EditNL -- Version 4.00

NODELIST Editor Program

by Ben Baker

Table of Contents

1. Introduction

EditNL is a program designed to edit an old nodelist with a distributed difference file to create a new up-to-date nodelist. If you don't know what a nodelist or a difference file are, you probably don't have much use for this program. Detailed descriptions of these files may be found in a document file named FSC002-x.PRN ("x" is the release number) maintained by the International FidoNet Association (IFNA), and will not be belabored here.

EditNL makes the following assumptions about its input files:

1) Input files are created, published and distributed on Friday of each week.

2) An input file name has a file extension which is the day-of-year of its publication.

3) An input file may be contained within an archive file bearing the same name as the file it contains, except that the file extension's first character is an "A" instead of a digit.

4) Nodelist and difference files conform to the specifications of the above mentioned IFNA document.

These assumptions are based on the standard practice of several amateur communications networks and sub-networks presently in existence.

When executed, EditNL will search for a nodelist not more than six weeks old to edit. On finding it, it will search for younger difference files a week at a time until has either reached the current week, or cannot find a difference file for a particular week. In these searches it will, if necessary, unpack archives. Finally it will apply the difference files it found to the old nodelist to create the newest nodelist it can. If more that one difference file is involved (as many as six are possible), they are applied in parallel. No intermediate lists are created.

There are options available to tell EditNL to delete any files it unpacks from archives, or even to delete all old files, and to create a distribution archive of the newly generated nodelist file.

1.1 Notation

The following notation is used to describe EditNL's command line and its various parameters:

UPPER_CASE represents fixed items. They are entered exactly as shown, except that switch names may be abbreviated.

<lower_case> items enclosed in angle brackets represent variable items. You substitute actual values (eg. a real file name) for them.

[OPTIONAL] items are enclosed in square brackets.

CHOICE1 │ CHOICE2 Choices are separated by a vertical bar. You select one or the other, not both.

2. EditNL Command Line

EditNL is invoked with the following command line:

```
     EDITNL [/OLD=<spec>] [/NEW=<spec>] [/EDIT=<spec>]
          [/ARCHIVE[=<spec>]] [/KILL │ /CLEAN]
          [/PK[=<pkxarc_alias>[,<pkarc_alias>]]]
```

All parameters are optional, and may appear in any order. Case is unimportant. Upper, lower or mixed case may be used. Executed without parameters, EditNL will create a new NODELIST.nnn from an old NODELIST.nnn and NODEDIFF.nnn files found in the current directory.

2.1 Path/File Specifiers

<spec> is a path/file specifier, and is used to define a path for a class of files and/or override the default name stem for that class. It has the following form:

```
     <path>[\<name_stem>.]

          or

     <name_stem>.
```

In the first form, <path> defines a relative or absolute path name (see your DOS manual for more information about path names) of the directory where a class of files may be found.

<name_stem> followed by a period (.) overrides the default name stem for a class of files. "Name stem" is defined here to mean that portion of a file name, exclusive of path, which precedes

the period (.) and extension. EditNL's default name stems are NODELIST and NODEDIFF.

2.2 /OLD=<spec>

The /OLD parameter defines path to the directory where the old nodelist file may be found. It may also be used to define an alternate name stem for the nodelist class of files. (There are two ways to do this. See /NEW.)

2.3 /NEW=<spec>

The /NEW parameter defines path to the directory where the new nodelist file is to be placed. It may also be used to define an alternate name stem for the nodelist class of files. (There are two ways to do this. See /OLD.)

Note that only one nodelist name stem is used for both old and new nodelist files. If /OLD and /NEW both define <name_stem>, only the last one is used.

2.4 /EDIT=<spec>

The /EDIT parameter defines path to the directory where the difference files may be found. It may also be used to define an alternate name stem for the difference class of files.

2.5 /ARCHIVE[=<spec>]

The /ARCHIVE parameter tells EditNL to create a distribution archive containing the newly created nodelist file. If <spec> is given, it defines the directory path where the archive will be placed. (NOTE: The syntax will accept a name stem, but in this case it is ignored. The plan in a subsequent version of EditNL is to deal with both list and difference archives, hence a name stem here would be ambiguous.)

2.6 /KILL

The /KILL parameter tells EditNL to delete any input files it created by unpacking archives. If a "free standing" input file is found, it is not deleted, even if there is an equivalent archive file available.

2.7 /CLEAN

The /CLEAN parameter is similar to the /KILL parameter, but tells EditNL to "clean up the debris" by deleting all files and archives that are no longer needed on successful completion.

/CLEAN implies /KILL, so both parameters need not be specified. This gets a little tricky when a CRC error occurs. When the /CLEAN parameter is given, unpacked files are always deleted.

Free standing files and archives are deleted only if no error occurred.

