

HTICK 1.9 manual

Husky Ticker

Georgi Fofanov @ 2:5050/29, fg@fg.udm.ru
Nestyurkin Nikolay @ 2:5020/3600, 3600@mail.ru
Max Chernogor @ 2:464/108, mche@ua.fm

Table of Contents

HTICK	1
1 An Overview of HTICK	3
2 Installation Procedures and Release Notes	5
2.1 Download the Source Code & Binary Files	5
2.2 Compiling the Source Code	5
2.3 Support	5
3 HTICK 1.9 Command Line OPTIONS and SWITCHES	7
4 HTICK 1.9 Configuration Reference	9
4.1 Keywords	9
4.1.1 FileAreaBaseDir	9
4.1.2 PassFileAreaDir	9
4.1.3 FileDescPos	9
4.1.4 fileLDescString	9
4.1.5 FileFixHelp	9
4.1.6 ExecOnFile	10
4.1.7 SaveTic	10
4.1.8 MaxTicLineLength	10
4.1.9 BusyFileDir	10
4.1.10 ConvertLongNames	10
4.1.11 ConvertShortNames	11
4.1.12 addDLC	11
4.1.13 DLCDigits	11
4.1.14 ticOutbound	11
4.1.15 filefixKillRequests	11
4.1.16 filefixReportsAttr	11
4.1.17 BBSarea	12
4.1.18 fileDescription	12
4.1.19 fileDescName	12
4.1.20 Note	13
4.1.21 FileAreaCreatePerms	13
4.1.22 FileFixFromName	13
4.1.23 AutoFileCreateFlag	13
4.1.24 FileFixNames	13
4.2 Link Keywords	13
4.2.1 TicPwd	13
4.2.2 fileEchoFlavour	14

4.2.3	fileAreaDefaults	14
4.2.4	RemoteFileRobotName	14
4.2.5	noTIC	14
4.2.6	autoFileCreate	14
4.2.7	AutoFileCreateFile	14
4.2.8	AutoFileCreateSubdirs	15
4.2.9	delNotReceivedTIC	15
4.2.10	FileFixFSC87Subset	15
4.2.11	TickerPackToBox	15
4.2.12	LinkFileBaseDir	15
4.2.13	ForwardFileRequests	15
4.2.14	ForwardFilePriority	16
4.2.15	ForwardFileRequestFile	16
4.2.16	FileFix	16
4.2.17	FileFixEchoLimit	16
4.3	FileArea Definition	16
4.3.1	FileArea	16
4.4	Announce System	18
4.4.1	AnnounceSpool	18
4.4.2	AnnAreaTag	18
4.4.3	AnnInclude	18
4.4.4	AnnExclude	18
4.4.5	AnnTo	18
4.4.6	AnnFrom	19
4.4.7	AnnSubj	19
4.4.8	AnnAddrTo	19
4.4.9	AnnAddrFrom	19
4.4.10	AnnOrigin	19
4.4.11	AnnMessFlags	19
4.4.12	AnnFileRFrom	19
4.4.13	AnnFileOrigin	20
4.4.14	Announcements example	20
4.4.15	Common husky keywords	21

Appendix A Configuration File Keyword Index .. 23

HTICK

This document describes HTICK 1.9, a Fidonet Fileecho Processor for OS/2, Windows, BeOS and Unix clones.

Because HTICK is a part of the HUSKY fidonet software project, it uses the `fidoconfig` library and many configuration statements described in a *Fidoconfig manual*.

1 An Overview of HTICK

HTICK is a Fidonet fileareas tosser with filefix. Now project is being supported by Husky Development Team (see <http://husk.sf.net/team.html>).

HTICK features are:

1. Tossing fileareas.
2. Filefix (on the fly, from command line, limit for areas...).
3. Autocreating on the fly.
4. Automatic forwarding requests for missing areas to the uplinks.
5. Switching areas into passive (send-only) and back into active mode.
6. Resending files from passthrough and none passthrough fileareas.
7. Groups & levels for personal and public access to fileareas.
8. Security check of tossgroup.
9. Link defaults.
10. Posting to net & echo areas.
11. Flexible file announcements via echo or netmail messages. Wildcards in filearea tags allow easy configuration of multiple announcement areas for different groups of fileareas.
12. Save announcements to disk, so you are protected from losing announcements because of critical HTICK termination (insufficient disk space, ^Break pressed, power off etc).
13. much, much more :)

The advantages of HTICK are:

1. Open Source (GPL)
2. Many supported platforms & operating systems
3. Quick bug fixing

2 Installation Procedures and Release Notes

This chapter provides you with information that is necessary to successfully install and use HTICK.

