

Le 18 novembre 1996 à 24:01.

SqTool

(c) Raphaël Vanney

The table of contents is at the end of the file.

There is also an index down there.

1 Preliminary stuff

1.1 Legal-like stuff

I take no responsibility from whatever results from your use of SqTool. If you are not prepared to handle the consequences, please don't use SqTool.

SqTool, SqFilter and other terms are unregistered trademarks of Raphaël Vanney, Exc.

1.2 What's SqTool ?

SqTool is meant to be a general-purpose utility for those of us who use Fidonet-style message bases. As of now, Squish and *.MSG bases are supported. If you run a BBS that uses Files.BBS files, SqTool can also provide file-oriented functions.

Here are a couple things you can do with SqTool :

- . Filter messages to you, and copy them to a specified message area ;
- . Filter messages of interest, based on a keyword search, and copy them to a specified area ;
- . Remove messages you don't want on your system (for instance, messages which contain adult content on a children-dedicated BBS), or move them to a specified area ;
- . Select messages on their content, and generate automatic replies from a disk file ;
- . Trigger execution of external commands based on message content (ie when a specific subject is found, run a .CMD or .EXE file and return its output to the message sender) ;
- . Set up a mail-based file-request system ;
- . Automatically announce new files on your system ;
- . Automatically answer filefind requests ;
- . Build a listing of all files on your BBS ;
- . Import and export SOUP packets ;
- . etc...

1.3 SqTool & SqFilter

To those of you who used SqFilter : the source of SqFilter has been lost in a disk crash (always back up, they say...) Since I wanted to add features to SqFilter, I started rewriting it, et voilà, it's now called SqTool.

Some of the features of SqFilter have been left out from SqTool. If you need some of them, or if you have a problem using SqTool replacement functions, please drop me a note (see "Support").

On the other hand, SqTool is now a 32-bit application (this does not mean a lot, except that it knows no memory constraints and runs a bit faster than SqFilter), and, tada!, it handles HPFS file names.

You will also note that the filters capabilities have been enhanced, in ways I quite like.

SqFilter configuration files won't directly plug into SqTool, but the areas and message filters definitions should not require too much work.

1.4 Support

SqTool is freeware, so don't expect professional-level, 24-hour support. Anyways, I usually answer requests for help and welcome enhancement requests. Of course, general comments or congratulations are welcome as well.

Please feel free to drop me a note at one of the following addresses :

- . Internet rvanney@ibm.net
- . Fidonet 2:320/222.7
- . Francomedia 101:173/209.7

Thanks to those who already did.

Those of you who are willing to translate SqTool messages to another language are welcome (SqTool is designed to easily support other languages).

I usually upload new versions of SqTool to the Hobbes ftp site (hobbes.nmsu.edu).

I have a robot that answers messages sent to "SqTool@ibm.net"; try it with "Info" in the subject.

1.5 Credits

Thanks to David Ahia and Roland Gifford, who both tried to run early versions of SqTool despite its numerous bugs, and provided a number of enhancement suggestions. Others will forgive me for not naming them here, I prefer not to make this doc an address book.

The German documentation and message files are the result of Mike Wolfram's work.

This documentation was typed in Microsoft Word 5.5, which is probably one of its (Microsoft's) last OS/2 applications.

SqTool is written in SpeedPascal, and compiled using SpeedPascal/2 version 1.5 beta V. As soon as the release version works on my system, I buy it.

1.6 System requirements

SqTool is a 32-bit OS/2 application. As such, it works on OS/2 version 2.0 or more.

You must also be running OS/2's WorkPlace Shell to run SqTool.

2 Configuring SqTool

SqTool operation is based on a configuration file, which contains several main parts :

- . Miscellaneous : various configuration options
- . Areas configuration : where you tell SqTool where and what your message and file areas are.
- . Filters configuration : where you tell SqTool what filters to apply.
- . Autopost configuration : SqTool can be set up to automatically post messages.
- . FileFind configuration : if you want SqTool to act as an AllFix filefind request processor, you must provide SqTool some extra information.
- . SOUP configuration : if you want SqTool to import/export SOUP packets, you must provide a few details.

The configuration file is named SqTool.Cfg, and must be stored in the same directory that holds the executable file (SqTool.Exe).

This is a text file, which can be created and edited with any text editor. Empty lines and lines beginning with a ";" sign are ignored (use a ";" to insert a comment in the file).

2.1 Miscellaneous

AddDebugKludges

If this keyword is used, the debugging kludges will be added. There are currently two debug kludges :

KWRD: <keyword> this kludge is added to messages that contain one or several keywords that triggered a filter. It is not added to the original message, but in the copy that's made of it, if the filter's action implies that a copy of the message is made (CopyTo, MoveTo...)

QUOTE%: <pct> this kludge is added to messages which's quote percentage triggered a filter. It is not added to the original message, but in the copy that's made of it, if the filter's action implies that

a copy of the message is made (CopyTo, MoveTo...)

AreasWrittenTo <filename>

The AreasWrittenTo keyword lets you define a file where SqTool will put a list of all the message areas where it wrote a message. The typical use for this is to have a message tosser process these areas after SqTool has done its job.

If the file already exists, SqTool appends data to it rather than overwriting it.

Example :

```
AreasWrittenTo c:\BBS\EchoToss.Log
```

Debug

Tells SqTool to generate much more verbose information in the log file. If you've got a misbehaviour (aka bug) to report, the debug log file often helps solve the problem.

Deutsch

Tells SqTool to use German messages (default when country code is 37, 49 or 43).

English

Tells SqTool to use English messages (default when country code as specified in CONFIG.SYS is 1 (U.S.) or 44 (U.K.)).

FidoUserNo <no>

FidoUserNo tells SqTool which user number to use to store lastread information for Fido *.MSG message bases. See comments for "SquishUserNo".

ForwardHdr

Defines the text that will be put in front of forwarded messages (see the "Forward" action of message filters). The text is terminated with an "EndTxt" keyword.

Example :


```
ForwardHdr
    "Message forwarded from {OrgArea}"
    "Originally from {OrgFrom}, to {OrgTo}"
    "-----"
EndTxt
```

Francais

Tells SqTool to use French messages (default when country code as specified in CONFIG.SYS is 33 or unknown).

Include <filename>

The Include keyword lets you include files in your configuration file. This might be handy if you use several configuration files and don't want to duplicate common information such as areas definition. You would then move the areas definition to another file and 'Include' this file in your main configuration files. The Include keyword may appear anywhere in the configuration file, except within a template text and filter definitions (where it would be considered as part of the text/filter). Nested include files are allowed, without any reachable limit.

InternetAddress <address>

Lets you define your Internet address. It will be used when SqTool writes messages to Internet-type areas or export SOUP messages which lack a From field.

LockTimeOut <delay>

Sets the maximum time SqTool will try to lock a message base. If a message base is kept locked by another process during more than this delay, SqTool will fail to lock it and halt with an error message. The delay is specified in thousands of a second. The default value is 10000 (ten seconds).

Example :

```
LockTimeOut 60000
```

Will try to lock a message base for up to one minute.

LogSizeLimit <bytes>

Defines the maximum size of the log file : it's such a PITA to have to shrink these log files periodically. SqTool will make sure that the log file does not grow over this size by removing older statements. Don't be surprised if one day the first line of your log file looks strange, SqTool does not check where it cuts the log file.

The default value is 256kb. It can be set to anything between 1 and 2^{31} bytes.

Warning : the log file shrinking is quite a straight process : the program just moves the last LogSizeLimit bytes of the file to the beginning of the file, then truncates the file. This may be a lengthy process if your log file is 10Mb in size !

The shrinking feature is disabled if this value is set to 0.

MainAddress <address>

Defines your main fidonet-like address, or aka. This address will be used when writing to message areas where no specific aka has been specified.

MaxAnnPerMsg <number>

Tells SqTool how many files, at most, may be announced in a single file announcement message (this relates to "Action Announce" in file filters). If more new files are to be announced, they will be separated over several messages. The default value is 100,000.

ReScan

Tells SqTool to ignore the lastread information and entirely rescan all message areas.

RunAtEnd <command>

Specifies a command that SqTool will execute after having performed its other tasks (SOUP import/export, filters). <command> should be a valid command for the current command processor.

