Technical Bulletin CS-19-05

Spectralink Versity Advanced Debugging

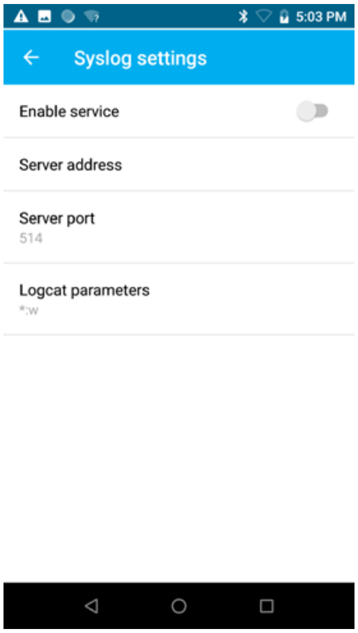
This technical bulletin explains how to use the Spectralink Versity Logging application feature, Advanced Debugging, to troubleshoot complex issues and understand how each function operates.

System Affected

All Spectralink Versity Models

Description

Spectralink Versity is a powerful device that introduces the potential for a lot of complexity when problems do arise. Fortunately, there is a built-in application that provides a way to reduce that complexity and improve your ability to troubleshoot situations that do occur. With the Spectralink Logging application that ships with the Versity handset, and is available via the Google Play Store, you can access the Advanced Debugging menu where you'll find a plethora of useful features. Just remember that you can configure most of these settings via the SAM Server, your EMM and via the Logging application screen. Be aware if you do configure the app via SAM and some EMM's the default behavior is for the application settings to be read-only on the handset. However, both SAM and an EMM can activate an option that will allow local configuration on the handset. See the *Versity Applications Administration Guide* for details.



Syslog settings

Enable service

Server address

Server port
514

Logcat parameters
*:w

Syslog Settings

When you open the Logging application you can press the three stacked dots, ⋮, in the top right corner of the screen. This will display a flyout menu where you can then select Syslog Settings. From this screen you can configure the syslog server details including the syslog server address and port along with enabling the syslog service in the phone. The final parameter is the filter used by the phone to determine what will be included in the output.

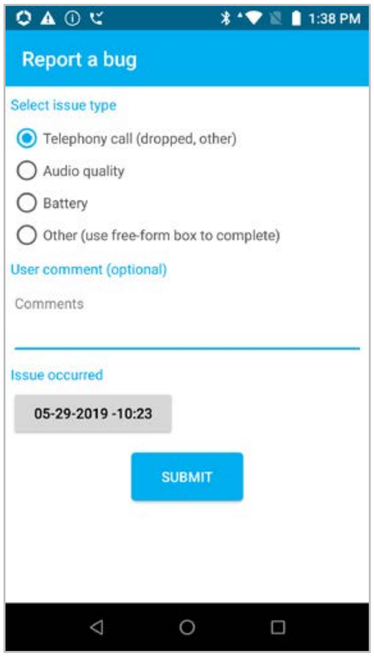
The default behavior is to only output warning level messages and above.

Tip

Syslog is a one-way protocol, so the phone has no way to know if there is a syslog server listening. If you configure syslog, the phone may continue to generate traffic even when the phone is off premises, even though the server is likely unreachable.

Report a Bug

The Report a bug option appears on the home screen. This feature allows an end user to report a failure they've experienced immediately after the failure occurs and provide feedback about the failure. The top radio buttons are for selecting the type of issue they experienced or using the Other field for free-form comments about the failure event. Use the date and time field for the event time and press the Submit button. The entered information along with a bug report will be captured and stored on the phone. We'll cover how to collect this data from the phone in a later section.



Report a bug

Select issue type

● Telephony call (dropped, other)

○ Audio quality

○ Battery

○ Other (use free-form box to complete)

User comment (optional)

Comments

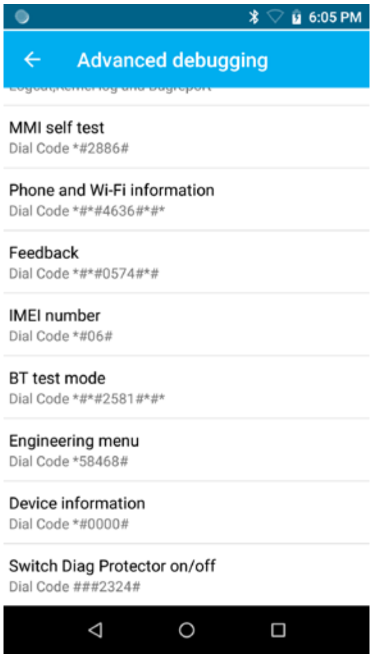
Issue occurred

05-29-2019 -10:23

SUBMIT

Advanced Debugging

From the home screen of the Logging application you can select the field that says "Password" immediately under the Advanced Debugging option. Enter the password---the default password is "admin", all lowercase without the quotes. The Advanced Debugging menu provides you with a list of functions that can be used for further debugging of the handset. We'll walk through each of these menus one at a time and explain how they can be used and what the information is best used for to help solve any problems you might encounter. We'll skip the Advanced Logging at the top for now and save that for last.



