

## **Better ABsolute Scanner (BABS)**

### **Disclaimer**

READ THE FLAMING MANUAL  
YOUR MILEAGE MAY VARY  
RESULTS NOT TYPICAL  
USE AT YOUR OWN RISK  
NO WARRANTIES EXPRESS OR IMPLIED  
BACK UP YOUR UNIVERSE BEFORE RESCANNING WITH THIS SCANNER

This scanner fixes problems I encountered with absolute numbered series. It disables certain features of the original scanner, but leaves most of the original scanner's features in place. See "What Changed (Short Version)" or "What Changed (Details)" for more information.

Backup your library before rescanning with this scanner. If your library ends up worse for wear, you can recover it from your backup. Information on backing up the library can be found in the Plex forums. You can use Google with site:forums.plexapp.com" before your search term to limit your search to the Plex forums. It's a lot easier than using the forum search.

Better yet, create a section to use this scanner to test it. That way, you can just delete the section if the scanner doesn't work out for you and the rest of your library (your "production" sections) are none the worse for wear.

If you find a problem, I may be willing to fix it, especially if it's something I can see possibly affecting my collection. :) I haven't tested every scenario, so I'm assuming there might be something I messed up that I missed. Then again, it may be a problem in the original Plex Series Scanner code that I started with.

THIS SCANNER IS FOR PLEX MEDIA SERVER 0.9.3.5.  
IT MAY NOT WORK FOR OTHER VERSIONS OF PLEX.

### **The Problem**

I set up Plex Media Server 0.9.3.5 (PMS) on an Ubuntu 10.04 box that is also my NAS so I could stream my media to a Roku box using the Roku channel version 1.4. I haven't been able to find anything else that can do that, so PMS is my solution. I've generally been pleased with the newest versions of PMS and the Roku channel so far. Huge thanks and kudos to everyone who worked on making this work.

There has been one fly in the ointment. I have a lot of series (TV shows) that are absolute numbered. This is common in the land of anime, where both fansubs and the shows themselves rarely number episodes in terms of seasons. You might get variant show names for different seasons, but TVDB doesn't handle that properly. Generally, the Plex Series Scanner (PSS) did an okay job of scanning my stuff in, except series with over 99 episodes, and series with numbers in the name like Galaxy Express 999 and 07-Ghost.

If an absolute numbered series has over 99 episodes, PSS will split it into seasons of 100 episodes. Which wouldn't be so bad, except episode 100 doesn't show up as episode 100, it shows up as episode 00. There was also some hinky stuff going on with metadata or possibly file matching for Season 2+. I didn't explore it too deeply. It was obvious PSS had hosed it up. I hypothesized I could rename

everything use SxxExx, but that would be a lot of work and it would make it much harder to track what I had vs. what I didn't have.

I saw the absolute ordering scanner (tweaked, v2, no less) Raindancing posted on the forums (<http://forums.plexapp.com/index.php/topic/21625-absolute-order-agent/>), but it didn't solve my problems. I also couldn't figure out how to install his agent (a plexapp file) in Linux.

I also had problems with series like Galaxy Express 999 and 07-Ghost. In the case of GE999, I saw one season (Season 9) and one episode (episode 99). Obvious what's going on there, right? 07-Ghost saw one episode. That's right, episode 7. So series with numbers in the name messed up the scanner. I could force the issue by renaming everything use SxxExx, but that would be a lot of work.

Posting on the forum got the usual, "Just name it all SxxExx," response, even though I'd made it clear that wasn't an option. I was further annoyed by the fact that the Plex wiki says Plex supports absolute ordering for series (right there in the Naming Guide, after the SxxExx and date-based naming sections). Well, since it wasn't working, it was time to dust off the little bit of Python I knew and read up on regular expressions (regexs) and figure out how PSS worked. I also looked at the absolute order scanner, which yielded a few valuable clues (thanks again to Raindancing).

The result of all this is a Better Absolute Scanner, or BABS for short. Corny, but I made it work, so I'm entitled to name it what I want. If the Plex guys want to take the ideas from this scanner and incorporate them into the PSS or an official absolute scanner, please do.

This still doesn't fix the fact that the TVDB agent doesn't know how to ask for absolute ordered data. I may get excited about that and see if it's in the Python somewhere and try to fix it too. Now that I know PMS keeps a copy of the base Plug-ins in `/usr/lib/plexmediaserver/Resources/Plug-ins`, I might be able to fake it out so it won't overwrite any code changes I might make.

### **Useful Tidbits**

Linux-specific documentation for PMS is sparse (to be generous). I had to figure out all kinds of stuff myself or by stringing together bits of information from various threads on the forums. It took me over an hour just to figure out how to get output from the scanner. Here are some of the most useful pieces of information I found. Remember I'm running Ubuntu 10.04. If you have a different distro, your mileage may vary.

For Ubuntu, the PMS Library folder (aka `~/Library`) is `/var/lib/plexmediaserver/Library`.

For Ubuntu, the PMS code is in `/usr/lib/plexmediaserver`. Note the pretty symmetry.