RunAtStartup <command>

Specifies a command that SqTool will execute before running filters or importing SOUP messages. <command> should be a valid command for the current command processor.

SquishUserNo <no>

SquishUserNo tells SqTool which lastread record it should use in the Squish *.SQL files. You should make sure that, if your reader uses this feature, it is set to use a different user no. The same applies if you run a BBS that uses the lastread information stored in Squish files.

The default value for SquishUserNo is 0. You may use anything between 0 and 64k.

UpperCaseFNames

If this keyword is specified, file names written by file filters will be uppercased. The default is to write file names exactly as they appear in the FILES.BBS file.

2.2 Areas configuration

Areas may be defined either "manually" or using existing, other software's configuration files.

2.2.1 Manual message areas configuration

Manual message areas definition uses the same syntax as GoldEd, the mail reader I use.

Area <tag> "<description>" <base type><msg type> <base init str>
<attr>[<atr>...]

where :

<tag> short name for the message area ; this is the name you will use when referring to this area in SqTool configuration.

<description> is an optional text describing the content of the area.

<base type> may be either "M" (without the quotes) for Squish areas, or "F" for *.MSG areas.

<msg type> May be "E" (without the quotes) for Echomail, "N" for NetMail, "L" for local areas. Defaults to Echomail.

Note that there is no space between <base type> and <msg type>.

<base init str> For Squish bases, path and file name without extension. For *.MSG bases, directory where the messages are stored.

<attr> Default attributes of messages entered in this area. May be none, one or several of :

- . ARQ :
- . ATT : a file is attached
- . RRQ : a return receipt is requested
- . RCT : this is a return receipt
- . CRA : crash mail
- . FRQ : file request
- . HLD : hold message
- . LOC : local (always set)
- . PVT : private message
- . SNT : sent
- . URQ :

Example :

```
Area OS2 "OS/2 main echo" ME e:\Msgs\OS2
```

```
AreaDef <tag> "<description>" <group> <msg type> <base type>  
<base init str> <aka> (<attr>) ["<origin>"]
```

AreaDef has the same purpose than the Area keyword, but allows a more detailed configuration :

<tag> See the "Area" keyword.

<description> See the "Area" keyword.

<group> Unused by SqTool (but must be present ; I suggest you use "*" (without the quotes)).

<msg type> May be either "Net" (without the quotes) for NetMail messages, "Echo" for echomail messages, "Local" for local mail messages or "News" for newsgroups.

<base type> May be either "Squish" (without the quotes) for Squish message bases, or "FTS1" for *.MSG bases.

<base init str> See the "Area" keyword.

<aka> Fidonet-style address (aka) to use when posting messages to this area. Use "." (a dot, without the quotes) to use your main address.

<attr> See the "Area" keyword. Note that the brackets must be there even if no attribute is specified.

<origin> Optional origin-line to use when writing messages in this area.

Examples :

```
AreaDef comp.os.os2.setup "UseNet : comp.os.os2.setup" *
Echo Squish e:\Msgs\comp.os.os2.setup 999:320/222.7 ()
```

```
AreaDef MsgsToMe "Messages to or about me" * Echo Squish
e:\Msgs\MsgsToMe 2:320/222.7 ()
```

```
AreaDef NetMail "NetMail" * Net Fido e:\Msgs\NetMail
2:320/222.7 (Pvt)
```

(Note that the definition should NOT be split over several lines).

2.2.2 Automatic areas configuration

SqTool can use several other-software configuration files to gather information relative to your areas.

LoraPath <path>

Tells SqTool where LoraBBS is installed. SqTool will use the files CONFIG.DAT, SYMSG.DAT and SYSFILE.DAT in this directory to configure the message and file areas.

See 4.2.3 for more details about Lora configuration files.

AreasBBS <file name>

Tells SqTool the name of a standard AREAS.BBS file where to find the message areas definition. Only Squish and *.MSG areas should be defined in this file.

SquishCfg <file name>

Tells SqTool the name of a Squish.Cfg file where to find the message areas definition. See 4.2.1 for more details about Squish.Cfg processing.

2.2.3 Manual file areas configuration

Manual file areas configuration is done using the FileArea keyword.

FileArea <tag> "<description>" <list name> <files path>

<tag> short name for the file area ; this is the name you will use when referring to this area in SqTool configuration.

<description> an optional description of the file area.

<list name> the name of the file that contains the list of the files in the area. This file must comply to the Files.BBS format (see "Technical, gory details").

<files path> the path where the files may be found.

Example :

```
FileArea OS2Drv "OS/2 Drivers"
e:\Lora\Files\OS2Drv\Files.BBS e:\Lora\Files\OS2Drv
```

(Note that the definition should NOT be split over several lines).

2.3 Message filters configuration

The way SqTool process filters is, hopefully, apparent in the way a filter is defined :

- . SqTool scans the specified message base(s) ;
- . Each message in the base(s) is compared against the "triggering" criteria ;

If the message matches the criteria (triggers the filter), the specified action is performed, and/or the specified reply is sent.

A filter definition is made up of the following parts :

2.3.1. Filter header

```
BeginFilter ["<description>"]
```

BeginFilter starts the definition of a new filter.

<description> is an optional description of the filter (this will help you identify the filter in the log file).

2.3.2 Definition of the areas to scan

The following keywords define which areas should be scanned when processing this filter. You may use as many Scan and Unscan keywords as necessary.

```
Scan <area_mask>
```

<area_mask> is either the name (tag) of a previously defined message area or a group of areas, specified using the "*" wildcard. The "*" character may appear before or after the root name, but not in the middle of it.

Examples :

Valid :

```
Scan os.os2.*
```

```
Scan *.fr
```

```
Scan *.os2.*
```

```
Scan os2.fr
```

Invalid :

```
Scan rec.*.fr
```

```
UnScan <area_mask>
```

Tells SqTool NOT to scan the specified area(s), which supposedly have been previously selected by the "Scan" keyword.

<area_mask> See the "Scan" keyword.

Example :

```
Scan comp.os.os2.*
UnScan comp.os.os2.misc
```

will scan all areas which's name (tag) starts with "comp.os.os2.", except the comp.os.os2.misc area.

2.3.3 Trigger

The following keywords define what conditions trigger the filter.

A filter may use several "Include" and "Exclude" keywords, which will be processed in the same order they appear in the configuration file.

Include|Exclude [<operators>] "<ref_string>"

Include trigger the filter if the message matches the condition that follows.

Exclude un-trigger the filter if the message matches the condition that follows.

<operators> are zero, one or several from :

- . Not negate the result of the search.
- . Exact perform an exact compare, rather than searching the reference string within the specified field(s).
- . Case : case-sensitive search.
- . From : search within the From field.
- . To : search within the To field.
- . Subject : search within the Subject field.
- . Header : search within the From, To and Subject fields.
- . Kludges : search within the kludge lines.
- . Message : search within the message text.
- . All : search within all the message fields, except kludges and addresses (this is the default).

- . Origin : compare to the originating address (Exact implicit).
- . Dest : compare to the destination address (Exact implicit).
- . QuotePct : check the percentage of quoted lines in the message against a specified range. There are three ways of specifying the range : ">n" (percentage must be greater than n), "<n" or "n-m" (percentage must be comprised between n and m). Blank lines, kludge lines, SEEN-BY, PATH, origin and tear lines are ignored (that is, they are not counted as quoted nor as non-quoted text).
- . LineCount : check the number of lines in the message against a specified range. The filter will be triggered if the number of lines in the message is within the specified range. The range is specified as for the "QuotePct" operator. Blank lines, SEEN-BY, PATH, origin and tear lines DO count in this line count.
- . Flags : triggers the filter if the corresponding flag(s) is/are set. See the "Area" keyword (2.2.1) for a list of valid flags (attributes).
- . And : the result of this search will be and'ed with the current trigger status. Not allowed in Exclude lines.
- . XOr : the result of this search will be xor'ed with the current trigger status. Not allowed in Exclude lines.
- . Or : the result of this search will be or'ed with the current trigger status (default).