Advanced debugging

MMI self test
Dial Code *#2886#

Phone and Wi-Fi information
Dial Code *#*#4636#*#*

Feedback
Dial Code *#*#0574#*#

IMEI number
Dial Code *#06#

BT test mode
Dial Code *#*#2581#*#*

Engineering menu
Dial Code *58468#

Device information
Dial Code *#0000#

Switch Diag Protector on/off
Dial Code ###2324#

MMI Self-Test

This function is extremely helpful for testing the hardware of the handset. When you first enter the menu, you will be presented with three options at the bottom of the screen, AUTO1, MANU, and AUTO2. The AUTO1 and AUTO2 tests are primarily used for manufacturing automation tests. But the MANU option provides you with a long list of self-test options. These include the following items and a description of what each of these tests does when it is run on the handset. Each menu will present you with a

Pass/Fail at the bottom of the screen. Pressing Pass or Fail has no immediate impact on the handset so it is for your reference only.

Self-Test	*Description*
Traceability	Provides information about the handset hardware including the IMEI number if it is an LTE capable device, the BSN or serial number of the handset, Bluetooth MAC address, Wi-Fi MAC address, Reference ID and Manufacturer Part Number. When you press Pass you are taken to a second screen where you are shown a short list of metrics used for quality tracking in the factory. If you press Fail, then you are returned to the self-test menu list.
NFC	NFC allows you to test the NFC read capabilities of the handset. You need to present the handset with an NFC readable item, such as an NFC card, credit card, hotel room key, retail tracking chip, etc. Place the NFC item behind the phone near the bottom of the battery and the handset will play a tone to indicate it has read the tag and will display information about the tag. If you fail to present a tag within 30 seconds the test will timeout automatically.
USB OTG	If you have a USB On-the-Go device, you can connect to the handset you can test the handset's ability to function with an OTG device. OTG allows you to attaching things like USB flash drives to the handset.
USB 3.0	This allows you to verify that the USB port on the bottom of the phone is operating correctly. You can connect a cable between your handset and the PC and confirm the status displayed on the screen.
Charger Test	When the handset attached to a charging source, either a wall wart, desktop charger or even a PC that is providing power, the handset will display whether the connection is working correctly and what the current in milliamps is flowing into the handset.
Fingerprint	You will be prompted to play your finger onto the fingerprint scanner on the back of the handset. Doing so will automatically cause the phone to complete the self-test. If you fail to place your finger on the sensor or the sensor has failed, then the test will timeout automatically and display a failure.
TPRaw Data	This will calculate the raw data input from the touch panel of the phone and confirm that it is operating. Output will happen automatically and report whether it passes or fails.
Touch Panel	You will be prompted to draw inside of yellow boxes on the screen to confirm that you are able to affect the screen properly. You'll be prompted if the line you draw is too short or another error occurs.
Pen Touch	Like the prior test, you can draw on the entire screen in this test as much as you desired. When you stop you are prompted whether the test passes or fails.
TP Lock	Here you are testing the locking function of the touch screen. You will be prompted to press the power button to lock the screen at which point you'll need to press it again and enter your PIN to unlock the handset.
LCD	For this test a set of three test stripes will display on the screen, red on the top, green in the middle and blue on the bottom. If you see these colors on the screen the first test has passed. If you press Pass you will then see a

Self-Test	*Description*
	gradient screen of grayscale colors. Pressing Pass again displays an empty black screen. Press Pass again to display a light grey screen. Pass again to display a slightly darker grey screen and which point pressing Pass one more time ends the test.
LCD Backlight	You are prompted to confirm whether the backlight of the handset is flashing, in other words, the screen going off and coming back on.
Key	You will be prompted to press each of the hard buttons on the handset, except the Power key. After pressing each key as directed the handset will return to the self-test menu list.
Front Camera	When activated this test will activate the front camera of the handset so you can verify that the camera is operational.
Main Camera	This activates the rear camera as in the previous test so you can verify the rear camera is operational.
Camera LED	When started the LED flash on the back of the handset should turn on and remain on until you select either Pass or Fail.
Charge LED	The LED on the front, top. Left of the handset will activate and cycle through green, red and blue to indicate each of the different states.
Audio	Plays a pre-recorded message first through the earpiece speaker and if you select Pass it will then switch to the rear speaker. Selecting Pass then causes the microphone on the front, bottom, center of the handset to be activated and looped to the earpiece speaker to test the microphone. Pass again will loop the secondary microphone to earpiece speaker. Pressing Pass will then loop the microphone on top of the phone to the earpiece speaker. Pass one more time loops the microphone on the rear of the handset to the rear speaker. After performing any audio tests on the handset, you should always reboot it as testing can leave the handset in a state where it is no longer optimized for voice. The reboot will restore the optimization.
Vibrator	Activating this test immediately turns on the vibrator in the handset.
Accessory	You are prompted to connect a headset to headset jack at which point you are prompted to confirm a discrete test in the left ear of the headset followed by a test in the right ear and then the microphone.
E-Compass	This test validates that the internal accelerometers are operating correctly. You will see the Angle displayed at the top of the screen to change as you move and shift the handset.
Gyroscope	Like the previous test, this reports the X, Y and Z position of the handset as you move it around.
G-Sensor	For this test you are presented with a series of tasks to conduct on the handset to validate the handset can experience rapid changes in acceleration and deceleration.
Light Sensor	The current ambient light value is displayed on the screen. As the ambient light changes the value displayed on the screen will change automatically. The light sensor is in the top, right corner on the front of the handset.

