEW

The architecture for OSX is based on the architecture of the Linux code for the station. The code for the station daemon and protocol library is fully shared between OSX and Linux. The socket library uses a different system application programming interface (API) (BSD sockets vs. Linux sockets), but the API exposed to the station daemon remains and functions the same between the two platforms.

Because of the limited amount of hardware that OSX Wi-Fi functionality natively supports, the current code only supports the data channel operating on an Apple AirPort W-Fi card and requires some manual setup using the command line.

The current code release includes a DCSTAD script that enables the Dual Channel Wi-Fi daemon to control an AirPort Wi-Fi card for the data channel interface by using the “networksetup” command (Figure 1).

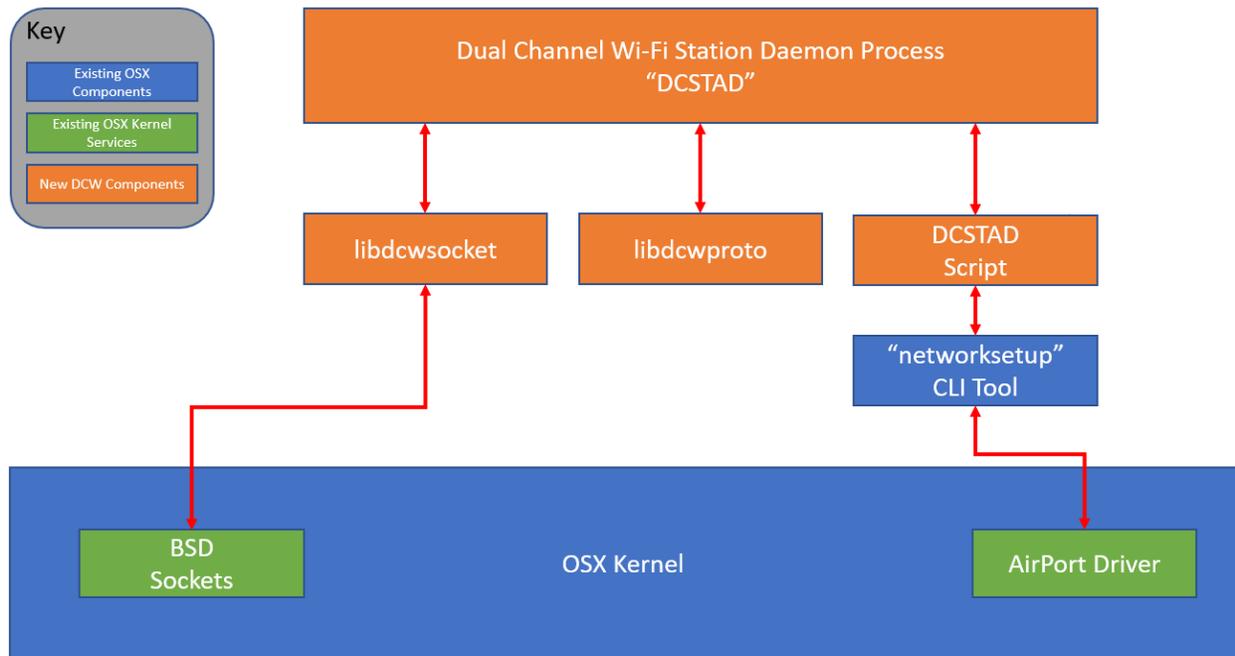


Figure 1 – OSX DCW Station S/W Architecture Overview

5.1 Component Responsibility Breakdown

The responsibility of each software component is separated as follows.

- DCSTAD—Controls all the logic and manages Dual Channel Wi-Fi. This component is not platform specific.
- DCSTAD Script—Implements hardware-specific details for setting up and managing the data channel Wi-Fi interfaces on the platform for which it resides.
- libdcwsocket—Implements platform-specific details for how to transmit and receive DCW signaling frames.
- libdcwproto—Implements DCW protocol for establishing connections with the access point (AP).