<ref_string> is the string that is to be searched in the message, or the parameter for QuotePct, LineCount and Flags operators. If the keyword "Origin" or "Dest" is used, this string should represent a valid FidoNet type address, where one or several of the fields may be replaced with a "*" to accept any value for this field.

Examples :

```
Include To Subject Message "Vannee"
Exclude From "Vannee"
```

The filter will be triggered if either the To field, the Subject field or the message text contain the string "Vannee", case-insensitively, except if the From field also contains this string.


```
Include To Exact "Log Robot"
Exclude Not Subject Exact "mypassword"
```

The filter will be triggered if the To field is exactly (but case-insensitively) "Log Robot", and if the subject is equal to "mypassword". The same could be expressed as :

```
Include To Exact "Log Robot"
Include And Subject Exact "mypassword"
```

```
Include Origin "2:320/222.*"
```

Triggers the filter for any message coming from 2:320/222 and its points.

```
Exclude QuotePct ">80"
```

Prevents messages with more than 80% of quoted text from triggering the filter.

```
Include Flags "RRQ"
```

Messages that have the Return Receipt Requested attribute set will trigger the filter.

2.3.4 Action

This keyword defines the action the filter should take when a message that matches the triggering conditions is encountered. There might be as many actions as you wish, including none.

Action <action> [params]

<action> may be one of the following :

Delete deletes the message.

CopyTo <area_name> [Public]

copies the message to the specified message area. If "Public" is specified, the PVT (private) attribute of the target message is cleared.

MoveTo <area_name> [Public]

moves the message to the specified message area. If "Public" is specified, the PVT (private) attribute of the target message is cleared.

ForwardTo <area_name> [From "<author>"]

forwards the message to the specified message area. You may optionally specify a string (<author>) that will be put in the "From" field of the forwarded message.

Ack (function not implemented yet)

Execute "<command>" shells to OS/2 and executes the <command>. See 4.4 for details.

AllFix <result_file>

Handles the message as a FileFind request, placing the result in the <result_file> file. See "FileFind configuration" for more details.

AllFixMore <result_file>

Handles the message as a FileFind second request, placing the result in the <result_file> file. See "FileFind configuration" for more details.

2.3.5. Reply

The following keywords define the reply SqTool should make to the original message, after having performed the action. This is optional.

ReplyWith <options> "<file_name>"

Send the content of <file_name> as a reply to the message. If the file does not exist, no reply is sent.

<option> is zero, one or several from :

- . uue : send the file uuencoded. Use this option to send binary files (binary files can't be sent as-is over mail ; UUE will convert them to a suitable text format. The receiver will need an uu-decoder to decode the file).
- . EAs : this option tells SqTool that the extended attributes of the file should be sent as well ; it is only meaningful when the "uue" option is used.
- . DelPrev : delete the file (if it exists!) before performing the action defined with the "Action" keyword. It is important to use the DelPrev option when the action is AllFix : if you don't, users that should not get an answer

(no file found) will receive the answer to the previous successful request.

- . DelAfter : delete the file after sending the reply.
- . Variables : replace the variables (see chapter 5) contained in the text with their value. The default action is to let the text as-is, without replacing variables.

ReplyArea <area_name>

Send the reply in the specified message area (the default is to send the reply in the area where the message came from).

ReplyAttr <attributes>

Set the specified attributes to the reply message (by default, only the LOC attribute is set). See the "Area" keyword for a list of valid attributes.

RepFooter

Defines a text that will be inserted in the reply after the content of the file specified with ReplyWith. The text must end with a line containing the string "EndTxt" (without the quotes).

When used in a filter which's action is AllFix, these header and footer are additive to those defined with AfxFoundHdr and AfxFoundFtr.

ReplyFrom "<string>"

SqTool will use <string> as the From field of reply messages for this filter.

RepHeader

Defines a text that will be inserted in the reply before the content of the file specified with ReplyWith. The text must end with a line containing the string "EndTxt" (without the quotes).

ReplyOrigin "<FidoNet_address>"

SqTool will use <FidoNet_address> as the origin address of reply messages for this filter.

ReplyTo "<string>"

SqTool will use <string> as the To field of reply messages for this filter. The default is, of course, to send the reply to the sender of the message that triggered the filter.

2.3.6 EndFilter

EndFilter

Terminates the definition of a filter.

2.3.7 Examples of filters

```
BeginFilter "Messages about/to me"
  Scan comp.os.os2.*

  Include   To Subject Message "Vanney"
  Exclude   From "Vanney"

  Action    CopyTo MsgsToMe
EndFilter

BeginFilter "Messages requesting the log file"
  Scan NetMail

  Include   To Exact "Log Robot"
  Exclude   Not Subject Exact "mypassword"

  ReplyWith "e:\Lora\Logs\Lora.Log"
EndFilter

BeginFilter "Offending messages"
  Scan comp.os.os2.*

  Include   Message "Windows"

  Action    Delete
EndFilter
```



```

BeginFilter "Externally handled messages"
    Scan      robot.requests

    Include   To "SqTool robot 2:320/222.7"
    Exclude   Subject "help"

    Action    Execute "e:\Stuff.Cmd"

; The reply is not generated if the file
; e:\Temp\Result.Txt does not exist. It is up to
; Stuff.Cmd to delete it if no reply should be
; made.

    ReplyWith "e:\Temp\Result.Txt"

    ReplyArea NetMail
; set the Crash and Private attributes to keep
; the answer confidential
    ReplyAttr CRA PRV
EndFilter

BeginFilter "Automatically reply to help requests"
    Scan      robot.requests

    Include   To Exact "SqTool robot 2:320/222.7"
    Include   And Subject "help"

    ReplyWith "e:\PreparedMessages\RobotHelp.Txt"
EndFilter

BeginFilter "Let's get funny"
    Scan      alt.binaries.pictures.for.teenagers

    Include   From "aol.com"
    Include   And Message "me too"
    Include   And LineCount "<10"
    Exclude   Subject "Re:"

    ReplyWith e:\PreparedMsgs\StopSendingMeToos.Txt
EndFilter

BeginFilter "Automatic UU-decode"
    Scan alt.binaries.*
    Include Message "begin"
    Action Execute "d: & cd \Temp & uudecode %MsgBody%"
EndFilter

```



```
BeginFilter "Flame Fast cash spammers"  
  Scan alt.*  
  Scan comp.*  
  Scan fr.*  
  Include Subject "cash"  
  Include Or Subject "money"  
  Include And Subject "fast"  
  Exclude Subject "Re:"  
  
; create the answer message : copy flame.cash, which  
; contains our answer, plus the original message to  
; the file Reponse.Txt, which we will send.  
  
  Action Execute "Copy d:\Textes\flame.cash+%MsgBody%  
  d:\Temp\Reponse.Txt "  
  
  ReplyArea SoupEMail  
  ReplyWith "d:\Temp\Reponse.Txt "  
EndFilter
```


2.4 File filters configuration

The way file filters operate is somehow different from the message filters :

- SqTool scans the specified file areas, and build a list of selected files ;
- The specified action is then applied to the list of files that was built.

A file filter definition is made up of the following parts :

2.4.1. Filter header

```
BeginFileFilter ["<description>"]
```

BeginFileFilter starts the definition of a new file filter.

<description> is an optional description of the filter (this will help you identify the filter in the log file).

2.4.2 Definition of the areas to scan

The following keywords define which file areas should be scanned when processing this filter. You may use as many Scan and Unscan keywords as necessary.

```
Scan <area_mask>
```

<area_mask> is either the name (tag) of a previously defined file area or a group of areas, specified using the "*" wildcard. The same rules as defined when Scan is used in message filters apply.

```
UnScan <area_mask>
```

Tells SqTool NOT to scan the specified area(s), which supposedly have been previously selected by the "Scan" keyword.

<area_mask> See the "Scan" keyword.

Example :

```
Scan OS2*
```

```
UnScan OS2Tools
```


will scan all file areas which's name (tag) starts with "OS2", except the OS2Tools file area.

2.4.3 Select

The following keyword define which files qualify for this filter.

Select All|New

New only new files will be processed.

All all the files will be processed.

SqTool considers file areas much like message areas. This means that, for SqTool, a "new" file is any file that has not been previously scanned. Thus, the first time you run SqTool to process a file filter, ALL files will be considered new, and processed as such.