Self-Test	*Description*
Proximity Sensor	This is the same sensor as the light sensor, but the screen will display the results as you cover the proximity sensor and uncover it.
Gesture Test	Gestures are a standard feature for things like zooming and scrolling. So, you can perform these types of tests on the screen to confirm that they are recognized. This test is like prior touch tests, but it will display the results as arrows on the screen.
Battery	A very useful test that reports the current battery details for temperature, current charge in percentage and the voltage in millivolts.
Backup Battery	Versity has the distinction of having a truly hot swappable battery due to the presence of this backup battery. This test allows you to see the same types of information as the prior test but about the backup battery.
Bluetooth	Activating the Bluetooth test shows a list of active Bluetooth devices that can be seen by the handset and the current signal strength of each of those devices.
WiFi	Wi-Fi displays the SSID’s the handset sees along with the signal strength and channel in MHz This displays only 2.4GHz values
WiFI5G	Just the same as the prior test, this test displays the same information but for the 5GHz frequency.
2D Scanner	This activates the 2D Scanner on handsets that have a barcode scanner. You will see what the barcode scanner sees on the screen to be able to confirm that the scanner is functioning.
HAC	This is a method for validating Hearing Aid Compatibility is operating correctly in the handset.
TOF Test	This test requires a manufacturing fixture to complete – reserved for internal Spectralink engineering and operations only.
EFuse Check	Here we are just checking whether the handset has been fused for production usage. An unfused device can run special engineering only software so a customer device should always pass the EFuse Check
Factory Reset	Not really a test per se but rather an actual factory reset method. Using this test will restore your handset to factory defaults!
Aging Mode	This is a stress test for the handset that will run multiple tests at a time to cause the handset to fail if it only fails under heavy load.
Top Mic to Receiver Loop	A subset of the previous audio tests, this audio test allows you to skip directly to the audio test where you can loop the microphone on top of the handset to the earpiece speaker for testing. After performing any audio tests on the handset, you should always reboot it as testing can leave the handset in a state where it is no longer optimized for voice. The reboot will restore the optimization.
Back Mic to Receiver Loop	A subset of the previous audio tests, this audio test allows you to skip directly to the audio test where you can loop the microphone on the back of the handset to the earpiece speaker for testing. After performing any audio tests on the handset, you should always reboot it as testing can leave the handset in a state where it is no longer optimized for voice. The reboot will restore the optimization.

Self-Test	*Description*
Switch Sub Mic	A subset of the previous audio tests, this audio test allows you to skip directly to the audio test where you can loop the two front microphones to the earpiece speaker based on which one you want to test. After performing any audio tests on the handset, you should always reboot it as testing can leave the handset in a state where it is no longer optimized for voice. The reboot will restore the optimization.

Phone and Wi-Fi Information

From this menu you can access some very detailed information about the handset and is current operation. This includes both LTE and Wi-Fi information and settings. When you first enter this feature, you are presented with Phone Info where you can look at the details on the LTE side of a handset. If your handset is LTE enabled you will be able to view your IMEI number, phone number, current network and so much more. There are also several tests available for validating LTE data functions are operating properly as well.

You next have Usage Statistics which is a list of applications that have been run on the handset, the last time they were run and for how long each of them ran.

The last menu is for Wi-Fi Information that provides you a view into the handset’s Wi-Fi Configuration and Wi-Fi Status. There’s a great deal of information available here particularly when the handset is connected to a Wi-Fi network. When it is disconnected the available information is reasonably limited. Within the Wi-Fi Status you can perform a ping test the validate IP connectivity by attempting to ping www.google.com.

IMEI Number

If you have an LTE capable handset you can use this menu option to display the IMEI number of your SIM card. A pop up will display on the screen with this information which you can then clear when you’re ready by tapping “OK”.

Bluetooth Test Mode

Bluetooth test mode allows you to perform more active tests of the Bluetooth radio in the handset. You can toggle Bluetooth on and off and select specific channels for the Bluetooth radio to operate on. This is most likely to only be used by developers.

Engineering Menu

From this menu it allows for special engineering testing of the LTE radio in both LTE and CDMA modes. These tests are selected based on the type of test to be performed and then you are prompted to reboot the handset to apply the changes. You are prompted for this regardless of whether there is an LTE radio present or not.

Device Info

This is an information only menu and provides details about the type of handset, vintage, radios, software versions and so on. If your phone has an LTE radio, then there is more information present in this screen than for those handsets without an LTE radio.

Switch Diag Protector

The last item on the list is a toggle only that enables and disables a Quallcomm diagnostic port that is used for manufacturing testing only.