We suppose that you already has compiled binaries. If not - read "Download" or "Compile the Source Code" chapters.

1. Read FIDOCONFIG documentation about location of config files and the statements of fidoconfig.
2. Create and edit config files for your purposes.
3. Run `tparser` from FIDOCONFIG package to test your config (read about `PublicGroup` or `AccessGrp` if you want to use groups for `FileEchoAreas`).
4. It is simply, isn't it? Enjoy! :-).

2.1 Download the Source Code & Binary Files

Main page (releases, source code, latest beta versions) -

<https://github.com/huskyproject/htick>

Direct link to 'stable' branch sources: <http://husky.sf.net/cvs2/stable/>

Direct link to 'current' (development) branch sources: <http://husky.sf.net/cvs2/current/>

2.2 Compiling the Source Code

1. The `smapi` and `fidoconf` packages are required for `htick`.
2. Put the `fidoconf` and `smapi` packages to the directory where the other packages of `fido` linux reside:

```
/usr/src/packages/  
-> smapi/  
-> fidoconfig/  
-> htick/
```

3. Compile and install `smapi` and `fidoconf` packages. Use "Makefile" for dynamic executables and `makefile.lnx` (or whatever you need) for static ones.

4. Compile and install HTICK:

```
$ make
```

```
$ make install
```

You should use the `_same_` makefiles in `smapi`, `fidoconf` and `htick`.

2.3 Support

There are numerous reasons why you might wish to establish contact with developers.

1. You have decided to use HTICK on a regular basis. In this case, please do send a netmail at the address listed below.
2. You have a general questions on how to configure or on how to use a certain feature of HTICK. In other words, you need support. In this case, you'd best post your question to one of the following echos:

`FIDOSOFT.HUSKY`

The international Husky conference. English is the preferred language here.

`RU.HUSKY` This Russian echo covers Husky Project.

If you have access to Internet, you may of course post your support request or bug report into form on husky development site: <http://sourceforge.net/projects/husky>, click the "Bugs" link if you want to post a bug report.

If you do not have access to any of these echos and internet, you may of course also contact developers via netmail or e-mail at the addresses listed below.

3. You want to report a bug. There are two sorts of bugs:
 - a. Normal bugs. You think that a certain function of HTICK does not work as expected, e.g. it is producing garbage, or doing strange things, or similar. In this case, either post to the echos listed above, bugrack on site or contact developers via netmail. Please do supply all information that is necessary to understand your problem.
 - b. Fatal bugs. A fatal bug occurs if HTICK crashes. Depending on your operating system, the symptom might be a core dump, or a SYS 3175, or a general protection fault, or a system lockup, or a spontaneous reboot. We do consider a crash intolerable. No matter how stupid things you do, you should not be able to crash HTICK.

If you are experienced user and get core dump, you can send developers gdb report. If you have a crash, locate `core` file that has been generated. Then run `$ gdb htick core`, type `where`. HTICK must be compiled with debug information (`DEBUG=1` in `huskymak.cfg` file). Then send report to addresses below.

If you are running any other binary version (like Windows), you will not get a core file on a crash. Write down as much information as you can, try to find a way to reproduce the crash and contact autors at the addresses below. Please attach log-file with `loglevels 1-0A-Za-z` (all numbers and letters).

4. You want to contribute to HTICK. If you are a programmer and have fixed a problem in HTICK on your own, please submit your changes to me. The preferred way for doing so is to send to developers a difference file in GNU diff format (with `-c` parameter). Your work will be highly appreciated and honored in an appropriate place. If you want to regularly work on HTICK, we also have a CVS server online that you can have access to if you like. If you want to write a new feature for HTICK, please contact developers beforehand to avoid that we do duplicate work. Again, we will appreciate and honor any efforts done by you. Please note that for writing a HTICK enhancement, you should be familiar with C. Also, HTICK uses a special indentation style throughout the source code, that we would like you to adhere to.

