

CreatePkt

COLLABORATORS

	<i>TITLE :</i> CreatePkt		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 17, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	CreatePkt	1
1.1	CreatePkt.guide	1
1.2	Purpose of this program	1
1.3	Disclaimer and License Agreement	2
1.4	Usage	2
1.5	Advanced Options	3
1.6	Example	3
1.7	How to contact the author	4
1.8	Program History	4

Chapter 1

CreatePkt

1.1 CreatePkt.guide

CreatePkt 2.0
Copyright © 1994 Johan Billing

Purpose

Disclaimer and License Agreement

Usage

Example

How to contact the author

Program history

1.2 Purpose of this program

Purpose of this program

=====

If you run a FidoNet point or node, perhaps you would like messages to be written to you to notify you of certain things. There are two ways to achieve this:

- 1) Write a message directly to the messagebase. This can be quite a big problem if you aren't using #?.msg and if the format of your messagebase is undocumented.
- 2) Create a standard FidoNet #?.PKT file that can be processed by your FidoNet point program or tosser. This is exactly what CreatePkt does.

CreatePkt can of course also be used in any other net using the same packetformat.

1.3 Disclaimer and License Agreement

Disclaimer and License Agreement
=====

Well, I do not guarantee that this program will work or that it won't make your HD go berzerk. It shouldn't be the case, as the program seems quite reliable, but if the worst should happen - Don't blame me! It's quite impossible, though, for plenty of bugs to sneak into a short program like this.

This program is released as FreeWare.
Use it, give it to your friends, upload it to any BBS, but DO NOT SELL IT FOR PROFIT! This is the programmer's explicit wish and you really should respect it!

1.4 Usage

Usage:

=====

Command template: FROMNAME=FN/K, FROMADDR=FA/K/A, TONAME=TN/K, TOADDR=TA/K/A,
SUBJECT=SUBJ/K, AREA/K, ORIGIN/K, PKTFROM=PF/K, PKTTO=PT/K,
DIR/K, TEXT/A

FROMNAME: The sender's name (maximum 35 characters).
Default: CreatePkt

FROMADDR: The sender's address in standard zone:net/node[.point] format.

TONAME: The recipient's name (maximum 35 characters).
Default: All

TOADDR: The recipient's address.

SUBJECT: Subject line of the message (maximum 71 characters).
Default: Information

AREA: The area in which the message should be written.
If this is omitted, a netmail will be generated.

ORIGIN: The Originline of the message. This will of course only be used
in Echomail areas.

DIR: The directory in which you would like the PKT-files to be created.
Default: MAIL:Inbound

TEXT: The textfile you wish to have sent.

Advanced options

1.5 Advanced Options

In order to understand these options, you need some knowledge about FidoNet packets. Being an ordinary point you most probably will not have to care about these switches, and can therefore just skip this part. ←

PKTFROM: This sets the origin address in the packet header. If you are a normal point, you shouldn't need to worry about this. If you are a sysop and run a tosser which is picky about these things, you may need to use this switch. E.g. you could set it to the address of your Hub to make your tosser accept the packet. Only use this if it is really necessary.

Default: The address entered as FROMADDR

PKTTO: This is the destination address in the packet header and has been implemented for the same reason as PKTFROM. If you want to create a netmail message to a node and your tossers only accepts packets that are addressed to your node, you will have to set PKTTO to your own address.

Default: The address you entered as TOADDR.

If you have any problems using either these or the other keywords don't hesitate to contact me (see the chapter for further information how to do this).
How to contact the author

1.6 Example

Example
=====

Let's say you want to write a netmail message to yourself everytime your system processes a new nodediff. This example will show you how to do it.

First of all, we assume that you have a script which is started everytime you process a nodelist. This script could, if you're using TrapList, look something like this:

--- Cut ---

.key diff

```
c:LhA x <diff> ; extract the file
copy amyndiff.[0-9]?? nodelist: ; copy the extracted file to your
; nodelist directory
TrapList ; process the new nodediff
```

--- Cut ---

Now, if you for some reason don't receive your nodediffs for a while you most certainly won't have a clue about this unless something goes wrong. E.g., it's not a very good thing to call a node which for some reason, perhaps he has left the Net and now uses that line for his voicecalls, is down. This is an excellent example where CreatePkt can help you realize something is wrong so that you in your turn can tell your uplink.

Just add the following line to your script, and - voilà! Everytime you receive a nodediff a message will be written to inform you.

--- Cut ---

```
CreatePkt FA 2:200/207.6 TA 2:200/207.6 FN TrapList TN "Johan Billing"  
SUBJECT "New nodediff" NODELIST:TrapList.report
```

--- Cut ---

Note: The above must appear on one line, otherwise it won't work.

1.7 How to contact the author

How to contact the author
=====

FidoNet: Johan Billing@2:200/207.6
InterNet: johan.billing@kcc.ct.se

SnailMail: Johan Billing
Östra Storgatan 22
260 60 Kvidinge

1.8 Program History

History
=====

1.00 - First release

1.01 - Buxfix: CreatePkt no longer generates corrupt dates.

2.0 - Almost everything is rewritten to use the new functions in Release 2 of the operating system. The program is now almost 4 KB smaller and uses standard arguments with ReadArgs(). Please note that because of this CreatePkt 2.0 is incompatible with earlier released versions.

- Now you can set the From/To addresses in the packet header too if you for some reason need it/them to differ from the addresses of the message itself.

-
- CreatePkt is a bit safer when generating packets.
 - CreatePkt now prints an error message if an invalid node number is passed on as an argument. In earlier versions, CreatePkt would just parse the nodenumber to 0:0/0.0.
 - CreatePkt now returns an error (errorcode 10) when it fails.
 - CreatePkt can be made resident.
 - CreatePkt now uses 0xfe as product code instead of 0x00.
-