Advanced Logging

We'll now go back to the first item in the menu which is for Advanced Logging. This is likely the most important tool in the application toolbox. From here you can capture bugreports, logcat, QXDM and even network captures. And best of all is the ability to leverage the upload server function for remote operation so you don't even have to touch a handset to get data from it.

There is no perceivable impact to audio while logging. For example, the uploads to the server are a lower priority traffic than WiFi. The files are also heavily compressed and uploading them as files is more efficient than syslog that send one uncompressed line at a time. Our own test users have continuous logging enabled with no observable performance impact. The main reason we don't recommend continuous logging on all phones in a live deployment is because of the amount of data the server needs to store. But if you have a reasonably large amount of disk space available on your upload server and a limited number of phones you can determine whether you are comfortable doing some logging such as logcat or network capture more continuously.

Before we get into the specifics on each item, we'll cover the basics of how this Advanced Logging works. Each of the four logging functions have a reserved space on the handset's SDCard of 512MB. There is a maximum total reserved space for all logging of 4GB. As a file is captured is stored on the SDCard and will be in the */sdcard/LoggingAppDebugData/* folder. We'll specify the exact path for each logging type with the description. Once a file is closed it will automatically be archived, put into a ZIP file and moved into the *ToBeUploaded* folder for the specific log type.

If you've configured an upload server then 5 minutes after the logs have been archived, they will be uploaded to the log server. Once they've been uploaded the files will be archived automatically and deleted only when the 512MB of space for that logging type has been completely consumed to free up space for a new file. The 512MB of space is shared between the Archives location and the ToBeUploaded location. The files will remain on the phone if there is no upload server defined. As logs reach their maximum sizes the oldest logs that haven't been uploaded will be deleted to make room for the new logs.

Logs can be password protected as well to ensure any data collected is safe from prying eyes. The password used is the same password used to access the Advanced Debugging menu so you may want to change it from the default of “admin” if you plan to use the password protection. The password is used for file encryption so you will be asked for the password by your chosen archive tool used to decompress the files.

You’ve also got the option to use MTP, which is just a local download option. This allows you to connect the handset to your PC via USB and to navigate the handset and transfer the files to your PC.

Tip

When using MTP, make sure USB is configured for "USB for file transfer", by connecting a USB cable to your PC, and pulling down the notifications at the top of the screen, and finding the notification "USB charging this device, TAP to change", tapping it, and selecting "Transfer Files". Then from the Logging application under the Advanced Debugging -> Advanced Logging, press the ⋮ icon and choose MTP Refresh. This will ensure what is displayed to your PC includes the files that were generated by the Logging application.

As logs are captured each logging type will have a unique name convention that can be used to help you identify when the log was created and for which device. For each logging type the specific file naming conventions will be defined to help you understand what to expect and what to look for. Don’t forget to use the file name and not the date and time of the file on the upload server. The date and time on the server are just when the file was uploaded, not when it was created by the handset.

Note

It’s important to note that in order to leverage the remote aspect of the Advanced Logging features you ***must*** have a HTTP server. That HTTP server ***must*** also have PHP installed and running as a service.

Capture Bugreport

Bugreports are an Android feature that allows the device to gather a great deal of data together into a single file for review. Bugreports contain logcat data, processor details, memory usage and much more. Bugreports are a snapshot in time since the device’s start point. So, if you had powered your handset on 6 hours ago the bugreport would contain nearly all the data since that point in time up to the moment the bugreport was triggered. Things like battery statistics are typically stored since boot up but some other things like logs will likely have been overwritten so there should be some sense of urgency to gather a bugreport when troubleshooting an issue. It’s important to also note that generating bugreports via other methods, such as via Developer Options menus does not place the bugreport in the same location as the Logging application and those bugreports will not get uploaded to the Logging uploads server. For the sake of simplicity, you should just use the “Report a Bug” option from the main screen of the Logging application since this gives the ability to enter more information about the event being captured.

Bugreports are stored in until upload /sdcard/LoggerAppDebugData/Bugreport/ToBeUploaded

Bugreports once uploaded /sdcard/LoggerAppDebugData/Bugreport/Archives

File name convention: mac_timestamp_command_platform_version_application version
Example: 00907aa7db2a_20181024_1227_bugreport_1.1.0.1785_4.0.5588.txt

Logcat

Logcat is the Android logging application where all system logging is stored. This includes kernel information, application logs and process logs. Logcat generated through this method is automatically captured using a verbose logging filter that disables the Chatty function. Chatty is a Google feature that suppresses logging sources that are too “chatty” or talkative to help reduce the amount of potential garbage that shows up in the logs. This is problematic when you’re trying to debug something as quite often the logging has a significant amount of output beyond what it might normally have. So, with Chatty disabled we can ensure that everything is captured. The default log buffer size is 1024 Kbytes, but you can change this to whatever size you want. Just remember than the maximum amount of data that can be capture is 512MB. The larger the buffer size you specify the larger the log file will be that is created. This can mean that if you want fewer files you can specify a larger buffer.

