

mTAIL

Olivier PHILIPP

Documentation revision (MM/DD/YYYY):

08/03/2003 [v0.1]: first release

...

02/01/2012 [v0.72]: update for mTAIL v1.8.3

02/02/2012 [v0.73]: update for mTAIL v1.8.3

02/02/2012 [v0.73]: update for mTAIL v1.8.3

02/14/2012 [v0.74]: update for mTAIL v1.8.5

06/13/2012 [v0.75]: update for mTAIL v1.8.6

06/13/2012 [v0.76]: update for mTAIL v1.8.12

Intro

mTAIL is a "Unix like" tail program working on Windows 32(/64) bits systems (95?/98?/NT4?/XP/2000/2003/Vista/2008/Seven → Desktop or Server).

Its purpose is to display text file changes (additions to the end of the file) in "real time".

mTAIL is compatible with UNICODE files.

mTAIL is 32bits program written in Delphi XE2 [*new in v1.8*].

The latest mTAIL version can be downloaded at this URL: <http://www.mtail.com>

If you have any ideas about how to improve mTAIL or you find a bug, please e-mail Olivier PHILIPP (ophilipp@free.fr).

mTAIL is **free** ! But, if you want to use mTAIL in a commercial/business environment, I really encourage ☺ you to register mTAIL making a Corporate Paypal donation. This will help me to buy new Delphi licenses.

Credits:

Regular expression documentation is copyrighted to Andey V. Sorokin

Thanks to Farid AMINI for all beta tests.

Thanks to every contributor that help me in mTAIL ☺

Minor documentation corrections made by Benjamin Cabell V, g.1@besiex.org.

Remarks

" " quotes stand for standard the "double quotes" characters ☺

First steps

Installation

mTAIL does not require any installation because there is only one executable (mTAIL.exe) and no libraries or additional files.

Unzip this file to any destination directory and launch the program. The mTAIL window should now appear on your screen.

You are not limited to one instance of mTAIL; you can launch any number of mTAIL windows to tail multiple file at the same time.

mTAIL is no more supported on Windows 9x. Nevertheless you could try to get it works by installing Unicode library from Microsoft. Also if you got an OLE32 error when launching mTAIL on Windows 95/98 system, you should update your DCOM library using Windows Update. *[new in v1.7]*

Tailing

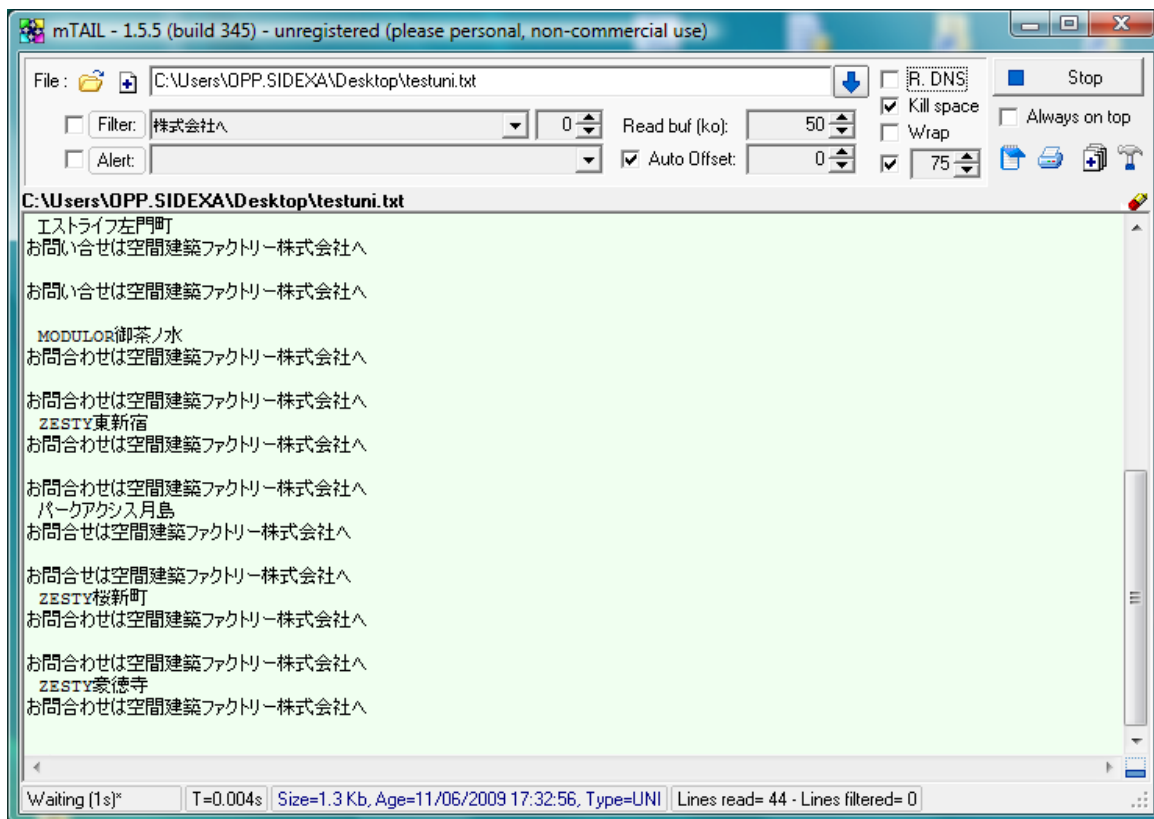
You should now see the mTAIL main window. Click on the button next to the "File:" label to select a file from the file system browser or enter a filename manually in the text combo box to the right of the button (example: C:\temp\log.txt).

Click the **Start** button (to the right of the window) and you should see the end of the text file you chose appear in the text display area.

To stop tailing, just click the stop button!

About Unicode *[new in v1.5]*

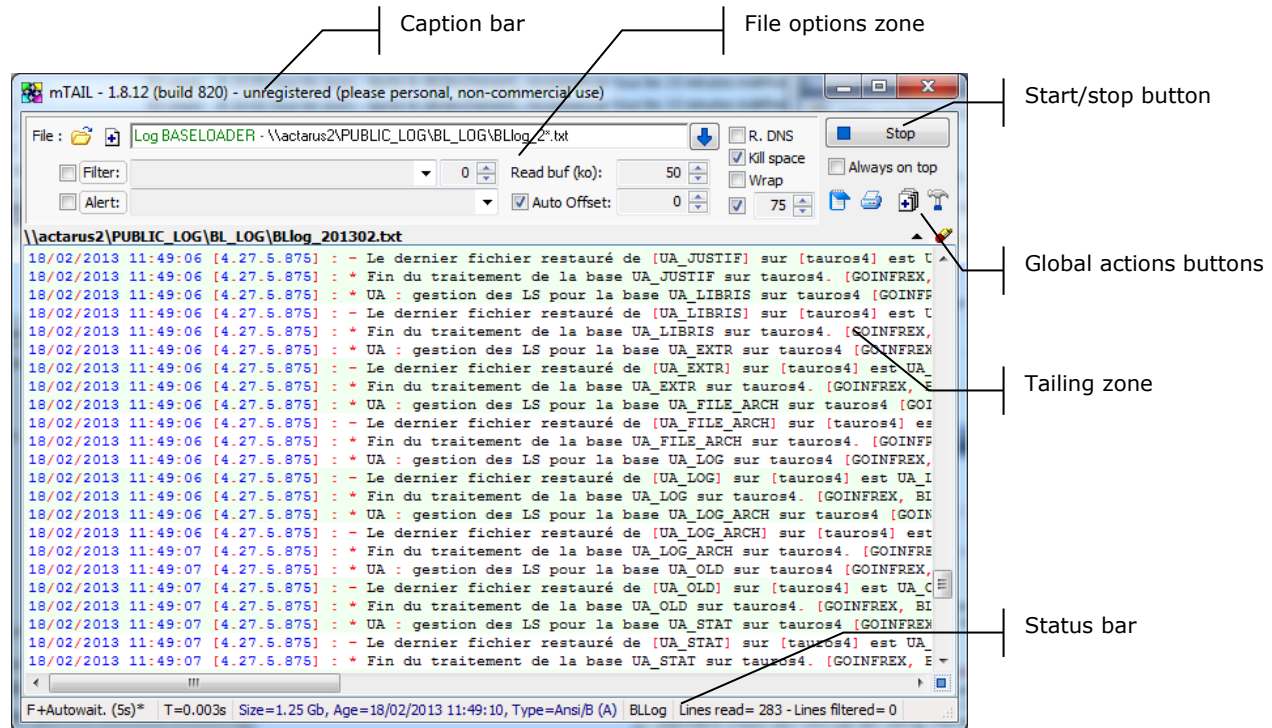
Beginning with v1.5, mTAIL can tail UNICODE files. UNICODE is supported also in Alert or Filter feature.



mTAIL support UTF-8, UTF-16 Big and Low Indian and standard Unicode. *[new in v1.7]*

mTail main screen area

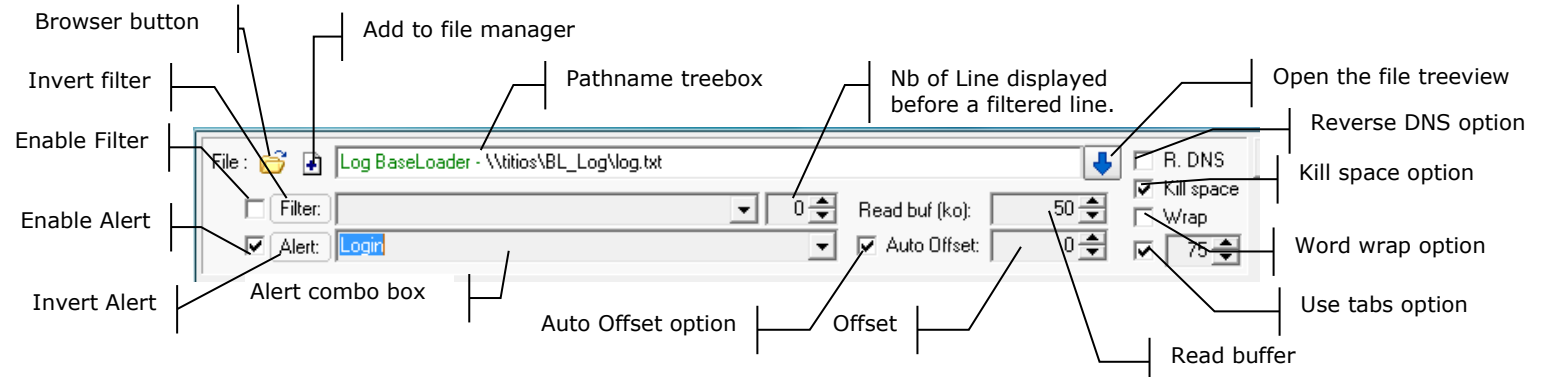
You will find below the description of each of mTail's "zones". Each zone is fully explained in the corresponding section.