6 HARDWARE COMPONENTS

The following hardware components are required for building and running this software.

- A MacBook capable of running OSX 10.10 or higher with an internal AirPort card
- An external Wi-Fi dongle capable of functioning as the primary channel Wi-Fi connection

7 SOFTWARE COMPONENTS

7.1 Individual Components

Dual Channel Wi-Fi station functionality comprises several software components.

7.1.1 libdcwproto

The *libdcwproto* component is a platform-independent C library responsible for marshalling and serializing the Dual Channel Wi-Fi signaling messages. The library models every Dual Channel Wi-Fi signaling message as a C struct and provides conversion to/from a raw byte-buffer, ready for transmission/reception.

This component is usable for both AP and station code.

7.1.2 libdcwsocket

The *libdcwsocket* component is a Linux and OSX specific C library which simplifies transmission and reception of Ethernet frames using the CableLabs Ethertype code of 0xB4E3, and CL3 protocol type of 0x00DC. More information regarding the specific details of the protocol can be found in the protocol specification document.

This component is usable for both AP and station code.

7.1.3 DCSTAD

The *DCSTAD* daemon component is the heart of the Dual Channel Wi-Fi business-logic implementation. The component handles all network protocol signaling for Dual Channel Wi-Fi. A “DCSTAD script” is required to grant this component the ability to configure the data channel Wi-Fi interfaces.

7.1.4 DCSTAD Script

The *DCSTAD Script* acts like a hardware-abstraction layer to the “DCSTAD” component. The script is called upon by the station daemon (DCSTAD) only when there is an event. The DCSTAD component itself has no knowledge of how to set up a service set identifier (SSID) on a specific adapter, but it does know which adapter needs to be joined to which SSID. For example, when the DCSTAD component has discovered a new AP capable of Dual Channel Wi-Fi and has received the parameters for the new data channel(s), it invokes the DCSTAD script to perform the actual join (channel-bond) operation of the data channel.

The DCSTAD script is responsible for the following:

- setting up the data channel Wi-Fi adapter parameters such as SSID, WPA, etc.;
- ensuring that traffic received on the data channel is routed into the system correctly;
- reporting back to the DCSTAD process with the result of the operation—success or failure; and
- restores the interface configuration to the original state when requested.

7.1.4.1 OSX “networksetup” DCSTAD Script “dcstad-osx-connect-datachan.sh”

A DCSTAD script that uses the “networksetup” CLI utility can be found in the DCSTAD repository under the `scripts/` directory. It is a Bash script that uses the “networksetup” CLI utility to set up the data channel on AirPort devices.

7.1.4.2 DCSTAD Script API

The API for this script is maintained in the DCSTAD (<https://github.com/cablelabs/dcstad>) repository as “dcstad-script-api.txt”.

7.2 Code Repositories

All Dual Channel Wi-Fi code is stored in the CableLabs GitHub team “DCW,” located at <https://github.com/orgs/cablelabs/teams/dcw/repositories>.

Each individual software component can be found in their respective git repositories.

libdcwproto <https://github.com/cablelabs/libdcwproto>

libdcwsocket <https://github.com/cablelabs/libdcwsocket>

DCSTAD <https://github.com/cablelabs/dcstad>

An additional third-party open-source dependency for building the station daemon on OSX is also required.

OSX fmemopen() port <https://github.com/NimbusKit/MemoryMapping>

7.3 Building

These are the build instructions for compiling the station daemon process for the local host.

7.3.1 Build Machine Preparation

Xcode may need to be installed in order to obtain a working C compiler. This can be downloaded from <https://developer.apple.com/xcode/>.

7.3.2 Building the Station Daemon

This example demonstrates how to build the station daemon for the local machine, statically linking in all Dual Channel Wi-Fi dependency libraries. However, if desired, the libraries could be built and linked as shared objects.

