

Palomar Transient Factory Archive Scripted Access Basic Use



In addition to an easy-to-use GUI interface, PTF also provides an interface to the archive which can be scripted, called “IBE”. This makes it easy to embed access to the archive directly in software. Two of the most common uses for this are to embed access to thumbnail images from PTF into a science tool, for example, a visualization tool for SNe. Another use is to quickly enable bulk nightly downloads of data, without requiring manual human intervention in order to use the GUI. This scripted interface’s capabilities are actually very similar to that of the GUI, primarily because both are application layers on top of a shared software framework.

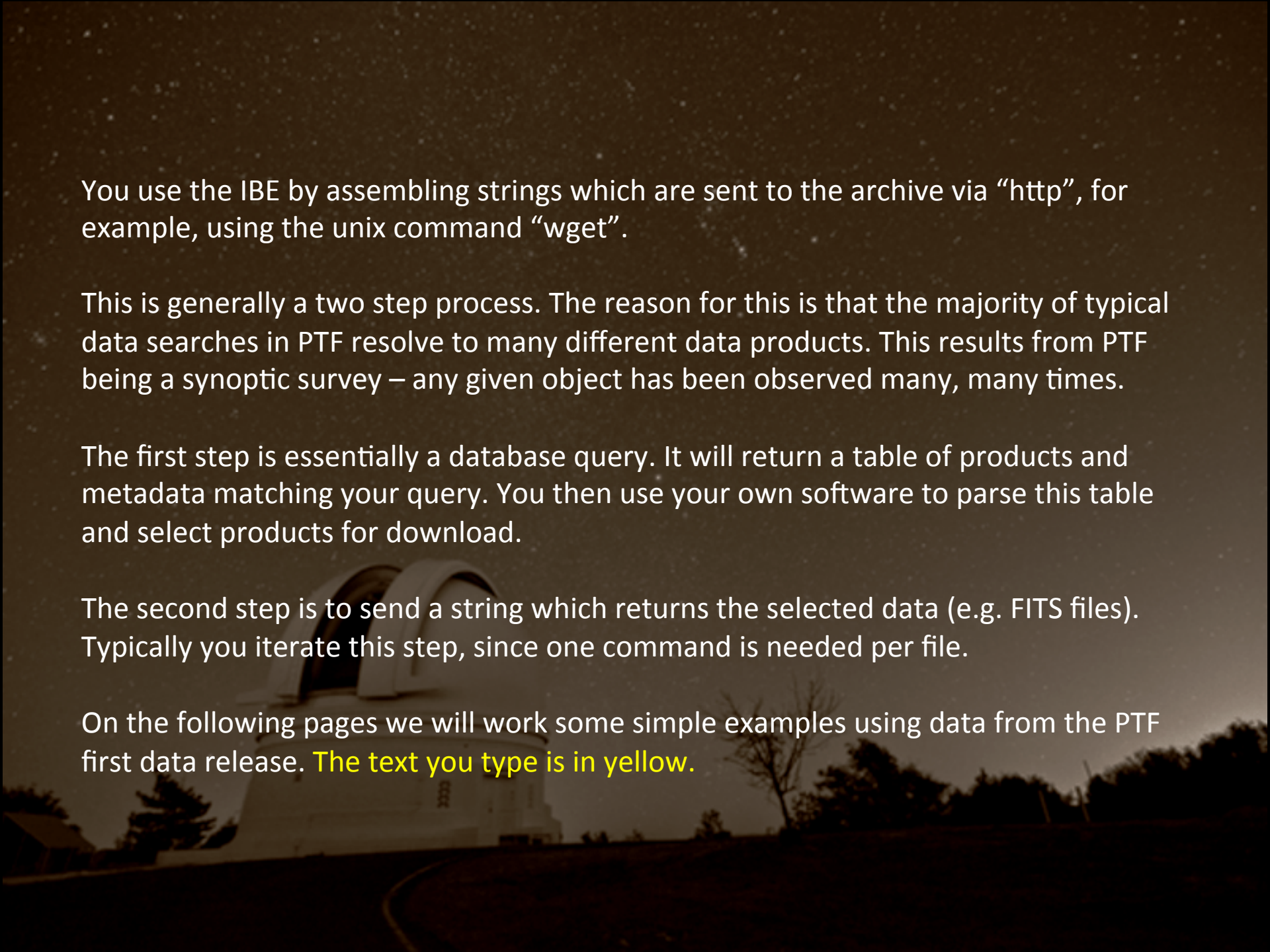
Detailed documentation for PTF is found at:

<http://irsa.ipac.caltech.edu/ibe/docs/ptf/images/level1/>

Generalized documentation on IBE capabilities, and in particular optional parameters, can be found at

<http://irsa.ipac.caltech.edu/ibe/index.html>





You use the IBE by assembling strings which are sent to the archive via “http”, for example, using the unix command “wget”.

This is generally a two step process. The reason for this is that the majority of typical data searches in PTF resolve to many different data products. This results from PTF being a synoptic survey – any given object has been observed many, many times.

The first step is essentially a database query. It will return a table of products and metadata matching your query. You then use your own software to parse this table and select products for download.

The second step is to send a string which returns the selected data (e.g. FITS files). Typically you iterate this step, since one command is needed per file.

On the following pages we will work some simple examples using data from the PTF first data release. **The text you type is in yellow.**

Simple Positional Search

Here is a simple positional search. The position is given in decimal degrees J2000 (and happens to be the starburst galaxy M82). I have redirected the output via the wget command to a file called “out.txt”. Note that public users only have access to the level 1 data at this time, and that is all we will cover here.

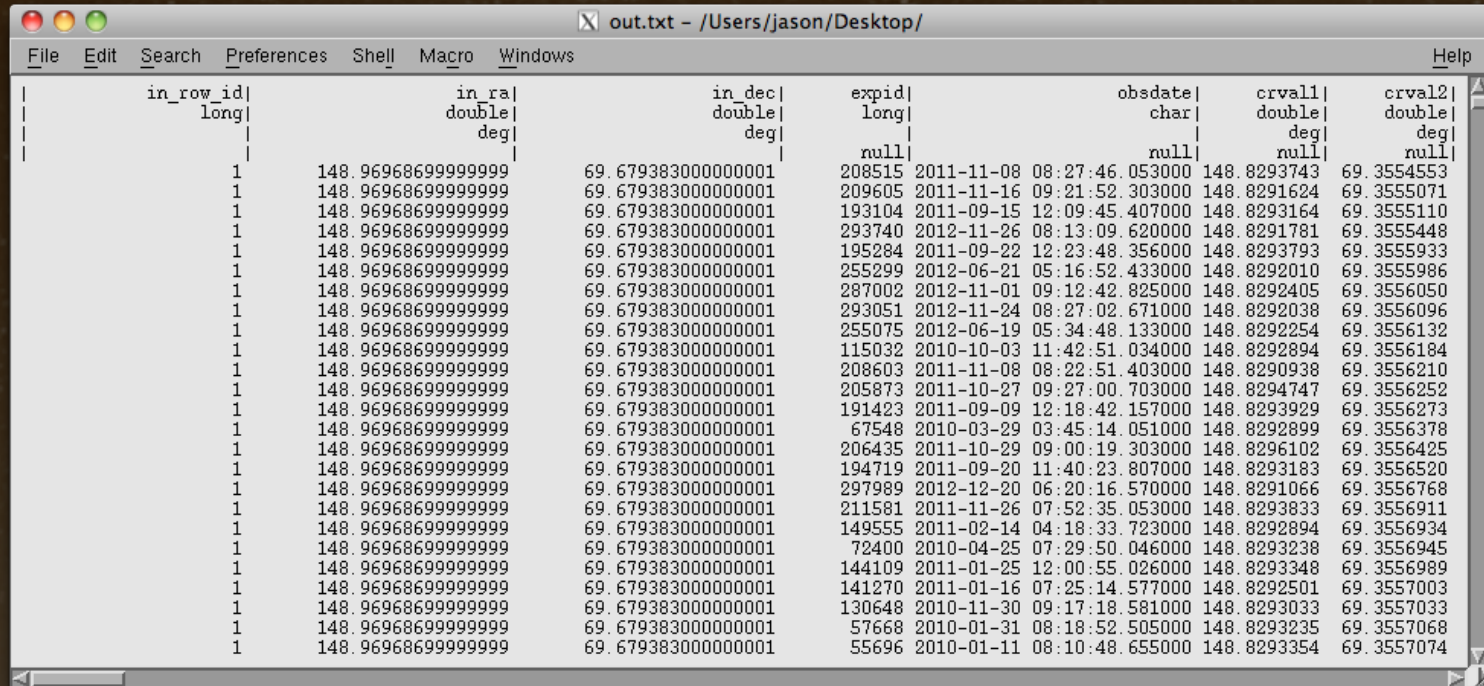
```
humu% wget "http://irsa.ipac.caltech.edu/ibe/search/ptf/images/level1?
POS=148.969687,69.679383" -O out.txt
--2014-05-01 16:43:09-- http://irsa.ipac.caltech.edu/ibe/search/ptf/images/level1?
POS=148.969687,69.679383
Resolving irsa.ipac.caltech.edu... 134.4.54.110
Connecting to irsa.ipac.caltech.edu|134.4.54.110|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]
Saving to: `out.txt'
```

```
[      <=>          ] 2,333,672  283K/s  in 8.0s
```

```
2014-05-01 16:43:22 (283 KB/s) - `out.txt' saved [2333672]
```

```
humu%
```

