



Installing Certificates on Spectralink 8400 Handsets

Introduction

For the purposes of this document we will be showing you how to load certificates onto the Spectralink wireless telephone in a couple of different ways. We will start with the preferred method, via XML configuration files; and follow that up with an explanation of how to perform the upload via the handset web interface.

Using XML Configuration Files

Depending on how you intend to use your certificate it is quite possible you will need to load the certificate during the initial provisioning process. This implies that the certificate may be required for either WLAN authentication or authentication with the central provisioning server.

There are several components to be configured in order for the certificate to be loaded and a few requirements that need to be considered. The first, and most important, is that the certificate must be in Base 64 format. This ultimately means that the certificate file is text readable and can be opened in a text editor. Because we will be copying the contents of the certificate directly into the configuration file it will hopefully be obvious why the certificate file needs to be in Base 64 format.

Likely the easiest way to convert the certificate file to Base 64 is to export it from Windows in this format. When viewing certificates in the Windows Certificate Store you have the option of exporting them. The export function provides a number of different certificate formats, one of which is Base 64. So, if the certificate you need to load onto the handset is in a Windows system it may be easiest for you to just use Windows to convert the certificate. An alternative method is to use OpenSSL which is available on most Linux systems but can also be accessed in Windows via a program called Cygwin which provides a Linux interface with support for OpenSSL.

You can use the following command string to convert certificates using OpenSSL to get them output in the most useable format:

```
openssl enc -base64 -A -in certificatename.cer -out newcertificate.txt
```

The “-A” in the command string above outputs the certificate contents without the header and footer and puts the contents on a single line. This is an important consideration as the certificate can only be loaded into the handset configuration files if it is on a single line with the header and footer removed. So, if you export the certificate from Windows into Base 64 format you will need to manually remove the header and footer and then remove the carriage return/line feed from each line so the certificate contents are one continuous string.

With later releases of handset software, it is now possible to leave the header and footer in place and simply paste the certificate contents directly into the configuration file. The phone will handle removing the header and footer and putting the certificate all on one line. You can see this if you export the phone’s configuration file and look at the certificate parameters.

Configuration File Parameters

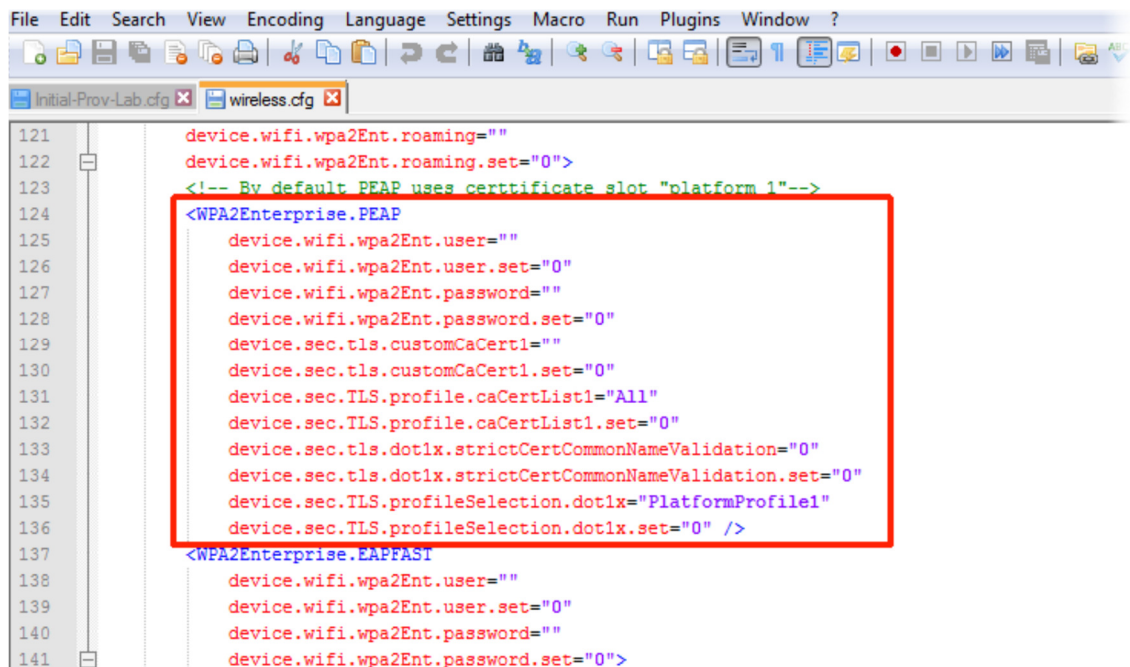
The initial provisioning of the Spectralink 8400 handsets is recommended to be done via the USB connection to a PC running an FTP server. Within the handset software packages, available from the support.spectralink.com website, is a folder with the configuration files for the handsets. There is a USB Setup folder included that has the initial wireless.cfg configuration file and the master configuration file, 000000000000.cfg that will be used to get the configuration onto the handset. Note that we will not cover these steps as they are well covered in the SpectraLink 8400 Series Deployment Guide - 4.2.0.

It is recommended that you use an XML/Text hybrid editor to edit the handset configuration files as this will offer you the greatest flexibility when managing your configuration files. Spectralink support engineers typically use either FOXE or Notepad++ for this purpose.

Let's start out by looking at the wireless.cfg file. There will be a few parameters that we will need to add into this configuration file but the setup will be relatively simple to complete. First, open the wireless.cfg configuration file in your editor; I'll be using Notepad++ in the examples shown.