Caption bar

The main window caption shows you the version number of mTAIL. You can also use this bar to move, minimize, maximize or close the mTAIL window.

File option zone



Pathname combo box

In the *Pathname combo box* you can enter the pathname you want to tail. Just enter it in the treebox edit zone. You can use standard shell wildcards: mTAIL will always take the last matching file.

Wildcard in pathname

If you enter wildcard in pathname, mTAIL will list the filename matching the wildcard and select the LAST in the list, according to the "File Sorting" value. The file sorting value can be set by default or, specifically for each file manager entry).

File sorting can be:

last alpha	mTAIL will tail the last alphabetic file
last fileage	mTAIL will tail the last modified file
last creation	mTAIL will tail the last created file
Last numeric	mTAIL will tail the last file according to the first numeric in its name
Last [RegExp]	mTAIL will tail the last file according to matching regexp.

Last alpha

Assuming your c:\log\ directory contains the following files:

Log20030117.txt
Log20030118.txt
Log20030119.txt

If you enter c:\log\log*.txt, the file tailed will be Log20030119.txt

Last fileage

Assuming your c:\log\ directory contains the following files:

Log20030117.txt, last modification 02/12/2013
Log20030118.txt, last modification **02/15/2013**
Log20030119.txt, last modification 02/01/2013

If you enter c:\log\log*.txt, the file tailed will be Log20030118.txt

Last creation

Assuming your c:\log\ directory contains the following files:

Log20030117.txt, last modification 02/12/2013, created 01/01/1999
Log20030118.txt, last modification **02/15/2013**, created 01/01/1998
Log20030119.txt, last modification 02/01/2013, created 01/01/1997

If you enter c:\log\log*.txt, the file tailed will be Log20030117.txt

Last numeric (new v1.8.11)

Only the first numeric in the filename will be used.

Assuming your c:\log\ directory contains the following files:

Log2_56_test.txt
Log55_test.txt
Log9_test.txt,

If you enter c:\log\log*.txt, the file tailed will be log55_test.txt

Last [RegExp] (new v1.8.121)

The RegExp must be added to path, with a pipe as separator: Pathname|regexp

Assuming your c:\log\ directory contains the following files:

Logb2b_test.txt → RegExp matches b2b
Log55_test.txt → RegExp matches nothing
Logaaaa9zzz_test.txt → ReegExp matches a9z

If you enter c:\log\log*.txt|D\d\D, the file tailed will be logb2b_test.txt

Notes:

1. The filename is checked every tailing cycle. So, if (while tailing) a NEW file appears in the path (ex: Log20030120.txt), the file name will change automatically to the new file. This is very useful to tail daily log.
2. If you try to tail a bad filename, mTAIL will display "File not found" in the tailing zone.
3. You can enter an *alias* to each pathname. The alias is displayed in green before the pathname. This is useful to give "explicit name" to path. To add an alias, see *Configuration Panel* section.
4. You can only enter or modify a pathname when mTAIL is stopped. When tailing, the name cannot be modified in the edit field, BUT you can select another path in the combo box choice.
5. You can use Windows wildcard in the name (see example above). When wildcards are used, the last file sorted alphabetically or by file age is tailed (this option can be set in the File manager)

Browser button

Click this button to open the file path browser. Select a file and click ok. The pathname field will be entered automatically. Don't forget to click Start to begin the actual tailing of the file once you've chosen it!

Add to file manager button [new in v1.4]

Click this button to quickly add the current tailed filename to the file manager.

Open the file treeview [new in v1.4]

Click this button to open the file treeview.

R DNS option [new in v1.4]

When active, this option allows mTAIL to resolve IP address in the tail Windows (ex: 64.125.45.100 → www.google.com).

IP are detected when they are leaded by space, "@", "/", ":", "=", "[" (new in v1.8.11)

Example:

Same lines without and with R DNS option (some info are hidden for security):

```
2009-06-11 12:05:41 10.0.1.28 255.255.255.255 Udp 138 138 BLOCKED 81.255.224.1
2009-06-11 12:07:12 10.0.1.28 255.255.255.255 Udp 138 138 BLOCKED 81.255.224.1
2009-06-11 12:07:21 81.255.224. 81.255.224. Udp 138 138 BLOCKED 81.255.224.
2009-06-11 12:07:21 81.255.224. 81.255.224. Udp 138 138 BLOCKED 81.255.224.
2009-06-11 12:08:29 81.255.224. 81.255.224. Udp 138 138 BLOCKED 81.255.224.
2009-06-11 12:09:26 81.255.224. 81.255.224. Udp 138 138 BLOCKED 81.255.224.
2009-06-11 12:09:26 81.255.224. 81.255.224. Udp 138 138 BLOCKED 81.255.224.

2009-06-11 12:05:41 [aphelie-v. .blr] 255.255.255.255 Udp 138 138 BLOCKED [aphelie-dmz. .com]
2009-06-11 12:07:12 [aphelie-v. .blr] 255.255.255.255 Udp 138 138 BLOCKED [aphelie-dmz. .com]
2009-06-11 12:07:21 [ftp-cps. .com] ?81.255.224.127? Udp 138 138 BLOCKED [aphelie-dmz. .com]
2009-06-11 12:07:21 [ftp-cps. .com] ?81.255.224.127? Udp 138 138 BLOCKED [aphelie-dmz. .com]
2009-06-11 12:08:29 [golgoth. .com] ?81.255.224.127? Udp 138 138 BLOCKED [aphelie-dmz. .com]
2009-06-11 12:09:26 [w1. .com] ?81.255.224.127? Udp 138 138 BLOCKED [aphelie-dmz. .com]
2009-06-11 12:09:26 [w1. .com] ?81.255.224.127? Udp 138 138 BLOCKED [aphelie-dmz. .com]
```

Kill line option *[removed in v1.4]*

The kill line option has been removed from the file zone option to the file manager option.

Kill space option

If this option is checked, mTAIL compresses any number of subsequent spaces to one space. You can combine this option with the *Kill Line* option.

Example:

Original file	Kill space not checked	Kill space checked
Line1 : this is an example Line2 : this is an example	Line1: this is an example Line2: this is an example	Line1: this is an example Line2: this is an example
Line3 : 1 2 3 4 5 Line4 : 1 2 4 5	Line3 : 1 2 3 4 5 Line4 : 1 2 4 5	Line3 : 1 2 3 4 5 Line4 : 1 2 4 5

Note:

- A default kill space option can be set for each path name via the *Configuration Panel*.

Use tabs

If this option is NOT set, tab characters (ASCII 9) will be replaced by a space character. If this option is set, tab characters are handled as tabs. The width of each tab is defined by the value in the spin edit selector (default is 75).

Note:

- A default tab option can be set for each path name via the *Configuration Panel*.

Read buffer *[change in v1.4]*

This value is the size of the buffer (in kilobytes) used to read the file. The file is always read from its end. The default value is 50 kilobytes, so mTAIL reads the last 50 kilobytes from the file and displays it.

You can increase or decrease this value to read more or less data.

Note:

- A default read buffer can be set for each path name via the *Configuration Panel*.
- When tailing a Unicode file, because each char is usually 2 bytes, the number of characters read is divided by 2 *[new in v1.5]*.

Offset

With Offset you control from which position in the buffer the file is read, relative to the end of the file.

Example: reading a 100 Kbyte Ansi files

Buffer :	50	50	200
Offset :	0	6000	10000
Result :	50 to 100	44 to 94	70 to 90

Note:

- A default offset can be set for each path name via the *Configuration Panel*.

Auto (offset)

This is certainly the strangest option in mTAIL. If *Auto* is checked, mTAIL tries to adjust the offset so that the *display area* is actually displaying text.

I have made this option especially for IIS (Microsoft Internet Information Server) logs. IIS is a bit unusual in the way it creates its log files. IIS allocates space for the log in chunks of 64 kb. So when the log is created, IIS creates a 64 kb file filled with 0x00 characters.

When logs are written in this file, IIS replaces the null character with the log text, but the file size does not change until that 64 kb chunk is full, at which point IIS allocates another 64 kb chunk.

Imagine what would happen if you were trying to tail a fresh new IIS log file, without this option enabled, mTAIL would read the last 50000 bytes and try to display it, displaying mostly screen full of blank lines (because those blanks are the 0x00 characters IIS has pre-allocated). The *Auto* option will automatically go back in the file to find non-null characters and tail from there. So each time the file is read, mTAIL will change to *offset* to go back, and back, and back...

Confused? Leave the option "on", it's the best!

Note:

- A default value for each pathname can be entered via the *Configuration Panel*.

Filter

If you check this option, mTAIL will only display lines that match the expression in the combo box. The *filter* accepts regular expressions, so you can make a powerful filter (see chapter on *Using Regular Expression*).

When filtering is on, the background color of the *display zone* is blue.

Example:

To display only lines containing ".com". Just type .com in the edit field (zone) and check the *Filter* checkbox.

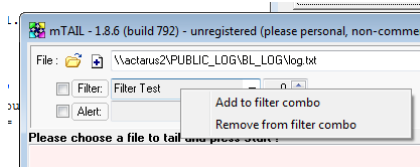
You can display information about line removed by the filter. To do so, you must check the *Show removed lines info when filtering is active* option. When this option is ON, mTail will display a double dot followed by the number of line filtered.