So here are developers' addresses if you want to get in contact with us:

- Fidonet: Max Chernogor @ 2:464/108
- e-mail: mche@ua.fm
- Fidonet: Stas Degteff @ 2:5080/102
- e-mail: stas_degteff@users.sf.net

3 HTICK 1.9 Command Line OPTIONS and SWITCHES

To get help about the command line syntax use HTICK without any arguments

Syntax: `htick [options] <command>`

where <command> is one of:

- `toss [annfecho <file>]`
Reading *.tic and tossing files. Optionally announce new fleechos in text file.
- `toss -b` Toss bad tics.
- `scan` Scan netmail area for mails to filefix and process filefix commands contained inside the mails
- `ffix <FTN address> <command>`
Process filefix command <command> for <FTN address> from command line
- `ffix! <FTN address> <command>`
Do the same job as 'ffix' and additionally notify the link of the processed filefix command as in example:
`htick ffix! 2:464/910 -910.TEST`
This command removes file echo link 2:464/910 from 910.TEST fleecho and produces a message to 2:464/910 with body "-910.TEST"
- `relink <pattern> <addr>`
In order to refresh subscription send to filefix at address <addr> commands to subscribe for fleechos with names matching <pattern>.
- `relink -f <file> <addr>`
In order to refresh subscription send to filefix at address <addr> commands to subscribe for fleechos with names contained in <file>.
- `resubscribe <pattern> <fromaddr> <toaddr>`
Move subscription of fileareas matching <pattern> from link with address <fromaddr> to another link with address <toaddr>
- `resubscribe -f <file> <fromaddr> <toaddr>`
Move subscription of fileareas listed in <file> from link with address <fromaddr> to another link with address <toaddr>
- `clean` Clean passthrough directory (remove sent files) and remove old files in fleechos
- `announce` Announce new files as defined in Ann* config statements (See Section 4.4 [Announce System], page 18.)
- `hatch <file> <area> [replace [<filemask>]] [desc <desc> [<ldesc>]]`
Hatch <file> into <area> optionally using description for the file; if `replace` is present in the command then add a replace field to TIC; if <filemask> is not present in the command then put <file> in the field.
A short (one line) description <desc> and a long description <ldesc> may be put to TIC.
One may use the following as <desc>:
`@@BBS` to use first line from files.bbs

`@@DIZ` to use first line from the contained file_id.diz
`@@<file>` to use first line from <file>
`@BBS` to load description from files.bbs
`@DIZ` to load description from the contained file_id.diz
`@<file>` to load description from <file>
and the last three items may be used as <ldesc>.

`send <file> <filearea> <address>`
Send <file> from <filearea> to the <address> with accompanying .tic

`filelist <file> [<dirlist>]`
Generate filelist containing all files in base.

- <file> filename to write output to.
- <dirlist> filename to which only list of fileecho directories is written.

and where options are:

`-q` Quiet mode (display only urgent messages to console)
`-c <config-file>`
Specify an alternative configuration file

4 HTICK 1.9 Configuration Reference

HTICK is based on FIDOCONFIG library, so read documentation of FIDOCONFIG about location of config file, keywords ideology and about majority of the keywords, See *Fidoconfig manual*.

4.1 Keywords

HTICK is based on FIDOCONFIG library, so read documentation of FIDOCONFIG about location of config file, keywords ideology and about majority of the keywords, See *Fidoconfig manual*.

4.1.1 FileAreaBaseDir

Syntax: `FileAreaBaseDir <directory>`

Example: `FileAreaBaseDir /var/spool/fido/fileareas`

Where the directories for all Fileareas shall be made at autocreate. Same as MsgBaseDir.

This statement cannot be repeated.

4.1.2 PassFileAreaDir

Syntax: `PassFileAreaDir <directory>`

Example: `PassFileAreaDir /var/spool/fido/transit`

Where shall be put files from Passthrough Fileareas (and TICs, if ticOutbound is not defined).

This statement cannot be repeated.

4.1.3 FileDescPos

Syntax: `FileDescPos <number>`

Example: `FileDescPos 13`

The number of spaces to indent long descriptions in FILES.BBS. Recommend: 13 (8.3+space)

This statement cannot be repeated.

