

Add WeeWX RSS feed to Hamclock

Mark Phillips, NI2O 07/2024 V1.03

Contents

Abstract.....	2
Assumptions.....	2
Caveats.....	2
Download the Hamclock-contrib archive.....	2
Install python/perl requirements.....	2
Edit WeeWX RSS template.....	2
Test the script.....	4
Update the information.....	4

Abstract

This article aims to instruct the reader on the required steps to create an RSS feed capable of displaying weather data from their own weather station onto their Hamclock display.

Assumptions

A functioning installation of Hamclock

A functioning installation of WeeWX

Network connectivity between both applications

Caveats

This install was performed on an Intel based PC rather than a Raspberry Pi or other SBC. While Debian is Debian, there may be issues between the various versions. This document is presented as-is. No guarantees are made as to this document resulting in a functioning RSS feed.

Download the Hamclock-contrib archive

<https://www.clearskyinstitute.com/ham/HamClock/hamclock-contrib.zip>

Unpack the file. It will create its own directory.

```
unzip hamclock-contrib.zip
```

Install python/perl requirements

The script we require from the above zip file requires a handful of helper libraries.

```
apt install python3-unidecode libtext-unidecode-perl libhtml-parser-perl libxml-libxml-perl
```

Edit WeeWX RSS template

For this project we used the “Standard” template supplied with the WeeWX software. It can be found in /etc/weewx/skins/Standard/RSS/weewx_rss.xml.tmpl Ensure to make a backup copy of this file before continuing.

The RSS stream display on the Hamclock only displays the titles of the RSS stories. We need to create (or edit) a story that will contain our WeeWX information.

Edit the file so that the information you want to see on the Hamclock RSS is between the <title></title> XML tags. There should be only one set of <item></item> tags too

```
nano /etc/weewx/skins/Standard/RSS/weewx_rss.xml.tmpl
```

An example of the one we created is shown below

```
<item>
    <title>$gettext("At") $current.dateTime, Temp: $current.outTemp, Pressure:
$current.barometer, Windspeed: $current.windSpeed, Direction: $current.windDir, Rainfall:
$current.rainRate</title>
    <link>$station.station_url</link>
    <description>
        $obs.label.outTemp: $current.outTemp;
        $obs.label.barometer: $current.barometer;
        $obs.label.wind: $current.windSpeed $gettext("from") $current.windDir;
        $obs.label.rainRate: $current.rainRate;
        $obs.label.inTemp: $current.inTemp
    </description>
    <pubDate>$current.dateTime.format("%H:%M:%S %Z")</pubDate>
    <geo:lat>$station.latitude_f</geo:lat>
    <geo:long>$station.longitude_f</geo:long>
    <content:encoded><![CDATA[
        <p>
            $obs.label.dateTime: $current.dateTime<br/>
            $obs.label.outTemp: $current.outTemp<br/>
            $obs.label.inTemp: $current.inTemp<br/>
            $obs.label.windchill: $current.windchill<br/>
            $obs.label.heatindex: $current.heatindex<br/>
            $obs.label.dewpoint: $current.dewpoint<br/>
            $obs.label.outHumidity: $current.outHumidity<br/>
            $obs.label.barometer: $current.barometer<br/>
            $obs.label.wind: $current.windSpeed $gettext("from") $current.windDir<br/>
            $obs.label.rainRate: $current.rainRate<br/>
        </p>
    ]]></content:encoded>
</item>
```

Save the edits to your file and then wait until your WeeWX instance does an update. Mine is set for every 5 minutes. WeeWX will create a new set of HTML pages along with a new RSS file containing the updated weather information. This can be viewed at

http://ipaddress_of_WeeWX/RSS/weewx_rss.xml

Test the script

We need to make sure that the script/RSS information works before we continue. In the above hamclock-contrib.zip file was a perl script called "hcrss.pl" (the one we needed the libraries for). Running the file from the command line will produce a short explanation of how it works (you may need to do chmod +x ./hcrss.pl first?)

```
root@km4wsk-aprs-wx:~/hamclock-contrib# ./hcrss.pl
```

Purpose: read an RSS feed and send the titles to hamclock.