First, clone all the code.

```
$ git clone git@github.com:cablelabs/dcstad.git
$ cd dcstad/
$ git clone https://github.com/NimbusKit/memorymapping.git
$ git clone git@github.com:cablelabs/libdcwproto.git
$ git clone git@github.com:cablelabs/libdcwsocket.git
```

Build the libdcwproto dependency as a static library.

```
$ cd libdcwproto/
$ ./configure --disable-shared --enable-static
$ make
$ cd ..
```

Build the libdcwsocket dependency as a static library.

```
$ cd libdcwsocket/
$ ./configure --disable-shared --enable-static
$ make
$ cd ..
```

Build the “dcwapd” daemon process.

```
$ CPPFLAGS="-I`pwd`/libdcwproto/include -I`pwd`/libdcwsocket/include" LDFLAGS="-L`pwd`/libdcwsocket/src/.libs -L`pwd`/libdcwproto/src/.libs" ./configure
$ make
```

8 USER GUIDE

This implementation currently supports only a single data channel because, as of publication of this document, all Apple products come with only one AirPort card.

8.1 Configuration and Setup

The recommendation is to first set up and connect to the desired AP’s primary channel SSID. The connection must use the external (non-Airport) adapter.

Because there is no standard API for working with each vendor-proprietary Wi-Fi adapter, the password for the primary channel must be manually updated in the DCSTAD script. The following line in the “dcstad-osx-connect-datachan.sh” script must be updated accordingly.

```
WIFI_PASSWORD="ABCDEFGH"
```

Change the “ABCDEFGH” string to match your configuration.

8.2 Starting the Dual Channel Wi-Fi CLI Client

Open a terminal, change to the station daemon build directory, and run the client daemon using sudo permission.

```
$ cd dcstad/
$ sudo src/dcstad -p en5 -d en0 -s scripts/dcstad-osx-connect-datachan.sh
```

IMPORTANT: This example assumes interface “en5” is the primary channel USB dongle and “en0” is the data channel internal AirPort card. Please adjust accordingly for your setup.

Figure 2 shows what the System Preferences Network pane looks like when the Dual Channel Wi-Fi client is running but the data channel is not yet bonded.

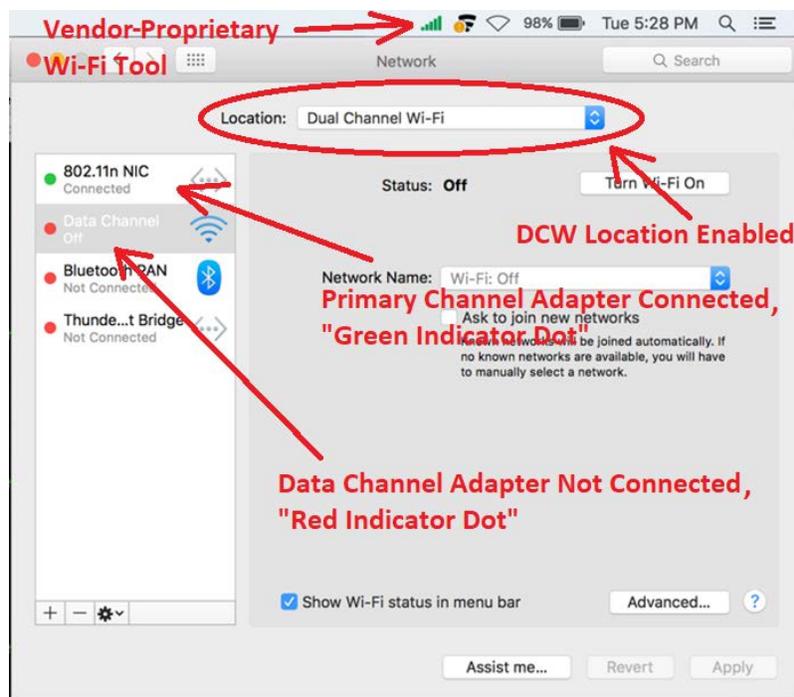


Figure 2 – System Preferences Network Pane Showing Unbonded DCW

The network location should automatically update to “Dual Channel Wi-Fi” when the station daemon is started. The location will be restored to “Automatic” when the station daemon is stopped.

