The IDL socket procedure is a powerful tool for connecting to and accessing content on remote servers. In this post, I will demonstrate how to use the socket procedure to download a file by sending an HTTP GET request.

The socket command syntax is:

```
socket,unit,host,port,/get_lun
```

where:

- Unit = logical unit number
- Host = remote host name
- Port = remote host port number [80 for HTTP]
- Get_lun = keyword to request a logical unit number

In the following example, I open a socket to the mission website for NASA's Solar Dynamics Observatory and send a GET request for a JPEG image of the Sun at http://sdo.gsfc.nasa.gov/assets/img/latest/f_211_193_171_512.jpg.

```
IDL> socket,lun,"sdo.gsfc.nasa.gov",80,/get_lun
IDL> printf,lun,"GET /assets/img/latest/f_211_193_171_512.jpg
HTTP/1.1"
IDL> printf,lun,"Host: sdo.gsfc.nasa.gov:80"
IDL> printf,lun,string(10b)
```

Note the following:

- the GET request is sent as text by using printf
- the remote file must include a full path name and be followed by a protocol, which is HTTP 1.1 in this example.
- the request must include a Host:Port keyword. Port 80 is assumed if not specified.
- the last command is a blank string (actually a LF).

Once the server receives and interprets the GET request, it responds with header text which I read into a string array using readf as follows:

```
header=""                                ;-- initialize strings
text="xxx"
 while text ne "" do begin               ;-- read each line of text
  readf,lun,text
```

```
  header=[header,text]                    ;-- append to header array
 endwhile
```

In this example, the HTTP header looks like this:

```
IDL> print, header

HTTP/1.1 200 OK
Date: Thu, 19 Mar 2015 18:37:02 GMT
Server: Apache/2.2.15 (CentOS)
Last-Modified: Thu, 19 Mar 2015 18:23:06 GMT
ETag: "35180008-f460-511a84a91b35f"
Accept-Ranges: bytes
Content-Length: 62458
Content-Type: image/jpeg
```

The header contains useful information about the remote file:

- HTTP/1.1 200 OK => the request was successful
- Content-Length: 62458 => the file size in bytes
- Content-Type: image/jpeg => the file type is a JPEG image

After reading the response header, I use readu to read (i.e. download) the actual byte data from the open socket. Since I know from the header that the JPEG file size is 62458 bytes, I initialize the byte array to this size before reading it:

```
IDL> data=bytarr(62458)                 ;-- create byte array for data
IDL> readu,lun,data                     ;-- read data from socket
IDL> close,lun                          ;-- close socket
```

After reading the data, I write it to a local file (e.g. output.jpeg):

```
IDL> openw,lun,"output.jpeg",/get_lun
IDL> writeu,lun,data
IDL> close,lun
```

Finally, I read and display the downloaded JPEG file:

```
IDL> read_jpeg,"output.jpeg",image,/true
IDL> tv,image,/true
```

Using the Socket procedure to download a file