Example:

```
07:57:24 0577CS 21 [16974]sent /2003_09_20/FULL/PROGRAM/DATA/BDDIDENT
... (5) < Five lines where removed by the filter
07:57:25 0577CS 21 [16974]sent /2003_09_20/FULL/PROGRAM/DATA/BDDIDENT
... (4) < Four lines where removed by the filter
07:57:26 81.50.142.10 0577CS 21 [16974]sent
```

Add/Remove filter in the filter combo box [new v1.8.6]

You can easily store default filter strings in the filter combo box. To do this, right-click on the filter edit zone and select the "add" or "remove" function in the popup.



Invert filter

You can invert the filtering condition by clicking on the filter label.

Example:



Invert filter is OFF. In this case, mTAIL will only display lines containing "FULL"

Invert filter is ON. In this case, mTAIL will only display lines that do NOT contain "FULL"

Nb of lines displayed before a filtered line

When you are filtering, mTAIL will only display lines matching the filter expression. If you want to display line BEFORE and/or AFTER a filtered line, you can use this spin-edit.

Example:

We are filtering line that contains "FULL". Extra displayed lines are in blue.

Original	<pre>sent /2003_09_20/FULL/PROGRAM/1 sent /2003_09_20/SPATCH/PROGRAM/2 sent /2003_09_20/FULL/PROGRAM/3 sent 2003_09_20/SPATCH/PROGRAM/4 sent /2003_09_20/HPATCH/PROGRAM/5 sent /2003_09_20/HPATCH/PROGRAM/6 sent /2003_09_20/FULL/PROGRAM/7 sent /2003_09_20/SPATCH/PROGRAM/8</pre>
nb lines = 0	<pre>sent /2003_09_20/FULL/PROGRAM/1 .. sent /2003_09_20/FULL/PROGRAM/3 .. sent /2003_09_20/FULL/PROGRAM/7</pre>
nb lines = 1	<pre>sent /2003_09_20/FULL/PROGRAM/1 sent /2003_09_20/SPATCH/PROGRAM/2 sent /2003_09_20/FULL/PROGRAM/3 .. sent /2003_09_20/HPATCH/PROGRAM/6 sent /2003_09_20/FULL/PROGRAM/7</pre>
nb lines = 2	<pre>sent /2003_09_20/FULL/PROGRAM/1 sent /2003_09_20/SPATCH/PROGRAM/2 sent /2003_09_20/FULL/PROGRAM/3 .. sent /2003_09_20/HPATCH/PROGRAM/5 sent /2003_09_20/HPATCH/PROGRAM/6 sent /2003_09_20/FULL/PROGRAM/7</pre>

Alert

If you check this option you activate the alert function. Alert general behavior depends on the general preference setting in the *Configuration Panel*.

Alert can be used to monitor file change or to be alerted when specific text appears in the file being tailed.

To be alerted only if the file changes

Just check *alert* WITHOUT entering any text in the edit zone. When the file changes, mTAIL will:
Bring the application to the front and restored And/or Play a sound (non-minimized)
And/or Play a sound (only if mTAIL is minimized)

...

A *speaker* icon displayed in front of the Alert checkbox indicates that an Alert sound is active.

To be alerted when specific text is found

Just check alert and enter text in the edit field. The text can be a regular expression like in the *filter*. When the text is found, mTAIL will:

- Display the alert window
- Bring back the application to front and restored
- And/or play a sound
- And/or send an eMail [new in v1.4]

The alert window displays the *line that triggered the alert* and when this alert occurred.

Note:

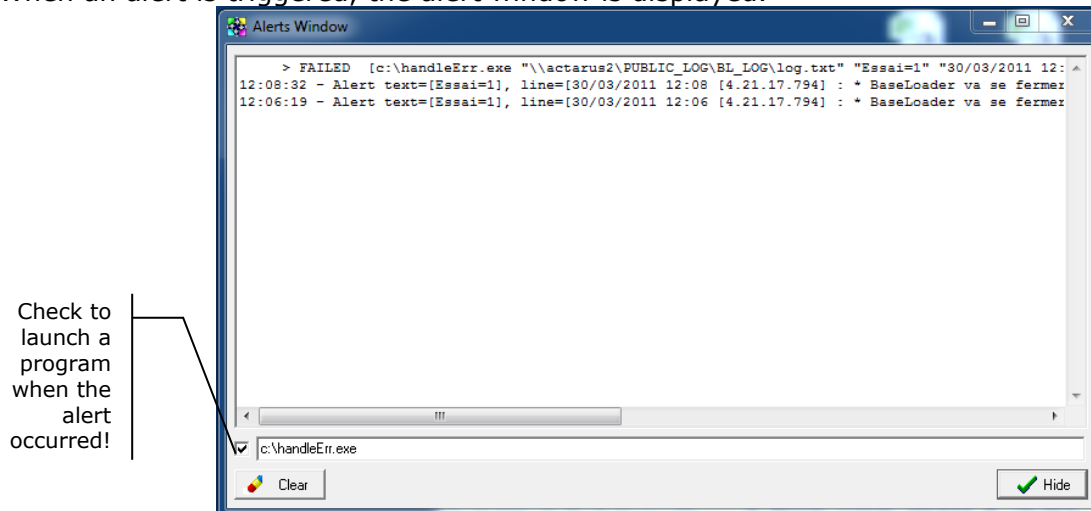
- The alert is only active for the lines displayed in the tail display window. So when the filter option is active, the alert will only search filtered lines (not all the lines in the actual file).

Invert Alert

You can invert the alert condition by clicking on the filter label.

Alerts Window

When an alert is triggered, the alert window is displayed.



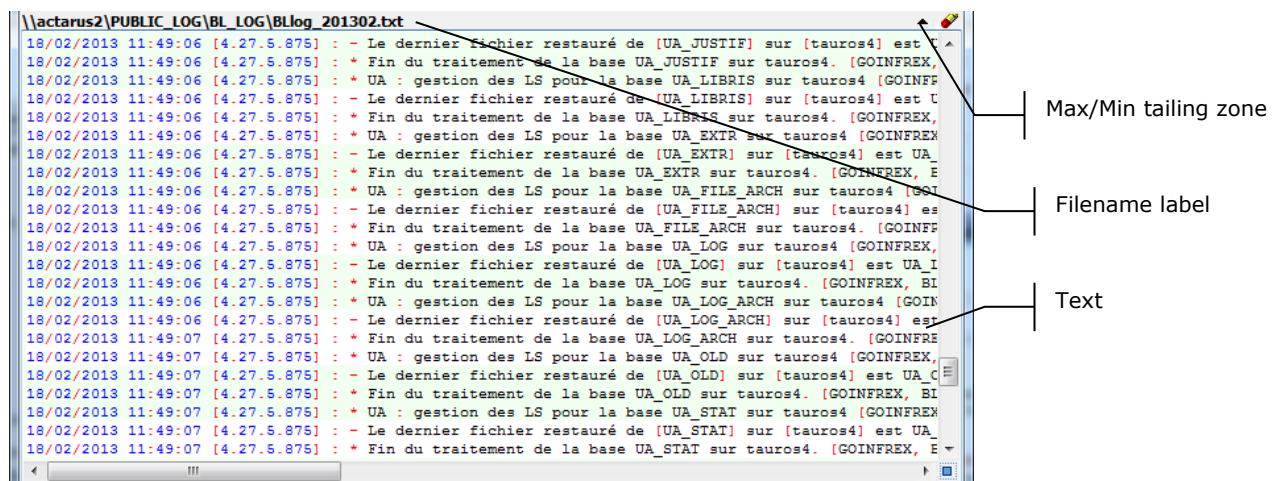
[new v1.8] – You can also launch a program each time an alert occurred. The called program will have the following parameters:

[program] "tailed filename" "alert text" "alert line"

Note:

If the program failed to launch, you will see a "> FAILED" information in the Alerts Windows. If it success, you will see a "> SUCCES" information.

Tailing Zone



Filename label

This field displays the filename of the file being tailed. This lets you see the file mTAIL chose if you used a wildcard in the pathname field.

[new v1.5] – By double clicking in the Filename label, mTAIL will open an explorer Windows in the directory of the file.

Max/Min tailing zone (new v1.8.9)

By clicking on this small arrow you can maximize or minimize text tailing zone.

Text

The text zone displays the current file content. The font can be chosen with the *Configuration Panel*.

The text zone background color is usually:

- Green when mTAIL is tailing
- Red when mTAIL is stopped
- Blue when mTAIL is tailing and a filter is active

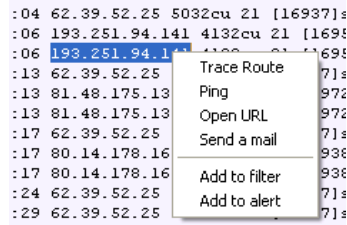
Also see

Selection text

If you click the left mouse button in the text zone, you can select text, like any other text editor. BUT, when a selection is active, mTAIL will automatically select the last occurrence of this selection. For example, if you select an IP address, mTAIL will always select and highlight the last occurrence of the address... this is very useful for monitoring easily some text!

If you click the right mouse button in the text zone, mTAIL will display a popup menu. This popup allows you to do some action with the current selection.

Here I have selected an IP address and I have clicked the popup. With the current selection I can now *Trace Route*, *Ping*, *Open the URL* or *Send a mail*. I can also add to selection to the filter or the alert.



Scrolling

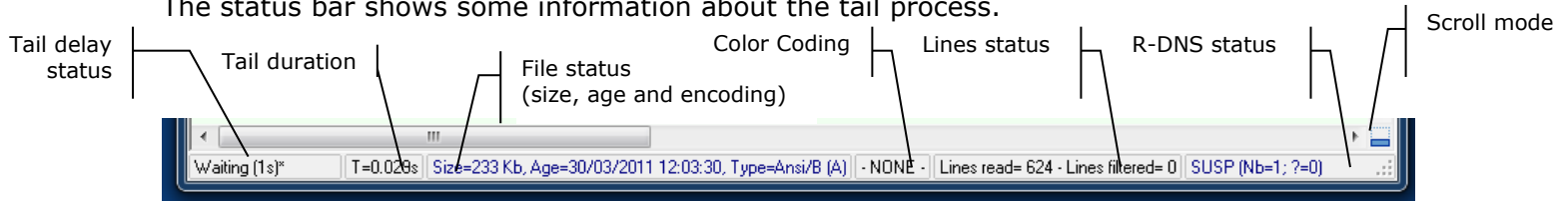
Scrolling is allowed while tailing! See the *Scroll mode* entry in the next section for more information about scrolling

Note:

- A default font for each pathname can be entered via the *Configuration Panel*.

