9-2009

Binary Usenet Application

Alexander J. Patterson
Grand Valley State University

Binary Usenet Application

By

Alexander J Patterson

September, 2009

Binary Usenet Application

By
Alexander J Patterson

A Project submitted in partial fulfillment of the requirements for the degree of
Master of Science in
Computer Information Systems

at
Grand Valley State University
September, 2009

____

Dr. Robert Adams **Date**

Table of Contents

Table of Figures

Table of Tables

ANZBC (Another News Bin Client) Synopsis

ANZBC software was written as a prototype to download small-encoded files form Usenet servers. This software allows anything from a text document to a HD movie to be downloaded and decoded into a single file. Often this is a single file that is split into multiple small files and then compressed.

Usenet

There is a great deal of talk in today's world of peer-to-peer networks because of Napster, Kazaa, and Skype. Many of us are familiar with these technologies and have probably used one of them. If you have ever downloaded anything using them you can understand how slow and difficult it can often become trying to locate and download large files.

An older technology that has been widely available since 1980 is Usenet taking its name from "user network." Usenet began in 1979 when Ellis and another Duke graduate student, Tom Truscott, thought of hooking computers together to share information. At the beginning of 1980, the network consisted of two sites at Duke and one at the University of North Carolina. Where peer-to-peer networks rely on the information contained on each individual computer, the idea behind a Usenet network is to have centrally located servers where clients can connect to upload and download articles.

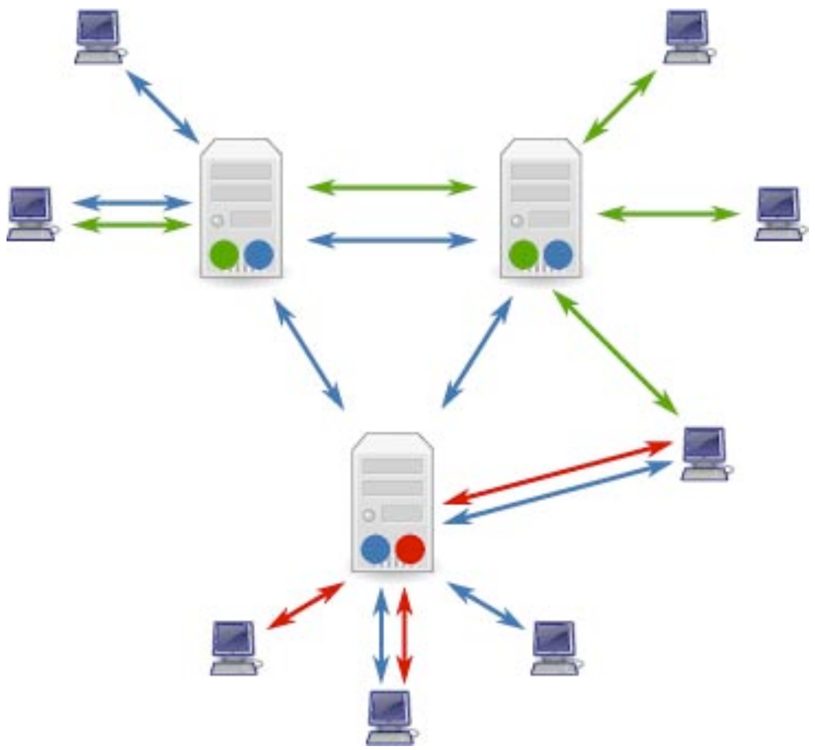


Figure 1: A common Usenet network setup (1)

In Figure 1 there are multiple clients that will connect to any given server in the Usenet network and upload files. These files are then replicated across the central servers, providing for great redundancy and access. These files are placed into groups on the network so that they can be easily searched and managed.

There are many "newsgroups" in Usenet but the focus of this paper is on alt.binaries.*, Figure 2 there is an example of taking a full DVD and uploading it to Usenet in the alt.binaries* group, typically this would be something like alt.binaries.movies. In the Figure 2 represents, but for now it is a good overview to look at the process. Usenet limits the file size of each article that is why the iso was split into several files and then compressed. The reason that the file parts are encoded is that Usenet was only meant for ASCII characters.