Again, I would like to point out that we will not be covering all the parameters in this file for setting up the WLAN; this is only about loading certificates.

Figure 1 below shows the section of the wireless.cfg file that we will be working. So scroll down to this section and we'll get started.



```
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
Initial-Prov-Lab.cfg wireless.cfg
121 device.wifi.wpa2Ent.roaming=""
122 device.wifi.wpa2Ent.roaming.set="0">
123 <!-- By default PEAP uses certificate slot "platform 1"-->
124 <WPA2Enterprise.PEAP
125     device.wifi.wpa2Ent.user=""
126     device.wifi.wpa2Ent.user.set="0"
127     device.wifi.wpa2Ent.password=""
128     device.wifi.wpa2Ent.password.set="0"
129     device.sec.tls.customCaCert1=""
130     device.sec.tls.customCaCert1.set="0"
131     device.sec.TLS.profile.caCertList1="All"
132     device.sec.TLS.profile.caCertList1.set="0"
133     device.sec.tls.dot1x.strictCertCommonNameValidation="0"
134     device.sec.tls.dot1x.strictCertCommonNameValidation.set="0"
135     device.sec.TLS.profileSelection.dot1x="PlatformProfile1"
136     device.sec.TLS.profileSelection.dot1x.set="0" />
137 <WPA2Enterprise.EAPFAST
138     device.wifi.wpa2Ent.user=""
139     device.wifi.wpa2Ent.user.set="0"
140     device.wifi.wpa2Ent.password=""
141     device.wifi.wpa2Ent.password.set="0">
```

Figure 1

You'll likely first note that this section appears to apply to WPA2-Enterprise PEAP configuration. And you'd be correct in assuming that is its intended purpose. However, the parameters in this section are

relevant for our purposes regardless of whether you are using them for WLAN authentication or for something like FTPS or HTTPS. The specific parameters in this section that we will be looking at are the following:

```
device.sec.tls.customCaCert1=""  
device.sec.tls.customCaCert1.set="0"  
device.sec.TLS.profile.caCertList1="All"  
device.sec.TLS.profile.caCertList1.set="0"  
device.sec.tls.dot1x.strictCertCommonNameValidation="0"  
device.sec.tls.dot1x.strictCertCommonNameValidation.set="0"  
device.sec.TLS.profileSelection.dot1x="PlatformProfile1"  
device.sec.TLS.profileSelection.dot1x.set="0"
```

As I mentioned earlier, we will also need to add a few parameters that are not here by default. Here are the parameters that need to be appended to this section. These parameters will be used primarily for provisioning server authentication:

```
device.sec.TLS.customCaCert2=""  
device.sec.TLS.cusomCaCert2.set="0"  
device.sec.TLS.profile.caCertList2="All"  
device.sec.TLS.profile.caCertList2.set="0"  
device.sec.tls.prov.strictCertCommonNameValidation="0"  
device.sec.tls.prov.strictCertCommonNameValidation.set="0"  
device.sec.TLS.profileSelection.provisioning="PlatformProfile2"  
device.sec.TLS.profileSelection.provisioning.set="0"
```

Each of these parameters can be added to the wireless.cfg file in the WPA2Enterprise.PEAP XML container. This means that just after the device.sec.TLS.profileSelection.dot1x.set="0" parameter and before the /> you can press Enter and add the new parameters. This will ensure that these additional parameters are included inside a valid XML container and will be read by the handset.

Using the Parameters

Now that we've identified the parameters we need to use for this process we will discuss how to actually use them. Let's start with setting up the parameters for use with WLAN authentication.

WLAN Authentication

The Spectralink 8400 handset uses certificates for WLAN authentication when using WPA2-Enterprise with PEAP or EAP-TLS¹. The certificate you will need for this will be the root CA certificate for the RADIUS server the phone is authenticating to. If the RADIUS server is using a self-signed certificate, then you will need to export that certificate from the RADIUS server so we can load it onto the handset.

¹ EAP-TLS support is available in the 84-Series Handset with software release 4.9 and later.

Installing Certificates on Spectralink 8400 Handsets



A brief note on root CA certificates and certificate chains; when a RADIUS server has a certificate that was issued and signed by another certificate server, often a customer owned Certificate Authority, then you will need use the right certificate. It is not at all uncommon for site to have what are known as Intermediate Certificate Authorities. These servers are issuing and signing certificates for servers, such as the RADIUS server, however this is not the certificate the phone needs to validate the chain of authority. Because the Intermediate CA is using a certificate that was itself issued and signed by another server, the Root Certificate Authority, we will need to have that certificate in order to validate the chain of authority. Please also note that just because other wireless devices in your network may not require a certificate to work with PEAP the Spectralink handset will still require one. This is because these other devices are operating in an insecure mode where they do not validate the RADIUS server's identity before sending it their user credentials. The Spectralink handset will always validate the RADIUS server's identity before ever sending sensitive user credentials.

