

Jozef Jarosciak

*Strive not to be a success, but rather to be of value.*

Converting UTZOO-Wiseman Usenet Tapes to Website with PostgreSQL backend using Python 3.8

October 7, 2020 by joe0



Recently, I came across a resource that allowed me to download the entire collection of UTZOO NetNews Archive of the earliest USENET posts. These were essentially the earliest available discussions posted to the Internet by people working at various Universities who were already connected to the Internet. There were approximately 2.1 million posts in these archives created between Feb 1981 and June of 1991. This article describes the journey of converting those tapes into fully searchable PostgreSQL database and later also into the usenetarchives.com website.

How & Why

Until 2001, these early Usenet discussions were considered being lost, but miraculously [Henry Spencer](#) from the University of Toronto, Department of Zoology

was backing it up onto magnetic tapes and kept them stored for all these years (apparently at a great cost).



H. Spencer had altogether 141 of these magnetic tapes, but there were of no use, so eventually, him and a couple of motivated people such as David Wiseman (who dragged 141 tapes back and forth in his a pickup truck), Lance Bailey, Bruce Jones, Bob Webber, Brewster Kahle, and Sue Thielen; embarked on a process of converting all of these tapes into

the regular format, accessible to everyone.

And that's the copy I downloaded. What a treasure, right?

Well, not so fast, once I unzipped the data, I realized that the TGZ format contains literally millions of small text files (each post in its own file). While it was certainly nice to have, it wasn't something that I or anyone else could read. Certainly not in a forum like discussion format. It wasn't obvious which post is the one that starts the discussion or which ones are the replies to the thread. And forget about searching through these files, that was utterly not possible. Just to put things into perspective, it took me over 5 hours to un-tar the archives.

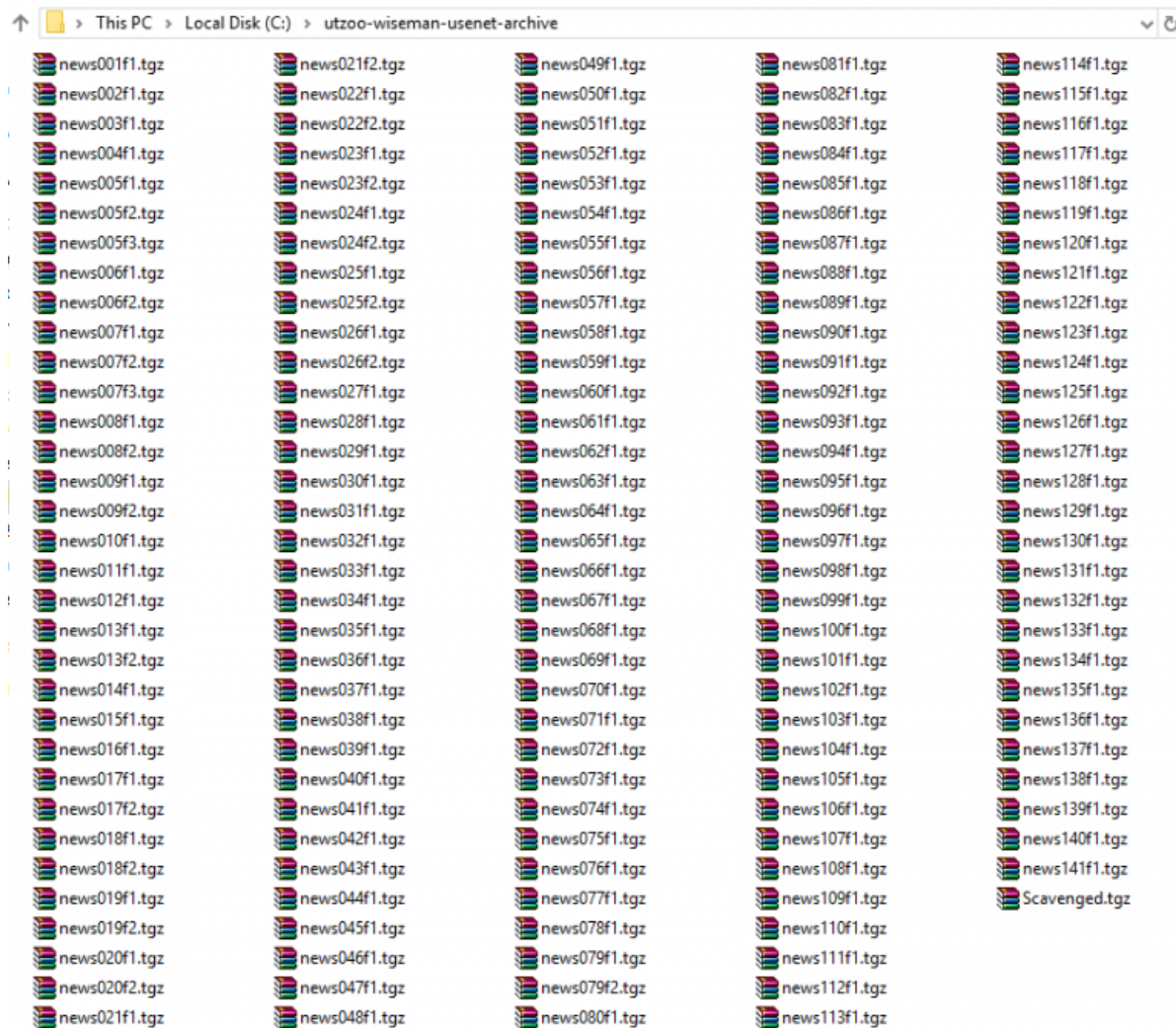
That said, it didn't take long for me to decide to develop a Python-based converter that would allow me to convert the entire collection from millions of flat files into a fully searchable PostgreSQL database. The following post talks about the process and also includes the Python code of the solution released as open source.

Archives

The UTZOO Usenet archive can be downloaded here:

- <http://www.skrenta.com/rt/utzoo-usenet/>
- <http://shiftright.com/mirrors/utzoo-usenet/>
- <https://ipfs.io/ipfs/QmTo7fRxpXwxv6Uw4TAAyLWEmvugKaggrHskNBTRHzWcA/>
- Or using this torrent: [utzoo-wiseman-usenet-archive_archive](#)

Once downloaded you'll see that archive contains 161 x TAR Archive files. It looks like this:



So, I grabbed a copy of the 7-Zip archiver from <https://www.7-zip.org> and started decompressing the files.

I ended up with over **2,104,828** flat text files in **56,988** folders, which was the entire copy of Henry Spencer's Usenet archive.