Status Bar

The status bar shows some information about the tail process.



[new in v1.5] – The file status displays the encoding of the file

Tail delay status

This status panel shows you what mTAIL is doing. The different states are:

STOPPED	The tail process is stopped.
Reading	mTAIL reads the file from the disk
Analyzing	mTAIL analyzes each line according to the options, filters, alarms
Analyze > memo	mTAIL transfers the result of previous phase into the memo control for display [new v1.4]
Waiting x s	mTAIL displays resulting lines and waits x seconds for next cycle.
Autowaiting x s	Same has "Waiting" but mTAIL has the autotimer functionality activated for this file.
F-Waiting x s	mTAIL displays resulting lines and waits x seconds for next cycle. Fast read is on. [new in v1.7]
F-Autowaiting x s	Same has "Waiting" but mTAIL has the autotimer functionality activated for this file. Fast read is on [new in v1.7]

If the *Autotimer* option is activated, you can click on the Tail delay status panel to reset the autotimer to original time value.

Note:

- If the * (star) character appears after Waiting or Autowaiting, this indicates that you are using the THREAD method to tail the file. See *Configuration Panel* for more info.
- If "F-" characters appear before Waiting of Autowaiting, this indicates that the Fast Reading option is enabled. [new in v1.7]

Tail duration [change in v1.4]

This value $T=s.mmm$ displays the total duration of the last tail process.

Example:

$T=0.132$ the whole tail process of reading the file, analyzing, etc. ... has taken 132 milliseconds.

File status

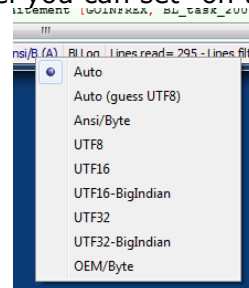
This status panel displays the current file size and its last modification date (age).

In most case, the modification date is equal to the last time the file content changed. But this is not always true (for example with IIS log file, because of the IIS allocation mode mentioned above).

[new in v1.5] – The type of encoding is displayed after "Type="

Ansi/B	Text is Ansi or Raw byte encoded
UTF8	Text is encoded in UTF8
UTF16	Text is encoded in UTF16
UTF16B	Text is encoded in UTF16 BigIndian
UTF32	Text is encoded in UTF32
UTF32B	Text is encoded in UTF32 BigIndian
OEM-Byte	Text is an OEM (Dos) Encoded (new v1.8.9)
RTF	Text is an RTF file
DOC	Text is an Microsoft DOC file

[new in v1.7] – By clicking on this panel you can set "on the fly" the file encoding:



Color Coding [new v1.8]

Right-click on this panel to choose the color coding scheme applied to the file. Changing the color coding scheme will not update the default value set in the File Manager.

[New v1.8.3] If the name is followed by a (?), this means that the color coding scheme is lacking in the ccini file. Please refer to Color Coding paragraph for more details of this new feature!

Left-click will force reloading the Color Coding Ini file (useful if you do some modifications)

Double-Click will launch the shell with the Color Coding Ini file. You can easy make modifications and reload the result with the Left-Click.

Lines status

This status panel shows the number of lines read and the number of lines filtered.

R-DNS Status

This panel shows you the status of the R-DNS process. R-DNS process is activated every time the file change, but you can click on the panel to force the refresh.

IDLE (Nb=x)	R-DNS process is starting. X shows the number of entries that will be checked
QUERY (Nb=x/y)	R-DNS is querying the entry x over y
CHECK (Nb=x/y)	R-DNS is checking entry x over y
SUSP (Nb=x; ?=z)	R-DNS has finished. X entries are solved, z entries are unknown.

Note:

If you right-click on the panel, you can copy the R-DNS resolve list to the clipboard.

Scroll mode

Just below the vertical scroll bar, you can see a small icon. This icon displays the scroll mode.



Always scroll to end mode

When this icon is displayed when you have scrolled to the end of the file. In this mode, mTAIL will always scroll down to last line, even if new lines appear.



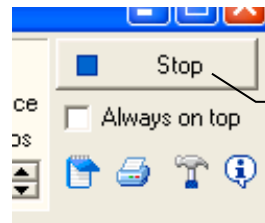
Keep scroll pos mode

When this icon is displayed, you have scroll to a specific line. mTAIL will keep this scroll position even if new lines appear at the end of the file.

Note:

You can avoid the *Keep scroll pos mode* by checking the option *Always scroll to end of file down when new lines appear* in the configuration dialog.

Start / Stop button



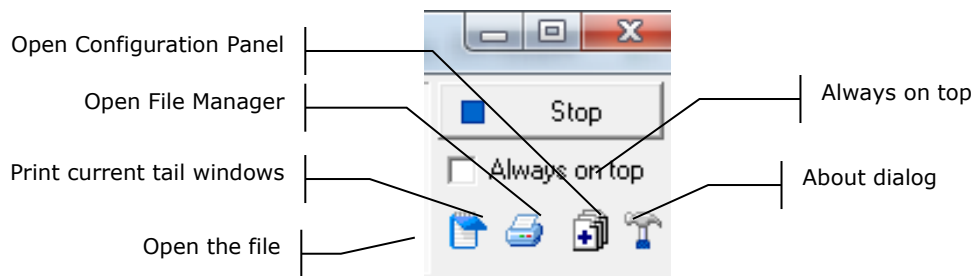
Start/Stop button

To start tailing, just click the start button.

Pressing the start button will add automatically set the current pathname in the *path combo box* (if you were using a wildcard).

To stop tailing, just click the stop button.

Global action/button



Always on top

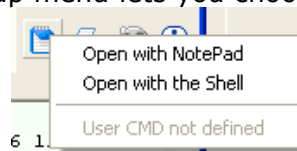
If you check *Always on top*, the mTAIL windows will always be displayed on top of other applications. This allows you to see mTAIL even if a window would have overlapped.

The default *Always on top* can be set via the Configuration Panel.

Open the file [change v1.8.6]

If you click this button, the current file will be opened with associated program in the Shell (usually NotePad).

If you left-click on this button, a popup menu lets you choose another way to open the file:



Open with NotePad

The current file will be opened with Windows NotePad. **Warning:** NotePad has problems loading large files, especially in W9x systems.

Open with the Shell

If you click this, mTAIL launches a *shellopen command* with the current filename. So, depending on the file association, the registered editor will be launched.

Example:

If DreamWeaver is registered for *html* file, and you are tailing *index.html*, clicking this button will launch DreamWeaver

User command

For each file, you can define a user command.

Configuration Panel

Click this button to open the *Configuration Panel* dialog. This dialog lets you choose default mTAIL setting and lets you configure the *pathname* combo box.

See *Configuration Panel* section for more info.

About [change in v1.7]

About Dialog is now the last tab of the Configuration Panel. You can also use the popup menu in the tray bar and choose *About*.

Using Filter and Alert expression

Introduction

Regular Expressions are a widely-used method of specifying patterns of text to search for. Special **metacharacters** allow you to specify, for instance, that a particular string you are looking for occurs at the beginning or end of a line, or contains **n** occurrences of a certain character. Regular expressions look ugly for novices, but really they are very simple (well, usually simple ☺), handy and powerful tool.

Let's start our learning trip!

[new v1.8.5] – mTAIL now use built-in Delphi Regular Expression Engine

BEWARE : mTAIL parses one line at a time, not the entire text being displayed!

Simple matches

Any single character matches itself, unless it is a **metacharacter** with a special meaning described below.

A series of characters matches that series of characters in the target string, so the pattern "bluh" would match "bluh" in the target string. Quite simple, eh ?

You can cause characters that normally function as **metacharacters** or **escape sequences** to be interpreted literally by 'escaping' them by preceding them with a backslash "\", for instance: metacharacter "^" match beginning of string, but "\"^" match character "^", "\\" match "\" and so on.

Examples :

foobar	<i>matchs string 'foobar'</i>
^FooBarPtr	<i>matchs '^FooBarPtr'</i>

Escape sequences

Characters may be specified using a **escape sequences** syntax much like that used in C and Perl: "\n" matches a newline, "\t" a tab, etc. More generally, \xnn, where nn is a string of hexadecimal digits, matches the character whose ASCII value is nn. If You need wide (Unicode) character code, You can use '\x{nnnn}', where 'nnnn' - one or more hexadecimal digits.

\xnn	<i>char with hex code nn</i>
\x{nnnn}	<i>char with hex code nnnn (one byte for plain text and two bytes for Unicode)</i>
\t	<i>tab (HT/TAB), same as \x09</i>
\n	<i>newline (NL), same as \x0a</i>
\r	<i>car.return (CR), same as \x0d</i>
\f	<i>form feed (FF), same as \x0c</i>
\a	<i>alarm (bell) (BEL), same as \x07</i>
\e	<i>escape (ESC), same as \x1b</i>

Examples:

foo\x20bar	<i>matchs 'foo bar' (note space in the middle)</i>
\tfoobar	<i>matchs 'foobar' predefined by tab</i>

Character classes

You can specify a **character class**, by enclosing a list of characters in [], which will match any **one** character from the list.

If the first character after the "[" is "^", the class matches any character **not** in the list.

Examples:

```
foob[aeiou]r    finds strings 'foobar', 'foober' etc. but not 'foobbr', 'foobcr' etc.
foob[^aeiou]r    find strings 'foobbr', 'foobcr' etc. but not 'foobar', 'foober' etc.
```

Within a list, the "-" character is used to specify a **range**, so that a-z represents all characters between "a" and "z", inclusive.

If You want "-" itself to be a member of a class, put it at the start or end of the list, or escape it with a backslash. If You want "]" you may place it at the start of list or escape it with a backslash.