Let's open the file and see what we got. It is an ascii IPAC table file. There are optional parameters that will return different formats. Here we use the classic unix "nedit" editor, which easily handles files of arbitrary width.



in_row_id	in_ra	in_dec	expid	obsdate	crval1	crval2
long	double	double	long	char	double	double
	deg	deg			deg	deg
			null	null	null	null
1	148.96968699999999	69.6793830000000001	208515	2011-11-08 08:27:46.053000	148.8293743	69.3554553
1	148.96968699999999	69.6793830000000001	209605	2011-11-16 09:21:52.303000	148.8291624	69.3555071
1	148.96968699999999	69.6793830000000001	193104	2011-09-15 12:09:45.407000	148.8293164	69.3555110
1	148.96968699999999	69.6793830000000001	293740	2012-11-26 08:13:09.620000	148.8291781	69.3555448
1	148.96968699999999	69.6793830000000001	195284	2011-09-22 12:23:48.356000	148.8293793	69.3555933
1	148.96968699999999	69.6793830000000001	255299	2012-06-21 05:16:52.433000	148.8292010	69.3555986
1	148.96968699999999	69.6793830000000001	287002	2012-11-01 09:12:42.825000	148.8292405	69.3556050
1	148.96968699999999	69.6793830000000001	293051	2012-11-24 08:27:02.671000	148.8292038	69.3556096
1	148.96968699999999	69.6793830000000001	255075	2012-06-19 05:34:48.133000	148.8292254	69.3556132
1	148.96968699999999	69.6793830000000001	115032	2010-10-03 11:42:51.034000	148.8292894	69.3556184
1	148.96968699999999	69.6793830000000001	208603	2011-11-08 08:22:51.403000	148.8290938	69.3556210
1	148.96968699999999	69.6793830000000001	205873	2011-10-27 09:27:00.703000	148.8294747	69.3556252
1	148.96968699999999	69.6793830000000001	191423	2011-09-09 12:18:42.157000	148.8293929	69.3556273
1	148.96968699999999	69.6793830000000001	67548	2010-03-29 03:45:14.051000	148.8292899	69.3556378
1	148.96968699999999	69.6793830000000001	206435	2011-10-29 09:00:19.303000	148.8296102	69.3556425
1	148.96968699999999	69.6793830000000001	194719	2011-09-20 11:40:23.807000	148.8293183	69.3556520
1	148.96968699999999	69.6793830000000001	297989	2012-12-20 06:20:16.570000	148.8291066	69.3556768
1	148.96968699999999	69.6793830000000001	211581	2011-11-26 07:52:35.053000	148.8293833	69.3556911
1	148.96968699999999	69.6793830000000001	149555	2011-02-14 04:18:33.723000	148.8292894	69.3556934
1	148.96968699999999	69.6793830000000001	72400	2010-04-25 07:29:50.046000	148.8293238	69.3556945
1	148.96968699999999	69.6793830000000001	144109	2011-01-25 12:00:55.026000	148.8293348	69.3556989
1	148.96968699999999	69.6793830000000001	141270	2011-01-16 07:25:14.577000	148.8292501	69.3557003
1	148.96968699999999	69.6793830000000001	130648	2010-11-30 09:17:18.581000	148.8293033	69.3557033
1	148.96968699999999	69.6793830000000001	57668	2010-01-31 08:18:52.505000	148.8293235	69.3557068
1	148.96968699999999	69.6793830000000001	55696	2010-01-11 08:10:48.655000	148.8293354	69.3557074