Logcat data is stored in /sdcard/LoggerAppDebugData/Logcat/ToBeUploaded

Logcat data once uploaded in /sdcard/LoggerAppDebugData/Logcat/Archives

File name convention: mac_timestamp_command_platform_version_application version
Example: 00907aa7db2a_20181024_1227_logcat_-v_threadtime_-b_all_1.1.0.1785_4.0.5588.txt

Network Capture

Likely the most powerful tool available to you here, this feature allows you to capture all network traffic coming and going through the handset’s radio. While not an actual wireless packet capture, it is a layer 3 capture of traffic to and from the handset. The best part of this is that traffic has already passed through the radio buffer so it has been decrypted so even if you’re using WPA2-Enterprise security you can see the contents of the packets. However, if you are using secure voice communication, e.g. SRTP, then the RTP frames would still be captured encrypted as the decryption of those frames does not happen in the radio but in the telephony stack. The only other thing you would be missing are the management frames. Another thing to be aware of is that Network Capture has a function called Snapshot Length. This limits the size of the frames captured if you want to prevent the amount of data being grabbed. If you want to capture the entire frame all the time then you would need to set the Snapshot Length to 0, the default is 114. Packet captures can be opened in your preferred capture viewer.

Another important note is that the network capture feature intentionally will not capture traffic for your Software Update (Sys Updater) server, Syslog server or your Advanced Debugging Upload server. This is completely on purpose as the traffic for these servers tends to be very large and would fill up the capture buffers very quickly. Additionally, there is an expectation that your SIP server and other application servers would reside on independent systems from these three server platforms. You could realistically use a single server for all three of these services, but regardless, just be aware that the Network Capture feature will not capture any data from these sources once they are configured in the handset.

Network Capture data is stored in /sdcard/LoggerAppDebugData/Slnkdump/ToBeUploaded

Network Capture data once uploaded /sdcard/LoggerAppDebugData/Slnkdump/Archives

File name convention: mac_timestamp_command_platform_version_application version
Example: 00907aa7db2a_20181024_1227_tcpdump_-i_wlan0_1.1.0.1785_4.0.5588.zip

QXDM

The QXDM (QUALCOMM eXtensible Diagnostic Monitor) is a tool primarily used by developers and manufacturers to troubleshoot issues with the Quallcomm reference design chipset. The tool generates a large amount of data and can be targeted to specific areas of the chipset. It is necessary to have a licensed QXDM application installed in order to open and interpret the files generated. The following are the list of logging types that can be collected and what each of them is focused on. QXDM logging should not be enabled or left running unless specifically directed by Spectralink Support personnel. QXDM generates a great deal of data and load on the system if left running and could impact system performance over time.

1. Audio_network.cfg – Provides detailed information on the audio performance and network operation of the handset for both Wi-Fi and LTE radios.
2. BT_WLAN_FM.cfg – Used for any Bluetooth specific issues.
3. Driver_Sensors.cfg – Allows for collection of data related to the accelerometer and other movement sensors in the handset.
4. GNSS_New.cfg – Any issues around GPS operation.
5. Network_related.cfg – Used for issues related to Wi-Fi throughput or inability to connect to a network.
6. NW_UIM_DS_WMS.cfg – Used for cellular/LTE related issues.
7. Protocol_audio_vocoder.cfg – Allows for collection of data related to audio performance issues.
8. Roaming_wlan.cfg – Can be used for capturing information about WLAN roaming performance issues.
9. Custom.cfg – Allows you to specify the exact filter to be used. This can be very complex and difficult to use without completely understanding what is necessary to enter and how to enter it. You should only use this if specifically told to by a Spectralink Support of Engineering member.

QXDM data is stored in /sdcard/LoggerAppDebugData/Qxdm/ToBeUploaded

QXDM data once uploaded /sdcard/LoggerAppDebugData/Qxdm/Archives

File name convention: mac_timestamp_command_platform_version_application version
Example: 00907aa7db2a_20181024_1227_qxdm_1.1.0.1785_4.0.5588.zip

Upload Server

The upload server used by the Advanced Logging feature requires some special setup and the use of PHP scripting to help control file size limits both on the handset and on the upload server. Spectralink recommends the Aprelium Abyss server for both Windows and Linux for small test environments or proof of concept deployments or whichever web server you prefer that will support PHP. Please note that we do not recommend using Abyss as your full-time solution as your Upload Server. Please consider using Microsoft IIS or Apache as they are designed to be enterprise web servers.

1. Follow the instructions for getting started https://aprelium.com/abyssws/start.html
   a. Take note of the "port" abyss set your server up on; HTTP "should" be 80 and HTTPS "should" be 443, but it can be 8000 and 4430. Take note as it must match in device settings.
2. Follow the instructions for adding PHP support https://aprelium.com/abyssws/php.html for your platform.
   a. (on Windows we recommend following the "important note" for php7)
3. Ensure that the default temp directory has read/write permissions
   a. (sudo chmod 1777 /tmp on Linux, right click /windows/temp for Windows, select properties, and deselect the read-only box and apply)
4. Create a folder named "uploads" in the htdocs folder of your Abyss installation and ensure that the folder has read/write permissions.
5. Edit php.ini (sudo gedit /etc/php/7.0/cgi/php.ini for Linux; C:/Program Files/PHP7/php.ini for Windows) and find and set the following variables:

   upload_max_filesize=10M;

   post_max_size=10M

   max_input_time=300

   max_execution_time=300
6. Copy and paste the PHP script below to a file named "fileUpload.php" in the htdocs folder of your Abyss installation.
   a. On Linux machine open a text editor and paste the script in, and save the file in your htdocs folder as "fileUpload.php" (note the uppercase U),
   b. On Windows using a text editor, copy and paste the script and save the file into your htdocs folder

Note

Debug file upload will work over VPN so remote workers can still log to their main company server over Wi-Fi or LTE data connections.