For those who like numbers, here is each Utzoo tape along with its size, number of files and folders:

Name	Size	Allocated	Files ▼	Folders	% of Parent
E:\Usenet\Utzoo\	4.9 GB	9.1 GB	2,104,828	56,988	100.0 %
news134f1	109.0 MB	211.6 MB	48,041	1,411	2.3 %
news130f1	110.2 MB	211.1 MB	47,292	1,294	2.2 %
news137f1	106.4 MB	208.8 MB	46,840	1,447	2.2 %
news131f1	104.9 MB	204.4 MB	46,548	1,304	2.2 %
news138f1	109.8 MB	208.4 MB	45,457	1,441	2.2 %
news138f1	106.7 MB	202.8 MB	44,860	1,454	2.1 %
news129f1	107.5 MB	201.5 MB	44,285	1,241	2.1 %
news141f1	105.1 MB	199.3 MB	44,011	1,479	2.1 %
news140f1	106.1 MB	195.8 MB	42,163	1,461	2.0 %

news139f1	108.4 MB	195.7 MB	41,914	1,475	2.0 %
news135f1	102.6 MB	190.2 MB	40,636	1,390	1.9 %
news128f1	106.3 MB	191.8 MB	40,228	1,238	1.9 %
news127f1	109.1 MB	193.0 MB	39,656	1,227	1.9 %
news132f1	92.1 MB	172.4 MB	37,627	1,275	1.8 %
news133f1	102.2 MB	181.2 MB	37,099	1,330	1.8 %
news118f1	77.9 MB	143.9 MB	31,352	962	1.5 %
news087f1	74.9 MB	138.6 MB	30,142	744	1.4 %
news120f1	66.0 MB	129.6 MB	30,138	927	1.4 %
Scavenged	66.6 MB	127.7 MB	29,680	769	1.4 %
news121f1	66.0 MB	126.8 MB	28,897	842	1.4 %
news115f1	55.7 MB	107.1 MB	24,004	656	1.1 %
news117f1	48.8 MB	92.5 MB	20,679	611	1.0 %
news116f1	51.4 MB	94.2 MB	20,198	608	1.0 %
news122f1	38.2 MB	72.5 MB	16,283	375	0.8 %
news011f1	24.8 MB	57.7 MB	16,198	210	0.8 %
news126f1	40.1 MB	74.3 MB	16,045	449	0.8 %
news119f1	31.4 MB	60.7 MB	13,868	493	0.7 %
news017f1	25.5 MB	54.6 MB	13,789	215	0.7 %
news012f1	20.6 MB	48.3 MB	13,116	199	0.6 %
news030f1	26.7 MB	54.7 MB	12,999	233	0.6 %
news029f1	25.2 MB	53.2 MB	12,996	234	0.6 %
news014f1	21.6 MB	48.8 MB	12,952	202	0.6 %
news028f1	27.4 MB	54.9 MB	12,851	240	0.6 %
news039f1	27.3 MB	54.0 MB	12,432	241	0.6 %
news032f1	25.9 MB	51.9 MB	12,226	230	0.6 %
news038f1	25.7 MB	51.9 MB	12,110	240	0.6 %
news034f1	24.9 MB	50.6 MB	12,100	245	0.6 %
news112f1	29.3 MB	54.2 MB	11,837	312	0.6 %
news033f1	25.9 MB	51.1 MB	11,814	248	0.6 %
news027f1	26.9 MB	52.0 MB	11,723	218	0.6 %
news031f1	25.5 MB	50.7 MB	11,706	224	0.6 %
news035f1	25.6 MB	50.2 MB	11,697	240	0.6 %
news110f1	25.1 MB	48.9 MB	11,629	311	0.6 %
news036f1	26.2 MB	51.2 MB	11,599	243	0.6 %
news002f1	12.9 MB	22.8 MB	11,522	503	0.5 %
news003f1	12.5 MB	23.2 MB	11,283	356	0.5 %
news109f1	24.3 MB	47.8 MB	11,237	300	0.5 %
news037f1	25.9 MB	50.3 MB	11,218	242	0.5 %
news098f1	25.6 MB	49.2 MB	11,175	265	0.5 %
news001f1	11.0 MB	18.5 MB	10,918	37	0.5 %
news040f1	26.5 MB	49.8 MB	10,909	240	0.5 %
news061f1	27.7 MB	51.2 MB	10,873	221	0.5 %
news099f1	25.0 MB	47.9 MB	10,864	275	0.5 %
news065f1	30.6 MB	52.3 MB	10,849	227	0.5 %
news113f1	26.5 MB	48.9 MB	10,808	312	0.5 %
news108f1	24.1 MB	46.5 MB	10,697	293	0.5 %
news064f1	27.1 MB	48.7 MB	10,676	236	0.5 %
news100f1	25.5 MB	48.3 MB	10,634	266	0.5 %
news091f1	24.0 MB	46.2 MB	10,605	257	0.5 %
news004f1	12.0 MB	22.2 MB	10,576	377	0.5 %
news094f1	26.1 MB	48.4 MB	10,554	258	0.5 %
news097f1	25.0 MB	47.3 MB	10,525	265	0.5 %
news058f1	28.8 MB	51.5 MB	10,522	165	0.5 %
news066f1	28.4 MB	49.5 MB	10,519	223	0.5 %
news046f1	27.9 MB	49.4 MB	10,416	210	0.5 %
news067f1	27.4 MB	48.1 MB	10,387	222	0.5 %
news093f1	24.4 MB	46.3 MB	10,352	250	0.5 %
news060f1	29.5 MB	51.7 MB	10,313	207	0.5 %
news125f1	27.5 MB	49.3 MB	10,301	296	0.5 %
news059f1	29.7 MB	52.0 MB	10,240	174	0.5 %
news055f1	29.0 MB	51.1 MB	10,216	156	0.5 %
news051f1	27.2 MB	49.3 MB	10,213	172	0.5 %
news020f1	21.5 MB	43.4 MB	10,131	222	0.5 %
news104f1	24.8 MB	45.7 MB	10,095	288	0.5 %
news114f1	24.7 MB	46.2 MB	10,073	288	0.5 %
news062f1	27.8 MB	48.7 MB	10,070	226	0.5 %
news123f1	26.8 MB	48.2 MB	10,065	273	0.5 %
news106f1	27.6 MB	48.5 MB	10,039	298	0.5 %
news101f1	25.2 MB	46.2 MB	9,991	266	0.5 %
news111f1	26.7 MB	47.4 MB	9,894	302	0.5 %
news068f1	28.3 MB	48.3 MB	9,890	229	0.5 %
news045f1	28.7 MB	49.3 MB	9,826	188	0.5 %
news010f1	14.4 MB	34.2 MB	9,795	201	0.5 %
news053f1	28.1 MB	48.9 MB	9,780	208	0.5 %