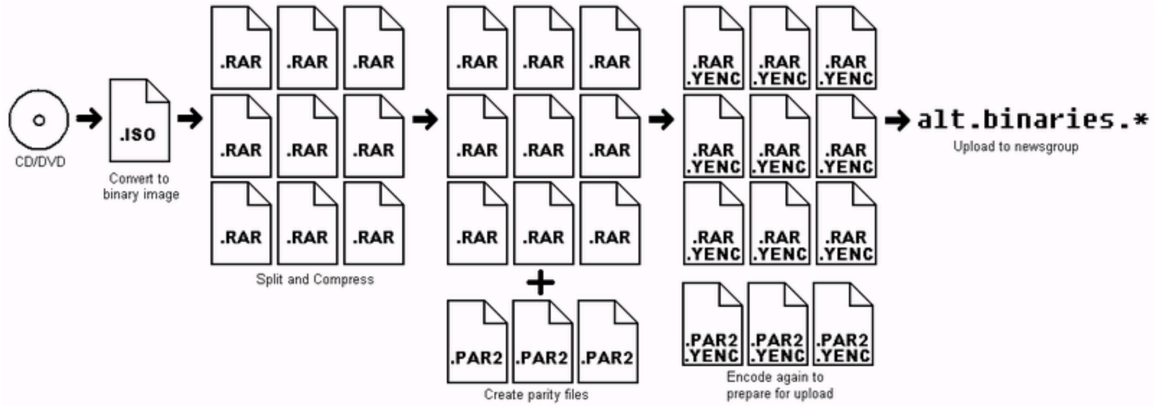


Figure 2: Usenet Binaries Upload process (2)

NNTP

News Network Transfer Protocol is the application protocol used for reading and posting articles on Usenet. As the IETF (Internet Engineering Task Force) RFC 3977 states, “For news-reading clients, NNTP enables retrieval of news articles that are stored in a central database, giving subscribers the ability to select only those articles they wish to read. (3)” This helps show the technology that ANZBC uses to capture specifically the articles requested and not a massive amount of other unused data.

“The Network News Transfer Protocol (NNTP) has been in use in the Internet for a decade, and remains one of the most popular protocols (by volume) in use today. (3)” Even though this is a very old protocol it still holds great value today offering an easy way to communicate and exchange data.

When using the NNTP protocol over TCP it is very similar to how the HTTP protocol works establishing a connection with the server waiting for a request. “When the connection is established, the NNTP server host must send a greeting. The client host and server host then exchange commands and responses (respectively) until the connection is closed or aborted. (3)” The official RFC 3977 for NNTP lists these as the message codes:

The first digit of the response broadly indicates the success, failure, or progress of the previous command:

1xx - Informative message
2xx - Command completed OK
3xx - Command OK so far; send the rest of it
4xx - Command was syntactically correct but failed for some reason
5xx - Command unknown, unsupported, unavailable, or syntax error

The next digit in the code indicates the function response category:

x0x - Connection, setup, and miscellaneous messages
x1x - Newsgroup selection
x2x - Article selection
x3x - Distribution functions
x4x - Posting
x8x - Reserved for authentication and privacy extensions
x9x - Reserved for private use (non-standard extensions)

Table 1: NNTP Response codes (3)

Instead of developing new code to handle all of the NNTP commands and responses, ANZBC uses a stable Java solution, written by Apache. Apache offers a NNTP solution as part of their Commons Net 2.0 API. In the package org.apache.commons.net.nntp it allows for an easy implementation in Java to make the connection to the server, download an article, and handle all of the response codes and situations. The implementation of this package is detailed further in the section.

Rar

Figure 2 it illustrates that a single file is broken into multiple rar files, also known as file spanning. Typically large files on Usenet are broken into rar files of 50MB size, however this can vary. As you will notice this provides for two things, a way to compress all of the data and reduce the size, and also a standard way of tracking all of the split files. RAR files of more recent times now have the ability to provide their own recovery files called .rev files. These still are not very popular on Usenet and par2 files are used.