4.1.4 fileLDescString

Syntax: `fileLDescString <string>`

Example: `fileLDescString ">"`

This string is put before long descriptions in FILES.BBS (Usually ">" or "+"). Default value: single space.

This statement cannot be repeated.

4.1.5 FileFixHelp

Syntax: `FileFixHelp <string>`

Example: `FileFixHelp /usr/local/fido/filefix.hlp`

Help file for FileFix, same as AreaFixHelp for HPT's AreaFix

This statement cannot be repeated.

4.1.6 ExecOnFile

Syntax: `ExecOnFile <filearea> <filemask> <command>`

Example: `ExecOnFile husky *.diff /usr/local/fido/update_source.sh`

Execute some command on receiving file. Command gets full filepath as parameter (space and file pathname should be appended to command string before execution).

This statement can be repeated.

4.1.7 SaveTic

Syntax: `SaveTic <FileEchoMask> <FileOption> <DirToSaveTic> <DaysToKeepTics>`

Example: `SaveTic husky -l /usr/local/husky 365`

<DirToSaveTic>

where to put TICs for fileecho[s] <FileEchoMask>

<FileOption>

save files with tics. May be:

empty files will not be copied

-l link files from fileechodir to <DirToSaveTic>

-c copy files from fileechodir to <DirToSaveTic>

Options -l and -c work only for non-Passthrough fileechos

<DaysToKeepTics>

<DaysToKeepTics> is optional. If it's defined tics saved in <DirToSaveTic> will be kept for <DaysToKeepTics> days.

This statement can be repeated.

4.1.8 MaxTicLineLength

Syntax: `MaxTicLineLength <number>`

Example: `MaxTicLineLength 79`

Set maximum line length in outgoing TICs to number.

This statement cannot be repeated.

4.1.9 BusyFileDir

Syntax: `BusyFileDir <directory>`

Example: `BusyFileDir /var/spool/fido/htick.busy/`

Where shall be put TICs for busy links (default <outbound>/busy.htk).

This statement cannot be repeated.

4.1.10 ConvertLongNames

Syntax: `ConvertLongNames <Upper|Lower|DontTouch>`

Example: `ConvertLongNames DontTouch`

What to do with long names when toss or hatch. Saying 'long name' we mean a mixed cased name or a name longer than 8.3

This statement cannot be repeated.

4.1.11 ConvertShortNames

Syntax: `ConvertShortNames <Upper|Lower|DontTouch>`

Example: `ConvertShortNames DontTouch`

What to do with short names when toss or hatch. Saying 'short name' we mean a DOS-like name: 8.3, without mixed case.

This statement cannot be repeated.

4.1.12 addDLC

Syntax: `addDLC <bool>`

Example: `addDLC yes`

Add Download counters ([000]) to files.bbs. Use with DLCDigits only!

This statement cannot be repeated.

4.1.13 DLCDigits

Syntax: `DLCDigits <number>`

Example: `DLCDigits 4`

How much digits DLC must be. DLC has [000] form.

This statement cannot be repeated.

4.1.14 ticOutbound

Syntax: `ticOutbound <directory>`

Example: `ticOutbound /var/spool/fido/ticOub`

TICs are stored here and killed after transmission. Default: use PassFileAreaDir value

This statement cannot be repeated.

4.1.15 filefixKillRequests

Syntax: `filefixKillRequests <bool>`

Example: `filefixKillRequests yes`

Kill filefix request after processing.

This statement cannot be repeated.

4.1.16 filefixReportsAttr

Syntax: `filefixReportsAttr <attr>`

Example: `filefixReportsAttr pvt k/s loc`

or `filefixReportsAttr pvt,k/s,loc`

Set flags to filefix replies.

Valid attributes are:

pvt

crash

read

sent

att
 fwd
 orphan
 k/s
 loc
 fwd
 xx2
 frq
 rrq
 cpt
 arq
 urq
 kfs
 tfs
 dir
 imm
 cfm
 npd

Default is "pvt loc k/s npd".

This statement cannot be repeated.

4.1.17 BBSarea

Syntax: `BBSarea <name> <bbsareapath>`

Example: `bbsarea bbs /var/spool/fido/bbsareas/bbs`

Filearea without subscribing, not shown in filefix reports. Use for filelist generation only ('htick filelist' command).

This statement cannot be repeated.