Note that occasionally you'll see some strange metadata – this is usually from the first year of operations and the camera software was not yet finalized.

Parsing that file is an exercise left to the reader. You can read it with shell tools (e.g. sed, awk), with IDL using “readcol”, or whatever tools you are used to. Let’s assume at the moment you used your eyes and brain, and decided you wanted the data from the first entry. Scrolling way over to the right, you’ll find the following columns:

- pfilename – this is the product name for the intensity or flux image. The “p” stands for “primary”.
- afilename1 – this is the mask file associated with the primary filename. That “a” stands for “ancillary”.
- afilename3 – this is the sextractor aperture photometry catalog, in FITS binary format.

For the first entry, the pfilename is:

proc/2011/11/08/f2/c2/p5/v2/PTF_201111083526_i_p_scie_t082746_u009907441_f02_p100037_c02.fits

Check the website for more information on the filename convention.

To actually retrieve the file, you first take the string

`http://irsa.ipac.caltech.edu/ibe/data/ptf/images/level1/`

And append to it the full filename

`proc/2011/11/08/f2/c2/p5/v2/PTF_201111083526_i_p_scie_t082746_u009907441_f02_p100037_c02.fits`

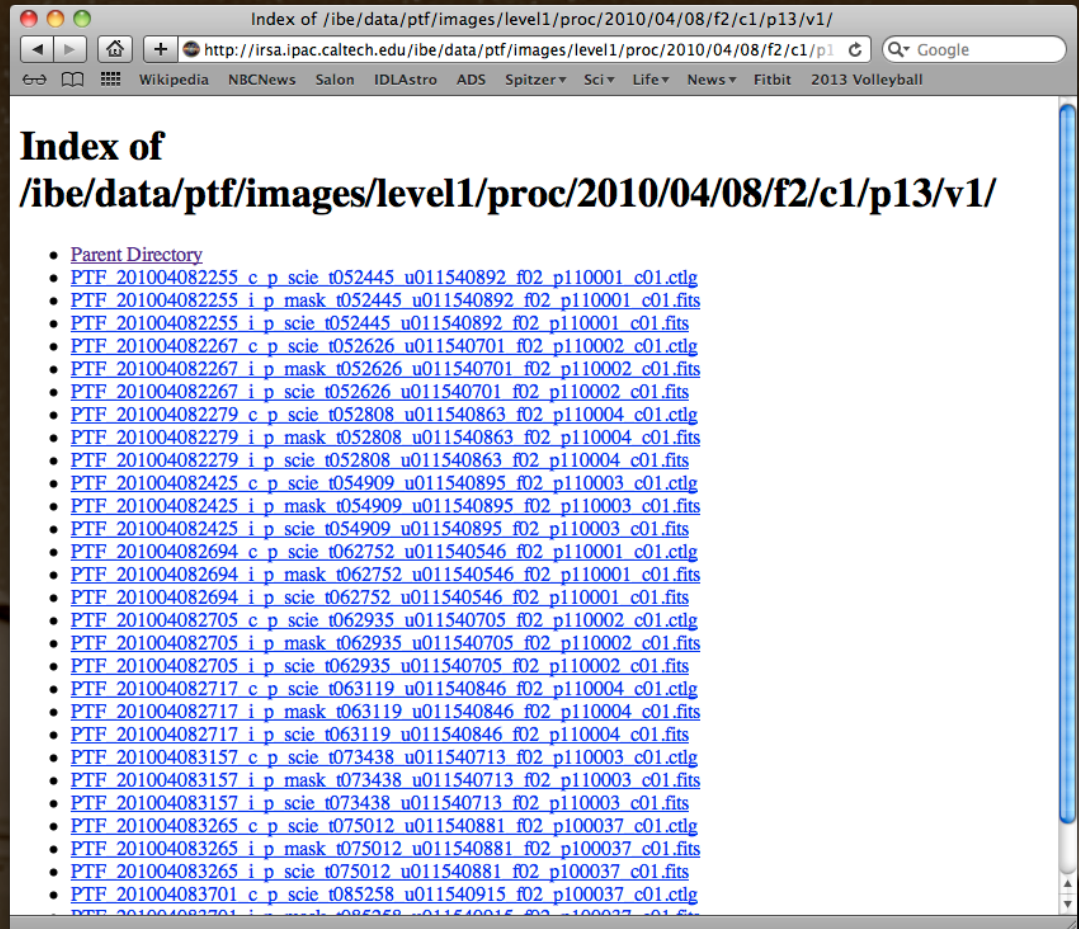
So you now execute:

```
humu% wget "http://irsa.ipac.caltech.edu/ibe/data/ptf/images/level1/proc/2011/11/08/f2/c2/p5/v2/PTF_201111083526_i_p_scie_t082746_u009907441_f02_p100037_c02.fits"