news069f1	27.6 MB	47.5 MB	9,780	238	0.5 %
news102f1	24.3 MB	44.6 MB	9,714	278	0.5 %
news050f1	29.5 MB	50.2 MB	9,712	169	0.5 %
news057f1	30.8 MB	51.4 MB	9,510	170	0.5 %
news009f2	14.4 MB	33.2 MB	9,508	190	0.5 %
news105f1	27.0 MB	46.8 MB	9,431	298	0.4 %
news018f1	17.7 MB	37.5 MB	9,430	213	0.4 %
news088f1	25.1 MB	44.9 MB	9,406	251	0.4 %
news103f1	26.8 MB	46.4 MB	9,348	292	0.4 %
news063f1	27.9 MB	46.3 MB	9,316	218	0.4 %
news054f1	27.9 MB	47.8 MB	9,274	189	0.4 %
news092f1	24.9 MB	44.0 MB	9,180	260	0.4 %
news015f1	15.4 MB	34.3 MB	9,163	191	0.4 %
news107f1	27.0 MB	45.9 MB	9,036	290	0.4 %
news044f1	28.8 MB	47.7 MB	9,003	169	0.4 %
news056f1	28.6 MB	47.7 MB	8,947	170	0.4 %
news089f1	25.2 MB	43.8 MB	8,945	251	0.4 %
news021f1	19.7 MB	38.8 MB	8,878	221	0.4 %
news090f1	24.8 MB	43.4 MB	8,861	240	0.4 %
news052f1	28.3 MB	47.2 MB	8,792	181	0.4 %
news042f1	21.8 MB	40.3 MB	8,715	83	0.4 %
news009f1	13.3 MB	30.1 MB	8,670	187	0.4 %
news013f2	14.3 MB	32.9 MB	8,642	96	0.4 %
news070f1	27.4 MB	44.8 MB	8,617	167	0.4 %
news047f1	23.7 MB	41.3 MB	8,286	253	0.4 %
news080f1	27.5 MB	44.6 MB	8,285	233	0.4 %
news049f1	29.1 MB	46.5 MB	8,222	165	0.4 %
news096f1	25.4 MB	42.5 MB	8,152	256	0.4 %
news022f1	14.6 MB	31.7 MB	8,053	199	0.4 %
news024f1	16.4 MB	33.5 MB	7,998	217	0.4 %
news083f1	18.9 MB	35.6 MB	7,949	214	0.4 %
news023f2	16.1 MB	33.1 MB	7,893	208	0.4 %
news008f1	11.5 MB	26.4 MB	7,890	183	0.4 %
news006f1	11.0 MB	24.2 MB	7,852	145	0.4 %
news048f1	24.2 MB	40.5 MB	7,752	229	0.4 %
news124f1	15.4 MB	31.9 MB	7,748	126	0.4 %
news079f2	23.5 MB	39.5 MB	7,731	215	0.4 %
news013f1	12.0 MB	27.5 MB	7,571	193	0.4 %
news016f1	14.2 MB	29.7 MB	7,555	197	0.4 %
news025f1	16.2 MB	32.3 MB	7,524	216	0.4 %
news006f2	11.8 MB	25.5 MB	7,433	147	0.4 %
news026f1	13.9 MB	29.6 MB	7,380	216	0.4 %
news086f1	18.2 MB	33.9 MB	7,344	229	0.3 %
news019f1	14.0 MB	29.6 MB	7,335	206	0.3 %
news008f2	11.7 MB	25.9 MB	7,332	189	0.3 %
news085f1	18.4 MB	34.0 MB	7,318	228	0.3 %
news025f2	13.6 MB	28.5 MB	7,174	216	0.3 %
news095f1	17.3 MB	32.2 MB	7,061	113	0.3 %
news026f2	14.1 MB	28.9 MB	7,012	215	0.3 %
news023f1	13.9 MB	29.0 MB	6,922	205	0.3 %
news084f1	20.6 MB	34.9 MB	6,860	215	0.3 %
news073f1	17.6 MB	31.4 MB	6,855	221	0.3 %
news007f2	9.8 MB	22.5 MB	6,848	170	0.3 %
news041f1	13.4 MB	28.2 MB	6,660	107	0.3 %
news007f3	9.2 MB	21.5 MB	6,512	126	0.3 %
news024f2	14.0 MB	28.1 MB	6,483	142	0.3 %
news007f1	9.7 MB	21.5 MB	6,472	151	0.3 %
news019f2	13.7 MB	27.1 MB	6,403	203	0.3 %
news043f1	15.8 MB	29.6 MB	6,393	94	0.3 %
news022f2	11.8 MB	24.9 MB	6,075	193	0.3 %
news005f3	7.2 MB	16.2 MB	5,583	136	0.3 %
news078f1	15.4 MB	26.2 MB	5,493	202	0.3 %
news018f2	10.1 MB	20.7 MB	5,113	207	0.2 %
news020f2	8.6 MB	19.9 MB	5,070	79	0.2 %
news082f1	13.1 MB	23.4 MB	4,897	106	0.2 %
news005f1	5.8 MB	10.9 MB	4,615	138	0.2 %
news021f2	9.0 MB	18.3 MB	4,408	202	0.2 %
news081f1	12.0 MB	21.1 MB	4,332	207	0.2 %
news074f1	11.9 MB	19.7 MB	4,043	196	0.2 %
news076f1	11.5 MB	19.4 MB	4,006	190	0.2 %
news072f1	11.4 MB	19.0 MB	3,963	216	0.2 %
news077f1	9.6 MB	16.5 MB	3,492	184	0.2 %
news071f1	8.5 MB	14.8 MB	3,283	202	0.2 %
news075f1	10.3 MB	16.7 MB	3,283	196	0.2 %
news005f2	3.5 MB	6.5 MB	2,635	138	0.1 %