Once you've exported and converted your certificate for the handset you should end up with something that looks like this when viewed in a text editor:

-----BEGIN CERTIFICATE-----

MIIDcTCCAlmgAwIbAQIqf7Y9uouwrFM37k8j7dKhZANBgkqhkiG9w0BAQUFADBL
MRUwEwYKCZImiZPyLGQBGRYFbG9jYWwxFTATBgoJkiaJk/IsZAEZFgVlc2xhYjE5
MBkGA1UEAxMSZXNsYWVtU09KT1VSTkVSLUNBMB4XDTEyMDEyMjAwMzE0OVoXDTES
MDEyMjAwNDE0OFowSzEVMBMGCgmSJomT8ixkArkWBWxvY2FsMRUwEwYKCZImiZPy
LGQBGRYFZXNsYWVxGzAZBgNVBAMTEmVzbGFilVNPsk9VUK5FU1DQTTCCASlwdQYJ
KoZlhvcNAQEBBQADggEPADCCAQoCggEBAJm+5t991773BgZqaY4wdEXBPwFA2CJC
B+WEumJWk58fxTDi/47jRld76gM4uq3n6om+tC6fgWycpVJc+F1ohRNIw2KOft07
R2abYNU+04clo9kitsrMDScI5h2ghWVuZO/Lv1teP+dtNgnnHwKK4T1mAlc+W0e
vpl6ED5Nwd+lsdNs+C59Q0BLf3DKtzTDbuDEPvVVaeF6hciBFbIY8j2ZEpJHoUNv
jedP184rrbOSlzEp8hflnkyb+0fk36nuPXaj5VyYoBNj1SsUi5xSjzfvueq3PgmtJ
NhgAb0b9vik0zNXCZCtS0M1SQjt0mQdw3zNv0WKXdyxsoaHy67TuM8sCAwEAAANR
ME8wCwYDVR0PBAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEF0i6+z
MSM4f4L/B9V4KHAdu/JMBAGCSsGAQQBgicVAQQDAgEAMA0GCSqGSIlb3DQEBBQUA
A4IBAQA8T02PELEdGmFheHNhQs3oCQcdRllsI0o0uozTHL5q3GG2xL+9fyHfl8Hm
a7NGZHzZsZaRJH006ApqYdg3OcOwRZ2PdBrhJctD6o1joo/SYIQAkbnBzS9r2rSU
GRjm4sZldkleJDDGiwL1XCr9ZalMNCdcf4nHf9/y0C4HVSOKH15yRzUXei9s8b9
qAiJmNq0Y/OA55eq1s0Nvo+h5U5QcgjtXbUjM4mxdsuW+vK8ICzMIA7hDMUK1MR
4QBirt80kYhmAkYN77I3dXYQH1ix3pa7S2DL3XTVsx7e8HY/8CDH9Yke88LGSaZj
kw5tzBdOlRcWjXRT36I3W48zR/PD

-----END CERTIFICATE-----

This certificate is from our support lab root CA. If you had used OpenSSL to convert the certificate using the command string provided in this document, then the output would look like this:

MlIDcTCCAlmgAwIBAgIQfd7Y9uouwrFM37k8j7dKhzANBgqhkiG9w0BAQUFADBLMRUWewYKCZImiZPyL
GQBGRYfB9jYWwxFTATBgoKJiaJk/I sZAEZFgVlc2xhYjEbmMBkGA1UEAxMSZXNsYWItU09KT1VSTkvSLUNB
MB4XDTEyMDEyMDIyMTA0MDUoXDTExNDA0ODU0OTcwSjJomT8ixkARKWBWxvY2F

Installing Certificates on Spectralink 8400 Handsets



sMRUwEwYKCZlmiZPyLGQBGryfZXNsYwIxGzAZBgNVBAMTEmVzbGFiLVNPSk9vUk5FU1iDQTCcASlwDQYJKoZIhvcNAQEBBQADgEPADCCAQoCggEBAJm+5t99I773BgZqaY4wdEXBPwFA2CJCB+WEumJWk58fxTDi/47jRld76gM4uq3n6om+tC6fgWypcVJc+F1ohRNIw2KOft07R2abYNu+04clo9kitsrMDScl5h2ghWVuZO/Lv1teP+dtNgnnHwKK4T1mAlo+cW0evpl6ED5Nwd+lsdNs+C59Q0BLf3DKtZTDbuDEPvVvVaeF6hcbfBY8j2ZEpJHoUNvjedP184rrbOSlzEp8hflnkyb+Ofk36nuPXaj5VyYoBNj1SsUI5xSjzfvuq3PgmtJNhgAb0b9vik0zNXCZcTS0M1SQjt0mQdw3zNv0WKXdyxsoaHy67TuM8sCAwEAAaNRME8wCwYDVR0PBAQDAGGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBByEFOiO6i+zMSM4f4L/B9V4KHAdtU/JMBAGCSsGAQQBgjcVAQQDAGEA MA0GCSqGS1b3DQEBBQUAA4IBAQA8T02PELEdGmFheHNhQs3oCQdcRllsI0o0uozTHL5q3GG2xL+9fyHfl8Hma7NGZHzZsZaRJHOO6ApqYdg3OcOwRZ2PdBrhjCtD6o1joo/SYIQAkbnBzS9r2rSUGRjm4sZldklekJDdGi wL1XCr9ZalMNCdcf4nHf9/y0C4HVSOKH15yRzUXei9s8b9qAijMnQ0Y/OA55eq1s0Nvo+h5U5QcgjtxBujM4 mxdsuW+vK8lCzZMIA7hDMUK1MR4QBirt80kYhmAkAYN77I3dXYQh1ix3pa7S2DL3XTVsX7e8HY/8CDH9Yke 88LGSaZJkw5tzBdOlRcWjXRT36I3W48zR/PD

This long form output is what you'll need to load into the handset configuration file so if you didn't use OpenSSL to convert the certificate then just use your text editor to modify it. Depending on the key size you've used the length of the certificate will vary greatly. The certificate we've used here has a 2048 key size which is why it has a longer length.

Now that we've got our certificate we can go back to `wireless.cfg` file to show you where to put this information and make it work for your deployment. Figure 2 below shows what your configuration file will look like once you've pasted in the certificate.