Par

Although NNTP is a useful protocol it is not very robust in nature, and only built for ASCII messages. So when binary files were starting to become more commonly used through Usenet a solution to data loss needed to be created. A par file “is a tool to apply the data-recovery capability concepts of RAID-like systems to the posting & recovery of multi-part archives on Usenet (4).” So along with the rar file set, also comes a set of par files, which can fill in the gaps or corrupted pieces of data that are lost during transfers and fix broken blocks.

yEnc

In order to convert data from 7 bit ASCII to 8 bit binary data an encoding scheme had to be developed. In the traditional approach to encoding binary messages to ASCII this increased the file size up to 40%, however with yEnc it is only around 1-2%. “The ASCII value of each output character is derived by the following simple formula: $O = (I+42) \% 256$. That is, the output value is equal to the ASCII value of each input character plus 42, all modulo 256. This reduces overhead by reducing the number of NULL characters (ASCII 00) that would otherwise have had needed to be escaped, since many binaries contain a disproportionately large number of NULLs). (5)” There are a few simple rules to yEnc but the basic process works like this:

1. Fetch a character from the input stream.
2. Increment the character's ASCII value by 42, modulo 256
3. If the result is a critical character (as defined in the previous section), write the escape character to the output stream and increment character's ASCII value by 64, modulo 256.
4. Output the character to the output stream.
5. Repeat from start.

Table 2: yEnc Encoding (5)

Figure 2 these files are broken apart again into smaller yEnc encoded multi-part files to make up a single rar file. The structure for these files will look like:

```
=ybegin part=2 total=2 line=128 size=19338 name=joystick.jpg
=ypart begin=11251 end=19338
```

Contains some data in yEnc encoded format and then…

```
=yend size=8088 part=2 pcrc32=aca76043
```

Table 3: yEnc Begin and End

Notice in Table 3 this is the second part, denoted part=2, of a multipart file. One of the downsides of this type of encoding is that it does not include the address of the rest of the multi-part files.

NZB (XML)

NZB is a fancy name for an XML file. The reason that this extension is used in most of my examples is because this is the glue that keeps all the yEnc encoded file parts together. Although nzb files are not required to download articles from Usenet they are required for ANZBC. Instead of reading through thousands of headers of a newsgroup the inventors of the site www.newzbin.com created a searchable site for content on Usenet. This site will build the nzb file that looks like Table 4 including all of the necessary components. The nzb files can be made up of a number of packaged files, or it could be from a raw Usenet read.

```
<?xml version="1.0" encoding="iso-8859-1" ?>

        <!DOCTYPE nzb PUBLIC "-//newzBin//DTD NZB 1.0//EN" "http://www.newzbin.com/DTD/nzb/nzb-1.0.dtd">
        <nzb xmlns="http://www.newzbin.com/DTD/2003/nzb">
                <file subject="joystick.rar&#34; -  [001/001] -  yEnc (1/1)"
                                date="1252174555" poster="anyone@anywhere”>
                        <groups>
                                <group>alt.binaries.pics </group>
                        </groups>
                        <segments>
                                <segment bytes="19338 " number="1">
                                        part1of2.AKjwgYdygm@powerpost2000AA.local
                                </segment>
                                <segment bytes="19338 " number="2">
                                        part2of2.AKjwgYdygm@powerpost2000AA.local
                                </segment>
                </file>
        </nzb>
```

Table 4: Sample NZB file

In Table 4 there is a base element called <file> with a subject attribute, which is the filename of the rar file. If you then look in the next node element <segment> this contains the locations on Usenet of all the yEnc encoded articles that make up the full rar file. This is something that is hard to describe and hopefully Figure 4 depicts this better visually.

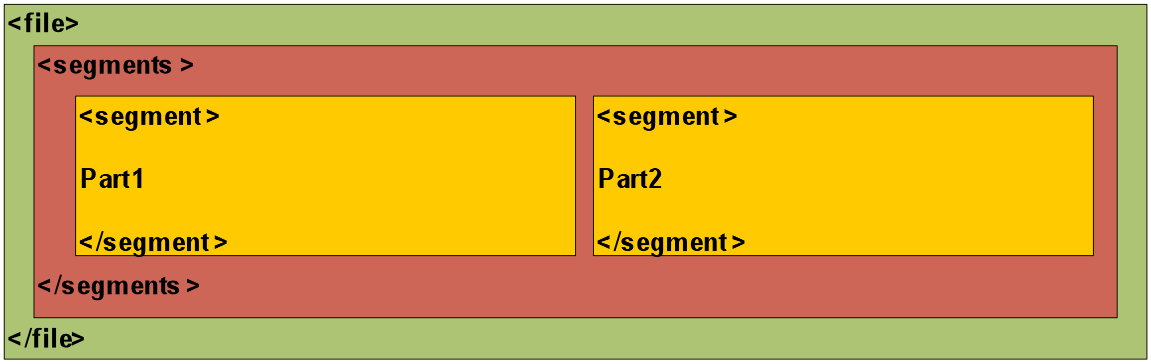


Figure 4: NZB Structure

ANZBC (Combining Technologies)

Even though the technologies listed above have been around for some time there are only a few “news readers” that download binary data. There are even less that will take an nzb file for specific files to be downloaded. ANZBC will combine the functions in Table 5 into one easy to use application. Otherwise it would take up to 5 different applications in order to make downloading binary data successful.