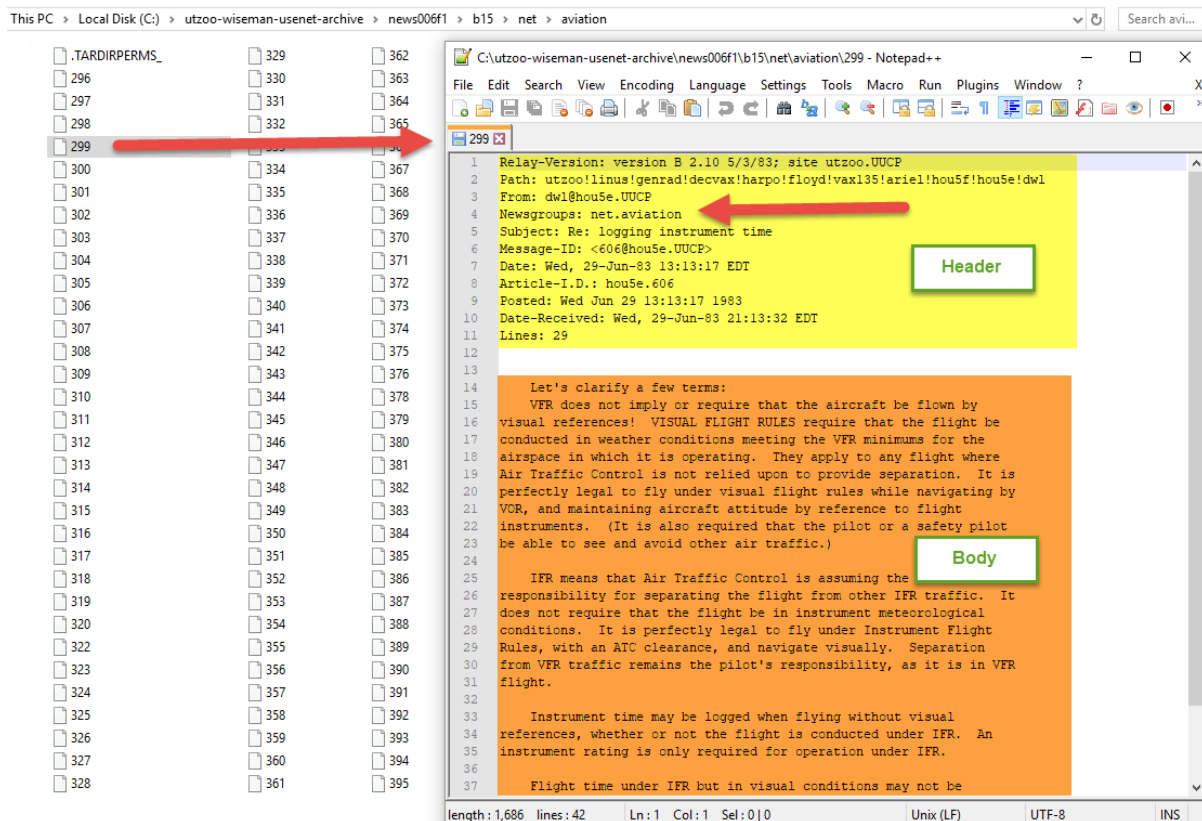
news017f2	3.9 MB	9.1 MB	2,385	50	0.1 %
news079f1	3.9 MB	6.5 MB	1,247	149	0.1 %

File Issues

While examining the extract, I realized that Magnetic Tape 118 is uncompressed in \utzoo-wiseman-usenet-archive\news118f1 folder, named tape118, so I had rename it to tape118.tar and extracted it manually, only to realize it's a copy of files which I already have. Someone creating the original archive forgotten to remove that file. There are 3 files in these folders that need to have .tar extension added and decompressed as well:

- \utzoo-wiseman-usenet-archive\news118f1\tape118
- \utzoo-wiseman-usenet-archive\news120f1\tape120
- \utzoo-wiseman-usenet-archive\news121f1\tape121

If you opened one of the folders and navigated down to one of the many subfolders, you'd find a file that contained the message. For example, going into \utzoo-wiseman-usenet-archive\news006f1\b15\net\aviation folder, I was now apparently in the **net.aviation** Usenet group. But the only way to find out was to open one of the files and look at the content. Here I highlighted what it looked like. As you can see, each file seems to consist of a header, then a single empty line and the body of the message:



Database Design

So, I decided to build a Python parser, that went through all these files reading the header portion of each message and grouping all unique results together, giving me all the possible headers such as (From, Subject, Newsgroup, etc.). I found that there were about 79 x different types of headers. So it appeared that not all messages adhered to the same basic structure. Going through the headers, all had the standard set that was common across all posts.

Once I had the common field, I've created a Postgres database called 'utzoo'

```
create database utzoo;
```

And a new schema called all_messages

```
create schema all_messages;
```

The above database and schema were the pre-requisites. Everything else, like table creation, inserting the posts, etc. is part of the Python script and fully automated.

In terms of table creation, the script automatically creates 5 tables for each detected newsgroup:

- headers – parsed headers
- references – references for each message
- body – text of the message
- from – who posted the message
- subjects – list of unique subject lines

This is what the script auto-creates for each unique Group name:

```
create table all_messages.GroupName_headers
(
    id          bigserial not null
    constraint GroupName_headers_pk primary key,
```

```
    dateparsed timestamp,
    subj_id    bigint,
    ref        smallint,
    msg_id     text,
    msg_from   bigint,
    enc        text,
    contype    text,
    processed  timestamp default CURRENT_TIMESTAMP
);
alter table all_messages.GroupName_headers
    owner to postgres;

create table all_messages.GroupName_refs
(
    id        bigint,
    ref_msg text default null
);
alter table all_messages.GroupName_refs
    owner to postgres;

create table all_messages.GroupName_body
(
    id    bigint primary key,
    data text default null
);
alter table all_messages.GroupName_body
    owner to postgres;

create table all_messages.GroupName_from
(
    id    serial not null
        constraint GroupName_from_pk primary key,
    data text
);
alter table all_messages.GroupName_from
    owner to postgres;

create table all_messages.GroupName_subjects
(
```



```
        id          serial not null
        constraint GroupName_subjects_pk primary key,
        subject text
    );
alter table all_messages.GroupName_subjects
    owner to postgres;
```

Those will be the tables where the Python parser will dump all the data and make sure posts are properly lined up between tables.