Figure 3 shows what the System Preferences Network pane looks like when the Dual Channel Wi-Fi client is running and the data channel is bonded.

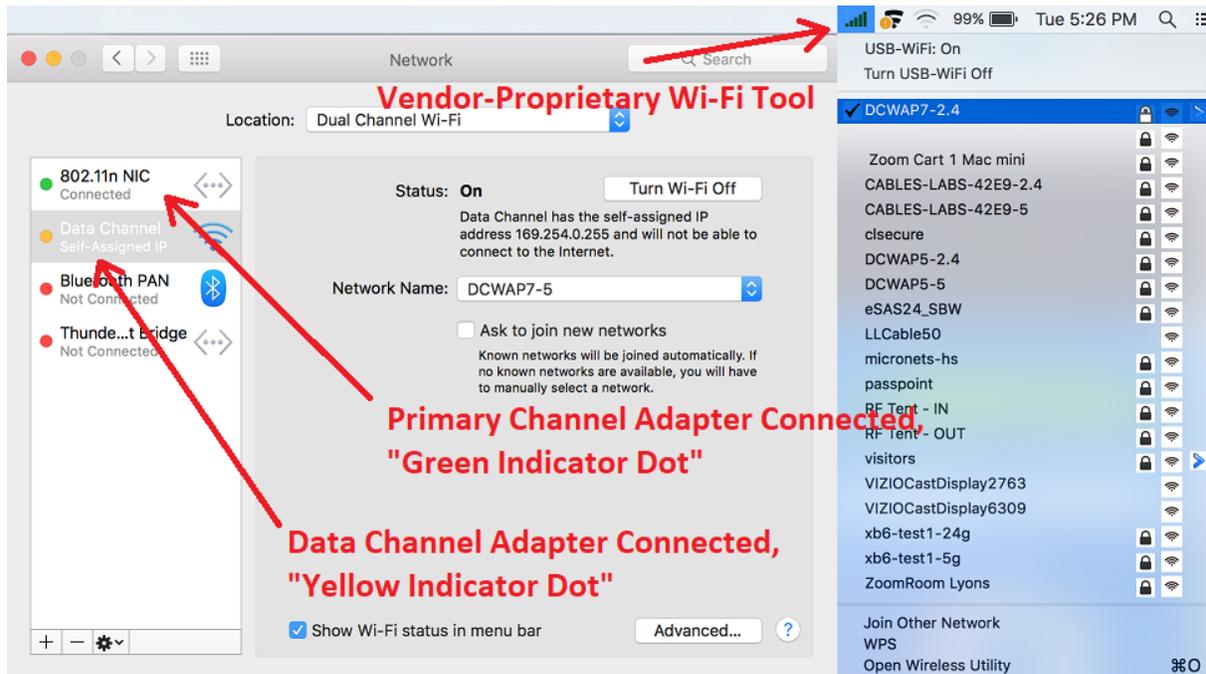


Figure 3 – System Preferences Network Pane Showing Bonded DCW

The vendor-proprietary Wi-Fi tool is used to select the primary channel Wi-Fi network. The data channel is automatically connected and disconnected by the station daemon and will show a “yellow indicator dot” when connected.

8.3 Stopping the Dual Channel Wi-Fi Client

Press Ctrl+C in the terminal windows in which the client was started to cleanly shutdown the client.