4.1.18 fileDescription

Syntax: `fileDescription <name>`

Example: `fileDescription descript.ion`

Name of file to hold file descriptions, by default files.bbs

This statement can be repeated.

4.1.19 fileDescName

Syntax: `fileDescName <name>`

Example: `fileDescName file_id.diz`

If LDESC is not defined in tic file, use <name> file from archive to fill it (usually file_id.diz).

This statement can be repeated.

4.1.20 Note

`unpack` action should be defined for archive and contain `$f` parameter (`$f` is replaced with file name while unpacking).

Examples:

```
Unpack "/path/to/unrar e -y -c- -o+ -inul $a $p $f" 0 52617221 # *.rar
Unpack "d:\path\unrar e -y -c- -o+ -inul $a $p $f" 28 52534658 # rar-sfx/DOS
Unpack "\path\pkzipc -ext -dir=none -over=all -nozip -silent $a $p $f" 0 504b0304
Unpack "unzip -joLqq $a -d $p" 0 504b0304
Unpack "pkunzip -njCL $a -d $p $f" 0 504b0304
Unpack "pkzip25 -ext -silent -over=all -nofix -nozip $a $p $f" 0
```

4.1.21 FileAreaCreatePerms

Syntax: `FileAreaCreatePerms <mode>`

Example: `FileAreaCreatePerms 644`

Define permissions for newly created filearea directories (UNIX only)

This statement cannot be repeated.

4.1.22 FileFixFromName

Syntax: `FileFixFromName <string>`

Example: `FileFixFromName FileFix`

Originator name in filefix replies.

This statement cannot be repeated.

4.1.23 AutoFileCreateFlag

Syntax: `AutoFileCreateFlag <file>`

Example: `AutoFileCreateFlag /usr/local/fido/flags/new_filearea`

Create flag when filearea has been autocreated.

This statement cannot be repeated.

4.1.24 FileFixNames

Syntax: `FileFixNames <string>`

Example: `FileFixNames allfix filefix htick`

Set of names separated by space on which FileFix will respond

This statement cannot be repeated.

4.2 Link Keywords

HTICK is based on FIDOCONFIG library, so read documentation of FIDOCONFIG about location of config file, keywords ideology and about majority of the keywords, See *Fidoconfig manual*.

4.2.1 TicPwd

Syntax: `TicPwd [<string>]`

Example: `TicPwd MySeCrET`

For full description look Section "ticpwd" in *Fidoconfig manual*

4.2.2 fileEchoFlavour

Syntax: `fileEchoFlavour <hold|normal|crash|direct|immediate>`

Example: `fileEchoFlavour direct`

This statement sets the flavour which outgoing files in fileechos for this link get. For example set `fileEchoFlavour` to hold for points and to crash for uplinks.

This statement can only be repeated for different links.

4.2.3 fileAreaDefaults

Syntax: `fileAreaDefaults [<options> [links]]`

Example: `fileAreaDefaults -lw 100 -g Z 2:5097/31.1`

Set default options for filearea definitions that follow. All options are possible except `areatag` and `path`. Settings in a filearea definition override the ones in `fileAreaDefaults`. A `fileAreaDefaults` setting is valid until a next `fileAreaDefaults`. Defaults are switched off with empty `fileAreaDefaults`.

See also Section “EchoAreaDefaults” in `fidoconfig`.

This statement can only be repeated for different links.

4.2.4 RemoteFileRobotName

Syntax: `RemoteFileRobotName <name>`

Example: `RemoteFileRobotName allfix`

Name of remote filefix robot (needed for `FileForwardRequest`)

This statement can only be repeated for different links.

4.2.5 noTIC

Syntax: `noTIC <bool>`

Example: `noTIC yes`

Disable TIC File Creation

This statement can only be repeated for different links.

4.2.6 autoFileCreate

Syntax: `autoFileCreate <on|off>`

Example: `autoFileCreate on`

Allow or deny autocreating fileechos received from this link.

This statement can only be repeated for different links.

4.2.7 AutoFileCreateFile

Syntax: `AutoFileCreateFile <file>`

Example: `AutoFileCreateFile /usr/local/fido/hpt/config.filefix`

Specifies the file new filearea definitions will be placed into

This statement can only be repeated for different links.