--2014-05-01 17:10:32-- http://irsa.ipac.caltech.edu/ibe/data/ptf/images/level1/proc/2011/11/08/f2/c2/p5/v2/PTF_201111083526_i_p_scie_t082746_u009907441_f02_p100037_c02.fits
Resolving irsa.ipac.caltech.edu... 134.4.54.110
Connecting to irsa.ipac.caltech.edu|134.4.54.110|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33586560 (32M) [application/octet-stream]
Saving to: `PTF_201111083526_i_p_scie_t082746_u009907441_f02_p100037_c02.fits'

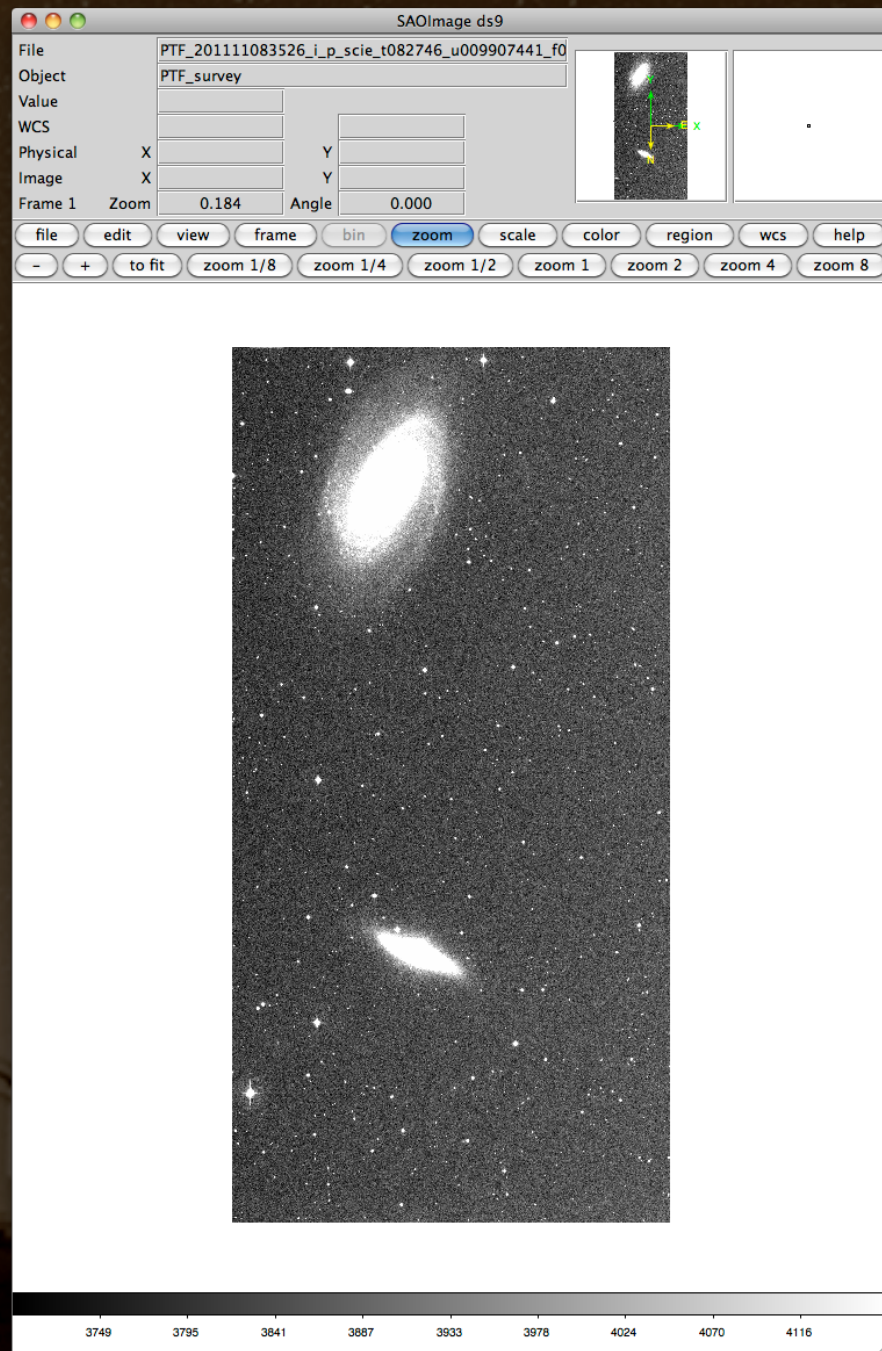
100%[=====>] 33,586,560 40.8M/s  in 0.8s

2014-05-01 17:10:34 (40.8 MB/s) -
`PTF_201111083526_i_p_scie_t082746_u009907441_f02_p100037_c02.fits' saved [33586560/33586560]
```