You can run the scanner from the console by copying `start.sh` in the code directory to `scan.sh` and editing `scan.sh` to run Plex Media Scanner instead of Plex Media Sever.

Scanner Python code (and a boatload of other things) lives beneath the Library folder. Official scanners are in `~/Library/Application Support/Plex Media Server/Plug-ins/Scanners.bundle` (a directory). User-built scanners go in `~/Library/Application Support/Plex Media Server/Scanners`. If you try to put them in Plug-ins, Plex will be angry and will delete them when it realizes what you've done.

The `~/Library/Application Support/Plex Media Server/Scanners/Series` (and all other specific scanner type directories) are NOT created by default. You must create them yourself with `mkdir`.

You must be the plex user (su plex) to run the scanner at the console (manually). If you try to run the scanner as root, the scanner will fail saying it can't find various libraries. You could put a bunch of symlinks in /lib32, but why expose all those libraries (and write all those link commands) when it's just a matter of becoming the plex user to get the right pathing?

The only way to see any output from the scanner code so you know what's going on is to use Python's print statement and run the scanner manually. Be sure you tag your lines so you can grep them (or find them) easily.

The following command will touch all the files and directories in a given directory and its subdirectories so the scanner will think they've been updated and will rescan them -- really useful if you need to force a rescan while testing from the console:

```
find * -exec touch {} \;
```

The debugging messages in this scanner's console output can help you identify duplicate file. If you're running the scanner from the command line, you can redirect your output to a file and grep the file for the\_episode to get a list of all episodes. The line of text you get back will include the value of show for each line (the name of the directory where the file is stored).

So, say you want to find duplicate files in a section. Do something like this:

1) In the directory containing all the subdirectories for the section (you do have everything neatly organized, right?) run the following Linux command or your OS's equivalent.

```
find * -exec touch {} \;
```

2) From the Plex code directory, run the scanner from the section redirecting output to a file. (Remember you must be the plex user.)

```
./scan.sh >/var/lib/plexmediaserver/scan.out
```

3) When the scan finishes, get the file into a spreadsheet. You may need to move it somewhere else first.

4) Use the spreadsheet's text parsing capabilities (I know Excel and LibreOffice can do this), and split out the episode number and the show name. You should end up with something like the table below.

Show	Episode
showname	1
showname	21
anothershow	92
anothershow	84

5) Sort by show and episode. In the next column over, put a 0 in the first row. In the second row, put the formula shown below. Then copy/paste that formula to the rest of the rows. It should automatically change to reflect the correct rows for comparison.

```
=if(b1=b2, 1, 0)
```

6) Now you can filter by the formula and find all 1's. These are you duplicate shows and episodes.

Assume you find show "showname" episode "2" is a duplicate. Open up your scanner output and search for

```
show = | showname | episode = 2
```

Every time you find it, look a few lines above it until you see the original file name. Now you know which files are duplicates and can choose which to keep.

Or maybe that will help you find a bug in the scanner. :)

### **What Changed (Short Version)**

I added a bunch of print statements you'll only see if you run the scanner from the command line.

If you use a 3 or 4 digit number to designate season (as 1-2 digits) and episode (as 2 digits), this scanner will NOT read your numbers correctly. You'll need to renumber them absolute or SxxExx or put them in a separate section that uses a different scanner.

If you use a tag to designate the episode (e.g., Blah Blah Ep 234), the only tags this scanner will recognize are "ep" and "episode" (not case sensitive) before the episode number. If you were using something else, you'll need to change it to ep, or episode. Based on a quick check of TVDB, this change fixes more problems that it seems likely to create. The tag must be preceded by a separator or be the first thing in the file name.

If you have an absolute numbered series, and the series name is only a number with no words or other characters, BABS will not process an episode with the same number correctly.

For example, if you have a series named "24" and an episode file named "24\_24.mkv", the scanner will not pick it up correctly. I recommend you rename the ONE problem episode to something like "24\_ep24.mkv" instead.

If your series name has a number in the name, for example, Galaxy Express 999, be sure the containing folder name will match the series name in the file. For example, if the file is Galaxy\_Express\_999\_01.mkv, you should name your folder Galaxy Express 999 or Galaxy\_Express\_999. If the name is GalaxyExpress999 - 01.mkv, name your folder GalaxyExpress999. CASE SENSITIVE.

### **What Changed (Details)**

Most of the code I added to the scanner was print statements so I could decipher where it was going wrong with my files. All my print output begins with BABS:. This makes it easy to find my output and identify what the scanner is doing with a given file. This print output will not show up anywhere except the console for manual scanner runs. See "Useful Tidbits" for how to run the scanner manually in Linux.

All the changes I made are tagged with comments that begin with BABS to make them easy to find.