2.8 /PK[=<pkxarc_alias>[,<pkarc_alias>]]

EditNL deals with archive files by using external utility programs. By default, it uses ARC from System Enhancement Associates.

Phil Katz has published a pair of utilities, PKARC for archive creation and updating, and PKXARC for archive extraction. The /PK parameter instructs EditNL to use version 3.5 of these programs instead of ARC. (Because command line syntax of these programs have presented a moving target, neither earlier nor later versions are guaranteed to work!)

It seems to be a common practice to rename these programs, giving them short one- or two-character names, hence the /PK parameter may optionally include alias names for these programs. Note that PKXARC's alias must be defined first, and if both are defined, they must be separated by a comma (,). No white space. If an alias is given for PKARC, EditNL assumes it is because you wish to create an output archive. In this case, the /ARCHIVE parameter need only be given if a path other than the /OUT path is to be used. In other words, defining an alias for PKARC forces an output archive to be generated. It will go in the same directory with the output file unless /ARCHIVE directs otherwise.

I cannot recommend the use of the Katz utilities, merely to save a few seconds (faster is not necessarily better). It has been my experience that they tend to be ill behaved in a cramped memory environment. More than once, they have crashed DoubleDOS on me, usually when called from within another program, as they are here. However, if you insist, EditNL will support you.

For compatibility with earlier versions of EditNL, /X is an alias for, and may be used interchangeably with /PK.

2.9 Abbreviations

All parameters may be abbreviated. Since they all have unique initial letters, they may be abbreviated to as few as one character. Longer representations of the parameter names will be recognized, but only if correctly spelled. In other words, any abbreviation of one of EditNL's parameters must be a proper prefix of that parameter's name.

3. Editing Error Detection

The first line of every difference file is an exact copy of the first line of the old nodelist file used to generate it. As a first level confidence check, EditNL compares these lines. If they do not compare exactly, the assumption is that the old file EditNL is using is not the old file MakeNL used to create the difference file. In that event, EditNL generates a fatal error and terminates without creating a new nodelist file.

The first line of every nodelist file contains a 5-digit number which is the decimal representation of a cyclic redundancy check word (CRC) generated by that file. EditNL accumulates a CRC on the output file as it is produced. If the generated CRC does not match the one in the first line of the file, a warning message is issued and another is appended to the output file. Earlier versions of EditNL treated a CRC error as fatal, but experience has shown that bad CRC's are frequently program errors, not data errors. However, a CRC error message is cause for concern. If an input file has been edited, the generated file could contain serious errors. You should satisfy yourself that this is not the case before using any file with a bad CRC. If there is any doubt, obtain a copy of the full nodelist.

4. Error Return Codes

On termination, EditNL sets the ERRORLEVEL return code, which can then be tested in a batch file. (See your DOS manual for more information on ERRORLEVEL return codes.)

These are the codes and their meanings:

0 A new nodelist was successfully created.
1 A CRC error was reported. Investigate before using the output file.
2 A fatal error occurred. No list created.
255 Command line error. Displays help screen.

There are four kinds of fatal error. The confidence check error mentioned in the previous section; an inconsistency in a difference file (edit command not found when expected, or an unexpected EOF); I/O errors while opening, reading or writing files; and error return codes from an archive utility program.

5. Examples

Update NODELIST.001 with NODEDIFF.008 and NODEDIFF.015 to create NODELIST.015. All files are in the current directory:

```
EDITNL
```

As above, but difference files are in archives in the directory MAILFILE. Don't keep files unpacked from the archives:

```
     EDITNL /E=MAILFILE /K
```

As above, but create an output NODELIST.A15 in the directory DOWNLOAD and delete all old files. (will not delete files in DOWNLOAD):

```
     EDITNL /E=MAILFILE /A=DOWNLOAD /C
```

As above, but file stems are ANETLIST and ANETDIFF. Use the Katz archive utilities:

```
     EDITNL /N=ANETLIST. /E=MAILFILE\ANETDIFF. /A=DOWNLOAD /C /PK
```

6. The Shareware Concept

EditNL is distributed under the "shareware" concept. It may be freely copied and distributed provided the program and its documentation are distributed as a unit. But EditNL is NOT FREE.

If you are a non-commercial user of EditNL, you are asked to support its development and maintenance both with your suggestions and bug reports and with your dollars. No particular fee is set for non-commercial use. Let your conscience be your guide.

Commercial users, and users participating in a commercial network are required to pay a license fee of twenty five (25) U. S. dollars for each computer on which EditNL is installed for use.

The Coordinator of a commercial network may obtain a network license for a one-time fee of $1,500 US. This will license all current and future nodes of the network for the use of EditNL. MakeNL and EditNL may be licensed as a set for a one-time fee of $2,250.

Please address all correspondence to:

Ben Baker
Baker's Acre
RR 1, Box 637
E. Alton, IL 62024