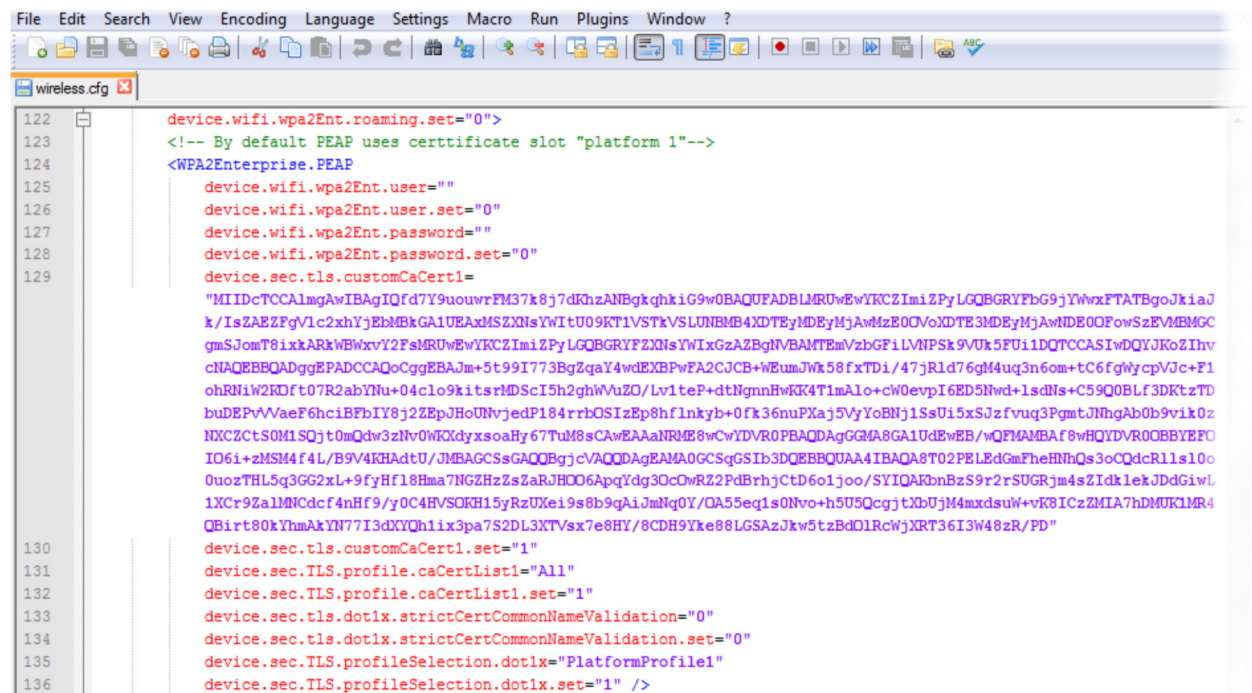


Figure 2

Be sure to note that one very important thing has been done in Figure 2 as well. That would be toggling the .set parameters for each of these parameters. All device parameters in the 8400 handset have an accompanying .set parameter that must also be set to “1” in order for the phone to use that parameter.

Installing Certificates on Spectralink 8400 Handsets



If the .set parameter is "0" then the handset will ignore the parameter. There is also a global device.set parameter at the very top of this wireless.cfg file but we've set it to "1" for you already by default.

Now let me go through each of the parameters so you can see what we did and why. First, we'll cover the certificate parameter.

device.sec.tls.customCaCert1=	This parameter is the actual certificate you are loading onto the handset. It provides the handset with the specific contents of the certificate body that are needed in order for the handset to validate server certificates sent to it during the authentication process.
device.sec.TLS.profile.caCertList1="All"	This parameter tells the handset which certificates in its certificate store it can utilize for authentication purposes. There are two certificate lists, 1 and 2, which will always point to All certificates by default. This is the preferred setting as it ensures the handset can attempt to use any certificate to validate the server certificate it receives. There are a large number of well-known certificate authorities pre-loaded into the handset as well.
device.sec.TLS.profileSelection.dot1x="PlatformProfile1"	This parameter tells the handset that for dot1x authentication it will need to look at the certificates applied to Platform Profile 1. There are two Platform Profiles, 1 and 2. Each can handle a single certificate for the specified application, in this case dot1x authentication.

These three parameters are all that is required, along with their .set partners that is, in order to load a certificate onto a Spectralink 8400 handset. The procedure we've just detailed will load the WLAN authentication certificate into slot 1 on the handset which can then be verified directly on the handset. From the carousel go to the Settings->Advanced Settings->456->Administration Settings->TLS Security->Custom CA Certificates and you should see your certificate in the first entry on the phone.

Don't forget that with EAP-TLS the phone requires a device certificate as well. You can choose to use the Spectralink loaded device certificate and download our PKI certificate chain which will significantly simplify your deployment. Or you can choose to use your PKI infrastructure and issue device certificates for each handset. When doing so, it is necessary to export the private key with the device certificates as you will be required to load the private key into the phone along with the device certificate. It's important to leave the private key header intact when putting it into the phone's configuration file.