4.2.8 AutoFileCreateSubdirs

Syntax: `AutoFileCreateSubdirs <on|off>`

Example: `AutoFileCreateSubdirs on`

Whether to autocreate areas like `/filebase/gfd.app.edit` (off) or `/filebase/gfd/app/edit` (on). If you want to configure this globally instead of on per-link basis, set this in the `deflink` section. For more information see 'AutoAreaCreateSubdirs' in the `hpt` manual.

This statement can only be repeated for different links.

4.2.9 delNotReceivedTIC

Syntax: `delNotReceivedTIC <on|off>`

Example: `delNotReceivedTIC on`

Remove TIC if file was not received

This statement can only be repeated for different links.

4.2.10 FileFixFSC87Subset

Syntax: `FileFixFSC87Subset <on|off>`

Example: `FileFixFSC87Subset on`

This boolean switch will change on a per link basis how TIC files are processed. By default this switch is ON. `htick` will then operate only on FSC87-compliant keywords and ignore unknown keywords in the TIC file processed. If this switch is set to OFF all implemented keywords will be used which can (and, in interaction with some tick processors, WILL) lead to severe problems. Unknown keywords found in a TIC file will make `htick` stop processing that file. If you don't know what all this is about set `FileFixFSC87Subset` to ON for all your links. It won't hurt.

This statement can only be repeated for different links.

4.2.11 TickerPackToBox

Syntax: `TickerPackToBox <on|off>`

Example: `TickerPackToBox on`

If link has file box - toss files and tics there

This statement can only be repeated for different links.

4.2.12 LinkFileBaseDir

Syntax: `LinkFileBaseDir <directory>`

Example: `LinkFileBaseDir /usr/local/fido/fileareas/2.5097.64.0`

Where autocreated fileareas will be placed to (may be "passthrough")

This statement can only be repeated for different links.

4.2.13 ForwardFileRequests

Syntax: `ForwardFileRequests <on|off>`

Example: `ForwardFileRequests on`

Allow fileareas forward-requests to this link.

This statement can only be repeated for different links.

4.2.14 ForwardFilePriority

Syntax: `ForwardFilePriority <number>`

Example: `ForwardFilePriority 1`

Priority in uplinks order for filearea subscribing forward-requests

This statement can only be repeated for different links.

4.2.15 ForwardFileRequestFile

Syntax: `ForwardFileRequestFile <file>`

Example: `ForwardFileRequestFile /usr/local/fido/hpt/uplink1.lst`

List of available fileareas from this link

This statement can only be repeated for different links.

4.2.16 FileFix

Syntax: `filefix <bool>`

Example: `filefix off`

By default filefix is "on". You can turn off using of filefix by this link.

This statement can only be repeated for different links.

4.2.17 FileFixEchoLimit

Syntax: `FileFixEchoLimit <number>`

Example: `FileFixEchoLimit 10`

Area subscribe limit (max fileareas number this link is allowed to subscribe to)

This statement can only be repeated for different links.

4.3 FileArea Definition

HTICK is based on FIDOCONFIG library, so read documentation of FIDOCONFIG about location of config file, keywords ideology and about majority of the keywords, See *Fidoconfig manual*.

4.3.1 FileArea

Syntax: `FileArea <name> <fileareapath> [options] <uplink> <downlinks>`

Example: `FileArea husky /var/spool/fido/fileareas/husky -a 2:5097/31 -p 28 -g Z -d "HUSKY fileecho" 2:5097/64 2:5097/303`

Same as EchoArea, most options are valid:

-a <address>

Our own address for this area.

-d "<string>"

Description for this filearea.

-g <group>

Group for this filearea.

-hide

Hide area (don't show in %LIST and in an announcement).

-nohide

Show area in %LIST and in an announcement if fileareadefaults set to `-hide`.