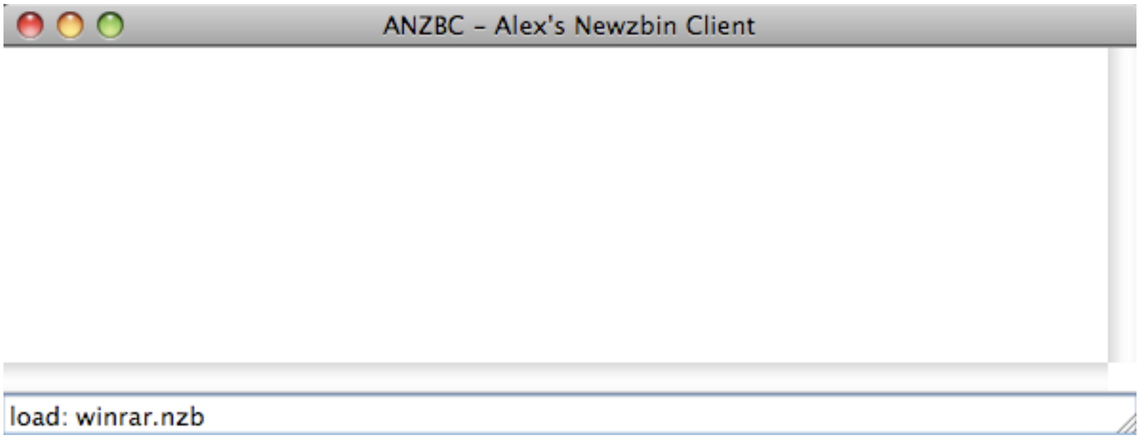
Step	Reason	Technology
1	Upload NZB File	www.newsbin.com
2	Download Articles	NNTP, Usenet
3	Decode and concatenate	yEnc
4	Check Parity	Par Files
5	Uncompress and join	Rar

Table 5: Why single tool is needed

ANZBC (Functionality)

Loading NZB File

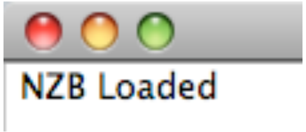
The prototype GUI was kept very simple in order to focus development on the underlying programming logic. To start the process of downloading Winrar’s 3.9 trial a nzb file must be downloaded from www.newzbin.com. The nzb file will be loaded into ANZB using the “load” command, in which starts the parsing of XML.



load: winrar.nzb

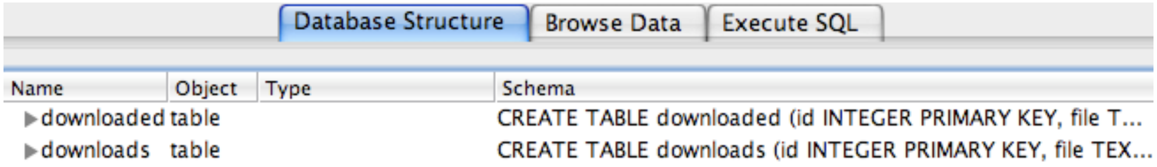
Figure 5: Loading NZB

When this is completed ANZBC issues the message in Figure 5. What actually happens during the parsing is that ANZBC finds the <file> and <segments> within <file> and stores them into the SQLite database called Downloads.db. In Figure 7 it shows the two tables in this database, downloads and downloaded.