Upload Server - Minimum Server Requirements

Storage

The upload server will be responsible for handling a significant amount of data from your handsets when they begin collecting data; whether it be from logcat, TCPDUMP, or bugreport, the amount of data uploaded, while compressed, will require a significant amount of space. Spectralink estimates that on average a handset with normal logging enabled will create approximately 1 gigabyte of data per month if a handset is used 24/7. We do not recommend running Advanced Logging on all the phone at your site all the time unless it is necessary.

The storage requirements on your upload server will need to be adjusted according to the number of handsets you have in your environment. If you have 100 handsets in your environment, then you would need to ensure you had 100 gigabytes of space available on your upload server at a minimum.

Memory

Memory usage will be highly dependent on whether other applications are running on the server. If this server is dedicated to being used as an upload server for Spectralink Versity handsets, then the minimum recommended memory is 12GB.

Processor

Processor requirements are based heavily on the number of handsets connecting to the upload server. It's estimated that each handset connection, when uploading generated content, will create a 7.023% processor impact over a 1 second period. The average transmission period for a single file upload of 145kb is 0.1 seconds.

For a handset that had been collecting and storing files, because it didn't have access to the upload server, there might be 21 files at an average size of 100kb. With the average upload rate of 145kb in 0.1 seconds we can estimate that all 21 files should be able to be uploaded in 1.4 seconds with an impact to the processor of just over 7% for that single session.

It's important to note that your mileage will vary. There are a lot of variable here that will impact performance and how your server's processor is impacted. System overhead is the biggest impact but there are network overhead challenges and the possibility of network transmission delays that could result in reduced transmission speeds. Anytime the transmission speeds are reduced, the amount of time needed to complete the upload will increase. This all means a much greater impact to the server's processor. Since we do not recommend that all phones be setup to run with Advanced Debugging you are unlikely to encounter issues with processor overload.

Note

All examples and times provided are based on a single processor dual core system operating with a gigabit ethernet connection on an internal network. All traffic is routed through one test environment into another to simulate a device existing in a voice environment with a server being hosted in a data center or other centralized location.

A typical web server should be run with a minimum of a single quad core processor. Given today's virtualization options we would highly recommend choosing a two to four processors for your upload server.

Setting up PHP

Since configuring PHP is a bit of mystery to many people and the content on the web is mixed at best, we thought it would be helpful to provide some tips and pointers on your PHP setup here. Please note that what we're providing here are just references and guidance. You ***must*** seek specific details on installation of PHP for your environment. We can only provide information based on what we've tested.

Windows IIS

Windows IIS is a beast simply because there are so many variations depending on which version of it you are running. We've done basic testing on version 6 through version 10 and had decent success. There are a few different configuration options from version to version because of what new features are added as you go from version 6 to 10. But for the most part the installation of PHP on IIS was similar enough to allow you to follow similar instructions. You may need to seek out some specific information based on your version if you do run into problems.

You'll obviously need to download PHP too and one of the easiest ways is to use the Windows Installer tool which will download and install it directly into the correct place on your server to make a lot of this easier. You can obtain that installer here:

https://www.microsoft.com/web/downloads/platform.aspx

Running the installer is simple and will walk you through the steps necessary to get everything into the right places on your server. Once you've installed the platform installer you will have an icon, Web Platform Installer, in your IIS system that you can double-click to launch the installer system.

When the window opens, click on the Products tab and then select the Frameworks section on the left side of the page. Then scroll down through the list until you find “PHP 7.3.7 (x64)” and click the Add button the right side. Then click the Install button and allow the installer to complete the installation of the software. You’ll find the PHP installation under C:\Program Files\PHP\v7.3.7 which is where you will then make any changes to your php.ini configuration file.

This link references how to configure PHP on IIS in reference to Windows 10 but all the information is transferable to all other versions of Windows Server too.
https://jamesmccaffrey.wordpress.com/2017/01/26/installing-php-on-windows-10-and-iis/

Here are some of the key configuration items from that site that you need to pay attention to when configuring your php.ini file:

```
error_log="C:\Windows\temp\PHP737_errors.log"
upload_tmp_dir="C:\Windows\temp"
session.save_path="C:\Windows\temp"
cgi.force_redirect=0
cgi.fix_pathinfo=1
fastcgi.impersonate=1
cgi.fix_pathinfo=0
cgi.force_redirect=0
fastcgi.logging=0
max_execution_time=300
date.timezone=America/Denver
extension_dir="C:\Program Files\PHP\v7.3.7\ext"
open_basedir="versity" <-- This will be the folder where your handsets are
uploading their logging files. So, enter the appropriate path which may
just be a folder name, e.g. “versity” or “uploads”.
```

Admin Tip

When configuring the folder on your Upload Server that will be the location where files are placed by the handsets you must define a base directory but then also include a sub-directory in that base directory called “uploads” in order for the handsets to actually perform their file uploads. For example, if your web server site specifies that the base directory for the Upload Server is C:\inetpub\wwwroot\versity then you need to include a sub-directory in that location called “uploads” as well.