Examples:

```
[-az]          matchs 'a', 'z' and '-'
[az-]          matchs 'a', 'z' and '-'
[a\-z]         matchs 'a', 'z' and '-'
[a-z]          matchs all twenty six small characters from 'a' to 'z'
[\n-\x0D]      matchs any of #10,#11,#12,#13.
[\d-t]         matchs any digit, '-' or 't'.
[]-a]         matchs any char from ']'..'a'.
```

Metacharacters

Metacharacters are special characters which are the essence of Regular Expressions. There are different types of metacharacters, described below.

Metacharacters - line separators

```
^          start of line
$          end of line
\A         start of text
\Z         end of text
.          any character in line
```

Examples:

```
^foobar      matchs string 'foobar' only if it's at the beginning of line
foobar$      matchs string 'foobar' only if it's at the end of line
^foobar$     matchs string 'foobar' only if it's the only string in line
foob.r       matchs strings like 'foobar', 'foobbr', 'foob1r' and so on
```

The "^" metacharacter by default is only guaranteed to match at the beginning of the input string/text, the "\$" metacharacter only at the end. Embedded line separators will not be matched by "^" or "\$".

The "." metacharacter by default matches any character.

"^" is at the beginning of a input string. Note that there is no empty line within the sequence \x0D\x0A.

"\$" is at the end of a input string. Note that there is no empty line within the sequence \x0D\x0A.

"." matches any character

Note that "^.*\$" (an empty line pattern) does not match the empty string within the sequence `\x0D\x0A`, but matches the empty string within the sequence `\x0A\x0D`.

Metacharacters - predefined classes

<code>\w</code>	<i>an alphanumeric character (including "_")</i>
<code>\W</code>	<i>a non-alphanumeric</i>
<code>\d</code>	<i>a numeric character</i>
<code>\D</code>	<i>a non-numeric</i>
<code>\s</code>	<i>any space (same as [\t\n\r\f])</i>
<code>\S</code>	<i>a non-space</i>

You may use `\w`, `\d` and `\s` within custom **character classes**.

Examples:

`foob\d r` matches strings like 'foob1r', 'foob6r' and so on but not 'foobar', 'foobbr' and so on
`foob[\w\s] r` matches strings like 'foobar', 'foob r', 'foobbr' and so on but not 'foob1r', 'foob=r' and so on

Metacharacters - word boundaries

<code>\b</code>	<i>Match a word boundary</i>
<code>\B</code>	<i>Match a non-(word boundary)</i>

A word boundary (`\b`) is a spot between two characters that has a `\w` on one side of it and a `\W` on the other side of it (in either order), counting the imaginary characters off the beginning and end of the string as matching a `\W`.

Metacharacters - iterators

Any item of a regular expression may be followed by another type of metacharacters - **iterators**. Using this metacharacters You can specify number of occurrences of previous character, **metacharacter** or **subexpression**.

<code>*</code>	<i>zero or more ("greedy"), similar to {0,}</i>
<code>+</code>	<i>one or more ("greedy"), similar to {1,}</i>
<code>?</code>	<i>zero or one ("greedy"), similar to {0,1}</i>
<code>{n}</code>	<i>exactly n times ("greedy")</i>
<code>{n,}</code>	<i>at least n times ("greedy")</i>
<code>{n,m}</code>	<i>at least n but not more than m times ("greedy")</i>
<code>*?</code>	<i>zero or more ("non-greedy"), similar to {0,}?</i>
<code>+?</code>	<i>one or more ("non-greedy"), similar to {1,}?</i>
<code>??</code>	<i>zero or one ("non-greedy"), similar to {0,1}?</i>
<code>{n}?</code>	<i>exactly n times ("non-greedy")</i>
<code>{n,}?</code>	<i>at least n times ("non-greedy")</i>
<code>{n,m}?</code>	<i>at least n but not more than m times ("non-greedy")</i>

So, digits in curly brackets of the form `{n,m}`, specify the minimum number of times to match the item `n` and the maximum `m`. The form `{n}` is equivalent to `{n,n}` and matches exactly `n` times. The form `{n,}` matches `n` or more times. There is no limit to the size of `n` or `m`, but large

numbers will chew up more memory and slow down r.e. execution.

If a curly bracket occurs in any other context, it is treated as a regular character.

Examples:

```
foob.*r    matchs strings like 'foobar', 'foobalkjdfikj9r' and 'foobr'
foob.+r    matchs strings like 'foobar', 'foobalkjdfikj9r' but not 'foobr'
foob.?r    matchs strings like 'foobar', 'foobbr' and 'foobr' but not 'foobalkj9r'
fooba{2}r  matchs the string 'foobaar'
fooba{2,}r matchs strings like 'foobaar', 'foobaaa', 'foobaaaa' etc.
fooba{2,3}r matchs strings like 'foobaar', or 'foobaaa' but not 'foobaaaa'
```

A little explanation about "greediness". "Greedy" takes as many as possible, "non-greedy" takes as few as possible. For example, 'b+' and 'b*' applied to string 'abbbbc' return 'bbbb', 'b+?' returns 'b', 'b*?' returns empty string, 'b{2,3}?' returns 'bb', 'b{2,3}' returns 'bbb'.

Metacharacters - alternatives

You can specify a series of **alternatives** for a pattern using "|" to separate them, so that fee|fie|foe will match any of "fee", "fie", or "foe" in the target string (as would f(e|i|o)e). The first alternative includes everything from the last pattern delimiter ("(", "[", or the beginning of the pattern) up to the first "|", and the last alternative contains everything from the last "|" to the next pattern delimiter. For this reason, it's common practice to include alternatives in parentheses, to minimize confusion about where they start and end.

Alternatives are tried from left to right, so the first alternative found for which the entire expression matches, is the one that is chosen. This means that alternatives are not necessarily greedy. For example: when matching foo|foot against "barefoot", only the "foo" part will match, as that is the first alternative tried, and it successfully matches the target string. (This might not seem important, but it is important when you are capturing matched text using parentheses.) Also remember that "|" is interpreted as a literal within square brackets, so if You write [fee|fie|foe] You're really only matching [feio|].

Examples:

```
foo(bar|foo)  matchs strings 'foobar' or 'foofoo'.
```

Metacharacters - subexpressions

The bracketing construct (...) may also be used for define r.e. subexpressions.

Subexpressions are numbered based on the left to right order of their opening parenthesis. First subexpression has number '1'.

Examples:

```
(foobar){8,10}  matchs strings which contain 8, 9 or 10 instances of the 'foobar'
foob([0-9]|a+)r matchs 'foob0r', 'foob1r', 'foobar', 'foobaar', 'foobaar' etc.
```

Metacharacters - backreferences

Metacharacters \1 through \9 are interpreted as backreferences. \<n> matches previously matched subexpression #<n>.

Examples:

(.)\1+ *matches 'aaaa' and 'cc'.*
(.+)\1+ *also match 'abab' and '123123'*
(['"]?) (\d+) \1 *matches '"13" (in double quotes), or '4' (in single quotes) or 77 (without quotes) etc*

Perl extensions

(?imsxr-imsxr)

You may use it into r.e. for modifying modifiers by the fly. If this construction inlined into subexpression, then it effects only into this subexpression

Examples:

(?i) Saint-Petersburg *matches 'Saint-petersburg' and 'Saint-Petersburg'*
(?i) Saint-(?-i) Petersburg *matches 'Saint-Petersburg' but not 'Saint-petersburg'*
(?i) (Saint-)?Petersburg *matches 'Saint-petersburg' and 'saint-petersburg'*
((?i) Saint-)?Petersburg *matches 'saint-Petersburg', but not 'saint-petersburg'*

(?#text)

A comment, the text is ignored. Note that the program closes the comment as soon as it sees a ")", so there is no way to put a literal ")" in the comment.

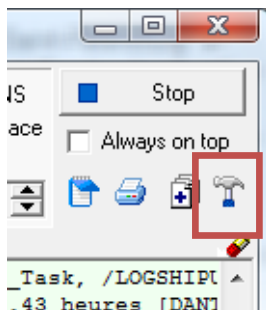
Some examples

Look the regular-Expressions.info web site for many useful examples!

<http://www.regular-expressions.info/examples.html>

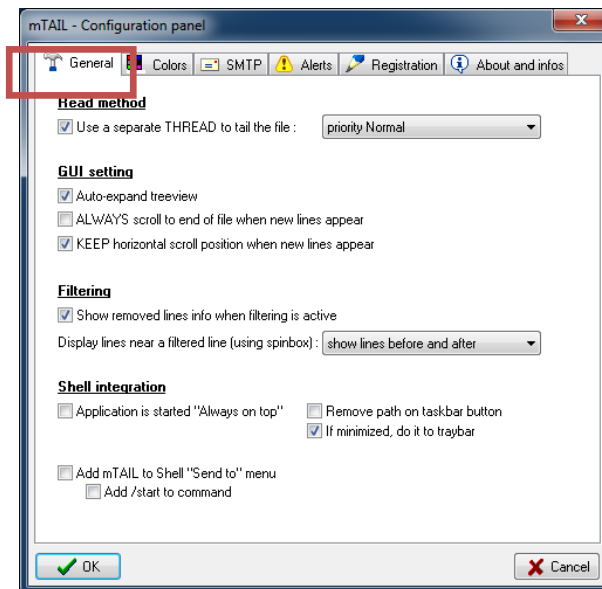
Configuration Panel

The dialog lets you manage some general settings. You can open the Configuration Panel by using the Hammer Icon:



General Tab

Click in the *General* tab sheet to display the general setting.



Use a separate THREAD to tail the file

If this option is checked, mTAIL will spawn a THREAD to read/analyze/display the file. You can select the priority of the THREAD by choosing an item in the combo box (*Idle* is the lowest priority and *time critical* the highest).

THREADs are very very useful because "reading on slow device" will not slow down your computer.

Auto-expand treeview

When opening the pathname treebox (with the blue arrow), the pathname treeview will be auto-expanded (all nodes and sub-nodes will be expanded)

Always scroll to end when new lines appear (default = false)

This option allows overriding the *Keep scroll pos mode*. When checked, mTAIL will scroll to end of file, each time a new line appears.