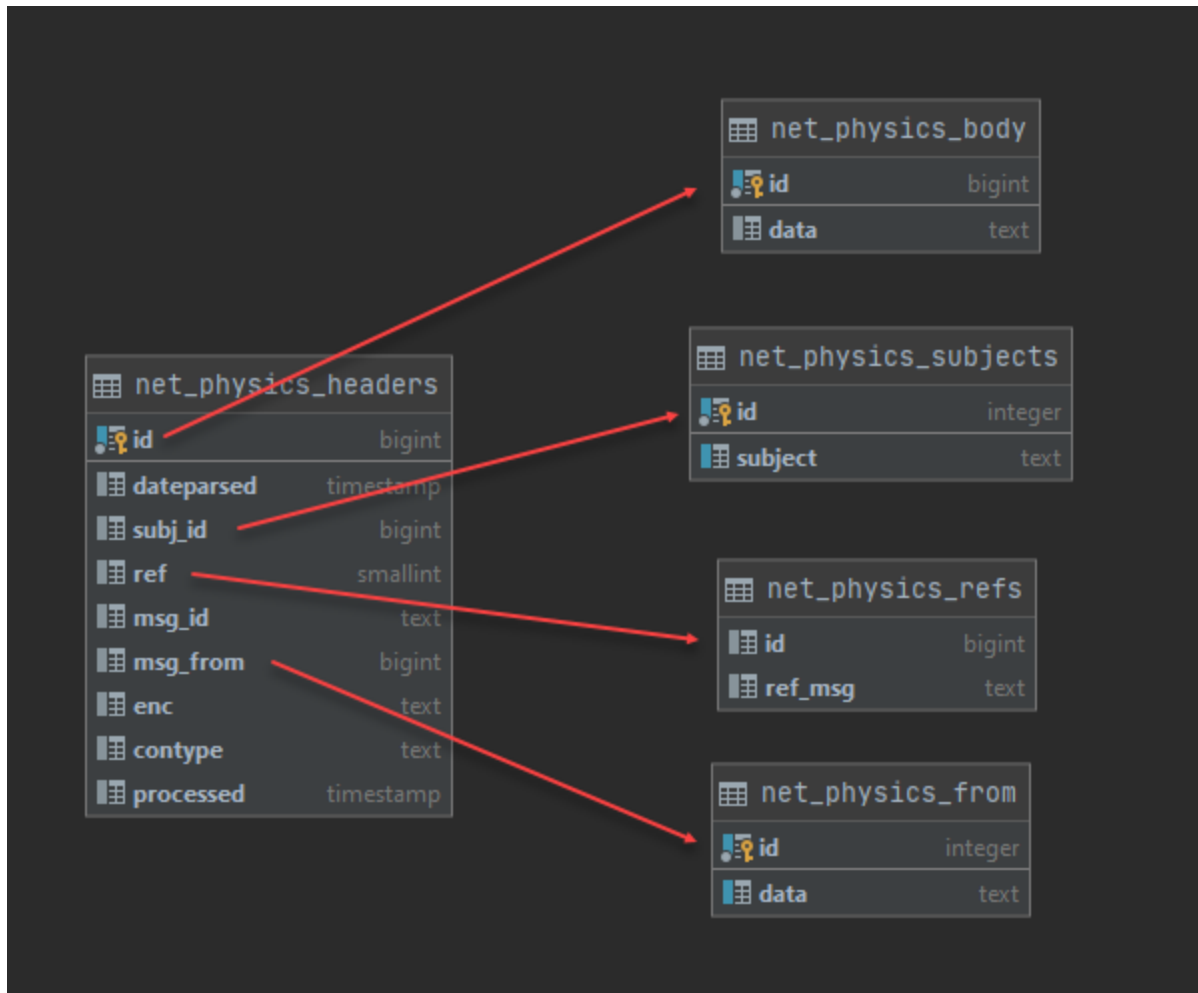
The python script also creates indexes to make the inserting and later reading of the posts faster:

```
create unique index GroupName_headers_uidx on all_messag
create unique index GroupName_headers_umidx on all_messag
create unique index GroupName_body_idx on all_messages.Gr
create unique index GroupName_from_idx on all_messages.Gr
create unique index GroupName_subjects_idx on all_message
```

Once created, the structure per group looks like this:

▼	net_physics_body
•	id bigint
■	data text
🔑	net_physics_body_pkey (id)
⌚	net_physics_body_idx (id) UNIQUE
⌚	net_physics_body_pkey (id) UNIQUE
▼	net_physics_from
•	id integer = nextval('all_messa...')
■	data text
🔑	net_physics_from_pk (id)
⌚	net_physics_from_idx (data) UNIQUE
⌚	net_physics_from_pk (id) UNIQUE
▼	net_physics_headers
•	id bigint = nextval('all_messa...')
■	dateparsed timestamp
■	subj_id bigint
■	ref smallint
■	msg_id text
■	msg_from bigint
■	enc text
■	contype text
■	processed timestamp = CURRENT_TIMESTAMP
🔑	net_physics_headers_pk (id)
⌚	net_physics_headers_pk (id) UNIQUE
⌚	net_physics_headers_uuidx (id) UNIQUE
⌚	net_physics_headers_umidx (msg_id) UNIQUE
▼	net_physics_refs
■	id bigint
■	ref_msg text
▼	net_physics_subjects
•	id integer = nextval('all_messa...')
■	subject text
🔑	net_physics_subjects_pk (id)
⌚	net_physics_subjects_idx (subject) UNIQUE
⌚	net_physics_subjects_pk (id) UNIQUE

The following screenshot explains how it's all wired up. I didn't do any hardcoded relationships, but you can change the script if you want that.



Date Related Issues

The date is an integral part of each message and I had to do some data conversion massaging in Python to get the proper date, as dates were coming in a variety of formats. I've tried various libraries but `dateutil.parser.parse` standard date and time library for Python did the best job.

However, I still needed to account for various labelling of data fields in the headers, so if data wasn't found in the 'date' header, I had to look into other header parts such as 'NNTP-Posting-Date', 'X-Article-Creation-Date', 'Posted', or 'Received' fields.

Python Source Code

Well and then it was all about creating a Python parser, start the PostgreSQL, point it to an archive directory, and wait :)

At the bottom of this article is the code of the Python solution. It's about 1,000 lines, and it took altogether about 1 day to create and test it. The script is smart enough to

keep the track of where it started, so if it needs to be interrupted, it'll know where to continue from to get the job done.

The source code is available on GitHub as open-source under MIT license:

https://github.com/JozefJarosciak/python_mbox_parser/blob/master/utzoo2postgres.py.

How to Run It?

The final solution artifact is called '**utzoo2postgres.py**', and it was tested on Python 3.8.

Open the script and define the path to un-tared Utzoo archive directories.

Examples:

```
# for Windows
positionFilePath = "E:\\Usenet\\Utzoo\\"
# for linux:
# positionFilePath = "/Usenet/Utzoo/"
```

Also, define the particulars of your PostgreSQL database:

```
db_connection = psycopg2.connect(host="localhost", user="",
```

And then just execute the script!