Example :

 Select New

Will select all the new files for processing by this filter.

2.4.4 Action

This keyword defines the action the filter should perform on the selected files.

Action <action> <params>

<action> may be one of the following :

List will create a list of all the selected files into a text file. In this case, <param> is the name of the file where the list will be written.

Announce will create a message set containing the list of the selected files. <param> is the name of a message area where the message will be sent. If no file is selected, the filter does nothing (ie SqTool does not make empty messages when there is nothing to announce).

2.4.5. Presentation

The following keywords define the presentation of the message or the file SqTool will generate.

Header

Following the Header keyword is a text that will be inserted on top of the file listing. The text ends with the keyword "EndTxt" alone in a line.

Example :

```
Header
Here is the list of the new files on Zlika :

EndTxt
```

Here is another way to get the same result :

```
Header
    "Here is the list of the new files on Zlika :"
```



```
EndTxt
```

Footer

Following the Footer keyword is a text that will be inserted at the bottom of the file listing. The text ends with the keyword "EndTxt" alone in a line.

Example :

```
Footer

These files may be freely downloaded from Zlika.

--- SqTool
EndTxt
```

or

```
Footer

    "These files may be freely downloaded from Zlika."

    "--- SqTool"
EndTxt
```

From "<sender>"

Defines the string that will be put in the From field of announcement messages. This keyword is useless when the Action is not Announce.

To "<recipient>"

Defines the string that will be put in the To field of announcement messages. This keyword is useless when the Action is not Announce.

Subject "<subject>"

Defines the string that will be put in the Subject field of announcement messages. This keyword is useless when the Action is not Announce.

2.4.6 EndFilter

EndFilter

Terminates the definition of a filter.

2.4.7 Examples of filters

```
BeginFileFilter "Announce new files"
  Scan *
  UnScan UserUploads

  Select New

  Action Announce NewFiles

  Header
    "Here is a list of new files on the BBS :"
```

This filter will announce all new files in the defined file areas, except those in the file area "UserUploads", to the NewFiles message area.


```

BeginFileFilter "Make a listing of all files"
  Scan *
  UnScan UserUploads

  Select All

  Action List e:\Lora\Files\AllFiles.Lst

  Header
    "List of all the files available on Zlika."

  EndTxt

  Footer
    "
                                     +- "

    "All these files may be freely downloaded."

    "Zlika is open from 10am to 6pm, on weekdays."
    "Phone number : +33-1-47748723"

    "
                                     +- "

    "This file listing was built by SqTool."
  EndTxt
EndFilter

```

This filter will build a listing of all the files on the system in the file "e:\Lora\Files\AllFiles.Lst".

```

BeginFileFilter "User uploads"
  Scan UserUploads

  Select New

  Action Announce SysAdmin

  Subject "Files recently updated by the users"
EndFilter

```

This filter will announce all the new files in the UserUploads file area with a message in the SysAdmin message area.

2.5 FileFind configuration

The following configuration options are necessary when using the "Action AllFix" and "Action AllFixMore" options in message filters.

2.5.1 File areas to scan

AfxScan <area_mask>

AfxScan tells SqTool which file areas should be scanned in search for files when answering to an AllFix filefind request.

<area_mask> is either the name (tag) of a previously defined file area or a group of areas, specified using the "*" wildcard. The "*" character may appear before or after the root name, but not in the middle of it.

AfxUnScan <area_mask>

Tells SqTool which file areas NOT to scan among those previously selected with AfxScan.

You may use as many AfxScan and AfxUnscan keywords as needed.

2.5.2 Answer setup

The following keywords configure the answer message that will be sent if SqTool finds files in response to a filefind request.

AfxFoundHdr

This keyword is followed by a text that will be inserted on top of the answer message. The text is terminated by the keyword "EndTxt".

Example :

```
AfxFoundHdr
The following {AfxFilesFound} file(s) were found on Zlika:

EndTxt
```

AfxFoundFtr

This keyword is followed by a text that will be inserted at the bottom of the answer message. The text is terminated by the keyword "EndTxt".

Example :

AfxFoundFtr

```
"> Zlika BBS, an OS/2-oriented BBS"
"SysOp      : Raphaël Vanney"
"Phone      : +33-1-47748723"
"Open hours : 10h00 - 19h00"
```

EndTxt

AfxMaxAnswers <number>

Defines the maximum number of files that will be listed in an answer to a filefind request. The default is 20.

AfxMaxDescLines <number>

Defines the maximum description lines per file. The default is 10.

AfxMoreToCome

This keyword is followed by a text that will be added to the answer message if more than AfxMaxAnswers files have been found. The text is terminated by the keyword "EndTxt".

Example :

```
AfxMoreToCome
"More than 20 files match your request. To receive a"
"list of the remaining files, reply to this message"
"with an empty message (your message should be sent"
"to "SqTool 2:320/222.7", the subject being unchanged)"
EndTxt
```

2.5.3 Example of FileFind setup

```
; define file areas where to search files
```

```
AfxScan OS2*
AfxScan DOS*
AfxScan Divers
```

```
; define answer message header
```

```
AfxFoundHdr
```

```
The following {AfxFilesFound} file(s) were found on Zlika:
```

```
EndTxt
```



```
; define answer message footer

AfxFoundFtr

> Zlika BBS, an OS/2-oriented BBS

SysOp          : Raphaël Vanney
Phone          : +33-1-47748723
Open hours     : 10h00 - 19h00

EndTxt

; define filter for first requests

BeginFilter "Process FileFind requests"
  Scan      wanted.fr

  Include   To Exact "AllFix"
  Include   To Exact "SqTool"
  Include   To Exact "SqFltr"

  Action    AllFix e:\Temp\AllFix.Res

  ReplyWith DelPrev DelAfter e:\Temp\AllFix.Res
EndFilter

; define filter for requests for more files

BeginFilter "Process FileFind second requests"
  Scan      wanted.fr

  Include   To Exact "AllFix 2:320/222.7"
  Include   To Exact "SqTool 2:320/222.7"
  Include   To Exact "SqFltr 2:320/222.7"

  Action    AllFixMore e:\Temp\AllFix.Res

  ReplyWith DelPrev DelAfter e:\Temp\AllFix.Res
EndFilter
```


2.6 The SOUP interface

The SOUP (Simple Offfling Usenet Packet) format is an increasingly popular format used for offline newsgroups and mail reading.

SqTool can import and export SOUP packets to and from your Fidonet message bases.

SoupEMail <area>

Tells SqTool in which message area it should put SOUP e-mail packets.

SoupExportPath <path>

Tells SqTool in which directory it should place the files exported for SOUP. If this variable is not defined, SqTool does not perform SOUP export.

IMPORTANT : if you are already using a SOUP scanner and plan to replace it with SqTool, please note that it is not impossible that your current scanner and SqTool use a different method for tagging messages that have been scanned. If this is the case, SqTool will export all messages from you the first time it exports your SOUP newsgroups. You should then delete the files "Replies", "SqTNews.Msg" and "SqTMail.Msg" in your SoupExportPath directory.

Other SOUP scanners also frequently use fake gateway addresses for importing/exporting SOUP packets. All references to these fake addresses must be removed or Squish will continue to export messages for the old gateway address.

SoupImportPath <path>

Tells SqTool where to look for a SOUP packet to be imported. If this variable is not defined, the SOUP import function is disabled.

SoupNewsrsrcFile <file_name>

Tells SqTool the name of your newsrsrc file. This file should contain the list of newsgroups you are connected to. SqTool uses it to know which message bases it should scan when exporting SOUP packets.

SoupOrganization <string>

If this keyword is defined, an "Organization" header line will be added to messages exported to SOUP packets. The organization line will contain <string> as the organization name.

2.6.1 Sample setup as a Squish<->SOUP converter

(courtesy of Kenneth Duke)

Here is a sample setup to use SqTool only as a Squish<->SOUP converter, using Souper to talk with the news server. Please check Souper's documentation for more information.

Here, Souper is installed in the g:\squish directory, SqTool in f:\sqtool.

