

Email Sender with Data Compression

¹Anjali Chauhan, ²Nimesh Sakhale, ³Chinmay Desai, ⁴Sanket Deore, ⁵Rajesh Gaikwad

^{1,2,3,4}B.E- CMPN (Pursuing) Atharva College of Engineering, Mumbai University, Mumbai, Maharashtra, India

⁵Professor at Atharva College of Engineering, Mumbai University, Mumbai, Maharashtra, India

Abstract: In this world of globalization where we need to transfer large amount of data across the internet that causes need of large amount of bandwidth as well data consumption. Our model reduces the data that is to be transferred which in turn reduces the required bandwidth and time to transfer data. Our model simplifies the basic process of mail sending by skipping the time and bandwidth consuming traditional steps of using browsers for authentication and thus can be used with slow internet connections. It comes with Innovative idea of combining all the e-mail service providers like gmail, hotmail, yahoo mail, etc for mail sending in a single Interface. It automatically compresses the file/folder which user wants to send, which decreases the size of the data to be sent. It also allows user to compress/decompress the data externally for his/her local machine's files.

Keywords: Data Compression, Zip, UnZip, GZip, Deflate Compression, etc.

1. INTRODUCTION

A. Description:

E-mail Sender With Data Compression , our model is used to send the mail where user doesn't need to use browser to authenticate to send mail instead he can directly enter his authentication details in our model and select attachment , our model will automatically compress the attachment and sent the mail to receiver. Furthermore, User can explicitly also compress/decompress data and bulk send the emails.

B. Problem Formulation:

E-mails have become an important part of life, ever since its inception in 1993. In today's world, where most of the work done by all companies is online, the importance of having an e-mail is even more. The objective of our model is to simplify the usage of e-mail, by providing access to multiple e-mail accounts in one software, accessing emails on android, compressing mails, etc. As we know marketing and advertising is a big part of any company, any company can use our model to send bulk emails to targeted users instead of sending individual mails. In an age where time is money, our model sends e-mails faster, and also consumes lesser bandwidth.

C. Motivation:

When user is connected to internet using slow connection, it is very difficult to login into email service provider's site and send a mail. This process involves high data transfer and bandwidth consuming. So, we have come up with the solution where the user can insert his email address respect to email service provider and puts his authentication details. Our model also compresses the attachment and it decreases the size of the attachment, this also reduces the data transfer. Bulk email sending reduces the work of a user by sending 100s of mails in one single click.

D. Proposed Solution:

Our model makes the sending of e-mail faster even with limited internet connectivity.

E-mails can be sent in bulk, which can be used very effectively by professional organizations to advertise and market their products. Our model also saves bandwidth by compressing the mails, which also helps it being sent faster. In today's

world where time and money is everything this feature is very important. Also our app can be accessed as an Android application for easier access.

2. METHODOLOGY

A. Introduction:

Do you need the ability to add data compression to your enterprise application? Well, look no further because Java provides a package for zip-compatible data compression. The new package, `java.util.zip`, allows software developers to read, create, modify and write PKZIP and GZIP compatible files.

This package will provide number of class that are useful to read data from a ZIP file and writing data to a ZIP file. Also, provide the necessarily methods available for accessing property information for the compressed entries in a ZIP file.

The classical input/output (I/O) library in Java contains classes that support I/O handling of streams with compressed data. You can easily compress and decompress any text or binary data to or from any I/O stream using either a file or any other stream (e.g., a servlet output stream). In this project, we can compress data streams in Java with GZIP and Zip data formats.

B. The Benefits of Data Compression: using Zip package:

By using the `java.util.zip` package, you will be able to incorporate sophisticated compression technology into your Java applications with minimal work. The package not only provides single file compression but you can also create multi-file archives. Since the ZIP files created by the package adhere to the ZIP standard, you will be able to use these files with the PKZIP utility.

Smart E-mailer: You can use ZIP data compression to create a smart e-mail program. The e-mail program can be developed to have the following features:

- automatic compression of outgoing attachments
- automatic uncompression of incoming attachments

Normally, users attach very large word-processing or spreadsheet files to e-mail message. The e-mail program assists the user with compressing the file. Before sending out the message the e-mail program checks to see if the attached files are in a compressed format. If not, then the e-mail program automatically compresses the file in ZIP format.