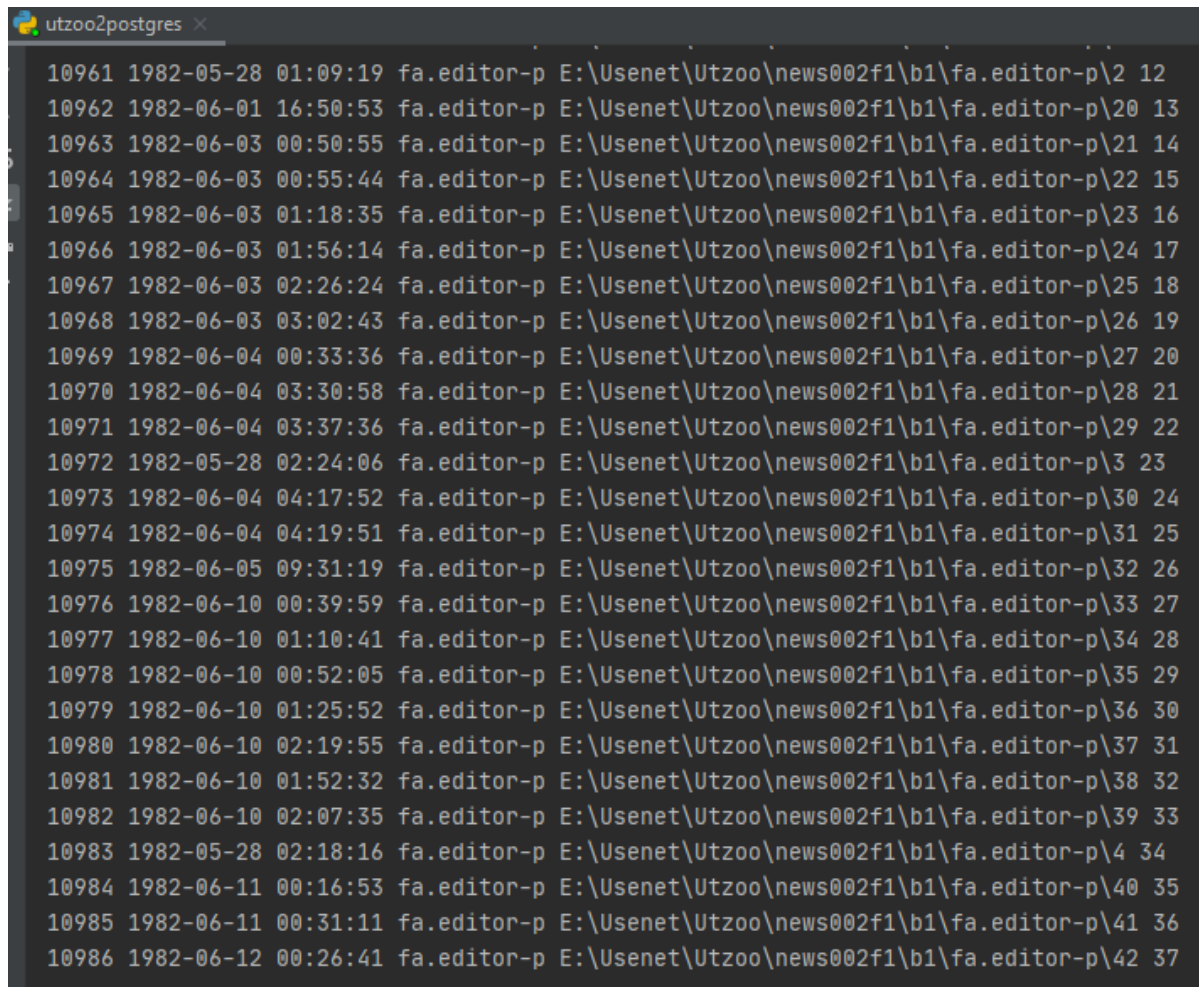
```
python 3 utzoo2postgres.py
```

Note: In case you need to stop the program and run it later, the script is smart to resume from the last spot it was processing.

Stats

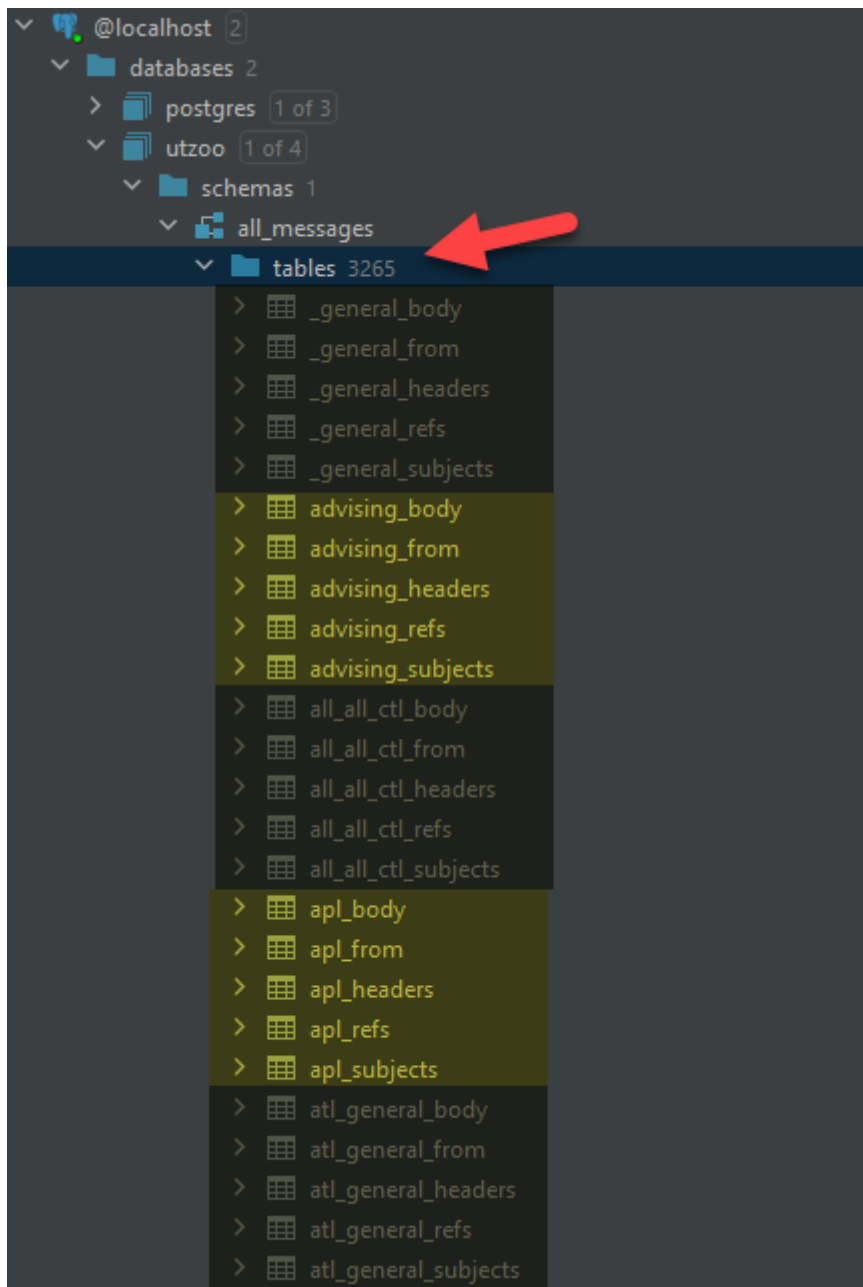
The script will process all Utzoo Archive messages in about 6 hours (depending on the speed of your machine).