Keep horizontal scroll position when new lines appear (default = true)

[new in v1.8.5]

When checked, mTAIL will keep horizontal scroll position even when new lines appear. If not, the horizontal scroll position will be reset to 0.

Show removed lines info when filtering is active

If you check this option, and you have an active filter, mTAIL will display information about "removed" lines not matching the filter.

The information is given by tree dot characters, followed by the number of removed lines in bracket.

Example:

```
07:57:24 0577CS 21 [16974]sent /2003_09_20/FULL/PROGRAM/DATA/BDDIDENT
... (5)  ← Five lines were removed by the filter
07:57:25 0577CS 21 [16974]sent /2003_09_20/FULL/PROGRAM/DATA/BDDIDENT
```

→ Display lines near a filtered line [new in v1.4]

This sub-option allows you to set the effect of the previous option. You can choose before, after or both.

Application is started "Always on top"

If this option is checked, mTAIL will be started with the *Always on top* option checked in the main windows.

If minimized, do it to traybar

If this option is checked, mTAIL will be minimized to tray bar when the minimized button is clicked. When minimized, you will see the mTAIL icon in the tray bar: the icon will animate if you are currently tailing a file. Each animation occurs when a tailing cycle starts.

If this option is unchecked, mTAIL will be minimized in the application bar.

Remove path on taskbar button [new in v1.5]

Because the taskbar buttons are small, is it possible with this option to remove the path of the file tailed. So, for example, if you are tailing several files in the same directory, it's easy to quickly.

With option on, you see that the filename is beginning by IPEXTD :



Add mTAIL to Shell "Send to" menu

Clicking this option will add mTAIL to the Windows "Send to" menu. You can call this menu by right clicking with the mouse in the Windows Shell.

Un-clicking the option removes it.

→ Add /start to command *[new in v1.5]*

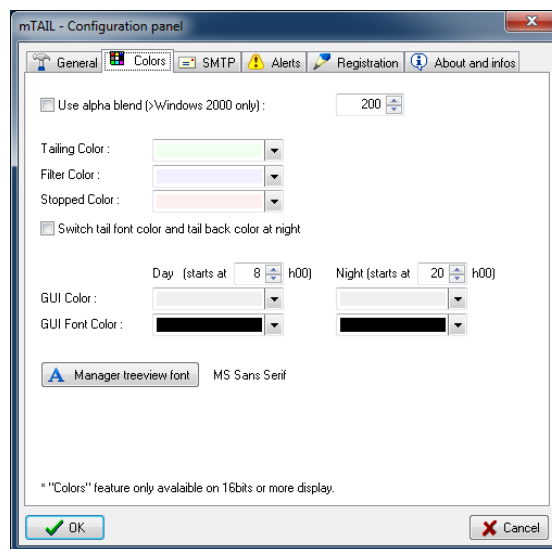
If you the option is checked BEFORE you use Add mTAIL to Shell, the shortcut with contain the /start command line option.

So, every time you use the "Send To", mTAIL will launch mTAIL and also START directly the tailing.

Note: you can also edit the "Send To" Shortcut with Windows to add or remove extra command line option.

Note: double clicking the shortcut path, will open an explorer windows in the shortcut directory.

Colors Tab



Use alpha blend (only available on 2000, XP)

By checking this option, mTAIL will use the directX alpha-blend. The value gives the transparency factor.

Using alpha-blend can slow down the computer drastically on poor graphic card.

Tailing Color

Color of the background tail windows when tailing is active.

Filter Color

Color of the background tail windows when tailing is active AND filtering is active.

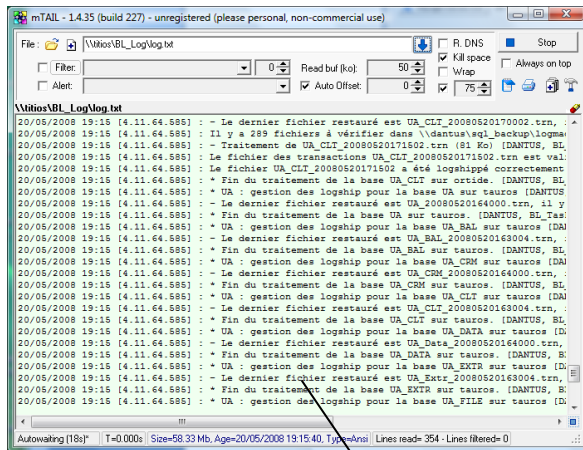
Stopped Color

Color of the background tail windows when no tailing.

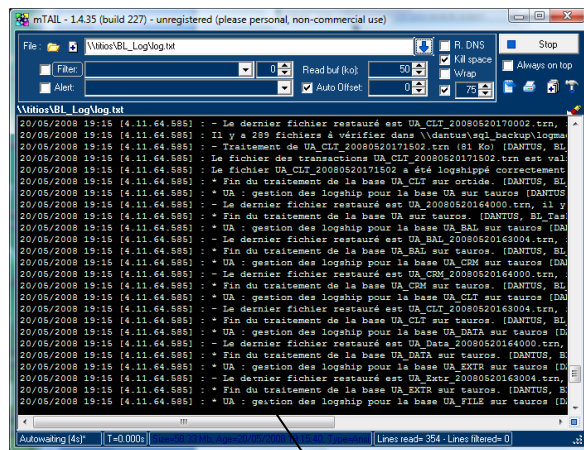
Switch tail font color and tail back color at night

If this option is checked, mTAIL will invert the background color with the font color during night (configured with Night start at spin box).

This option is useful if you use mTAIL on poor light environment to minimize the luminosity.



It is the day !
Font is black, back is green.



It is the night!
Font is green, back is black

Day start / Night start

Those two spin boxes allows you to define at which hour the days begins and the night begins.

GUI Color

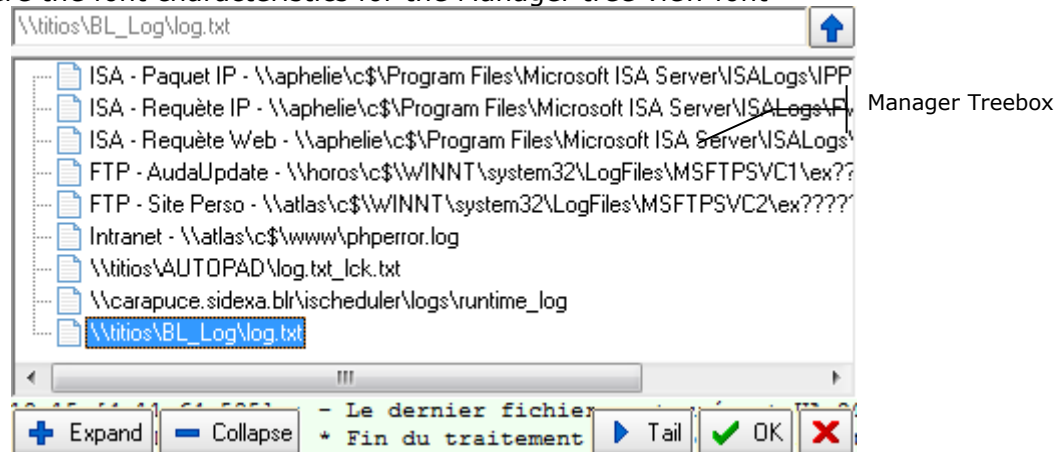
You can define here the color of the GUI background during day and night period. You can choose the same color for both periods.

GUI Font Color

You can define here the color of the GUI font during day and night period. You can choose the same color for both periods.

Manager tree view font

You can select here the font characteristics for the Manager tree view font



SMTP Tab

SMTP Settings

mTAIL alerts can be sent to a specific eMail. eMail are sent by using the SMTP settings in the general tabs.

SMTP Server: IP address or Name of the SMTP server

SMTP server Port: You can specify the TCP port of the SMTP Server *[new in v1.8.2]*

Login: Login of the user that will be used to send to mail (don't forget domain qualifier for Windows SMTP)

Password: Password of the login. The Password is stored encrypted **BUT WEAKLY.** Don't use critical account/password.

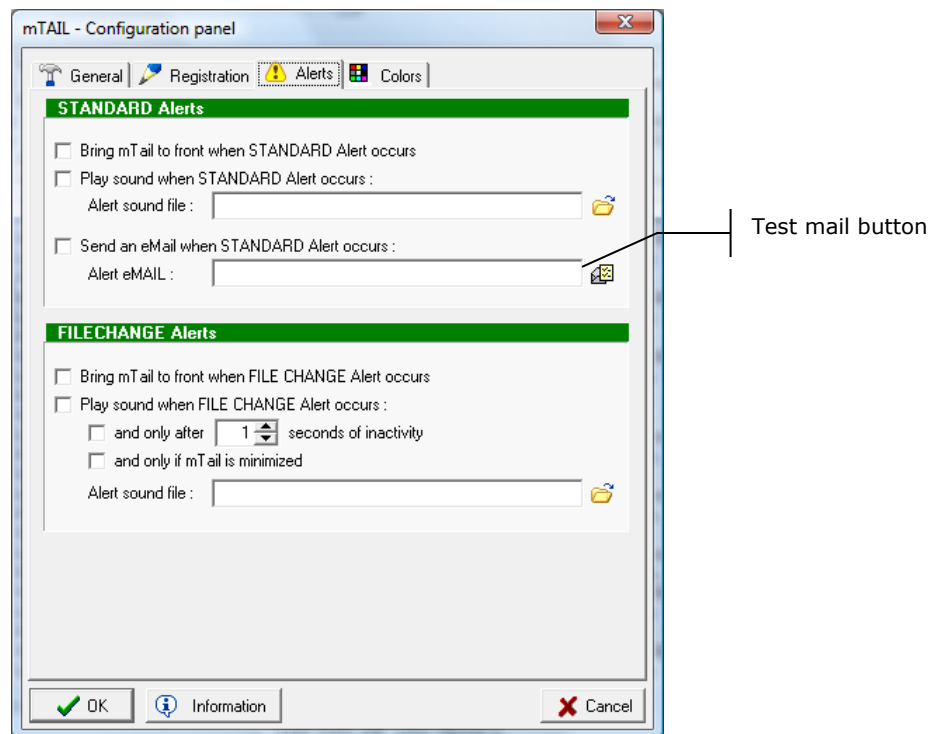
Authentication: Specify the authentication method when posting the mail *[new v1.8.3]*

From: eMail address used to send to mail. This address will appear as the sender of the Alert.