The e-mail program also automatically unzips incoming attachments. The user simply sets a program option for unzipping files in a specific directory. Also, in order to conserve disk space, an archive feature is available to compress old e-mail messages. This will allow the user to conserve valuable disk space. The user will benefit because the e-mail program has ZIP data compression built-in. Also, the user no longer has to fumble with an external ZIP utility.

C. Gzip Compression with Java:

GZIP is a program application used for file compression. Gzip (GNU zip) is a compression utility designed to be a replacement for compress. Its main advantages over compress are much better compression and freedom from patented algorithms.

GZIP, the simplest compressing method, is ideal for situations when you have knowledge stream that you require to compress. Here they have compress and then decompress a file with the help of Java classes located in the `java.util.zip` package. The `gzip` utility makes use of Lempel-Ziv (LZ77) compression techniques to reduce the size of the specified files. For each file specified, the original file is deleted and replaced with the compressed version which has the same file name as original with the file extension `.gz` appended. The compressed file has the same possession modes, access time, and manipulation time as the original. When you do not specify any files to be compressed or specify - as a file name, `gzip` reads from the standard input, compresses what is read, and writes the result out to the standard output. The `gzip` utility won't try to compress special files.

Gzip” is often also used to refer to the `gzip` file format, which is:

D. a 10-byte header, containing a magic number, a version number and a timestamp

- E. optional extra headers, such as the original file name,
- F. a body, containing a DEFLATE-compressed payload
- G. an 8-byte footer, containing a CRC-32 checksum and the length of the original uncompressed data.

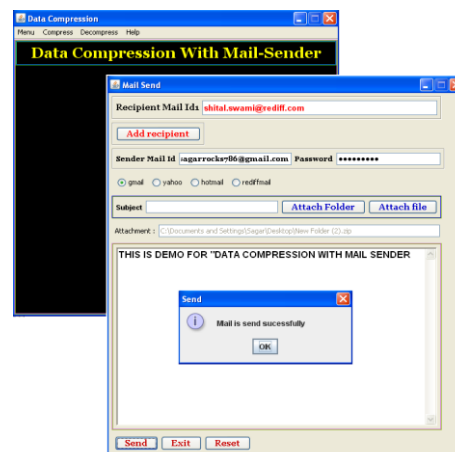
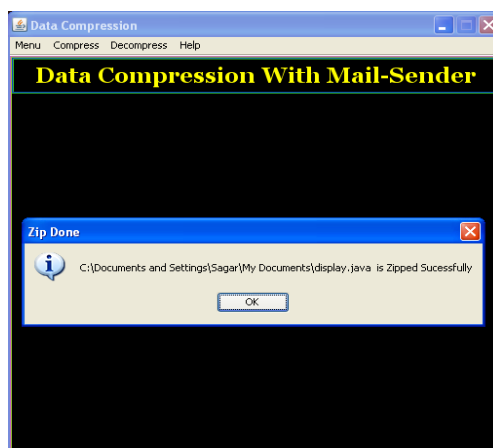
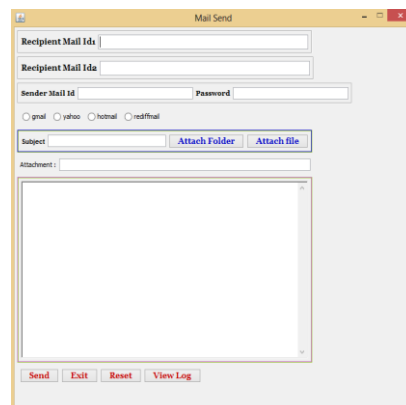
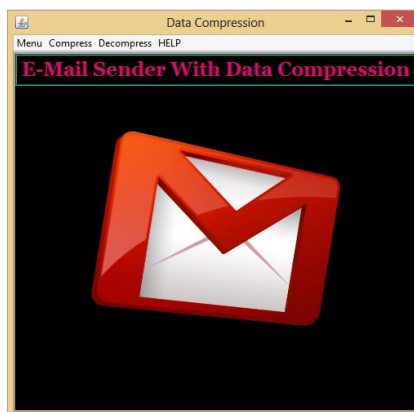
To decompress files, use `gzip -d`, `gunzip` or `gzcat`. When decompressing a file on a different file method with `gunzip` or `gzip -d`, there is a possibility that the original name for the file (if stored in the compressed by using `gzip -N`) may not be legal on the new file method. In this case, the utility decompressing the file creates a brand new legal file name from the original. Files compressed with the `gzip` utility can only be decompressed with `gunzip` in the event that they contain member and it was compressed using the 'deflation' method.