NZB Loaded

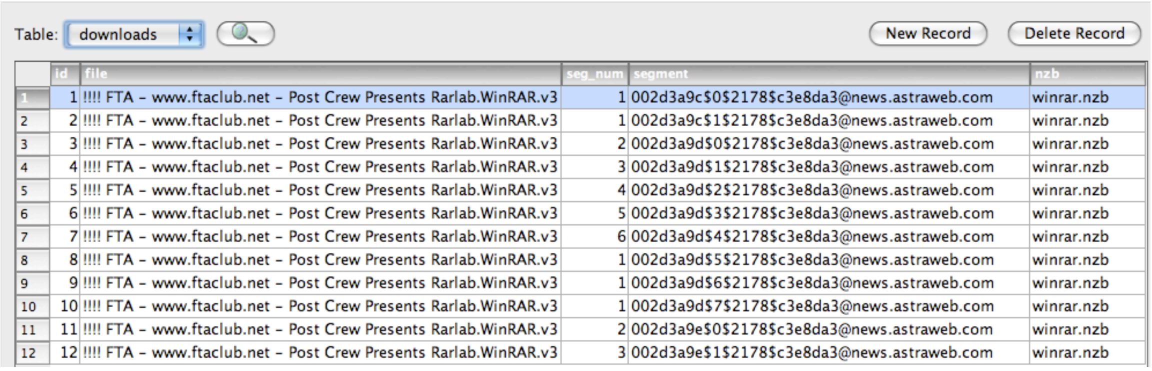
Figure 5: Successful NZB load



Name	Object	Type	Schema
▶downloaded	table		CREATE TABLE downloaded (id INTEGER PRIMARY KEY, file T...
▶downloads	table		CREATE TABLE downloads (id INTEGER PRIMARY KEY, file TEX...

Figure 7: Downloads.db tables

When ANZBC stores the originally parsed nzb file it is placed into the "downloads" table like in Figure 8. Saving all of the valuable segments and associating them with their file and overall nzb file from the original creation.

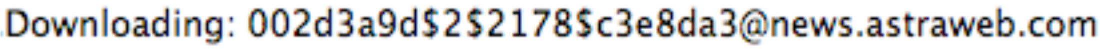


	id	file	seg_num	segment	nzb
1	1	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	1	002d3a9c$0$2178$c3e8da3@news.astraweb.com	winrar.nzb
2	2	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	1	002d3a9c$1$2178$c3e8da3@news.astraweb.com	winrar.nzb
3	3	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	2	002d3a9d$0$2178$c3e8da3@news.astraweb.com	winrar.nzb
4	4	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	3	002d3a9d$1$2178$c3e8da3@news.astraweb.com	winrar.nzb
5	5	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	4	002d3a9d$2$2178$c3e8da3@news.astraweb.com	winrar.nzb
6	6	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	5	002d3a9d$3$2178$c3e8da3@news.astraweb.com	winrar.nzb
7	7	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	6	002d3a9d$4$2178$c3e8da3@news.astraweb.com	winrar.nzb
8	8	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	1	002d3a9d$5$2178$c3e8da3@news.astraweb.com	winrar.nzb
9	9	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	1	002d3a9d$6$2178$c3e8da3@news.astraweb.com	winrar.nzb
10	10	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	1	002d3a9d$7$2178$c3e8da3@news.astraweb.com	winrar.nzb
11	11	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	2	002d3a9e$0$2178$c3e8da3@news.astraweb.com	winrar.nzb
12	12	!!!! FTA – www.ftaclub.net – Post Crew Presents Rarlab.WinRAR.v3	3	002d3a9e$1$2178$c3e8da3@news.astraweb.com	winrar.nzb

Figure 8: Downloads Table

Downloading Segments

After loading the nzb file the "write" command along with the name of the nzb file can be issued to start downloading. The reason that the nzb filename is used is so that multiple nzb files can be loaded then when the time is appropriate the "write" command is issued for a single nzb file. ANZBC will start to list each segment that it is downloading like Figure 9.



```
Downloading: 002d3a9d$2$2178$c3e8da3@news.astraweb.com
```

Figure 9: Segment Download

The downloading is multi-threaded so that several segments can download at a single time, not having to wait until the socket is done blocking. However a thread limit can be set so that the amount of socket connections you have to the Usenet server is reduced. Most Usenet servers only allow up to 30 connections from a single IP address. When each segment completes at the very end this entry moves from the "downloads" table and into the “downloaded” table. The files in Figure 10 are still in a cached location because they need to be sorted, decoded and combined.