Tip

Make sure that you've configured your website where the Versity handsets will be uploading their Advanced Debugging files to have the proper permissions to allow anonymous uploads. This typically requires adding "Everyone" as a permitted security group to the folder location and allowing read/write permissions. There is no method available for the Versity handsets to authenticate to the server rather than use anonymous authentication.

Linux/Apache

The process for installing a web server and PHP on Linux will depend a great deal on the Linux distribution you are running and the version of that distribution as well. However, the overall process is similar. You will need to consider the requirements noted above for server sizing just as you would for a Windows server, but with Linux there is a bit more flexibility in how you allocate your resources.

For the purposes of this document we will assume that you've already installed your desired Linux distribution and desired web server platform. The most common we encounter is Apache/Tomcat but if you have another platform you would like to use then feel free to do so. All testing Spectralink has conducted has been on Ubuntu using Apache Tomcat8.

Installing the necessary packages is typically straightforward but does require that the server have internet access to perform the installation process. PHP 7.3 is the latest stable release of PHP. Perform the steps below to install PHP 7.3 on Ubuntu 18.04.

Start by enabling the Ondrej PHP repository:

```
sudo apt install software-properties-common
sudo add-apt-repository ppa:ondrej/php
```

Install PHP 7.3 and some of the most common PHP modules:

```
sudo apt install php7.3 php7.3-common php7.3-opcache php7.3-cli php7.3-gd
php7.3-curl php7.3-mysql
```

To verify the installation, run the following command which will print the PHP version:

```
php -v
PHP 7.3.1-1+ubuntu18.04.1+deb.sury.org+1 (cli) (built: Jan 13 2019
10:19:33) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.1, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.3.1-1+ubuntu18.04.1+deb.sury.org+1, Copyright (c)
1999-2018, by Zend Technologies
```

Once everything is setup you can move onto the last step of adding the PHP script to the server.

Admin Tip

When configuring the folder on your Upload Server that will be the location where files are placed by the handsets you must define a base directory but then also include a sub-directory in that base directory called “**uploads**” in order for the handsets to actually perform their file uploads. For example, if your web server site specifies that the base directory for the Upload Server is *C:\inetpub\wwwroot\versity* then you need to include a sub-directory in that location called “**uploads**” as well.

PHP Script – Upload Server

Note

The following PHP script is not optional and ***must*** be installed on your upload server in the root directory. You cannot place it anywhere else on the server as the phone is only going to look for it in the root directory. Without the fileUpload.php script placed in the root directory of your HTTP server you will not see any logging files appear on the upload server at all.