Screenshot from processing:



```
utzoo2postgres X
10961 1982-05-28 01:09:19 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\2 12
10962 1982-06-01 16:50:53 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\20 13
10963 1982-06-03 00:50:55 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\21 14
10964 1982-06-03 00:55:44 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\22 15
10965 1982-06-03 01:18:35 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\23 16
10966 1982-06-03 01:56:14 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\24 17
10967 1982-06-03 02:26:24 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\25 18
10968 1982-06-03 03:02:43 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\26 19
10969 1982-06-04 00:33:36 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\27 20
10970 1982-06-04 03:30:58 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\28 21
10971 1982-06-04 03:37:36 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\29 22
10972 1982-05-28 02:24:06 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\3 23
10973 1982-06-04 04:17:52 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\30 24
10974 1982-06-04 04:19:51 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\31 25
10975 1982-06-05 09:31:19 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\32 26
10976 1982-06-10 00:39:59 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\33 27
10977 1982-06-10 01:10:41 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\34 28
10978 1982-06-10 00:52:05 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\35 29
10979 1982-06-10 01:25:52 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\36 30
10980 1982-06-10 02:19:55 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\37 31
10981 1982-06-10 01:52:32 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\38 32
10982 1982-06-10 02:07:35 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\39 33
10983 1982-05-28 02:18:16 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\4 34
10984 1982-06-11 00:16:53 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\40 35
10985 1982-06-11 00:31:11 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\41 36
10986 1982-06-12 00:26:41 fa.editor-p E:\Usenet\Utzoo\news002f1\b1\fa.editor-p\42 37
```

Here is a screenshot of the database after only a couple of minutes of conversion:



As you can see, the conversion utility produces a database with 5 tables per group where messages are linked to each other through auto-created indexes.

Let's say we want to look up all discussions in the **net.physics** discussions; and sort them out by the number of replies.

This is how you can do that:

```

select headers.id, headers.dateparsed, subj.subject, count(*) as replies
from all_messages.net_physics_refs refs
  join all_messages.net_physics_headers headers on refs.ref_msg = headers.msg_id
  join all_messages.net_physics_subjects subj on headers.subj_id = subj.id
where headers.ref = 0
group by refs.ref_msg, headers.msg_from, headers.id, ref_msg, headers.dateparsed, headers.enc, subj.subject
having count(*) >= 0
order by dateparsed desc

```

Output Result 17

245 rows

	id	dateparsed	subject	replies
1	1578	1984-10-31 09:59:25.000000	Could someone explain why FTL is illegal? In small words?	17
2	1705	1984-12-08 00:48:50.000000	Floating a battleship in a gallon of water	11
3	1648	1984-11-23 17:59:03.000000	Question on FTL and quantum mechanics	10
4	894	1984-01-03 14:59:48.000000	More on Cold Bottles of Coke	8
5	1712	1984-12-09 17:15:55.000000	Big Bang Impossible	6
6	1669	1984-11-29 08:09:25.000000	"big bang" a big bust?	6
7	1537	1984-10-10 08:54:23.000000	Those funny lines	6
8	1589	1984-11-02 18:25:16.000000	Re: C as speed limit	5
9	1404	1984-06-11 11:05:01.000000	RE: Unix for physicists (attn:finn)	5
10	730	1983-10-26 03:43:31.000000	M = E/C^2 ??? How??? - (nf)	5
11	1760	1984-12-15 13:28:00.000000	deflecting laser	4
12	1739	1984-12-13 01:33:08.000000	Is the universe predictable?	4
13	1711	1984-12-09 17:12:55.000000	quo vadis gravity?	4
14	1583	1984-11-01 17:17:50.000000	A Question on Ballistics	4

Now, we can look up a particular discussion by the ID. For example, we want the ID: 1648 from the screenshot above, the discussion with the subject: **“Question on FTL and quantum mechanics”**. That’s not so hard either:

```

select headers.dateparsed, subjects.subject, ffrom.data, body.data, headers.enc, headers.msg_id
from all_messages.net_physics_headers headers
  JOIN all_messages.net_physics_body body ON headers.id = body.id
  JOIN all_messages.net_physics_subjects subjects ON headers.subj_id = subjects.id
  join all_messages.net_physics_from ffrom on headers.msg_from = ffrom.id
where (
  headers.id in
    (select refs.id from all_messages.net_physics_refs refs where refs.ref_msg = (select hd.msg_id as selectmsg FROM all_messages.net_physics_headers hd WHERE hd.id = 1648))
  or (headers.id = 1648)
)
ORDER BY headers.dateparsed::timestampz ASC, LENGTH(subject) ASC

