

# Managed Firmware Plugin

For devices that read firmware updates from HTTP, TFTP or FTP directories, you can create a settings descriptor file that will be used to generate a web page to upload files for the device and deliver them to the appropriate directory.

## Step 1: Declare upload specification

Add a new upload specification to any existing Spring bean file. For example *sipXconfig/neoconf/src/org/sipfoundry/sipxconfig/phone/phone.beans.xml*

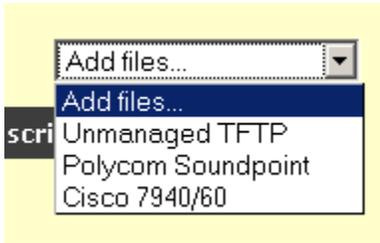
```
<bean id="polycomFirmware"
  class="org.sipfoundry.sipxconfig.upload.UploadSpecification">

  <!-- Use upload spring bean -->
  <constructor-arg value="upload"/>

  <!-- Directory to find upload.xml file -->
  <constructor-arg value="wayCool"/>

  <!-- Label -->
  <constructor-arg value="Way Cool Device"/>

</bean>
```



## Step 2: Describe upload files

Create following file

*sipXconfig/neoconf/etc/wayCool/upload.xml*

```
<?xml version="1.0"?>
<!DOCTYPE model PUBLIC "-//SIPFoundry//sipXconfig//Model specification 2.0//EN"
"http://www.sipfoundry.org/sipXconfig/dtd/setting_2_0.dtd">
<model>
  <group name="firmware">
    <label>Firmware</label>
    <description>Firmware for Way Cool Device</description>
    <setting name="wayCoolFirmware.dat">
      <label>wayCoolFirmware.dat</label>
      <type><file><contentType>application/binary</contentType></file></type>
      <description>Firmware binary.</description>
    </setting>
  </group>
</model>
```

The group and setting names and labels are irrelevant, but pick meaningful values incase you need to process the settings.

## Advanced

- You can write your own code to deliver or process the uploaded files. Extend Upload.java, add an entry in Spring file similar to Spring bean "upload". Change the first argument of the upload specification entry to your spring bean name.
- Declare firmware, ring tones, dial plans, directories, etc, in your upload.xml file. Break into logical groups for better presentation in web UI.

- To add support for multiple firmware versions, you can create a series of upload.xml files to organize and extend other versions, much like phone.xml and line.xml files. For example:  
*sipXconfig/etc/wayCool*

```
upload.xml – all firmware files
upload_3.x.xml – describe firmware files unique to version 3.x
upload_4.x.xml – describe firmware files unique to version 4.x
```

and multiple upload specification entries in the spring file for 3.0 and 4.0

## Future

- Phone and gateways will be able to select from a list of available upload entries. At minimum this will enable new setting for the device to show up based on that firmware version.