You can download the Spectralink root CA and issuing certificates here:

<http://pki.spectralink.com/aia/Spectralink%20Root%20CA.crt>
<http://pki.Spectralink.com/aia/Spectralink%20Issuing%20CA.crt>

Installing Certificates on Spectralink 8400 Handsets



You may need to extract the private key on its own from each device certificate. This can be accomplished using OpenSSL but you will need to know the passphrase that was used to lock the private key. You'll then need to remove that passphrase so that you can paste the key into the phone's configuration file. Most device certificates are contained in a PFX package which contains the certificate and key protected by the passphrase. Then you'll have to change the key into a readable, unencrypted, format. Here are the commands you'll need to make that PFX package usable by the phone.

```
openssl pkcs12 -in yourP12File.pfx -passout pass:yourPassphrase -nocerts -out privateKey.pem  
openssl rsa -in privateKey.pem -out rsaKey.pem  
openssl pkcs12 -in yourP12File.pfx -clcerts -nokeys -out publicCert.pem
```

Provisioning Server Authentication

When using FTPS or HTTPS for your provisioning server protocol you will also need to load a certificate onto the handset to allow for the SSL encryption to occur. The procedure is essentially identical to what we just did for WLAN Authentication but we will be using a few different parameters to complete the process. The primary reason for this is that there may be a situation where you need to do both WLAN authentications with certificates and your provisioning server setup. By using slightly different parameters we're ensuring that these two can easily coexist.

For this section we're going to be using those parameters I mentioned earlier that will need to be added to the wireless.cfg file. Here they are again:

```
device.sec.TLS.customCaCert2=""  
device.sec.TLS.cusomCaCert2.set="0"  
device.sec.TLS.profile.caCertList2="All"  
device.sec.TLS.profile.caCertList2.set="0"  
  
device.sec.tls.prov.strictCertCommonNameValidation="0"  
device.sec.tls.prov.strictCertCommonNameValidation.set="0"  
device.sec.TLS.profileSelection.provisioning="PlatformProfile2"  
device.sec.TLS.profileSelection.provisioning.set="0"
```

Figure 3 below shows what the configuration file might look like with these parameters added to it.

Installing Certificates on Spectralink 8400 Handsets



```
119 device.wifi.wpa2Ent.method=""
120 device.wifi.wpa2Ent.method.set="0"
121 device.wifi.wpa2Ent.roaming=""
122 device.wifi.wpa2Ent.roaming.set="0">
123 <!-- By default PEAP uses certificate slot "platform 1"-->
124 <WPA2Enterprise.PEAP
125     device.wifi.wpa2Ent.user=""
126     device.wifi.wpa2Ent.user.set="0"
127     device.wifi.wpa2Ent.password=""
128     device.wifi.wpa2Ent.password.set="0"
129     device.sec.tls.customCaCert1=""
130     device.sec.tls.customCaCert1.set="1"
131     device.sec.TLS.profile.caCertList1="All"
132     device.sec.TLS.profile.caCertList1.set="1"
133     device.sec.tls.dot1x.strictCertCommonNameValidation="0"
134     device.sec.tls.dot1x.strictCertCommonNameValidation.set="0"
135     device.sec.TLS.profileSelection.dot1x="PlatformProfile1"
136     device.sec.TLS.profileSelection.dot1x.set="1"
137     device.sec.TLS.customCaCert2=""
138     device.sec.TLS.cusomCaCert2.set="0"
139     device.sec.TLS.profile.caCertList2="All"
140     device.sec.TLS.profile.caCertList2.set="0"
141     device.sec.TLS.profileSelection.provisioning="PlatformProfile2"
142     device.sec.TLS.profileSelection.provisioning.set="0" />
143 <WPA2Enterprise.EAPFAST
144     device.wifi.wpa2Ent.user=""
145     device.wifi.wpa2Ent.user.set="0"
146     device.wifi.wpa2Ent.password=""
147     device.wifi.wpa2Ent.password.set="0">
```

Figure 3

With these additional parameters entered into the wireless.cfg file we can proceed with setting them appropriately. To start out, you will need to get the certificate from your provisioning server. In most cases this will likely be a self-signed certificate but it could also be a signed certificate. If it is a self-signed certificate you will need to follow the same steps outlined above for converting it to Base 64. Likewise, for a certificate signed by a root CA, we will want to get the root CA certificate and again convert it to Base 64.

The certificate will be copied into the customCaCert2 parameter this time which we are doing to ensure there is no conflict with a potentially loaded WLAN authentication certificate. You could still use customCaCert1 if desired. For the example setup, I used FileZilla for FTPS and generated a self-signed certificate with a 1024 key size.

-----BEGIN CERTIFICATE-----

```
MIICsDCCAhmGAWIBAgIBADANBgkqhkiG9w0BAQUFADCBNTEQMA4GA1UEAxMHamlT
c2VydjELMAkGA1UEBhMCMDExETAPBgNVBAGTCENvbG9yYWRvMRwDgYDVQQHEwdC
b3VsZGVyMRQwEgYDVQQKEwtTcGVjdHJhbGluazEQMA4GA1UECjMHU3VwcG9ydE
VwMC0GCSqGSIb3DQEJARYganVzdGluLmJvcnRod2lja0BzcGVjdHJhbGluay5jb2
wHhcNMTMwNDAzMTU0OTQ3WhcNMTQwNDAzMTU0OTQ3WjCBNTEQMA4GA1UEAxMHamlT
```