The regex in episode\_regexps that was tagged "#.602." causes major problems for absolute numbered series. It looks for a string of 1-2 digits followed by 2 more digits. The first 1-2 digits is the season. The second is the episode. This causes problems for long series. If you have episodes numbered 001, 002, etc., they'll end up in Season 0. When you hit episode 100, it becomes episode 00 in Season 1. But what really kills you is when no one guessed the series would run that long and started numbering 01, 02,

etc. Now Episodes 01-99 are the equivalent of S01E01-E99, and Episodes 101-199 are S01E01-E99. Obviously, this is a problem. I concluded that this regex is the bane of absolute ordered series everywhere, especially long-running absolute ordered series.

I noted that Raindancing's absolute ordering scanner deactivated this regex, so I just removed it completely. I only found one series in my collection that used this numbering scheme for part of the episodes. Everything else was absolute ordered or season-based. I just renumbered that one series (<100 episodes) by hand. A small price to pay compared to fixing thousands of other episodes.

After fixing that, I found another problem. The regexes in `just_episode_regexs` were having problems. Specifically, Galaxy Express 999 was kicking out on the "Blah Blah ep234" regex. This seemed odd, so I dug into the Python regex documentation and found that the first term is, in my opinion, NOT what the comment advertises. The first term will look for an e followed by 0-n letters followed by 0-n separators (space, dot, dash, underscore) followed by a 2-3 digit episode number. So in the case of Galaxy Express 999, the underlined text there fits the regex. I recoded the regex so it will look for either e, ep, or episode instead of e followed by 0-n letters. This may create language support issues for some languages, but I doubt all language begin the word for episode with an e, so they existed before.

Not satisfied that these changes would handle everything, I made one more change. Late in the scanner process, when it's down to the `just_episode_regexs` list as the only option left, this scanner attempts to strip the show name out of the episode name twice -- once before doing a `CleanName` call and once after. It does this twice because the `CleanName` call could mess up the name so it won't match the show name or could fix it so it does, so either place could be right.

The potential problem this creates is if you have a show whose name is only a number, and whose numeric name matches an episode number. See "What Changed (Short Version)" for some examples of what this means for a file name.

The fix for this is fairly simple. For the one episode of an absolute numbered series that duplicates the show name, put an "e" or "ep" in front of the number so it hits the "ep234" case described above. This only affects absolute numbered series.

SxxExx and all the other flavors of season/episode based processing happen before this, so shouldn't hit this case. In fact, at this point in the scanner, we're down to the "it must be an episode-only number" cases.

PSS (which I copied to start with) has comments that suggest it specifically deals with the traditional "day x hour y-z" numbering used for the series "24," and that happens before this so should not be affected.

After some more testing, I found that the "Blah Blah Ep234" regex was still causing problems with some oddly named files. Specifically, if the checksum is in parentheses, clean name will remove the parentheses, but not the checksum. If something ending in an "e" is left immediately before the checksum after cleaning, and the checksum begins with a digit, the regex was picking up the first part of the checksum as the episode number. So file `blah_123_name_(456checksum).mkv` was cleaned to `blah 123 name 456checksum` and the part underlined there was matching the e-only form of the regex. I tweaked the regex again to allow only ep and episode and to require the tag to either be at the beginning of the name or preceded by a separator (to deal with words ending in "ep").

And that is it. Except for the changes described above, this is the stock PSS scanner.

### Installation and Use

IF YOU HAVEN'T READ THE DISCLAIMER AND THE SHORT VERSION OF WHAT CHANGED, DO NOT INSTALL THIS SCANNER. IT ISN'T LIKE THAT'S A LOT TO READ.

To install the scanner, put the BABS.py file in the following folder.

~/Library/Application Support/Plex Media Server/Scanners/Series

NOTE:

On Ubuntu 10.04, that translates to

/var/lib/plexmediaserver/Library/Application Support/Plex Media Server/Scanners/Series

On Windows 7, that translates to

C:\Users\yourusername\AppData\Local\Plex Media Server\Scanners\Series

NOTE: replace yourusername in the path above with your Windows user name.

I don't have a Mac, so can't verify, but the Plex web site says it is

~/Library/Application Support/Plex Media Server/Scanners/Series

YOU MAY NEED TO CREATE THE SERIES SUBDIRECTORY BY HAND.

I STRONGLY RECOMMEND YOU TEST THE SCANNER IN A NEW (TEST) SECTION. IF YOU LIKE IT, REPLACE YOUR PRODUCTION SECTION.

EITHER WAY, YOU SHOULD BACK UP YOUR LIBRARY, JUST IN CASE.

To use the scanner, create a TV Shows section and select BABS as your media scanner option. See screen shot. Then let BABS scan your media.

To use the scanner on an existing TV Shows section, edit the section settings and change the Media Scanner to BABS. Force the section to be rescanned. (Usually I just touch all the files with the command shown in "Useful Tidbits," but there's some kind of shift-click or ctrl-click on the refresh button that supposedly does it too.)

Section type: TV Shows

Section name: Test

Media scanner: BABS

Primary metadata agent: TheTVDB

Language: English

Add the folders containing media for this section below. Any media files found inside these folders will be scanned into your library using the above settings.

Test

\mnt1rd0\m\Video

+ - Drag folders into the list, or click the '+' button to browse.

Cancel Update Section