You can download a copy of the fileUpload.php script from the Spectralink Support Portal by using the following link: https://support.spectralink.com/system/tdf/resource_files/Versity_Advanced_Debugging_PHP_Script.zip?file=1&type=node&id=13584

```
----------Start Copy Immediately Below-----------
<?php
$form = "<form action='fileUpload.php' method='post'
enctype='multipart/form-data'>
    <input type='file' name='fileToUpload' id='fileToUpload'
</form>";
function _GetMaxAllowedUploadSize()
{
    $Sizes = array();
    $Sizes[] = ini_get('upload_max_filesize');
    $Sizes[] = ini_get('post_max_size');

    for($x=0;$x<count($Sizes);$x++){
        $Last = strtolower($Sizes[$x][strlen($Sizes[$x])-1]);
        if($Last == 'k'){
      $Sizes[$x] = str_replace('K','',$Sizes[$x]);
      $Sizes[$x] = str_replace('k','',$Sizes[$x]);
            $Sizes[$x] *= 1024;
        } elseif($Last == 'm'){
      $Sizes[$x] = str_replace('M','',$Sizes[$x]);
      $Sizes[$x] = str_replace('m','',$Sizes[$x]);
      $Sizes[$x] *= 1024;
      $Sizes[$x] *= 1024;
        } elseif($Last == 'g'){
      $Sizes[$x] = str_replace('G','',$Sizes[$x]);
      $Sizes[$x] = str_replace('g','',$Sizes[$x]);
            $Sizes[$x] *= 1024;
            $Sizes[$x] *= 1024;
            $Sizes[$x] *= 1024;
        } elseif($Last == 't'){
      $Sizes[$x] = str_replace('T','',$Sizes[$x]);
      $Sizes[$x] = str_replace('t','',$Sizes[$x]);
            $Sizes[$x] *= 1024;
            $Sizes[$x] *= 1024;
            $Sizes[$x] *= 1024;
            $Sizes[$x] *= 1024;
        }
    }
    return min($Sizes);
}

function delete_older_than($dir, $max_age) {
  $list = array();

  $limit = time() - $max_age;

  $dir = realpath($dir);

  if (!is_dir($dir)) {
    return;
  }

  $dh = opendir($dir);
  if ($dh === false) {
    return;
  }

  while (($file = readdir($dh)) !== false) {
    $file = $dir . '/' . $file;
    if (!is_file($file)) {
      continue;
    }

    if (filemtime($file) < $limit) {
      $list[] = $file;
      unlink($file);
    }

  }
  closedir($dh);
  return $list;
}

function makeDir($path)
{
     return is_dir($path) || mkdir($path,0755);
}
```

```
# Set to false for a flat uploads/ directory structure
# Set to true for uploads/mac_address directory structure
$mac_dir = false;
$target_root = "uploads/";
$baseFileName = basename($_FILES["fileToUpload"]["name"]);

if($mac_dir === true)
{
     $macIndex = strripos($baseFileName, "00907a");
     if ($macIndex === false)
     {
  $target_dir = $target_root;
     }
     else
     {
       $macAddress = substr($baseFileName,$macIndex,12);
       makeDir($target_root . $macAddress);
       $target_dir = $target_root . $macAddress . '/';
     }
}
else
{
     $target_dir = $target_root;
}

$target_file = $target_dir . $baseFileName;
echo nl2br("file is " . $target_file . "\n");
$uploadOk = 1;

$fileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
echo nl2br("file type is " . $fileType . "\n");

#echo ini_get('upload_max_filesize');
#echo ini_get('post_max_size');
#echo ini_get('memory_limit');
#echo _GetMaxAllowedUploadSize();

// Check if file already exists
if (file_exists($target_file))
{
    echo nl2br("Sorry, file already exists.\n");
    $uploadOk = 0;
}

// Check file size
if ($_FILES["fileToUpload"]["size"] > _GetMaxAllowedUploadSize())
{
    echo nl2br("Sorry, your file is too large.\n");
    $uploadOk = 0;
}

// Allow certain file formats
if($fileType != "zip")
{
    echo nl2br("Sorry, only ZIP files are allowed.\n");
    $uploadOk = 0;
}

// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0)
{
    echo nl2br("Sorry, your file was not uploaded.\n");
}
else
{
// if everything is ok, try to upload file
    if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"],
$target_file))
    {
        echo nl2br("The file ". basename( $_FILES["fileToUpload"]["name"]).
" has been uploaded.\n");
    }
    else
    {
        echo nl2br("Sorry, there was an error uploading your file.\n");
    }
}

// Added for aging out old files
// Delete backups older than 30 days
$deleted = delete_older_than($target_dir, 3600*24*30);
$txt = "Deleted " . count($deleted) . " old backup(s):\n" . implode("\n",
$deleted);
echo nl2br($txt . "\n");
?>
----------------------End Copy Immediately Above---------------------------
```

Within the PHP script, there is an option you can enable that will automatically create directories based the handset’s MAC address. Then each handset’s files that are uploaded will be placed into the folder for their specific MAC Address. Using this option will significantly improve the manageability of the directory structure of your Upload server. The following is the section of the script that needs to be modified to enable this option.

```
# Set to false for a flat uploads/ directory structure
# Set to true for uploads/mac_address directory structure
$mac_dir = true;
```

The highlighted value above defaults to “false” in the script. This needs to be changed to “true” to enable the directory creation process.

Document Status Sheet

Document Control Number: CS-19-05

Document Title: Spectralink Versity Advanced Debugging

Revision History: I01 – Review *May 31, 2019*
I02 – Released *June 7, 2019*
I03 – Released *November 21, 2019*

Date: *November 21, 2019*

Status: ☐Draft ☒Issued ☐Closed

Distribution Status: ☐Author Only ☐Internal ☐Partner ☒Public

Copyright Notice

Notice

Spectralink Corporation has prepared this document for use by Spectralink personnel and customers. The drawings and specifications contained herein are the property of Spectralink and shall be neither reproduced in whole or in part without the prior written approval of Spectralink, nor be implied to grant any license to make, use, or sell equipment manufactured in accordance herewith.

Spectralink reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult Spectralink to determine whether any such changes have been made.

NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING CAPACITY, RESPONSE-TIME PERFORMANCE, SUITABILITY FOR USE, OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY SPECTRALINK FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY OF SPECTRALINK WHATSOEVER.

Warranty

The *Product Warranty and Software License and Warranty* and other support documents are available at http://support.spectralink.com.

Contact Information

US Location
+1 800-775-5330

Spectralink Corporation
2560 55th Street
Boulder, CO 80301
USA

info@spectralink.com

Denmark Location
+45 7560 2850

Spectralink Europe ApS
Bygholm Soepark 21 E Stuen
8700 Horsens
Denmark

infoemea@spectralink.com

UK Location
+44 (0) 20 3284 1536

Spectralink Europe UK
329 Bracknell, Doncastle Road
Bracknell, Berkshire, RG12 8PE
United Kingdom

infoemea@spectralink.com