Test to: Read only file that shows the destination mail address.

Hint: you can test those settings by sending a test eMail using the small mail icon *[new v1.8.3]*.

Alerts Tab



Two types of alert is handled by mTAIL. The first are standard alert and the other are File change alert.

Standard Alerts

Standard alert occurs when you enter a regular expression in the alert combo box. When the expression is found is the alert occurs.

Bring mTAIL to front when STANDARD Alert occurs

If you check this option, mTAIL go back to front and the alert dialog pop up will be shown.

Play sound when STANDARD Alert occurs

If this option is checked, the sound file stated in the edit box will be played EACH TIME the alert occurs.

Send an eMail when STANDARD Alert occurs

If this option is checked, an eMail is sent at the specified address EACH TIME the alert occurs. You can send a test email using the test mail button.

File change Alerts

File change alert occurs when you check Alert without entering a regular expression.

Bring mTAIL to front when FILE CHANGE Alert occurs

If you check this option and an alert occurs, mTAIL will go back to front.

Play sound when FILE CHANGE Alert occurs

If this option is checked, the sound file stated in the edit box will be played EACH TIME the alert occurs, according to the following options

And only after x seconds of inactivity

The sound will be played only if the file has change after the file has NOT change since x seconds.

This is useful to be alerted when "batch modification" occurs.

And only if mTAIL is minimized

The sound will only be play if mTAIL is minimized.

Registration Tab

mTAIL is free. If you want to use mTAIL in a commercial/business environment, I really encourage you to register mTAIL. mTAIL registration donation is usually a 20 EUR. Corporate License is a Lifetime License.
Registered version of mTAIL has NO additional feature.

To register mTAIL, just enter the registration info in the corresponding boxes.

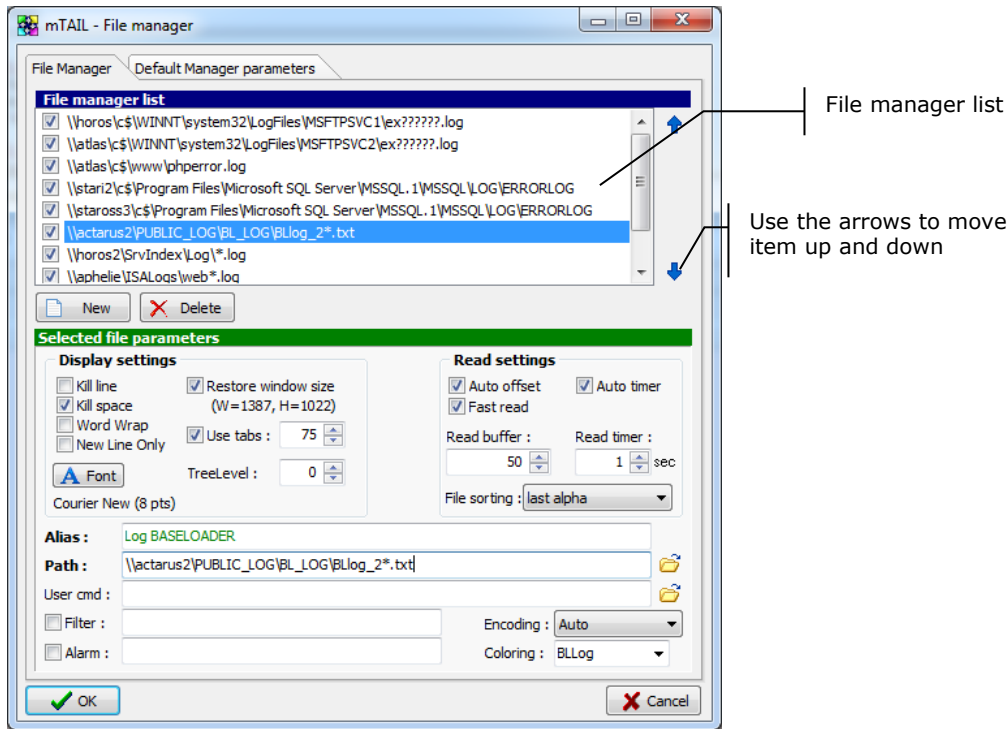
If you lose your registration infos, contact me by eMail (ophilipp@free.fr)

About and info tab [new in v1.5]

You will find useful information here 😊 like:

- ➔ Some info about the program
- ➔ The OS [new in v1.7]
- ➔ The command line at launch
- ➔ The path of forced ini ou ccini if any [new in v1.8.2]
- ➔ The command line ID (see Command line option for more info about that!)

File Manager



Files manager list

In this list box you find all the items found the Pathname combo box. The order in the list box will be the order in the Pathname treebox.

You can move items up and down using the dark blue arrow button right to the scroll bar. You can also use the TreeLevel spinebox to create hierarchical view in the treebox.

If an item is checked, it will appear in the Pathname treebox. If not, the item will not appear.

To edit an item, just click on it: his parameters will be show in the *Selected file parameters* panel.

New button

Click the *new* button to add a new item in the list box.

Delete button

Click the *delete* button the delete selected items in the list box.

Selected file parameters

Just enter the parameters of the selected item in this panel. You can define for each entry:

- The display settings (including The font, size and attributes)
- The read settings
- The Alias
- The Pathname to log
- The user command
- The filter
- The Alarm

The Alias is only a "simple name" of the pathname. This is very convenient to give "explicit" name. Alias are displayed in green just before the pathname.

The user command MUST be in the Windows Path or have the full path. The file currently tailed can be transmitted to the usercmd by the %1 parameter.

Display setting - Kill line / Kill Space / Word Wrap / Use Tabs

Set the according option for the file.

More about kill line option

If this option is checked, mTAIL does not display blank lines (or lines containing only spaces) in the display area.

Example:

Original file	Kill line not checked	Kill line checked
Line1 : this is an example Line2 : this is an example Line3 : 1 2 3 4 5 Line4 : 1 2 4 5	Line1: this is an example Line2: this is an example Line3 : 1 2 3 4 5 Line4 : 1 2 4 5	Line1: this is an example Line2: this is an example Line3 : 1 2 3 4 5 Line4 : 1 2 4 5

Note:

- A default kill line option can be set for each path name via the *Configuration Panel*.

Display setting - New Line Only

When the file is started for tailing, only news lines are displayed. Without, mTail will "back" tail and display old lines according to the buffer size.

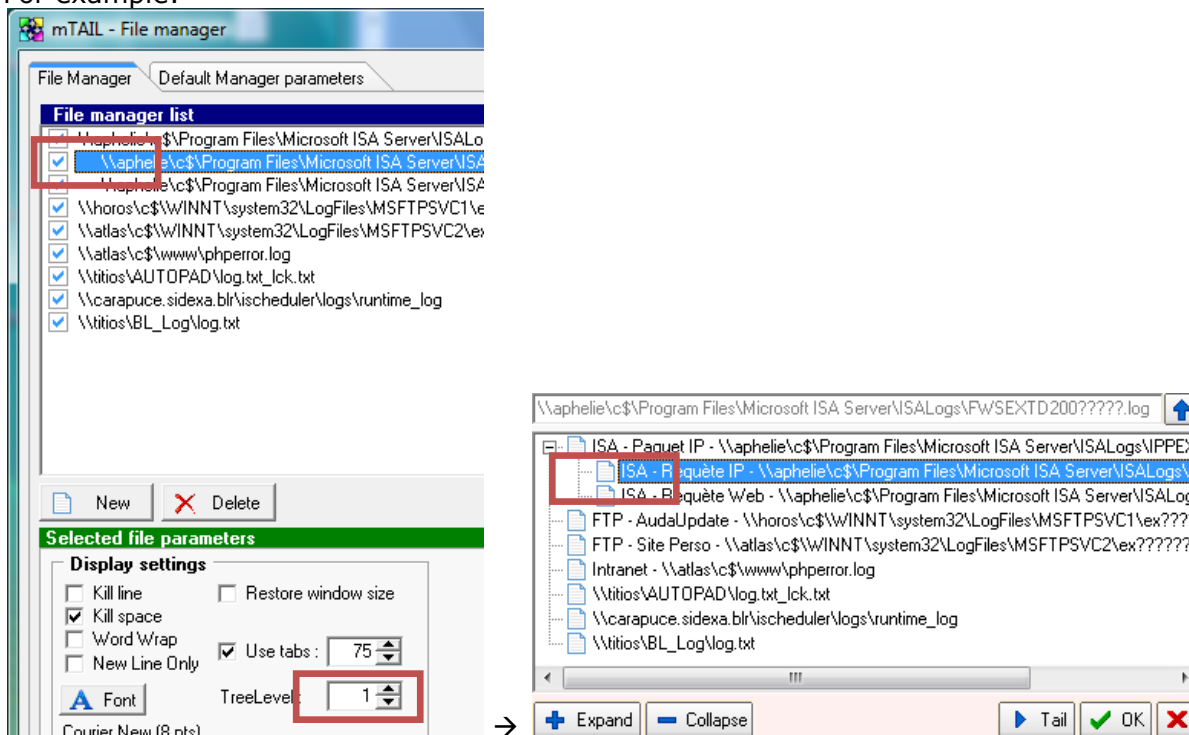
Display setting - Font

You can choose here a specific font for the text display in the tail zone.

Display setting - TreeLevel

You can choose here the level of the item in the treebox manager. This can help you to organize all your files in a hierarchical order.

For example:



Display setting - Restore windows size

If you activate this option, mTAIL will save the current mTAIL windows size as you STOP to tail the file (or exit mTAIL).

The next time you START to tail, mTAIL will restore the size.

Note: the windows is restore when you hit the button, not when you launch mTAIL !