- lr <number>**
 "Read level": min link level for sending files from this filearea to this link. Value can't be negative.
- lw <number>**
 "Write level": Min link level allowed to post into this filearea files received from link. Value can't be negative.
- mandatory**
 Disallow remote unsubscribe (disable unsubscribe filefix command).
- nomandatory**
 Allow remote unsubscribe (enable unsubscribe filefix command) if fileareadefaults set to **-mandatory**.
- manual** Disallow remote subscribe (disable subscribe filefix command)
- nomanual** Allow remote subscribe (enable subscribe filefix command) if fileareadefaults set to **-manual**.
- noCRC** Disable CRC check for incoming files.
- CRC** Enable CRC check for incoming files if fileareadefaults set to **-noCRC**.
- nodiz** Do not try to get file description from <fileDescName> (for packed files).
- diz** Try to get file description from <fileDescName> (for packed files) if fileareadefaults set to **-nodiz**.
- pause** %PAUSE has effect to this filearea.
- nopause** %PAUSE has no effect to this filearea if fileareadefaults set to **-pause**.
- noreplace** Don't replace existing files in this filearea.
- replace** Replace existing files in this filearea if fileareadefaults set to **-noreplace**.
- norename** Don't rename new file if name already exists in filearea (default, if not overwritten in fileareadefaults).
- rename** Rename new file if name already exists in filearea. The file will be renamed only in the storage directory, not for further distribution to other nodes.
 Renaming scheme:
- long filename
 file.more.ext -> file.<number>.more.ext
 e.g. long-name.tar.gz -> long-name.01.tar.gz
 - DOS style filename
 file.ext -> file.<1st char of ext><number>
 e.g. file.zip -> file.z01
 - <number> is 01 - 99
- p <integer>**
 Purge after n days. Setting to 0 to disables purging (this is default value).
- sendorig** Send files from PassFileAreaDir for non passthrough areas.
- nosendorig**
 Send files from FileArea directory for non passthrough areas if fileareadefaults set to **-sendorig**.

- r Default read-only: set link read-only at subscribing.
- w Default write-only: set link write-only at subscribing.

4.4 Announce System

Htick announcements are selective, they can be different for different fileareas. Token **AnnAreaTag** starts a new section of announce definition, other tokens define lists of fileareas included into announcement and describe announce message.

To post announcements run **htick announce**.

You may turn announcements off: just remove **AnnounceSpool** definition from config.

HTICK is based on FIDOCONFIG library, so read documentation of FIDOCONFIG about location of config file, keywords ideology and about majority of the keywords, See *Fidoconfig manual*.

4.4.1 AnnounceSpool

Syntax: **AnnounceSpool** <path>

Example: **AnnounceSpool** /var/spool/fido/announce

Directory for accumulating report information.

This statement cannot be repeated.

4.4.2 AnnAreaTag

Syntax: **AnnAreaTag** <AREATAG|@filename>

Example: **AnnAreaTag** local.announce

Begins new announcement definition section. <AREATAG> is name of an area where announcement will be placed to; <@filename> is filename with leading '@' where announcement will be placed to.

This statement can be repeated.

4.4.3 AnnInclude

Syntax: **AnnInclude** <fileecho mask list>

Example: **AnnInclude** *

Array of fileecho names|masks that will be announced in this annAreaTag. Default: all fileareas.

This statement can be repeated only for different AnnAreaTag.

4.4.4 AnnExclude

Syntax: **AnnExclude** <fileecho mask list>

Example: **AnnExclude** aftnged

Array of filearea names|masks that won't be announced in this annAreaTag

This statement can be repeated only for different AnnAreaTag.

4.4.5 AnnTo

Syntax: **AnnTo** <string>

Example: **AnnTo** All

Announcement message destination name.

This statement can be repeated only for different AnnAreaTag.

4.4.6 AnnFrom

Syntax: `AnnFrom <string>`

Example: `AnnFrom Announce Robot`

Announcement message originator name.

This statement can be repeated only for different AnnAreaTag.

4.4.7 AnnSubj

Syntax: `AnnSubj <string>`

Example: `AnnSubj New files arrived`

Announcement message subject.

This statement can be repeated only for different AnnAreaTag.

4.4.8 AnnAddrTo

Syntax: `AnnAddrTo <address>`

Example: `AnnAddrTo 2:5097/31.1`

Announcement message destination address (use for netmail area).

This statement can be repeated only for different AnnAreaTag.

4.4.9 AnnAddrFrom

Syntax: `AnnAddrFrom <address>`

Example: `AnnAddrFrom 2:5097/31`

Announcement message originator address.

This statement can be repeated only for different AnnAreaTag.

4.4.10 AnnOrigin

Syntax: `AnnOrigin <string>`

Example: `AnnOrigin New files!`

Origin line in announcement message.