Installing Certificates on Spectralink 8400 Handsets



```
c2VydjELMAkGA1UEBhMCMDExETAPBgNVBAGTCENvbG9yYWRvMRAwDgYDVQQHEwdC
b3VsZGVyMRQwEgYDVQQKEwtTcGVjdHJhbGluazEQMA4GA1UECXMHU3VwcG9ydDEv
MC0GCSqGSIb3DQEJARYganVzdGluLmJvcnRod2lja0BzcGVjdHJhbGluay5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANxeWRFR2fuTHKuDgD3c4RfD7u15
o+ToeW4rqzegIIISGoDQIXFNByeZDIXc5mRtmZvZGbnxYmij9iWLFWJZtzOKo4pQC
QLTLRAG1sOgc0P7uj9ZhX1GjKyAVALWtJrWp2mBxZURJb8Ba9gJqa3CD7v7xB8tH
h8caC2RBABoXZhhVAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAeSXbNu2rZ4GXRgxl
AWi1Mtr4f93ulea/i1x4QkeUSuEnoj22FvT99b/eZTy5Gll6QDjgWjfdOG8n/Atn
mwNwRwz8r4NVG0FrXyrQlsuWhk2L6ex3BiKt//SZWTjctGdSWY9bodT4fsEcOrrg
ebCPhecTkwwysB5wNcFRpvJt/mlM=
-----END CERTIFICATE-----
```

The above certificate is what we'll be loading into the handset, but don't forget that we still need to remove the header and footer along with the line feeds so it is one continuous line. The following is what the same certificate looks like with those changes made to it.

```
MIICsDCCAhmGAWIBAgIBADANBgkqhkiG9w0BAQUFADCbnTEQMA4GA1UEAxMHamltc2VydjELMAkGA1
UEBhMCMDExETAPBgNVBAGTCENvbG9yYWRvMRAwDgYDVQQHEwdCb3VsZGVyMRQwEgYDVQQKEwtTc
GVjdHJhbGluazEQMA4GA1UECXMHU3VwcG9ydDEvMC0GCSqGSIb3DQEJARYganVzdGluLmJvcnRod2lja0
BzcGVjdHJhbGluay5jb20wHhcNMTMwNDAzMtU0OTQ3WhcNMTQwNDAzMtU0OTQ3WjCBnTEQMA4G
A1UEAxMHamltc2VydjELMAkGA1UEBhMCMDExETAPBgNVBAGTCENvbG9yYWRvMRAwDgYDVQQHEwdC
b3VsZGVyMRQwEgYDVQQKEwtTcGVjdHJhbGluazEQMA4GA1UECXMHU3VwcG9ydDEvMC0GCSqGSIb3D
QEJARYganVzdGluLmJvcnRod2lja0BzcGVjdHJhbGluay5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAo
GBANxeWRFR2fuTHKuDgD3c4RfD7u15o+ToeW4rqzegIIISGoDQIXFNByeZDIXc5mRtmZvZGbnxYmij9iWLF
WJZtzOKo4pQCQLTLRAG1sOgc0P7uj9ZhX1GjKyAVALWtJrWp2mBxZURJb8Ba9gJqa3CD7v7xB8tHh8caC2R
BABoXZhhVAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAeSXbNu2rZ4GXRgxlAWi1Mtr4f93ulea/i1x4QkeUS
uEnoj22FvT99b/eZTy5Gll6QDjgWjfdOG8n/AtnmwNwRwz8r4NVG0FrXyrQlsuWhk2L6ex3BiKt//SZWTjctG
dSWY9bodT4fsEcOrrgebCPhecTkwwysB5wNcFRpvJt/mlM=
```

With the certificate in this format we can now import it directly into the handset configuration file for use in the handset. As mentioned, we'll be putting this into customCaCert2 this time which will look something like figure 4 below.

Installing Certificates on Spectralink 8400 Handsets



```

122 device.wifi.wpa2Ent.roaming.set="0">
123 <!-- By default PEAP uses certificate slot "platform 1"-->
124 <WPA2Enterprise>
125     device.wifi.wpa2Ent.user=""
126     device.wifi.wpa2Ent.user.set="0"
127     device.wifi.wpa2Ent.password=""
128     device.wifi.wpa2Ent.password.set="0"
129     device.sec.tls.customCaCert1=""
130     device.sec.tls.customCaCert1.set="0"
131     device.sec.TLS.profile.caCertList1="All"
132     device.sec.TLS.profile.caCertList1.set="0"
133     device.sec.tls.dot1x.strictCertCommonNameValidation="0"
134     device.sec.tls.dot1x.strictCertCommonNameValidation.set="0"
135     device.sec.TLS.profileSelection.dot1x="PlatformProfile1"
136     device.sec.TLS.profileSelection.dot1x.set="0"
137     device.sec.TLS.customCaCert2=
        "MIICsDCCAhmgAwIBAgIBADANBgkqhkiG9w0BAQFADCBnTEQMA4GA1UEAxMHAmItc2VydjEIMAAkGA1UEBhMCMDExET
        APBglNVBAGTCENvbgG9yYWRvMRAwDgYDVQQHEwdCb3VsZGVyMRQwEgYDVQQKEwtTcGVjdHJhbGluazEQMA4GA1UECXMHU
        3VwcG9yZDEvMCOGCSGqSIB3DQBJJARYganVzdGluLmJvcnRod21ja0BzcGVjdHJhbGluay5jb20wHhcnMTAwMDAzMTU0
        OTQ3WhcnMTAwMDAzMTU0OTQ3WjCBnTEQMA4GA1UEAxMHAmItc2VydjEIMAAkGA1UEBhMCMDExETAPBgNVBAGTCENvbgG9
        yYWRvMRAwDgYDVQQHEwdCb3VsZGVyMRQwEgYDVQQKEwtTcGVjdHJhbGluazEQMA4GA1UECXMHU3VwcG9yZDEvMCOGCS
        qSIB3DQBJJARYganVzdGluLmJvcnRod21ja0BzcGVjdHJhbGluay5jb20wZG8wDQYJKoZIhvcNAQEBBQADgY0AMIGJA
        oGBANxeWRFRI2fuTHKuDgD3c4RFD7u15o+ToeW4rqzegIISGoDQ1XFBNbyeZDIXc5mRtmZvZGbnxYmij9iWLfWJZtZOkO
        4pQCQLTLRAG1sOgcOP7uJ9zhx1GjKyAVALWtJrWp2mBxZURJ8b8a9gJqa3CD7v7xB8tHh8ca2CRBABoXzhVAgMBAAE
        wDQYJKoZIhvcNAQEFBQADgYEAeSXBu2rZ4GXRgx1AWi1Mtr4f93ulea/i1x4QkeUSUEnoj22FvT99b/eZTy5G1l6QD
        jgWjfdOG8n/AtmmwNRwz8r4NVG0FrXyrQ1suWhk2L6ex3BiKt//SZWTjctGdSWY9bodT4fsEccOrrgbeCPhecTkWysB
        5wNcFRpvJt/mLM="
138     device.sec.TLS.cusomCaCert2.set="1"
139     device.sec.TLS.profile.caCertList2="All"
140     device.sec.TLS.profile.caCertList2.set="1"
141     device.sec.TLS.profileSelection.provisioning="PlatformProfile2"
142     device.sec.TLS.profileSelection.provisioning.set="1" />

