

[English](#) • [Deutsch](#) • [Español](#) • [Français](#) • [Italiano](#) • [???](#) • [Polski](#) • [Português](#) • [??????](#) • [Svenska](#) • [???\(????\)?](#) • [???\(??\)?](#)

The contents of this howto will show you how to do the following:

- Create a script that will output information to an accessible location in a format that is easy to modify.
- Transform the data into RSS format and ready for reading by a program.

## Contents

- 1
  - Configuration
    - ◆ 1.1
      - Router
        - ◇ 1.1.1
          - Startup Script
    - ◆ 1.2
      - Client
    - ◆ 1.3
      - Shell Script

## Configuration

### Router

#### Startup Script

You will need to have this script load on a startup script:

```
mkdir -p /tmp/www
while [ 1 ];
do
sed -n 's%.* src=(192.168.[0-9.]*\).*%\1%p' /proc/net/ip_contrack | sort | uniq -c | awk '{t++
awk '/MemTotal:/ {mt = $2} /MemFree:/ {mf = $2} /Buffers:/ {mb = $2} /^Cached:/ {mc = $2} END {
awk '{ printf( "Load: %3d%%\n", $1*100); }' /proc/loadavg | tee -a /tmp/www/stat.html
awk '{ printf( "Uptime: %2.2f Hours\n", $2/3600); }' /proc/uptime | tee -a /tmp/www/stat.html
sleep 30;
done;
```

This script will parse out various peices of status information from the files located on the router. If you need more or less of the information or different formatting, you can edit this down to please you. The other script that will read this is independent so it can take more or less lines of information as needed without a hicup. The only syntax requirement for proper reading in the other script is that each status must be fully contained on its own line (but even that isn't really mandatory). The output is stored in /tmp/www/stat.html and can be accessed at <http://192.168.1.1/user/stat.html> (substitute IP if needed).

This is a sample of what this file will contain:

## RSS\_Feed\_Output

```
Connection 1: 192.168.1.1 has 476 connections open.  
Connection 2: 192.168.1.143 has 267 connections open.  
Free: 9%  
Used: 90%  
Buffers: 12%  
Cached: 35%  
Load: 45%  
Uptime: 1.30 Hours
```

## Client

This article assumes you have a linux operating system. This can probably also be done in windows, and I will attempt to explain the process so that if anyone wants to they can easily understand what is going on and reproduce it in windows if they have the knowledge.

## Shell Script

```
#!/bin/bash  
rm /tmp/rssfeed.xml;  
echo '<?xml version="1.0" ?><rss version="2.0"><channel><title>Router Connection Status</title><link>http://192.168.1.1/user/stat.html</link><description>WRT54G</description><item><title>"$0"</title><link>http://192.168.1.1/</link><description></description></item></channel></rss>' | tee -a /tmp/rssfeed.xml;
```

I have it the script run every minute using Cron. Basically what it does is it downloads the stat.html file you have hosted on your computer and uses that to create a /tmp/rssfeed.xml file. The created /tmp/rssfeed.xml file that conforms to RSS markup standards for input into a program of your choice. Firstly it adds an RSS header to the start of the file:

```
<?xml version="1.0" ?>  
<rss version="2.0">  
<channel>  
<title>Router Connection Status</title>  
<link>http://192.168.1.1/</link>  
<description>WRT54G</description>
```

Then it will add the RSS tags around each status line, so this:

```
Connection 1: 192.168.1.1 has 476 connections open.
```

will become:

```
<item>  
<title>Connection 1: 192.168.1.1 has 476 connections open.</title>  
<link>http://192.168.1.1/</link>  
<description></description>  
</item>
```

And then a footer markup to the end of the file:

```
</channel></rss>
```

Once it completes you have a /tmp/rssfeed.xml file that looks something like this:

## RSS\_Feed\_Output

```
<?xml version="1.0" ?>
<rss version="2.0">
<channel><title>Router Connection Status</title><link>http://192.168.1.1</link><description>WRT5
<item>
<title>Connection 1: 192.168.1.1 has 71 connections open.</title>
<link>http://192.168.1.1/</link>
<description></description>
</item>
<item>
<title>Connection 2: 192.168.1.143 has 65 connections open.</title>
<link>http://192.168.1.1/</link>
<description></description>
</item>
<item><title>Free: 11%</title>
<link>http://192.168.1.1/</link>
<description></description>
</item>
<item>
<title>Used: 88%</title>
<link>http://192.168.1.1/</link>
<description></description>
</item>
<item>
<title>Buffers: 12%</title>
<link>http://192.168.1.1/</link>
<description></description>
</item>
<item>
<title>Cached: 36%</title>
<link>http://192.168.1.1/</link>
<description></description>
</item>
<item>
<title>Load: 30%</title>
<link>http://192.168.1.1/</link>
<description></description>
</item>
<item><title>Uptime: 0.29 Hours</title>
<link>http://192.168.1.1/</link>
<description></description>
</item>
</channel>
</rss>
```

This is basically an all purpose script that will change any input file into a format that an RSS