Sending Binary Files Across Non-Transparent Comms Networks

Connect, Inc.
1701 Quincy Avenue, Suites 5 & 6, Naperville, IL 60540
Ph: (630) 717-7200 Fax: (630) 717-7243
www.connectrf.com

Introduction

The following documents the methods used to send binary files over a network that does not allow or tolerate binary data.

It has been reported that when executables and other UNIX type files are sent out over the Internet, they are frequently received corrupted. This is because compressed UNIX files are stored in a binary format, and it is possible for this data to be misinterpreted by the messaging application (e.g. e-mail package) as invalid ASCII symbols (such as the DC1 & DC3 symbols). This can screw-up the messaging protocol, resulting in the message becoming corrupted.

This problem can be avoided using the UNIX commands *uuencode* and *uudecode*, which were designed to convert binary files into a valid textual equivalent.

The following note describes a procedure that will avoid this problem. As is usual with UNIX systems, the syntax should be followed EXACTLY. For safety, it is suggested that you use a normal login rather than root.

Method for Encoding Attachment Files

The following steps should be followed in the order presented. For this example, assume that the files that you wish to send have the names *file1.HEX*, *file2.dat*, *file3.sh* and *file4*, although more (or less) files than this can be sent out together. (These files can be executables or text files.) The important lines in the following screen dump are *italicized*. Other lines are simply for information.

Description

All of the commands in this screen dump have been keyed in at the UNIX command line. The first command (ls) lists the contents of the current working directory. The next command (tar) combines the named files into a single tar file named *tarfile*. The next command (sum) gives a check-sum for the tar file. This number should be recorded in the main body of the e-mail message so that the recipient can check for possible corruption of the files during the mailing process. The next command (compress) uses maximum compression to reduce the size of the tar file and renames it with the suffix .Z. The next command (uuencode) is responsible for converting the binary into a text format file and has three arguments. The first argument is the name of the file to encode (*tarfile.Z*). The second argument is the name that the file will be given when it is later unencoded on the remote system. (This may be a different file name with or without a path, although you should bear in mind that this may result in an identically named file in the same location on the remote system being overwritten). This encoded file will be stored on the system as the file *uencoded*. (Bear in mind that DOS file names should not exceed 8 characters.) The last command uses the Z-Modem application (sz) to upload the file from the NCU into the attached PC.

Method for Decoding Attachment Files

This is achieved using the uudecode command and then reversing all of the other commands in the above sequence. If the name of the file when it is decoded on the remote system has not been reported in the e-mail message, it can be seen by using the pg command to look at the top of the file *uencoded*. The first line of the file will be in the format ‘begin *mode filename*’ with the mode in numeric format and the filename will include any pathing that was set up by the sender. This line can be modified if necessary (using a text editor) if it is unsuitable for the remote system.

Description

The first command () calls the Z-Modem Receive application to download the file onto the NCU into the current working directory. The next command (uudecode) converts this file back to a binary file giving it the name (and path) that is stated in line one of the file *uencoded*, in this case *tarfile.Z.* The next command (compress) decompresses the file also removing the .Z suffix from the file name. The next command (sum) calculates the check-sum of the file, and this should be checked against the originally stated check-sum to check for file corruption. The next command (tar) separates out the original files from the tar file, overwriting any identically named files that already exist in the working directory.

About This Document

This document is based on the following Technical Document in our Lotus Notes database that has been made obsolete: T1068.

Please let us know about any errors in this document at:
http://207.241.78.223/isoxpert/calltrak.nsf/WebTracking?OpenForm.