This command file is used to fetch news, toss them in using SqTool, then send replies.

```
SET NNTPSERVER=NEWS.CRIS.COM      <- Sets News Server
G:                                  <- Changes to proper drive
cd \squish                          <- and directory
souper -m -l300 -k0 -h G:\squish <- Using Souper to Get
                                   news in soup format.
                                   -m is for news only.
                                   -l300 limits each piece
                                   of news to 300 lines
                                   or less
                                   -k0 doesn't care how
                                   many messages are in
                                   this group.
                                   g:\squish is where the
                                   soup formatted messages
                                   are being stored.

f:
cd\sqtool                           <- Change to the SQT00L
                                   directory
sqtool                               <- Run Sqtool with NO
                                   parameter.

g:                                   <- Change back to the
cd\squish                            <- Squishdirectory to send
                                   mail back out
souper -s                             <- Sends mail back to
                                   internet.
exit                                  <- Exits the OS/2 window.
```


Here is the corresponding SqTool configuration file (SqTool.Cfg). Note that area definitions should NOT be split over two lines as they appear here.

```
; Example configuration file for SqTool.
; Do not use without adapting it.
; -----
;

; Define which LastRead information to use
;SquishUserNo 8
;FidoUserNo 8

; My return address
InternetAddress Guest@4000.trapper.isdn.net

; Define main FidoNet address

MainAddress 1:116/4000

SoupExportPath G:\squish
SoupImportPath G:\squish
SoupNewsrsrcFile newsrc

; Define message areas.
; All newsgroups defined in the newsrsrc file should have
; a matching message area.

Area comp.os.os2.misc "comp.os.os2.misc" ME
d:\Msgs\Posted\comp.os.os2.misc

Area comp.os.os2.setup.misc "Setup OS/2" ME
d:\Msgs\Posted\comp.os.os2.setup.misc

Area comp.os.os2.moderated "comp.os.os2.moderated" ME
d:\Msgs\Posted\comp.os.os2.moderated
```


2.7 Autopost configuration

Autoposts are used to tell SqTool to automatically post the content of a file to a message base. Each time SqTool is run, it will attempt to post the content of the specified files, if it exists.

2.7.1 Autopost beginning

Autopost ["<description>"]

Autopost starts the definition of a new automatic post setup.

<description> is an optional description of the automatic post.

2.7.2 Message configuration

This section describes the message that will be posted. A minimum setup must include the following keywords : Area, File.

Area <area_name>

<area_name> is the name of a message area that has been previously defined.

Tells SqTool in which message area the message will be posted.

File [<options>] ["<file_name>"]

<options> none, one or several from :

- . UUE : the file will be UU-encoded before being posted.
- . EAs : the extended attributes will be UU-encoded as well.
- . DelAfter : the file will be deleted after having been posted. If this option is not set, the file will be sent each time SqTool is run, unless it is deleted by other means.
- . Variables : the variables contained in the file will be replaced by their values in the message (see chapter 5).

<file_name> is the (full) name of the file that will be posted.

This keyword tell SqTool the name of the file that will be posted. If the file does not exist at run time, no action is taken (no message is posted).

From ["]<name>["]

<name> is a string that will be put in the From field of the message. If the string contains blank characters, the enclosing quotes are required.

To ["]<name>["]

<name> is a string that will be put in the To field of the message. If the string contains blank characters, the enclosing quotes are required.

Origin ["]<address>["]

<address> is a Fidonet-like address that will be put in the Origin field of the message. This keyword is generally not necessary, since SqTool already knows which aka to use for NetMail areas, and addresses are not necessary for other areas.

Dest ["]<address>["]

<address> is a Fidonet-like address that will be put in the Destination field of the message. This keyword is necessary when posting to NetMail areas.

Subject ["]<subject>["]

<subject> is a string that will be put in the Subject field of the message.

Attr <attributes>

<attributes> is a list of attributes that will be set for the message. The LOC attribute is always set. See the Area keyword for a list of valid attributes.

Header

Defines a text that will be inserted in the message before the content of the file specified with File. The text must end with a line containing the string "EndTxt" (without the quotes).

Footer

Defines a text that will be inserted in the message after the content of the file specified with File. The text must end with a line containing the string "EndTxt" (without the quotes).

Both the Header and Footer text may contain variables (see chapter 5).

2.7.3 Autopost ending

EndAutopost

terminates the definition of an automatic post setup.

2.7.4 Example Autopost setup

```
BeginAutopost "PopUpLog"
  Area Test
  File DelAfter "d:\PopUpLog.OS2"
  From "Système"
  To "Raphael Vanney"
  Origin "2:320/222.7"
  Subject "Contenu du fichier PopUpLog.OS2"
  Attr PVT

  Header
    The file d:\PopUpLog.OS2 has the following content:
    -----
  EndTxt

  Footer
    -----
    This seems to indicate that an error has happened,
    while you were looking elsewhere.

    Message sent by SqTool version {VersionNumber}
  EndTxt

EndAutopost
```

3 Running SqTool

Installing SqTool is pretty straightforward : just copy the files in the archive to a directory. The following files are needed to run SqTool : SqTool.Exe, KBDVIO32.DLL and SqTool??.Msg (messages).

Before running SqTool, you must create a suitable SqTool.Cfg configuration file. If you're not sure what a SqTool.Cfg file should look like, please take a look at the included SqTool.1st file, and build your own file after it, with the help of this documentation.

The first time you run SqTool, you might want to use the "-InitPtr" command-line option. This will pre-set pointers so that SqTool does not scan all your messages and files bases.

SqTool should be run periodically, for instance once a day, or each time you insert messages in your message bases (after a poll to your uplink).

3.1 Command-line options

SqTool recognizes the following command line parameters :

-Debug

Activates the debug mode. SqTool will generate a much more verbose log file.

-ReScan

Tells SqTool to rescan all MESSAGE areas, ignoring lastread information.

-InitPtr

Runs SqTool in a special mode, in which it will not proceed the filters but initialize the lastread information for all areas. This option should only be used once, the first time you run SqTool, so that it does not process all your messages and files (consider them as "new", or "unscanned") the first time it processes filters. It's pretty much the opposite of the "-ReScan" option.

-Idle

Runs SqTool at a low priority level.

-Log=<log_file_name>

Tells SqTool to use <log_file_name> as a log file, instead of the default SqTool.Log.

-NoFileFlt

Tells SqTool not to process file filters. Ignored when "-InitPtr" is used at the same time.

-NoMsgFlt

Tells SqTool not to process message filters. Ignored when "-InitPtr" is used at the same time.

-NoSOUPIn

Don't import SOUP packets.

-NoSOUPOut

Don't export SOUP packets.

3.2 Troubleshooting

3.2.1 Run-Time Error 216

This is a serious error, which means SqTool has encountered a situation I had not planned.

Though there are potentially many reasons for this error, I have spotted one on my system: using the TRIOS2.ADD driver (replacement for IBM1S506.ADD for TRITON-chipset motherboards) causes seemingly random Run-Time Error 216, which do not appear when using the stock IBM1S506.ADD. If you encounter such an error and are using a replacement IDE disk driver, please try replacing it with IBM1S506.ADD before reporting the error to me.

3.2.2 Error codes

Under certain circumstances, most often due to a problem accessing a file, SqTool may stop with a message indicating an

error number. Here's a list of error numbers and their meaning :

OS/2 error codes

- 15 An invalid drive was specified. Check the path of the file you try to use.
- 32 Sharing violation. The file SqTool tried to gain exclusive access to is already locked by another process. Stop that process, or retry later.
- 110 A file was not found. This error may happen if the files needed by SqTool (*.Msg, *.Cfg) are not in the current directory.

SqTool error codes

- 1000 You should never encounter this error ; it means SqTool has attempted writing to a message base without locking it first. Please report if you see it happen.
- 2000 (Squish) The index file is too small when compared to the number of messages indicated in the base header. The base needs be fixed.
- 2001 (Squish) The end_frame field of the header is incorrect. The base needs be fixed.
- 2002 (Squish) The num_msg and high_msg fields are different, while they should always be equal. The base needs be fixed.
- 2003 (Squish) The value of the sz_sqhdr field is not what was expected. If you systematically encounter this error, it means that the version of Squish that was used to create the message base is different of the one I know of, and incompatible with it. This should not happen until the Squish format is seriously changed by its author.
- 2004 (Squish) A frame was not identified as such, due to an invalid Id. The base needs be fixed.
- 2005 (Squish) An unknown frame_type was encountered. Either the base is damaged, or the format is incompatible with the Squish 2.00 API.