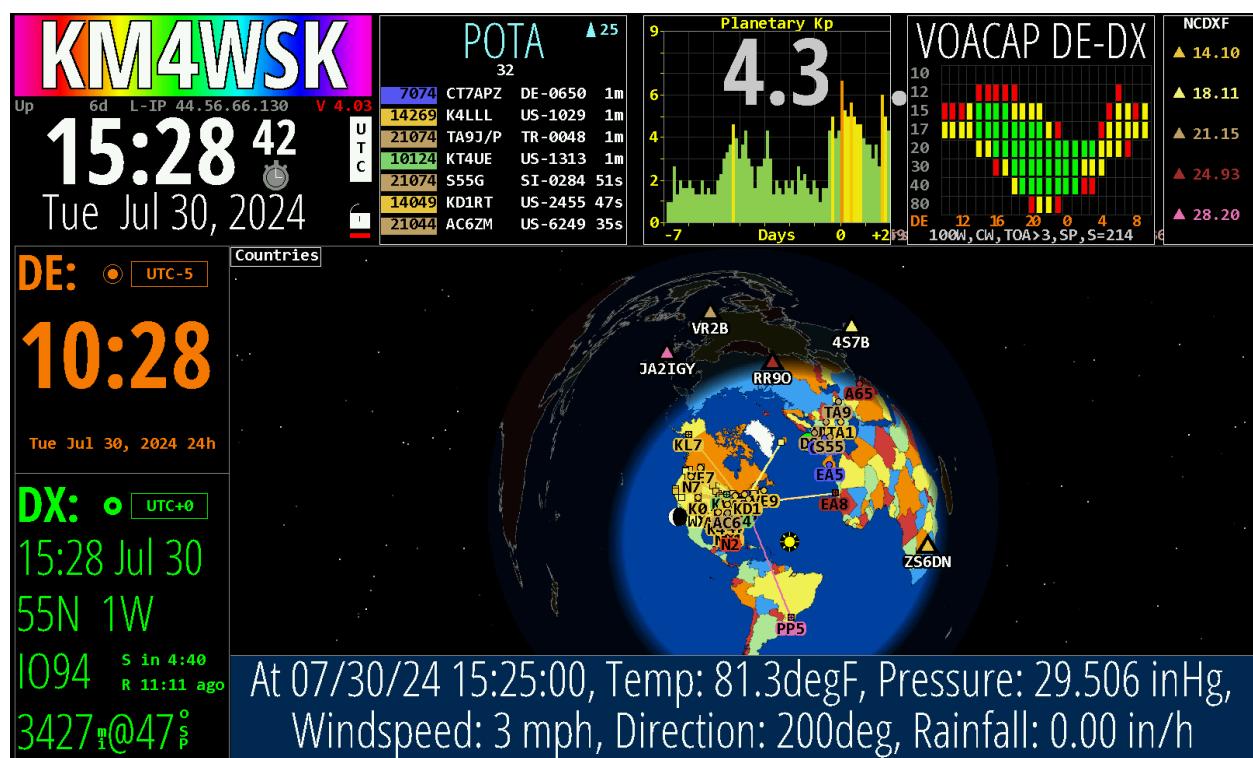
Usage: hcrss.pl hamclock_ip feed_url

Example: hcrss.pl 192.168.7.117 https://www.arrl.org/news/rss

So now you know how to execute the command. In our case we used the below:

```
./hcrss.pl 44.56.66.130 http://weather.km4wsk.ampr.org/weather/RSS/weewx\_rss.xml
```

The result was an RSS banner as seen below



Update the information

This new RSS source needs to be updated regularly if you want it to be anywhere near relevant to your actual weather station. The simplest way to do this is by using a cron job to run the RSS update script every 5 minutes.

Edit your crontab and paste in the below entry

```
# update the RSS every 5 minutes with the weather from WeeWX RSS template
*/5 * * * * /root/hamclock-contrib/hcrss.pl 44.56.66.130
http://weather.km4wsk.ampr.org/weather/RSS/weewx_rss.xml
```

It should be noted that there will be no other RSS headline displayed. You have exchanged the supplied RSS headlines with your own. If you want rotating headlines AND weather data you will have to create or call other RS feeds in addition to the WeeWX feed. This can be done by calling the hcrss.pl script from the crontab on a different schedule and injecting the RSS feed you want.

```
# update the RSS every hour with ARRL headlines at 20 past the hour
20 * * * * /root/hamclock-contrib/hcrss.pl 44.56.66.130
https://www.arrl.org/arrlletter/audio/aan.rss
```