The `gzip` utility always compresses the files it reads even when the resulting compressed file is larger than the input file. The actual number of disk blocks used for a file never increases. `gzip` is based on the DEFLATE algorithm, which is a mix of LZ77 and Huffman coding. DEFLATE was intended as a replacement for LZW and other patent-encumbered knowledge compression algorithms which, at the time, limited the usability of compress and other popular archivers.

Mail Protocols: Protocols are rules that define an exact format for communication between systems. In the case of mailing systems, there are:

1. **SMTP:** - Simple Mail Transfer Protocol, transfers mail between the mail servers. It defines a delivery mechanism for email.
2. **POP:** - Post Office Protocol allows users to retrieve email when connected, and then act on the retrieved messages without needing to stay "on-line."
3. **IMAP:** - Internet Message Access Protocol. While POP is for offline access of messages, IMAP is for online access. Some of POP clients provide an option to leave copies of email on the server.

3. SCREENSHOTS



4. RESULTS AND DISCUSSIONS

A. Compressing a folder/file:

The users just need to select a folder/file which he/she wants to compress. Our model will automatically compress the folder/file using java.util.zip package in Java. The attachment path is also shown when successfully compressed. Finally the file will be compressed with minimum file size.

B. Uncompressing a folder/file:

One just has to select the folder to be uncompressed or unzip from the desired location. It automatically generates the unzipped file in the location.

One does not need to download winTool for compression. There is no need for pre-compression for the folder to be sent. Data transmission rates over the networks are high.

5. CONCLUSION

When a user is connected to the internet using a slow connection, it is very difficult to login into an email service provider's site and send a mail. This process involves high data transfer and bandwidth consuming.

So, we have come up with the solution where the user can insert his email address respect to email service provider and puts his authentication details.

It compresses the attachment and it decreases the size of the attachment, this also reduces the data transfer.

Our model can be used in corporate offices and residents where sending a mail is a crucial task.

It will be available for mobiles on android OS which will increase the usability of our model.

If a user has multiple accounts over different email providers then it is overcome by our model which gives the user access to every account within our model.

We are also developing an android app, so that sending of mails can be user from phones, on their edge connection. This is really important, especially in rural areas, as 3G connection is not available there's easily. Our model functions easily on edge data.

We have also developed a website, which offers free download of our app. The website is hosted, and is live. The application can be downloaded by anyone. The only requirement to run the application is JRE. Virtually any PC can run this application, as it only requires 128 MB RAM. It also works on Linux systems. Since the ZIP algorithm does not work on we have also provided GZIP algorithm.

ACKNOWLEDGMENT

We take this opportunity to express our sincere thanks and gratitude to Principal Dr S.P.Kallurkar of ATHARVA Engineering College, for providing the facilities and infrastructure for the project.

We also thank Prof. Mahendra Patil, Head Of Department, for his approval and support, further extending our thanks to Prof. Rajesh Gaikwad, for his constant guidance, support and critical review of the project.

We also thank the BE Computer lab assistants for providing the necessary assistance throughout the course of development.

REFERENCES

- [1] Research Paper on Text Data Compression Algorithm using Hybrid Approach Amandeep Singh Sidhu [M.Tech]¹, Er Meenakshi Garg [M.Tech]² C.S.E. & Guru Kashi University, Talwandi Sabo, Bathinda, Punjab, India
- [2] High Performance DEFLATE Compression on Intel® Architecture Processors November 2011 White Paper Vinodh Gopal Jim Guilford Wajdi Feghali Erdinc Ozturk Gil Wolrich IA Architects Intel Corporation

- [3] <http://www.oracle.com/technetwork/articles/java/compress-1565076.html>
- [4] <http://www.c-sharpcorner.com>
- [5] <http://ixrevisions.com>
- [6] <http://oasis-open.org>
- [7] <http://xmpp.org>
- [8] <http://emailaddressmanager.com>
- [9] <http://java.sun.com>
- [10] <http://www.developer.com>
- [11] <http://www.javaworld.com>
- [12] <http://www.go4expert.com>