```

Figure 4

With the certificate loaded we can look at the differences between this setup and the WLAN Authentication example. Other than the obvious differences, the most important one is the last parameter as this defines which certificate to use for the provisioning process.

```
device.sec.TLS.profileSelection.provisioning="PlatformProfile2"
```

The main difference here is that this parameter includes the word “provisioning” which is what helps you to identify its purpose.

The last thing we would need to do is identify the provisioning server type being used. Fortunately, that is also done in the `wireless.cfg` file which makes it easy to complete this process. Figure 5 below shows the parameters in the `wireless.cfg` file that you'll be making a change to for this last step.

Installing Certificates on Spectralink 8400 Handsets



```
28 device.auth.localAdminPassword.set="0" />
29 <!-- ***** -->
30 <!-- * Configuration Provisioning * -->
31 <!-- ***** -->
32 <!--Provisioning Server types: 0-FTP, 1-TFTP, 2-HTTP, 3-HTTPS, 4-FTPS-->
33 <ProvisioningServer
34 device.dhcp.bootSrvUseOpt="CustomAndDefault"
35 device.dhcp.bootSrvUseOpt.set="0"
36 device.dhcp.bootSrvOpt="160"
37 device.dhcp.bootSrvOpt.set="0"
38 device.prov.serverType="0"
39 device.prov.serverType.set="0"
40 device.prov.serverName="0"
41 device.prov.serverName.set="0"
42 device.prov.user="0"
43 device.prov.user.set="0"
44 device.prov.password="0"
45 device.prov.password.set="0" />
46 <!-- ***** -->
```

Figure 5

The server type for the parameter is based on an index value, which is provided in the green comment within the file. For FTPS we'll be using server type "4". And don't forget, there is that pesky .set parameter again so that will need to be changed to "1" in order for the server type parameter to be used.

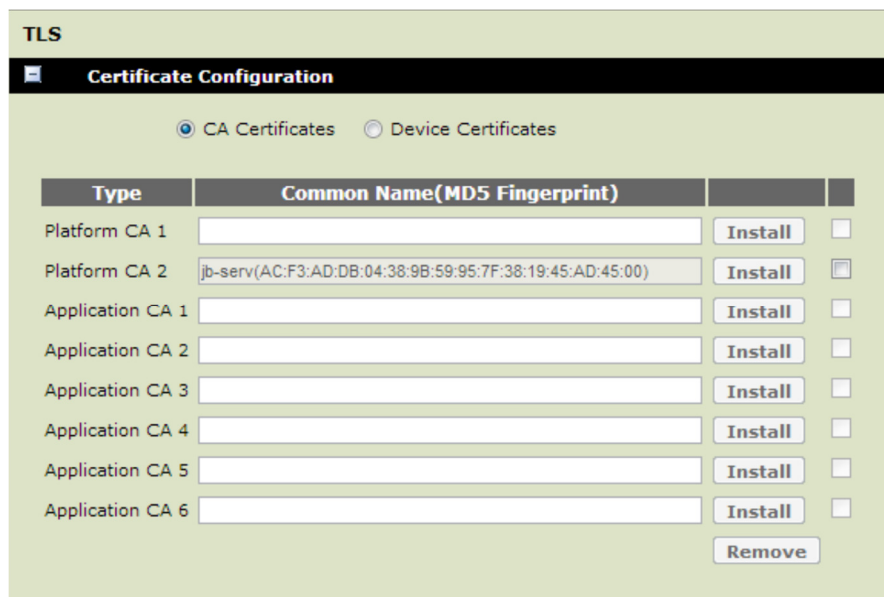
This is a good opportunity to point out the provisioning server username and password as well. The default value for both is PlcmSplp; which is case sensitive. You may desire to change this for greater security. To do so you'll just need to change the device.prov.user and the device.prov.password parameters along with their .set partners.

One final note on this topic relates to certificate names. In most situations certificates have a name associated with them that is tied to the server that it has been issued to. In most cases this won't be much of a problem, but it could be an issue if you are using self-signed certificates. This all boils down to what is known as Common Name Validation. When a server sends its certificate to the client to be validated one of the default steps the phone will do is to verify that the Common Name tied to the certificate matches the name of the server it is talking to. If they don't match, the default behavior is to deny the connection and disconnect. What we've included in the parameters provided above is a way to allow the phone to ignore the name of the server and the name in the certificate. Note that this is less secure but for many environments it will be perfectly acceptable. For the purposes of this document we've disabled Common Name Validation as it will often cause issues with connecting to the server. What you will likely see is in the log of your FTPS or HTTPS server the device will establish a connection but will not send its authentication credentials. This should be a dead giveaway of what the issue may be.