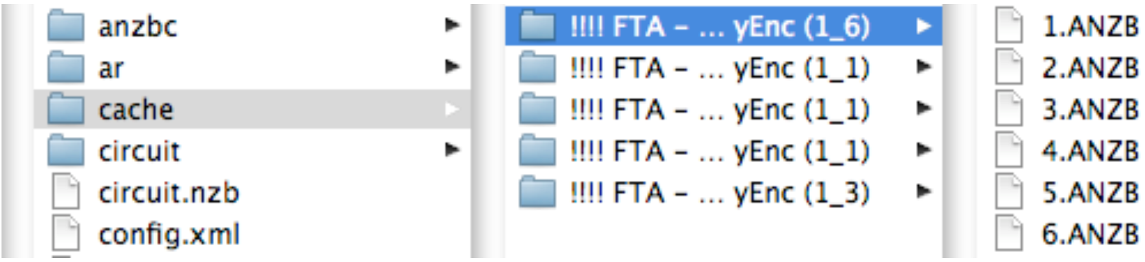
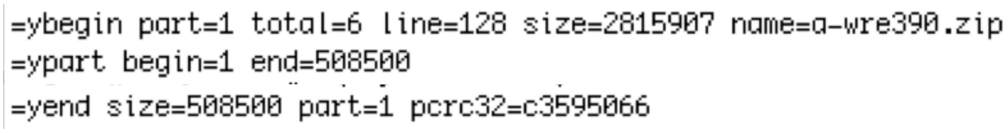

anzbc
ar
cache
circuit
circuit.nzb
config.xml
!!!! FTA - ... yEnc (1_6)
!!!! FTA - ... yEnc (1_1)
!!!! FTA - ... yEnc (1_1)
!!!! FTA - ... yEnc (1_1)
!!!! FTA - ... yEnc (1_3)
1.ANZB
2.ANZB
3.ANZB
4.ANZB
5.ANZB
6.ANZB

Figure 10: Cached segments

These files are still encoded in the yEnc format as shown in Figure 10 the first file shows the yEnc description of a multi-part file, this information also shows that it is looking for a total of 6 parts and this is part 1. Looking again at Figure 10 we can see that all 6 parts are available for the decoding process.

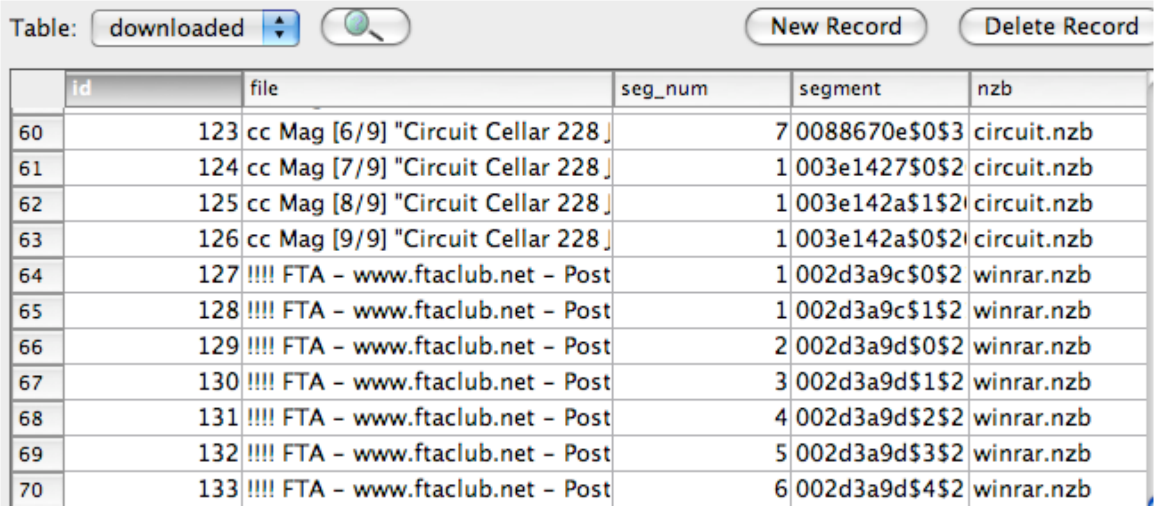


```
=ybegin part=1 total=6 line=128 size=2815907 name=a-wre390.zip
=ypart begin=1 end=508500
=yend size=508500 part=1 pcrc32=c3595066
```

Figure 10: yEnc Cached File

Decoding yEnc Files

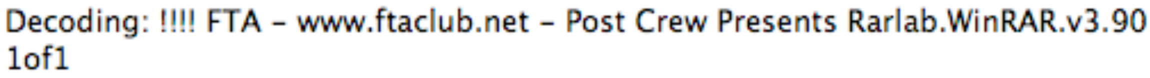
The "decode" command along with the name of the nzb file will now sort each one of the files in the cache. In order to find all of the correct files to sort, decode and combine ANZBC looks at the downloaded table in the Downloads.db database. You will notice in the nzb column that there is more than one nzb file already in cache, circuit.nzb and winrar.nzb, this is why we must again specify which one we would like to decode.