If you're not sure you have structured your URL correctly, you can actually "walk" a virtual directory tree to see the files you are trying to get. Note that this is account-aware, so you will only see the files actually available to you.



And there's the image we just downloaded. You retrieve the mask and catalog files exactly the same way, only you substitute their full filenames for the image we just used.



More Advanced Retrieval Options: Thumbnails

Let's try something a little more sophisticated. PTF images are pretty large, and often you just want to see the region of interest. In the following example, we've:

- Requested only a small thumbnail using the “?center=148.97,69.68&size=600arcsec” option.
- Turned off the thumbnail default file compression with “&gzip=false”.
- Renamed the output file to something sensible.

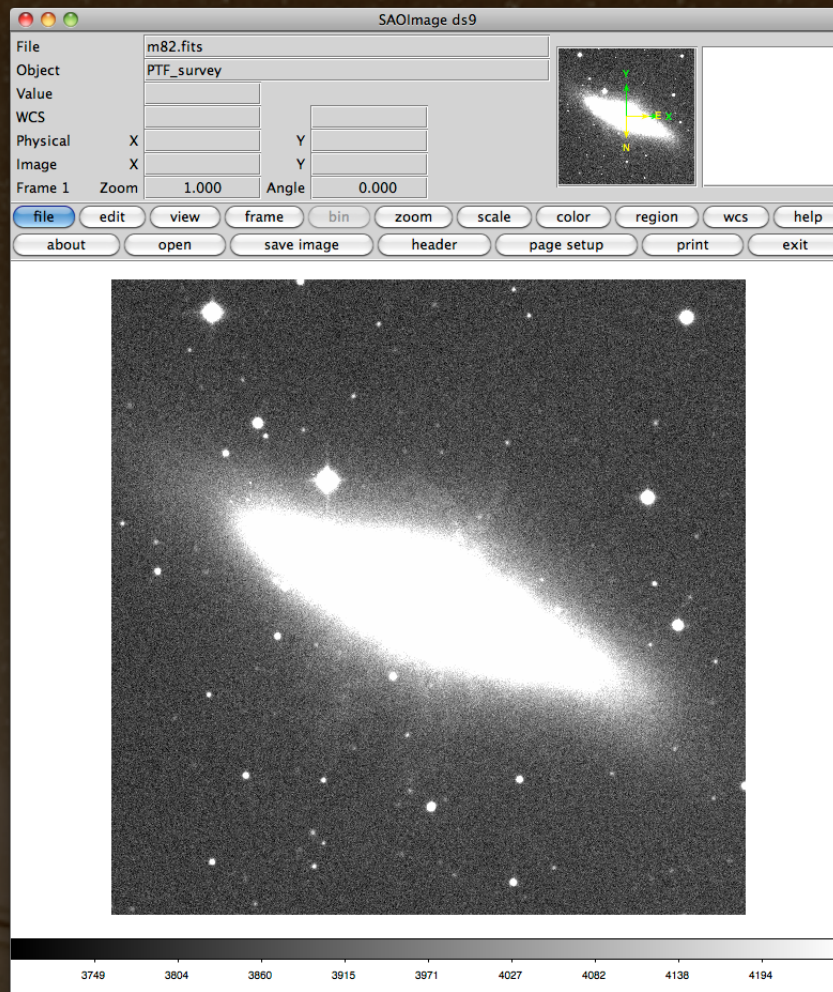
```
humu% wget "http://irsa.ipac.caltech.edu/ibe/data/ptf/images/level1/proc/2011/11/08/f2/c2/p5/v2/PTF_201111083526_i_p_scie_t082746_u009907441_f02_p100037_c02.fits?center=148.97,69.68&size=600arcsec&gzip=false" -O m82.fits
--2014-05-02 14:00:47-- http://irsa.ipac.caltech.edu/ibe/data/ptf/images/level1/proc/2011/11/08/f2/c2/p5/v2/PTF_201111083526_i_p_scie_t082746_u009907441_f02_p100037_c02.fits?center=148.97,69.68&size=600arcsec&gzip=false
Resolving irsa.ipac.caltech.edu... 134.4.54.110
Connecting to irsa.ipac.caltech.edu|134.4.54.110|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1442880 (1.4M) [application/fits]
Saving to: `m82.fits'

100%[=====>] 1,442,880  --.-K/s  in 0.02s

2014-05-02 14:00:47 (61.3 MB/s) - `m82.fits' saved [1442880/1442880]

humu%
```

And here's the output from
the thumbnail request.



Search by PTF Field ID

Here we perform a search by PTF FieldID using the “where” clause. This is particularly useful when finding data from the PTF first data release. Because this release is limited to a subset of areas around the sky, finding this data by field id (which we supply on the data release web pages) will ensure that you get all the data.

```
humu% wget "http://irsa.ipac.caltech.edu/ibe/search/ptf/images/level1?where=ptffield=4808" -O out.txt
--2014-05-02 14:38:14-- http://irsa.ipac.caltech.edu/ibe/search/ptf/images/level1?where=ptffield=4808
Resolving irsa.ipac.caltech.edu... 134.4.54.110
Connecting to irsa.ipac.caltech.edu|134.4.54.110|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]
Saving to: `out.txt'

[ <=> ] 35,154 --K/s in 0.04s

2014-05-02 14:38:15 (954 KB/s) - `out.txt' saved [35154]

humu%
```