```

Output Result 18

11 rows

	dateparsed	subject	ffrom.data	body.data	enc	msg_id
1	1984-11-23 17:59:03.000000	Question on FTL and quantum mechanics	barry@ames.UUCP (Kenn Barry)	[]<< The following bit of speculation came to me a v ANSI	<654@ames.UUCP>	
2	1984-11-24 21:24:59.000000	Re: Question on FTL and quantum mechanics	jln@hplabs.UUCP (Tai Jin)	i've heard that the problem in attaining ftl speeds is.. ANSI	<1150@hplabs.UUCP>	
3	1984-11-27 19:51:01.000000	Re: Question on FTL and quantum mechanics	fons@mcvax.UUCP (Fons Kuijk)	In article <654@ames.UUCP> barry@ames.UUCP (Kenn Barr.. ANSI	<6201@mcvax.UUCP>	
4	1984-11-29 17:33:09.000000	Re: Question on FTL and quantum mechanics	abeles@huxm.UUCP (abeles)	//>You ought to read up on tachyons which are particle.. ANSI	<274@huxm.UUCP>	
5	1984-11-30 02:37:34.000000	Re: Question on FTL and quantum mechanics	act@pur-phy.UUCP (Alex C. Tselis)	> ----- ANSI	<1530@pur-phy.UUCP>	
6	1984-11-30 16:58:07.000000	Re: Question on FTL and quantum mechanics	act@pur-phy.UUCP (Alex C. Tselis)	> <=> => You ought to read up on tachyons which are pa.. ANSI	<1531@pur-phy.UUCP>	
7	1984-12-02 18:59:23.000000	Re: Re: Question on FTL and quantum mechanics	gjk@talcott.UUCP (Greg J Kuperberg)	> You ought to read up on tachyons which are particle.. ANSI	<155@talcott.UUCP>	
8	1984-12-05 02:45:21.000000	Re: Re: Question on FTL and quantum mechanics	act@pur-phy.UUCP (Alex C. Tselis)	> > You ought to read up on tachyons which are partic.. ANSI	<1550@pur-phy.UUCP>	
9	1984-12-05 15:56:16.000000	Re: Re: Question on FTL and quantum mechanics	cpf@lasspva.UUCP (Courtenay Footman)	In article <=> gjk@talcott.UUCP (Greg J Kuperberg) wrt.. ANSI	<146@lasspva.UUCP>	
10	1984-12-06 04:37:01.000000	Re: Re: Question on FTL and quantum mechanics	gwyn@brl-tgr.ARPA (Doug Gwyn <gwyn>)	> > You ought to read up on tachyons which are particles moving faster > > than light. They can't slow down in the same way that we can't speed > > up to c. There is no known evidence of tachyons. > > Tachyons are dead. Some respected physicist published a paper on them > once. There were many replies, to the effect of, "you made a mistake in > your physics." The guy then said, "oops", and that was the end of it. > > The some non-physicists discovered the original article and published much > literature about it.		
11	1984-12-06 05:10:30.000000	Re: Re: Question on FTL and quantum mechanics	gwyn@brl-tgr.ARPA (Doug Gwyn <gwyn>)	Oops! I meant Gerald Feinberg, not Weinberg (I've be.. ANSI	<639@brl-tgr.ARPA>	

The Final Product

It's nice to have a database full of posts, but it's hardly usable that way. I needed something that would allow me to easily access these posts.

So, once everything was done, I built a PHP script around this code and registered <https://usenetarchives.com> to make all these archives available online, in an easy to

The PHP code is not part of this article, but you can head over to <https://usenetarchives.com/groups.php?c=utzoo> to see how it all works:

Usenet Archives

[Alt](#) - [Comp](#) - [Humanities](#) - [Microsoft](#) - [Misc](#) - [News](#) - [Rec](#) - [Sci](#) - [Soc](#) - [Talk](#) - **Utzoo**

Henry Spencer's UTZOO NetNews Archive

In the following groups, over 2.1 million posts created between Feb 1981 and June of 1991 originate from Henry Spencer's UTZOO NetNews Archive of earliest USENET posts.

Page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [Next](#)

Utzoo Groups	Description	Posts
Alt. Flame	Flame	168
Alt. Folklore. Computers	Folklore Computers	268
Alt. Sources	Sources	261
Alt. Sources. D	Sources D	160
Bionet. General	Bionet General	136
Bionet. Molbio. Genbank. Updates	Bionet Molbio Genbank Updates	1,217
Comp. Admin. Policy	Discussions of site administration policies.	2,034
Comp. Ai	Artificial Intelligence. (Moderated)	5,045
Comp. Ai. Digest	Comp Ai Digest	150
Comp. Ai. Neural - Nets	All aspects of neural networks.	46,337
Comp. Ai. Philosophy	Philosophical aspects of Artificial Intelligence.	337,477
Comp. Arch	Computer architecture.	264,139
Comp. Archives	Comp Archives	700
Comp. Archives. Admin	Issues relating to computer archive administration.	321
Comp. Benchmarks	Discussion of benchmarking techniques and results.	1,200
Comp. Binaries. Apple2	Comp Binaries Apple2	140
Comp. Binaries. Ibm. Pc	Comp Binaries Ibm Pc	318
Comp. Binaries. Ibm. Pc. D	Comp Binaries Ibm Pc D	1,367
Comp. Binaries. Mac	Comp Binaries Mac	185
Comp. Binaries. Os2	Comp Binaries Os2	226
Comp. Compilers	Compiler construction, theory, etc. (Moderated)	27,983
Comp. Compression	Data compression algorithms and theory.	76,108
Comp. Databases	Database and data management issues and theory.	60,359
Comp. Dcom. Lans	Comp Dcom Lans	589
Comp. Dcom. Modems	Data communications hardware and software.	6,902
Comp. Dcom. Sys. Cisco	Info on Cisco routers and bridges.	290,481
Comp. Dcom. Telecom	Telecommunications digest. (Moderated)	155,015
Comp. Dsp	Digital Signal Processing using computers.	389,768
Comp. Editors	Topics related to computerized text editing.	48,282
Comp. Edu	Computer science education.	2,141

Conclusion

So now that it's all done, I have to say, it was a great journey.

For those who want to play with the code, you can grab it from [Github](#) and adjust it to your liking. Please don't judge the code, it's not pretty, nor formatted or commented out (for the most part) as I wasn't exactly planning to release it. I did so primarily for posterity reasons. But now that it's out there you're more than welcome to fork the repo, clean it up though and commit your changes, so others can benefit from your work too.

To conclude this article, this is the illustrated process of getting information from each of the files into the PostgreSQL database.

Screenshot Description:

- 1. Henry Spences stores early internet posts on Magnetic Tapes
- 2. Downloaded copy of tar files is extracted into millions of flat files
- 2. Testing Headers and Body example of each of the flat file posts
- 3. Writing and running Python code to parse out all header and body fields
- 5-6. The Python script auto creates tables and indexes
- 7. The result: PostgreSQL fully searchable database of all lost Usenet posts Feb 1981 and June of 1991
- 8. Making the whole Utzoo archive available online at <https://usenetarchives.com/groups.php?c=utzoo>

The screenshots illustrate the following steps:

- Henry Spencer with magnetic tapes.
- A large directory listing of extracted flat files.
- A terminal window showing Python code for parsing headers.
- A terminal window showing the execution of a Python script.
- A diagram of the database schema showing tables like net_physobj_body, net_physobj_headers, net_physobj_refs, net_physobj_subjects, net_physobj_from, and net_physobj_to.
- A terminal window showing the output of the Python script.
- A screenshot of the PostgreSQL database interface showing the results of a query.
- A screenshot of the Usenet Archives website showing the Henry Spencer's UTZOO NetNews Archive.