Web Interface and Certificates

The alternative to using the configuration files to perform a certificate upload to the handset is to use the phone's built in web interface. The web interface is extremely limited in what it allows you to configure for the handset but it can be helpful in small deployments and for troubleshooting. The major disadvantage to this approach is that it requires you to access each phone individually after they have been provisioned onto the WLAN. Hopefully you see the advantage to use the configuration files over this more intensive approach.

Once you've connected your browser to the handset's IP address you can go to Settings->Network->TLS to access the Certificate Configuration screen. This interface can be a little bit confusion because it anticipates that you will be loading the certificate from a server in the network. Figure 6 shows the Certificate Configuration screen:



TLS

Certificate Configuration

☒ CA Certificates ☐ Device Certificates

Type	Common Name(MD5 Fingerprint)		
Platform CA 1		Install	<input type="checkbox"/>
Platform CA 2	jb-serv(AC:F3:AD:DB:04:38:9B:59:95:7F:38:19:45:AD:45:00)	Install	<input checked="" type="checkbox"/>
Application CA 1		Install	<input type="checkbox"/>
Application CA 2		Install	<input type="checkbox"/>
Application CA 3		Install	<input type="checkbox"/>
Application CA 4		Install	<input type="checkbox"/>
Application CA 5		Install	<input type="checkbox"/>
Application CA 6		Install	<input type="checkbox"/>

Remove

Figure 6

In the above image I have already loaded a certificate into slot 2 which was accomplished using the procedure detailed in the previous section. This is another method for validating that a certificate has been loaded into your handset.

To load a new certificate into the phone you will need to decide which slot you want to use. It is recommended that for WLAN Authentication or for Provisioning Server Authentication that you only use Platform CA 1 and Platform CA 2. This is only because it simplifies the overall configuration. The certificate that you will be uploading to the phone will still need to be in Base 64 format. However, it will need to have the header and footer and retain its original format rather than being condensed onto a single line. To complete this process, the phone will need to be on the WLAN and ideally on a network

Installing Certificates on Spectralink 8400 Handsets



that can access a system that will host the certificate. The phone will use HTTP or FTP to get the certificate from the user.

In the slot you want to install the certificate you will need to enter the connection string with the path to the certificate. Figure 7 shows what this string might look like if you are using FTP to host the certificate for the handset.

The string should include the protocol, the user credentials, server address and file path and name.

For example, if you were using HTTP with user authentication the connection string might look something like this:

`HTTP://UserID:Password@serveraddress/filepath/filename`

If you did not opt for user authentication, then the connection string might look like this:

`HTTP://serveraddress/filepath/filename`

When using FTP for provisioning the certificate to the phone your connection string might look like this:

`FTP://UserID:Password@serveraddress/filepath/filename`

Since anonymous FTP is not allowed with the Spectralink 8400 handsets you will always need to provide a username and password when performing this step with FTP.

TLS

Certificate Configuration

☒ CA Certificates ☐ Device Certificates

Type	Common Name(MD5 Fingerprint)		
Platform CA 1	eslab-SOJOURNER-CA(18:46:99:B0:5C:B9:B2:7C:1D:C9:10:27:32:5)	Install	<input type="checkbox"/>
Platform CA 2	ftp://ESSplp:ESSplp@192.168.142.71/FTPCert.cer	Install	<input type="checkbox"/>
Application CA 1		Install	<input type="checkbox"/>
Application CA 2		Install	<input type="checkbox"/>
Application CA 3		Install	<input type="checkbox"/>
Application CA 4		Install	<input type="checkbox"/>
Application CA 5		Install	<input type="checkbox"/>
Application CA 6		Install	<input type="checkbox"/>

Remove

Figure 7

Once you've entered the connection string into the slot you can then press the Install button in the browser. This will cause the handset to attempt to download the certificate and install it into the slot

Installing Certificates on Spectralink 8400 Handsets



you've specified. Figure 8 is an image of the message that will display on the screen while this process is taking place.

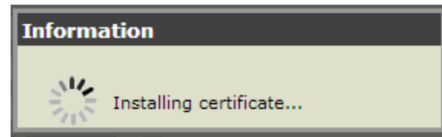


Figure 8

Once the process is complete the browser will display a notification message to indicate whether it was successful in loading it or not. Figure 9 below is the successful message that will display when the certificate is properly loaded.

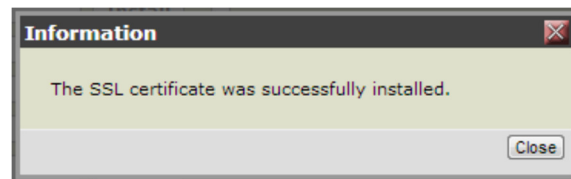


Figure 9

Completing the Process

Now that the certificate is loaded onto the handset you will be able to proceed with connecting the handset securely to the central provisioning server. The procedure for completing this process is obviously contingent on getting the handset onto the WLAN first, at least when using the web interface. When you load the certificate via the configuration files, via the USB interface, it is unnecessary to start with a WLAN connection.

Hopefully this document has helped you to better understand how to utilize certificates with the Spectralink 8400 Series handsets. If you need additional information, please contact the Spectralink technical support center.