9 TROUBLESHOOTING

9.1 Validating Interface Packet Counters

The interface packet counters can be displayed by using the following command.

```
$ netstat -ian
```

10 KNOWN ISSUES AND LIMITATIONS

10.1 Third-Party Wi-Fi Hardware Support

Because OSX includes limited OS driver support for third-party Wi-Fi adapters, the Dual Channel Wi-Fi client on OSX cannot be fully integrated to support generic third-party Wi-Fi drivers. A third-party Wi-Fi adapter is required, however, because at the time of publication, Apple does not offer an external AirPort adapter. Though it may be possible to integrate the DCSTAD script with a third-party Wi-Fi driver or utility, the process would have to be repeated for every driver or utility intended for use with Dual Channel Wi-Fi.

10.2 System Preferences Locks Up on Network Pane

It has been observed that, after using Dual Channel Wi-Fi, the system preferences network pane locks up and ceases to function even after a system reboot (Figure 4).

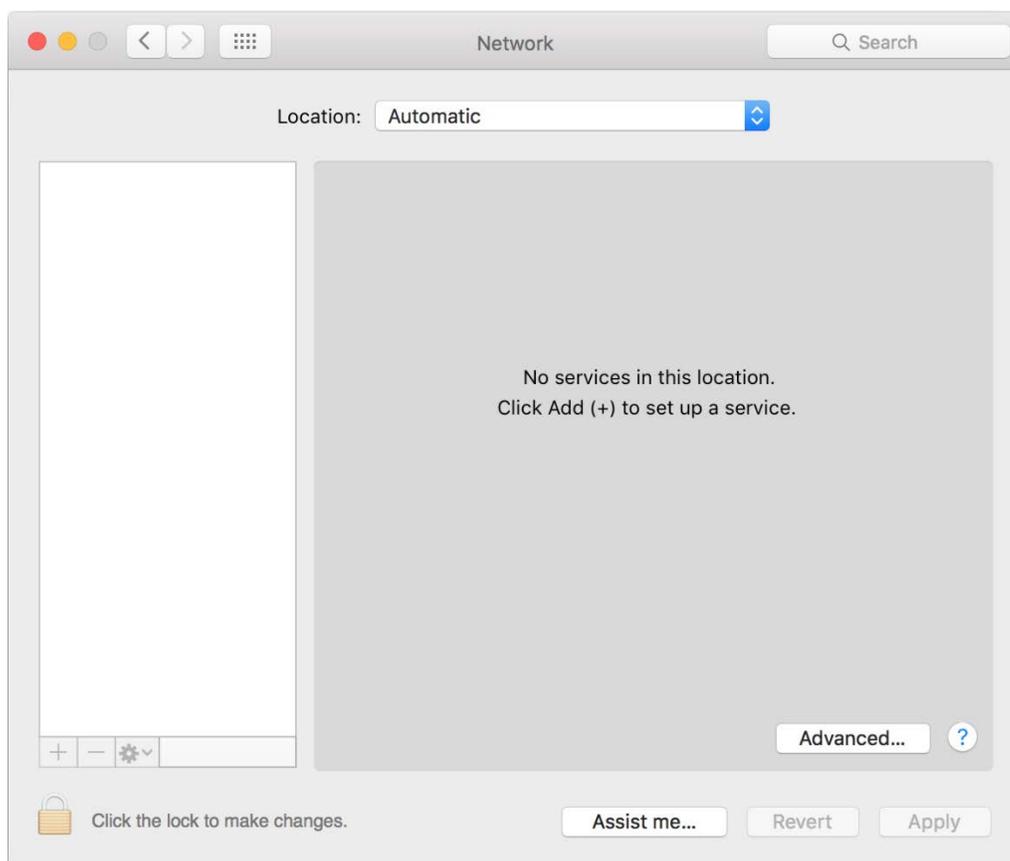


Figure 4 – System Preferences Network Pane When Locked Up

If this happens, completely remove most of the system preferences—

```
$ sudo rm -Rf ~/Library/Preferences/com.apple.systempreferences.plist  
/Library/Preferences/SystemConfiguration
```

—then reboot the system.

* * *