- 2006 (Squish) This error should not happen. It means that SqTool has tried to read a message that does not exist.
- 3001 (*.MSG) This is the same as 2006, but for *.MSG bases.

4 Technical, gory details

4.1 UUEncode and extended attributes

As far as I know, the UUEncode "standard" does not make any provision to uuencode extended attributes, so I decided to make my own.

When using the "uue EAs" options of the ReplyWith keyword, SqTool builds and uuencodes a file which contains two files, each enclosed in its own pair of "begin"/"end" statements. The first file contains the actual source file, the second file its extended attributes. The second file is named "EAs.for.<filename>", where <filename> is the name of the main file.

Depending on the decoder used to retrieve the file, the result is as follows :

- . If the decoder does not handle several files in one archive, it will create the original file without the extended attributes.
- . If the decoder does handle multiple files in one archive, but does not handle HPFS file names, the result is unknown to me ; the main file will be decoded, but there will be a problem with the EA's file.
- . If the decoder handles multiple files in one archive and knows about HPFS, it will create two files, the main file and the EAs file. If the target file system is HPFS, the EA's file name will be as indicated earlier, otherwise unknown to me. Anyways, the user will then be able to reinstate the extended attributes with the EAUtil.Exe command (comes with OS/2).
- . If you use my own uu-decoder, the file will be automatically decoded with its extended attributes. My decoder is freeware and should be available from hobbes.nmsu.edu as UUCode???.Zip (current version is UUCode19.Zip).

4.2 Foreign configuration files

SqTool supports some of the popular BBS configuration files. If you would like SqTool to support another configuration file format, please get in touch with me (see "Support").

4.2.1 Squish.Cfg

The following keywords are recognized in a Squish.Cfg configuration file :

Address Your main Fidonet address.
AreasBBS The name of an Area.BBS file.
NetArea Definition of a Netmail area.
EchoArea Definition of an Echomail area.
LocalArea Definition of a local area.

Only the "-f", "-p" and "-\$" flags of area definitions are used by SqTool.

4.2.2 Areas.BBS

The following information is retrieved from an Areas.BBS configuration file :

- . All Squish and *.MSG message areas configuration. Areas are assumed to be of Echomail type. The alias (aka) is not used, since it does not appear to be accurate for point systems.

4.2.3 LoraBBS

SqTool can read Lora 2.4x configuration files. I *may* add support for the new Lora format, though I'm clearly not making any promise. Actually, I'm not very enthusiastic about that.

The following information is retrieved from LoraBBS configuration files :

- . All Squish and Fido (*.MSG) message areas configuration, including Aka, description and origin line ;
- . All file areas, except those that don't have a short name defined : YOU MUST DEFINE A SHORT NAME with Lora's configuration program if you want SqTool to use the file areas. The short name of file areas is the equivalent of the "Echo-tag" for message areas.

4.2.4 Files.BBS

Files.BBS files are text files containing a list of files and their descriptions. They are used by many BBS softwares to store a list of files in a file area.

However, there does not appear to be any real standard format for this file.

SqTool will understand any file that follow these rules :

- . a line that begins with one or several characters " " (space), ">", "<" or "+" will be considered as an extended description line. Any line that does not begin with one of these characters will be considered as a "normal" line.
- . a normal line contains a file name, a space, then a description for the file. This means that no file name should contain a space.
- . empty lines are ignored.

Example of a valid Files.BBS file :

```
Dial153.zip Dialer v1.53, beta
FWADP28.ZIP This is the patched FAX.ADP file for
>Faxworks Lite and the USR Sportster
>28800 FAX Modems.
ic10390.zip Correctif pour OS/2 Warp Connect/MPTN
>\mptn\protocol\afinet.sys
>\mptn\bin\inetcfg.exe
web103.zip WebExplorer V1.03
WKICKR.ZIP Corrective Service Facility disks for
>installing OS/2 FixPaks.
```

4.3 FileFind requests processing

SqTool accepts the same request syntax as defined by Allfix, including handling of boolean operations defined with "And", "Or" and "XOr".

SqTool being HPFS-aware, a request such as :

```
"*abc*def*"
```

is valid and would return any file which's name contains the strings "abc" and "def", in this order.

4.4 External execute

4.4.1 Environment variables

This section relates to the "Execute" option of the "Action" keyword of message filters.

The process may use (retrieve) the following environment variables :

- . MsgArea name (tag) of the message area
- . MsgFrom "from" field of the message
- . MsgTo "to" field of the message
- . MsgSubject "subject" field of the message
- . MsgOrigin origin address of the message
- . MsgDest destination address of the message
- . MsgWritten date at which the message was written
- . MsgBody The name of a temporary file where SqTool has stored the message content (without header information).

Note that invalid characters ("@", "=", "<", ">", control characters) will be replaced with a "*" when setting the environment variables. This is important to keep in mind if your filter processes messages with Internet addresses in the header (the "@" of the Internet address will be replaced with a "*").

The process will always be executed through the command processor (as defined by the COMSPEC environment variable).

The "Execute" option is a very powerful one, which enables you to do virtually anything through mail, specially when used together with REXX scripts. Beware, however, that this might be a widely open security hole. Imagine what could be done with a filter such as :

```
BeginFilter
  Scan NetMail
  Include To "Executor"
  Action Execute "%MsgSubject% >c:\Execute.Res"
  ReplyWith DelPrev DelAfter "c:\Execute.Res"
  ReplyAttr Cra
EndFilter
```


This filter, when it encounters a message to "Executor" in the NetMail area, executes the command specified in the Subject field of the message and crash-replies with the output of the command.

4.4.2 Format of the message file

When dumping the message content to the %MsgBody% file, SqTool uses a CR only to end each line of the message. Some programs, specially OS/2 text editors, like it better to have a CR/LF combination to terminate lines. Here's a REXX script that will convert CR-terminated lines to CR/LF-terminated lines :

```
/* REXX */
/* Converts a text file where lines end with a CR only to a
   text file where lines end with a CR/LF (better suited
   for OS/2 text editors)
*/

Parse Arg Source Dest

call RxfuncAdd 'SysLoadFuncs','RexxUtil','SysLoadFuncs'
call SysLoadFuncs

AppendMode=1          /* set to 0 to overwrite Dest */

If Dest="" Then
Do
  Say "CR2CRLF <source_file> <destination_file>"
  Say ""
  Say "CR2CRLF converts a file where lines end with a"
  Say "CR only to a text file where lines end with a"
  Say "CR/LF (better suited for OS/2 text editors)."
```


Another issue to consider is the fact that some messages do not include formatting line-ends. This means that CR characters are found only at the end of a paragraph. SqTool will not modify such messages.

4.5 The SOUP interface

The soup format documentation is available from Rhys Weatherley at "rhys@cs.uq.oz.au" (well, so says the doc).

SqTool (partly) supports version 1.2 of the SOUP format. Within this specification, the following message file formats are handled :

- . u (import) USENET news articles
- . m (import) Unix mail
- . b (export) Binary mail
- . B (export) Binary news

Please drop me a note if you think others should be supported as well. SqTool appears to work correctly with Souper and VSoup.

SqTool only uses the files AREAS and *.MSG when tossing a SOUP packet. If an ERRORS file exists, it is displayed and dumped to the log file.

SqTool only exports messages to newsgroups which you are connected to, according to your newsrc file. That means that, if you write a message to a group for which you don't receive news using SOUP (ie, the group's name in newsrc is not followed by a colon), this message won't be exported by SqTool's SOUP interface.

5 Variables

Variables are special codes that SqTool will replace with values that are not known before execution time. They are used in configuration texts, when you don't know yet what the value will be (for instance, the first name of the addressee of a message, etc.)

A variable code is composed of a variable name, enclosed in accolades : "{" and "}".

Some variables are always available, other depend on what action SqTool is performing.

5.1 Always available

VersionNumber

Of SqTool.