	id	file	seg_num	segment	nzb
60	123	cc Mag [6/9] "Circuit Cellar 228	7	0088670e$0$3	circuit.nzb
61	124	cc Mag [7/9] "Circuit Cellar 228	1	003e1427$0$2	circuit.nzb
62	125	cc Mag [8/9] "Circuit Cellar 228	1	003e142a$1$2	circuit.nzb
63	126	cc Mag [9/9] "Circuit Cellar 228	1	003e142a$0$2	circuit.nzb
64	127	!!!! FTA - www.ftaclub.net - Post	1	002d3a9c$0$2	winrar.nzb
65	128	!!!! FTA - www.ftaclub.net - Post	1	002d3a9c$1$2	winrar.nzb
66	129	!!!! FTA - www.ftaclub.net - Post	2	002d3a9d$0$2	winrar.nzb
67	130	!!!! FTA - www.ftaclub.net - Post	3	002d3a9d$1$2	winrar.nzb
68	131	!!!! FTA - www.ftaclub.net - Post	4	002d3a9d$2$2	winrar.nzb
69	132	!!!! FTA - www.ftaclub.net - Post	5	002d3a9d$3$2	winrar.nzb
70	133	!!!! FTA - www.ftaclub.net - Post	6	002d3a9d$4$2	winrar.nzb

Figure 13: Downloaded Table

When the decoding begins ANZBC sends the message to the GUI for each file it is working on along with the segment and total number of segments like Figure 14.



```
Decoding: !!!! FTA - www.ftaclub.net - Post Crew Presents Rarlab.WinRAR.v3.90
1of1
```

Figure 14: Decode Message

When decoding finishes all of the split files and parity files are produced. External tools can be used to check the parity and possibly complete missing blocks of data. Then Winrar or another tool can be used to decompress and combine the files creating the original source file. This completes the entire download process.

Conclusions

The ANZBC application is very successful at taking in an nzb file and parsing it to collect all of the necessary articles located on Usenet. It also downloads, decodes, and combines those articles getting the original binary split files. Along with the functionality of ANZBC, it is written in Java which allows for it to be used on several OS types, which was another one of the main goals. Where ANZBC is lacking is in its fault tolerances and error handling. As discussed earlier the NNTP protocol is not very robust and a lot of issues need to be handled. Another piece that needs to be matured is the decoding and concatenating of yEnc files. If a piece of the data is missing it should still try to piece the data together so that a parity file can successfully be used later in the process to fix the missing blocks of the file.

Future Work

Functionality

Within the same application add automatic parity checks once a file is completely downloaded. Then when a parity check is successful (whether nothing is done or blocks are filled) automatically decompress and concatenate the rar files. There should be an option for throttling the number of socket connections to a server, while allowing downloads from multiple Usenet servers at the same time to get data.

Appearance

ANZBC was built as a prototype and therefore lacks in GUI design. In the future showing the download progress of a single segment, along with the progress of that segment in the full file should be represented. Allowing for a drag-and-drop feature to load the nzb files could be added. A graphical representation of all the download files that are in the queue could be added to the GUI.

Bibliography

1. **bdesham.** [Online] http://upload.wikimedia.org/wikipedia/commons/f/f4/Usenet_servers_and_clients.svg.

2. **DMahalko.** Binaries Upload. [Online] http://upload.wikimedia.org/wikipedia/commons/7/71/Usenet_Binaries_Upload_process.PNG.

3. **Society, The Internet.** Network News Transfer Protocol (NNTP). [Online] October 2006. http://tools.ietf.org/html/rfc3977.

4. **The Parchive Project.** Parchive: Parity Tool Archive. [Online] 2001. http://parchive.sourceforge.net/.

5. **Helbing, Juergen.** yEncode - A quick and dirty encoding for binaries - Version 1.2. [Online] February 28, 2002. [Cited: September 01, 2009.] http://www.yenc.org/yenc-draft.1.3.txt.

6. **Feather, C.** Network News Transfer Protocol (NNTP). [Online] October 2006. http://tools.ietf.org/html/rfc3977.