out.txt - /Users/jason/Desktop/

expid	obsdate	crval1	crval2	filter	ccdid	ptffld	seeing	airmass	moonillf	moonesb	ph
long	char	double	double	char	long	long	double	double	double	double	
null	null	deg	deg		null	arcsec					
127594	2010-11-14 04:26:29.181000	1.1848435	53.2393005	R	1	4808	2.81	1.06	0.55	-1.35	
127550	2010-11-14 05:29:25.930000	5.0510475	51.9823967	R	11	4808	2.60	1.08	0.55	-1.37	
127550	2010-11-14 05:29:25.930000	2.1332479	53.5261220	R	2	4808	2.60	1.08	0.55	-1.37	
127550	2010-11-14 05:29:25.930000	1.1848697	53.2389299	R	1	4808	2.60	1.08	0.55	-1.37	
127550	2010-11-14 05:29:25.930000	0.2325093	53.2096191	R	0	4808	2.60	1.08	0.55	-1.37	
127594	2010-11-14 04:26:29.181000	0.7162224	52.5063097	R	6	4808	2.81	1.06	0.55	-1.35	
127550	2010-11-14 05:29:25.930000	5.5050368	53.5314676	R	5	4808	2.60	1.08	0.55	-1.37	
127594	2010-11-14 04:26:29.181000	2.1332198	53.5258941	R	2	4808	2.81	1.06	0.55	-1.35	
127594	2010-11-14 04:26:29.181000	5.5050267	53.5313809	R	5	4808	2.81	1.06	0.55	-1.35	
127594	2010-11-14 04:26:29.181000	0.3146904	53.1225898	R	0	4808	2.81	1.06	0.55	-1.35	
127550	2010-11-14 05:29:25.930000	1.4485193	52.0658046	R	7	4808	2.60	1.08	0.55	-1.37	
127550	2010-11-14 05:29:25.930000	0.7162121	52.5063603	R	6	4808	2.60	1.08	0.55	-1.37	
127594	2010-11-14 04:26:29.181000	1.4485129	52.0658763	R	7	4808	2.81	1.06	0.55	-1.35	
127594	2010-11-14 04:26:29.181000	5.0510653	51.9827781	R	11	4808	2.81	1.06	0.55	-1.35	

Here we see that the IBE search (above) has returned the same products as the GUI (below).

Search by PTF Field ID ID(s)=4808; Product Level=1

Level-1

Prepare Download 1 of 1 (1 - 14 of 14) as Text Save Filters

expid	obsdate	crval1	crval2	filter	ccdid	ptffld	seeing	airmass	moonillf	moonesb
☐ 127594	2010-11-14 04:26:29.181000	0.3147	53.1226	R	0	4808	2.8100	1.0600	0.5500	-1.3500
☐ 127594	2010-11-14 04:26:29.181000	1.1848	53.2393	R	1	4808	2.8100	1.0600	0.5500	-1.3500
☐ 127594	2010-11-14 04:26:29.181000	2.1332	53.5259	R	2	4808	2.8100	1.0600	0.5500	-1.3500
☐ 127594	2010-11-14 04:26:29.181000	5.5050	53.5314	R	5	4808	2.8100	1.0600	0.5500	-1.3500
☐ 127594	2010-11-14 04:26:29.181000	0.7162	52.5063	R	6	4808	2.8100	1.0600	0.5500	-1.3500
☐ 127594	2010-11-14 04:26:29.181000	1.4485	52.0659	R	7	4808	2.8100	1.0600	0.5500	-1.3500
☐ 127594	2010-11-14 04:26:29.181000	5.0511	51.9828	R	11	4808	2.8100	1.0600	0.5500	-1.3500
☐ 127550	2010-11-14 05:29:25.930000	0.2325	53.2096	R	0	4808	2.6000	1.0800	0.5500	-1.3700
☐ 127550	2010-11-14 05:29:25.930000	1.1849	53.2389	R	1	4808	2.6000	1.0800	0.5500	-1.3700
☐ 127550	2010-11-14 05:29:25.930000	2.1332	53.5261	R	2	4808	2.6000	1.0800	0.5500	-1.3700
☐ 127550	2010-11-14 05:29:25.930000	5.5050	53.5315	R	5	4808	2.6000	1.0800	0.5500	-1.3700
☐ 127550	2010-11-14 05:29:25.930000	0.7162	52.5064	R	6	4808	2.6000	1.0800	0.5500	-1.3700
☐ 127550	2010-11-14 05:29:25.930000	1.4485	52.0658	R	7	4808	2.6000	1.0800	0.5500	-1.3700
☐ 127550	2010-11-14 05:29:25.930000	5.0510	51.9824	R	11	4808	2.6000	1.0800	0.5500	-1.3700

And that's it! This document will be updated with more examples as they arise. Questions about the mechanics of the archive should be directed to the helpdesk at IRSA. Questions about PTF and it's products should be directed to the PTF/iPTF project.