This statement can be repeated only for different AnnAreaTag.

4.4.11 AnnMessFlags

Syntax: `AnnMessFlags <string>`

Example: `AnnMessFlags loc hld`

Announcement message flags (k/s loc hld ...).

This statement can be repeated only for different AnnAreaTag.

4.4.12 AnnFileRFrom

Syntax: `AnnFileRFrom <on|off>`

Example: `AnnFileRFrom on`

Show in announcement: address of system which file was received from.

This statement can be repeated only for different AnnAreaTag.

4.4.13 AnnFileOrigin

Syntax: AnnFileOrigin <on|off>

Example: AnnFileOrigin on

Show in announcement: address of system that has hatched file.

This statement can be repeated only for different AnnAreaTag.

4.4.14 Announcements example

The following text is taken from a real config file.

```

#= Announcements definitions =====
AnnounceSpool   E:\FIDO\EXCH\report

# report all arrived files to file e:\fido\rep.txt
AnnAreaTag      @e:\fido\rep.txt
AnnFileOrigin   on

# report into area 'seumbieky.news'
# files posted to 'union.*' group of fileechos
AnnAreaTag      seumbieky.news
AnnInclude      union.*
AnnSubj         New mp3's arrives
AnnTo           United mp3 fans
AnnFrom         Fullfeed Distribution Site
AnnOrigin       *** seumbieky forever ***
AnnFileOrigin   on
AnnFileRfrom    on

# report 'uee.*' group of fileechos into area 'seumbieky.news'
AnnAreaTag      seumbieky.news
AnnInclude      uee.*

# report into area 'seumbieky.news'
# all other except of 'uee.* and union.*' groups
AnnAreaTag      seumbieky.news
AnnExclude      uee.* union.*

# report via netmail to defined address
AnnAreaTag      netmail
AnnInclude      union.*
AnnSubj         New mp3's arrives
AnnTo           Max Inda Houze
Annaddrto       2:464/108
AnnFrom         Fullfeed Distribution Site
Annaddrfrom     2:464/910.108
AnnOrigin       *** seumbieky forever ***
AnnMessFlags    k/s loc hld
AnnFileOrigin   on
AnnFileRfrom    on
#= Announces definitions ends =====

```


4.4.15 Common husky keywords

List of keywords used in many husky programs. For details see fidoconfig manual.

BBSArea

LogLevels

...

HTICK is based on FIDOCONFIG library, so read documentation of FIDOCONFIG about location of config file, keywords ideology and about majority of the keywords, See *Fidoconfig manual*.

Appendix A Configuration File Keyword Index

A

addDLC	11
AnnAddrFrom	19
AnnAddrTo	19
AnnAreaTag	18
AnnExclude	18
AnnFileOrigin	20
AnnFileRFrom	19
AnnFrom	19
AnnInclude	18
AnnMessFlags	19
AnnOrigin	19
Announcements example	20
AnnounceSpool	18
AnnSubj	19
AnnTo	18
autoFileCreate	14
AutoFileCreateFile	14
AutoFileCreateFlag	13
AutoFileCreateSubdirs	15

B

BBSarea	12
BusyFileDir	10

C

Common husky keywords	21
ConvertLongNames	10
ConvertShortNames	11

D

delNotReceivedTIC	15
DLCDigits	11

E

ExecOnFile	10
------------	----

F

FileArea	16
FileAreaBaseDir	9
FileAreaCreatePerms	13
fileAreaDefaults	14
fileDescName	12
FileDescPos	9
fileDescription	12
fileEchoFlavour	14
FileFixEcho	16
FileFixEchoLimit	16
FileFixFromName	13
FileFixFSC87Subset	15
FileFixHelp	9
filefixKillRequests	11
FileFixNames	13
filefixReportsAttr	11
fileLDescString	9
ForwardFilePriority	16
ForwardFileRequestFile	16
ForwardFileRequests	15

L

LinkFileBaseDir	15
-----------------	----

M

MaxTicLineLength	10
------------------	----

N

noTIC	14
-------	----

P

PassFileAreaDir	9
-----------------	---

R

RemoteFileRobotName	14
---------------------	----

S

SaveTic	10
---------	----

T

TickerPackToBox	15
ticOutbound	11
TicPwd	13