DateVersion

This should be the date (month year) I produced the program.

SquishUserNo

The value of the SquishUserNo configuration keyword.

FidoUserNo

The value of the FidoUserNo configuration keyword.

MaxLogSize

The value of the LogSizeLimit configuration keyword.

DefaultPath

The path where SqTool.Exe resides, including a trailing backslash.

AfxMaxAnswers

The value of the AfxMaxAnswers configuration keyword.

AfxMaxReAnswer

The value of the AfxMaxReAnswer configuration keyword (not implemented; always 50).

AfxMaxDescLines

The value of the AfxMaxDescLines configuration keyword.

5.2 Context-dependant

5.2.1 Message

The following variables represent elements of a message, either the message that triggered a filter (OrgXXXX) or a message SqTool is ready to write (DstXXXX).

OrgFrom (DstFrom)

Content of the "from" field of the message.

OrgFromFirst (DstFromFirst)

First name of the "from" content.

OrgFromLast (DstFromLast)

Last name of the "from" content.

OrgTo (DstTo)

Content of the "to" field.

OrgSubject (DstSubject)

Content of the "subject" field.

OrgLineCnt (DstLineCnt)

Number of lines in the message.

OrgAttrib (DstAttrib)

Attributes of the message.

5.2.2 Message area

The following variables describe a message area, either the one that is being processed (OrgXXXX) or the one SqTool is ready to write a message in (DstXXXX).

OrgArea (DstArea)

Short name (tag, echoid, etc.) of the message area.

OrgBaseType (DstBaseType)

Message base type (Squish, *.MSG).

OrgMsgType (DstMsgType)

Content of the message base (Echomail, Netmail, Internet).

OrgInitStr (DstInitStr)

Either the root file name (Squish bases) or the directory (*.Msg).

5.2.3 File area

The following variables relate to a the file area being processed.

FilArea

Name of the area (Tag).

FilFileList

Name of the file that contains the file list for this area (Files.BBS).

FilFilePath

Path where the files of this area are stored.

FilDescript

Description of the area.

FilCount

Number of files in the area.

FilTotSize

Total size of the files in this area.

5.2.4 FileFind requests

The following variables are relative to an answer to a filefind request.

AfxRequest

Expanded version of the request (content of the Subject field of the original message).

AfxFilesSearched

Number of files searched. This number is either the number of files contained in all the AfxScan areas, or the number that were searched to reach AfxMaxAnswers matches.

AfxFilesFound

Number of matching files found, limited to AfxMaxAnswers (first request) or AfxMaxReAnswer (second request).

5.3 Variables summary

This table lists where each variable may be used. The variables that are always available are not mentioned.

A- ForwardHdr

B- AfxFoundHdr, AfxFoundFtr, AfxMoreToCome

C- (nowhere yet ; would help to customize area headers in file listings (made by file filters) and Allfix requests answers. Request if you want it implemented.)

D- RepHeader, RepFooter, ReplyWith content (note that variables present in the ReplyWith file are replaced with their values only if the Variables keyword is specified after ReplyWith in the configuration file. RepHeader and RepFooter are always processed).

E- Header, Footer, File content of Autoposts (note that variables in File are replaced with their values only if the Variables keyword is specified after the File keyword).

	A	B	C	D	E
OrgFrom	x	x		x	
OrgFromFirst	x	x		x	
OrgFromLast	x	x		x	
OrgTo	x	x		x	
OrgSubject	x	x		x	
OrgLineCnt	x	x		x	
OrgAttrib	x	x		x	
DstFrom		x		x	x
DstFromFirst		x		x	x
DstFromLast		x		x	x
DstTo		x		x	x
DstSubject		x		x	x
DstLineCnt		x			x
DstAttrib		x		x	x
OrgArea	x	x		x	
OrgBaseType	x	x		x	
OrgMsgType	x	x		x	
OrgInitStr	x	x		x	
DstArea		x		x	x
DstBaseType		x		x	x
DstMsgType		x		x	x
DstInitStr		x		x	x

A B C D E

FilArea				x
FilFileList				x
FilFilePath				x
FilDescript				x
FilCount				x
FilTotSize				x

AfxRequest				x
AfxFilesSearched				x
AfxFilesFound				x

History

- 0.2 first public beta.
- 0.3 Added support for the FileFind requests processor. The "LoraBBS" keyword now reads file areas configuration. Added keywords "AfxScan", "AfxUnscan", "AfxFoundHdr", "AfxFoundFtr", "AfxMaxDescLines", "AfxMaxAnswers", "FileArea". Implemented "Action AllFix", added "Action AllFixMore".
- Fixed a bug with Scan and Unscan wildcard handling.
- Implemented the "Include" keyword (when used outside a filter).
- Fixed a number of various bugs.
- Added the file filters features (which replace SqFilter's new file announce and all files lister features).
- 0.4 Fixed a bug when an empty line follows a "BeginFilter" keyword.
- Added the "From", "To" and "Subject" keywords to file filters definitions.
- Added the "AreasWrittenTo" keyword.
- Added the command-line options "-Debug", "-InitPtr" and "-ReScan".
- The documentation (SqTool.Prn) should now have form-feed characters where appropriate (if not, replace all occurrences of the string "< TOPOFPAGE >" with a form-feed :-)).
- 0.5 (Feb 96) Added the "-Idle", "-NoFileFlt" and "-NoMsgFlt" command-line options.
- "LocalArea" keyword now recognized in Squish.Cfg.
- Added the "ForwardTo" action to message filters.
- Changed the behavior when "Area" or "AreaDef" define an already defined message base : the

old definition is now replaced with the new one. A warning message is still issued.

Added the "LockTimeOut" keyword.

Added "RepHeader" and "RepFooter" keywords to message filters.

SqTool now always sets the MSGID kludge line when the origin of the message is known.

Added the "From" option to the "ForwardTo" action.

Added the "MaxAnnPerMsg" keyword.

Sped-up the file-areas related code.

Added the "UpperCaseFNAMES" keyword.

Now retrieves origin lines from .SQO files.

0.6 (Mar 96) Fixed a problem with external execute when a field of the message contains invalid characters ("@", "=", ...).

Added the "QuotePct" keyword.

Added the "/LOG=" command-line parameter.

Added support for SOUP packets. "SoupEMail", "SoupImportPath", "SoupExportPath", "SoupNewsSrcFile" keywords.

Fixed a bug handling Squish messages with an empty control area.

Now correctly sets the DateWritten field to 0 for Squish locally entered messages.

Added the ability to enclose texts in quotes (") within configuration files, which should help keep them readable (see the new headers/footers definitions in SqTool.1st).

SqTool.Exe and KbdVio32.DLL are now compressed using LxLite by Andrew Zabolotny.

0.7 (Mar 96) Improved handling of Internet mail header fields.

Added the "InternetAddress" keyword.

Fixed a bug when exporting SOUP packets.

08 (Jun 96) The SOUP import function now displays the ERRORS file, if it exists.

Various bugfixes and enhancements.

Added the "AfxMoreToCome" keyword.

Now handles the situation where a FILES.BBS file is in use by another process.

Added the "-NoSOUPIn" and "-NoSOUPOut" command-line parameters.

Started implementing the Variables' concept.

Added the "ForwardHdr" keyword.

Fixed a bug that prevented SqTool from exporting SOUP messages in different newsgroups in the same run.

MIME header lines are now exported to SOUP packets.

When answering a FileFind request, SqTool used to put its interpretation of the request on top of the answer. This is no longer the default, you must now use the {AfxRequest} variable in the AfxFoundHdr or AfxFoundFtr texts.

The text defined with AfxMoreToCome is now inserted (if needed) BEFORE the one defined with AfxFoundFtr.

Added the "Deutsch" keyword, support for German messages.

Read-only Files.BBS files (such as those found on CD-ROMs) are now correctly handled.

09 (Aug 96) Fixed a couple bugs.

Added the "AddDebugKludges" keyword.

The log file shrinking feature can now be disabled by setting LogSizeLimit to 0.

Added the "RunAtEnd" and "RunAtStartup" keywords.

Added the "SoupOrganization" keyword.

It is now possible to have more than one action per message filter.

10 (Sep 96)

Macros now available in RepHeader, RepFooter and ReplyWith content. "Variables" option added to the ReplyWith keyword.

Exporting to SOUP now exports all RFC header lines found in the source messages.

SOUP-imported messages now have a SEEN-BY line added, if a valid FidoNet address has been defined.

Added the "LineCount" and "Flags" filter operators.

Added the "ReplyTo" and "ReplyOrigin" keywords.

Added the "MsgArea" environment variable (Action Execute).

Message exported to SOUP from Squish bases now have their MsgScanned flag set.

SOUP-imported e-mails now have the private flag set.

11 (Nov 96)

Added the "Autopost" concept.

Fixed a bug handling FILES.BBS files with long file descriptions.

Index

*.MSG	6, 9, 10, 11
.SQO	56
Ack	17
Action	16, 23, 44
AddDebugKludges	5
Address	8, 11, 15, 42
AfxFilesFound	52
AfxFilesSearched	52
AfxFoundFtr	18, 27
AfxFoundHdr	18, 27
AfxMaxAnswers	49, 52
AfxMaxDescLines	49
AfxMaxReAnswer	49, 52
AfxMoreToCome	28
AfxRequest	52
AfxScan	27
AfxUnScan	27
Aka	8, 11
All	14, 23
AllFix	17, 26, 27
AllFixMore	17, 26
And	15
Announce	8, 23
Area	9, 34
AreaDef	10
Areas	9
Areas definition	7
AREAS.BBS	12, 42
AreasBBS	12, 42
AreasWrittenTo	6
ARQ	10
ATT	10
Attr	10, 11, 35
Attribute	11, 18
Attributes	35
Autopost	34, 53
Base type	9, 11
BeginFileFilter	22
BeginFilter	13
Binary	17
Case	14
Command line	37
COMSPEC	44
CONFIG.SYS	6, 7
Configuration file	5, 7, 9
CopyTo	5, 6, 16
Country	7
CR	45
CRA	10

DateVersion	48
Debug	6, 37
DefaultPath	48
DelAfter	18, 34
Delete	16
DelPrev	17
Dest	15, 35
Deutsch	6
DstArea	51
DstAttrib	50
DstBaseType	51
DstFrom	50
DstFromFirst	50
DstFromLast	50
DstInitStr	51
DstLineCnt	50
DstMsgType	51
DstSubject	50
DstTo	50
EAs	17, 34, 41
EchoArea	42
Echomail	10, 11
EndAutopost	36
EndFilter	19, 25
EndTxt	24
English	6
Environment variable	44
Exact	14
Exclude	14
Execute	17, 44, 45
Extended attribute	17, 41
FidoUserNo	6, 48
FilArea	51
FilCount	51
FilDescript	51
File	34
File area	12, 42
File filters	22
FileArea	12
FileFind	17, 27
Files.BBS	42
FilFileList	51
FilFilePath	51
Filters	12
FilTotSize	52
Flags	15
Footer	24, 36
Forward	6
ForwardHdr	6
ForwardTo	17
Francais	7
From	14, 17, 35
FRQ	10

FTS1	11
GoldEd	9
Header	14, 24, 36
HLD	10
Idle	38
Include	7, 14
InitPtr	37
Installing	37
InternetAddress	7
Kludge	5
Kludges	14
KWRD	5
Language	3
Lastread	6, 8, 9
LineCount	15
List	23
LOC	10
Local	10, 11
LocalArea	42
LockTimeOut	7
Log	8, 38
LogSizeLimit	8, 48
LoraBBS	11, 42
LoraPath	11
MainAddress	8
MaxAnnPerMsg	8
MaxLogSize	48
Message	14
MoveTo	5, 6, 16
Msg type	10, 11
MsgArea	44
MsgBody	44
MsgDest	44
MsgFrom	44
MsgOrigin	44
MsgScanned	58
MsgSubject	44
MsgTo	44
MsgWritten	44
NetArea	42
NetMail	10, 11
New	23
Newsgroups	11
NoFileFlt	38
NoMsgFlt	38
NoSOUPIn	38
NoSOUPOut	38
Not	14
Operators	14
Or	15
Organization	31
OrgArea	51
OrgAttrib	50

OrgBaseType		51
OrgFrom		50
OrgFromFirst		50
OrgFromLast		50
OrgInitStr		51
OrgLineCnt		50
OrgMsgType		51
OrgSubject		50
OrgTo		50
Origin	11, 15,	35
Origin-line		11
Percentage		15
Public		16
PVT	10,	16
QUOTE%		5
Quoted lines		15
QuotePct		15
Range		15
RCT		10
RepFooter		18
RepHeader		18
Reply		17
ReplyArea		18
ReplyAttr		18
ReplyFrom		18
ReplyOrigin		18
ReplyTo		19
ReplyWith	17,	41
ReScan	8,	37
RFC		58
RRQ		10
RunAtEnd		8
RunAtStartup		9
Scan	13,	22
Select		23
Short name		42
Shrink		8
SNT		10
SOUP		58
SoupEMail		30
Souper	32,	47
SoupExportPath		30
SoupImportPath		30
SoupNewsrsrcFile		30
SoupOrganization		31
SqFilter		3
SqTool.lst		37
SqTool.Cfg	5,	37
Squish	9, 10,	11
Squish.Cfg	12,	42
Squish<->SOUP		32
SquishCfg		12
SquishUserNo	9,	48

Subject	14, 35
Support	3
Tag	9, 10, 12, 13, 14, 22, 23, 27
To	14, 35
Trigger	14, 15
TRIOS2.ADD	38
UnScan	13, 22
UpperCaseFNAMES	9
URQ	10
Uue	17, 34, 41
Uuencode	17, 41
Variables	18, 34, 53
VersionNumber	48
VSoup	47
Wildcard	13, 22, 27
WorkPlace Shell	4
XOr	15

Table of contents

1	Preliminary stuff.....	2
1.1	Legal-like stuff	2
1.2	What's SqTool ?	2
1.3	SqTool & SqFilter	3
1.4	Support	3
1.5	Credits	4
1.6	System requirements	4
2	Configuring SqTool.....	5
2.1	Miscellaneous	5
2.2	Areas configuration	9
2.2.1	Manual message areas configuration	9
2.2.2	Automatic areas configuration	11
2.2.3	Manual file areas configuration	12
2.3	Message filters configuration	12
2.3.1.	Filter header	13
2.3.2	Definition of the areas to scan	13
2.3.3	Trigger	14
2.3.4	Action	16
2.3.5.	Reply	17
2.3.6	EndFilter	19
2.3.7	Examples of filters	19
2.4	File filters configuration	22
2.4.1.	Filter header	22
2.4.2	Definition of the areas to scan	22
2.4.3	Select	23
2.4.4	Action	23
2.4.5.	Presentation	23
2.4.6	EndFilter	25
2.4.7	Examples of filters	25
2.5	FileFind configuration	26
2.5.1	File areas to scan	27
2.5.2	Answer setup	27
2.5.3	Example of FileFind setup	28
2.6	The SOUP interface	30
2.6.1	Sample setup as a Squish<->SOUP converter	32
2.7	Autopost configuration	34
2.7.1	Autopost beginning	34
2.7.2	Message configuration	34
2.7.3	Autopost ending	36
2.7.4	Example Autopost setup	36
3	Running SqTool.....	37
3.1	Command-line options	37
3.2	Troubleshooting	38

3.2.1	Run-Time Error 216	38
3.2.2	Error codes	38
4	Technical, gory details.....	41
4.1	UUEncode and extended attributes	41
4.2	Foreign configuration files	41
4.2.1	Squish.Cfg	42
4.2.2	Areas.BBS	42
4.2.3	LoraBBS	42
4.2.4	Files.BBS	42
4.3	FileFind requests processing	43
4.4	External execute	44
4.4.1	Environment variables	44
4.4.2	Format of the message file	45
4.5	The SOUP interface	47
5	Variables.....	48
5.1	Always available	48
5.2	Context-dependant	50
5.2.1	Message	50
5.2.2	Message area	50
5.2.3	File area	51
5.2.4	FileFind requests	52
5.3	Variables summary	53
	History.....	55
	Index.....	59
	Table of contents.....	64