Read setting – Autotimer, Fast Read, Read Timer

If you activate AutoTimer option, mTAIL will dynamically adjust the "Read Timer" value according the following rules:

- Initial timer value is given by "Read Timer value"
- If the file doesn't change after a tail process, the timer value is increased by one second (max 60 sec).
- If the file changes, the timer is reseted to the initial value.
- If the global tailing process cycle time (read/analyze/display) is lower than the current time value, the timer value is set to the global tailing process cycle time + 1 sec.

Hint: Clicking in the status panel will reset the initial timer value.

[NEW in v1.7] Fast Read is a new option that enables mTAIL to use Operating System modification date to know if the file has changed. Fast Read can be used with Autotimer to reduce drastically I/O and processor time.

The back draw of Fast Read is that the OS modification date is sometime cached or updated at the end of the modification transaction. This could produce small display lag.

	Autotimer OFF Fast Read OFF	Autotimer OFF Fast Read On	Autotimer On Fast Read OFF	Autotimer On Fast Read On
Positive	Fastest way to see modifications!	Modifications displayed near real-time. But modification lag can occur.	Modifications are displayed only after each autotimer.	Modifications are displayed only after each autotimer. Modification lag can occur.
Negative	I/O, network and processor consumption	Low I/O, network and processor consumption	Medium to Low I/O, network and processor consumption	Lowest I/O, network and processor consumption

Nevertheless, Fast Read could be turned ON for near every tailing. Only ultra-speed critical monitoring (or when tailing file on OS that does not support modification date) should turn off this option.

Read setting – Auto offset, Read Buffer

If you activate Auto Offset option, mTAIL will dynamically adjust the "Offset Value"

Read setting – File sorting

This option controls the way mTAIL will select the file to tail, when wildcard is used.

last Alpha	mTAIL will tail the last alphabetic file
last fileage	mTAIL will tail the last modified file
last Creation	mTAIL will tail the last created file
Last numeric	mTAIL will tail the last file according to the first numeric in the filename

Alias

This is an alias you can give to the filename. This can help you to clarify some obscure filename. The alias is written in Green color.

Encoding [new v1.7]

Choose here the specific charset encoding for this file.

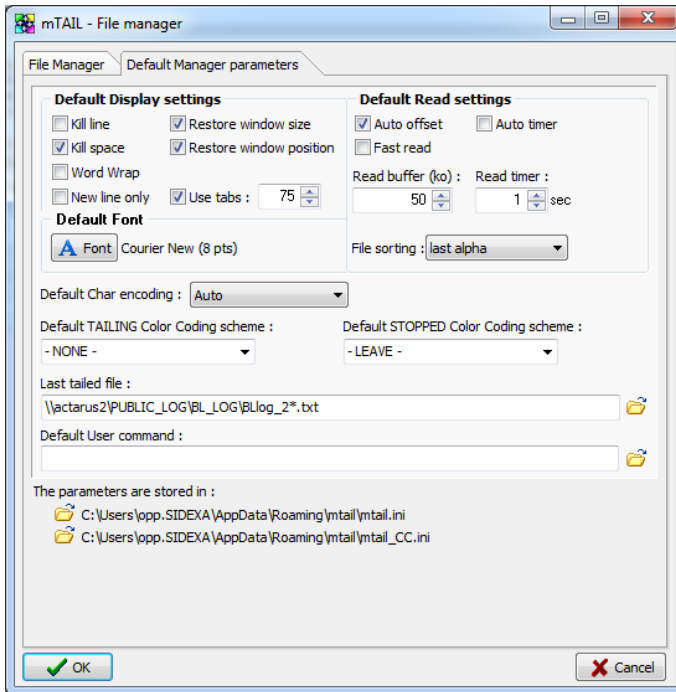
Coloring [new v1.8]

Choose here the specific color coding scheme for this file.

Edit default parameters

If you click this check box, the panel below the list box caption will change to *Default Parameters*.

You can change default parameter option for mTAIL. All new items will inherit from those default parameters until redefined at item level.



You can also:

- set a "STOPPED" Color Coding Scheme to force a coloring scheme when tailing is stopped [new v1.8.3]
- force NO COLORING in v1.8.2 (- NONE -),
- keep the current scheme (- LEAVE -)
- keep also mTAIL windows position (and monitor) [new v1.8.7]

Color coding scheme [new v1.8]

You can now supply to mTAIL a color coding scheme *ini* file to allow color coding in the tail windows.

For now, there is NO editor within mTAIL to create a new Color coding ini file. You will have to use a notepad or any other editor to do that.

Color Coding Ini file should be located at the same place as mTail.ini file. The common name is **mTail_cc.ini**

mTail_cc.ini can hold any number of Color Coding scheme. Each Color Coding scheme can hold up to 100 rules.

A *scheme* is defined by an *ini* section (like [Name]). The section name is the scheme name.

[Name]		Header / Single
Hint=	An hint that describe the color scheme	Single, optional
WordSeparator=	Contains all the character the mTAIL will use the separate word. Each Word will be submitted to the Rule engine.	Single, mandatory
LittDelStartX	Characters that begin a literal string. No word are searched within literal string	Loop X=0...99, optional
LittDelEnd	Character that end a Literal String	Loop X=0...99, optional
RULES list	<i>See structure below</i>	

Remark: If the scheme name is `default`, the scheme will be applied by default for all files that haven't a specific color scheme.

Note: don't forget that the rules are applied on each "word". "Words" are delimited by the WordSeparator character list.

RULES list will allow you to specify the coloring of each "line" or "word". You can use SimpleMatch (like MS-DOS joker) or complex RegExp.

First, for each line, "line" scope rules are checked to determine default line text and background color.

Second, for each keyword, "word" scope rules are processed.

Rules are processed in the INI file ordering. If the rule matches, the following rules of the current scope are not processed.

For each rule, see the structure below. X is the rule number.

Rules Liste		Header / Single	Default
RuleName X =	Describe the rule	Single, mandatory	
RuleScope X =	Indicated the scope of the rule. If line , the rule applies on the whole line. If word , the rule applies on the current keyword. line rules are checked first for selecting default background and word color. word rules are applied after and supersede line rules.	Single, optional	word
RuleMode X =	Only "match" for the moment	Single, optional	match
RuleStopOnOk X = [new v1.8.3]	If Y, stops processing following rules if the rule is triggered. Useful to avoid word rules if a line rule is triggered.	Single, optional	N
AllowedChar X =	List of allowed char in the WORD. If not present, all characters are allowed	Single, optional	
ForbiddenChar X =	List of forbidden char in the WORD. If present rule not OK. If not present no characters are forbidden	Single, optional	
SimpleMatch XxY =	Simple match list. If one entry match, the rule is OK. * is any number of char ? is one char [is the escape character	Loop, optional	
RegExpMatch XxY =	Reg Exp match list. If one entry match, the rule is OK	Loop, optional	
LineModulo X =	The line number in the editor will be check against the modulo of this value. If <i>LineModuloValX</i> , the rule is OK (look line color alternation example below)	Single, optional	1 (will be ok for each line)
LineModuloVal X = [new v1.8.3]	Rule is OK if : (Line Number MOD LineModuloX) = LineModuloValX	Single, optional	0 (will be ok for each line)

TextColorX=	Color of the text if the rule matches. If not present, the default text color is used (can be in hex or decimal)	Single, optional	
BackColorX=	Color of the background if the rule matches. If not present, the background color is the default background color (can be in hex or decimal)	Single, optional	

Look bellow for some examples:

[Sample1]

```
Hint='Simple Sample showing color coding'
WordSeparator='.;,:!.''^+*/\ `[]()e{}?|%=<>'
```

```
RuleName0=Separator in Red
RuleMode0=match
AllowedChar0='.;,:!.''^+*/\ `[]()e{}?|%=<>'
TextColor0=255
```

```
RuleName1=Digit in blue only in odd line
RuleModel=match
LineModulo=2
AllowedChar1='0123456789'
TextColor1=$FF0000
```

```
RuleName2=Key word in Green
RuleMode2=match
SimpleMatch2x0='mail*'
SimpleMatch2x1='internet*'
TextColor2=$8000
BackColor2=$FFFFFF
```

[Sample2]

```
Hint='Bracket highlightment and hours'
WordSeparator=' '
LittDelStart0=[
LittDelEnd0=]
```

```
RuleName0=Hour
RuleMode0=match
AllowedChar0='0123456789:'
SimpleMatch0x0='?:??'
SimpleMatch0x1='?:?:??'
TextColor0=$F0
```

```
RuleName1=Bracket
RuleModel=match
SimpleMatch1x0='[[]*]'
TextColor1=$FFFFFF
BackColor1=$FF0000
```

[Sample3]

```
Hint='Alternate each line color'
```

```
RuleName0=Line alternation 1
RuleScope0=line
LineModulo0=2
BackColor0=$CEF6CE
```

```
RuleName1=Line alternation 1
RuleScope1=line
LineModulo1=2
LineModuloVal1=1
```

```
BackColor1=$CEF6EC
```

Remarks:

You can easily color each line containing a specific keyword (look example below that will mark the line in red when the line contains **alert**, **warning** or **opp**).

```
[Sample4]
Hint='Color coding'

RuleName0=Line in green
RuleScope0=line
RuleMode0=match
SimpleMatch0x0='*warning*'
SimpleMatch0x1='*alert*'
SimpleMatch0x2='*opp*'
TextColor0=$FFFFFF
BackColor0=$80

RuleName1=Line alternation 1
RuleScope1=line
LineModulo1=2
BackColor1=$CEF6CE

RuleName2=Line alternation 1
RuleScope2=line
LineModulo2=1
BackColor2=$CEF6EC
```

You can mix **line** and **word** scope, modulo and regexp to do really complex color coding scheme!!!!

Command line parameters

mTAIL can be started with the command line or the shell.

The syntax is: mTAIL.exe filename [/start] [/minimized] [/recycle] [/f=xxxx]
[/a=yyyy]

[] indicates optional parameters

- Filename is the path name to tail. If Filename = # , the last tailed file.
- If /start is specified, mTAIL will begin to tail at launch
- If /minimized, mTAIL windows is minimized to traybar
- *[new v1.5]* If /recycle, mTAIL will launch a new tail windows except if an existing Windows with the same command line ID already exists. This is useful for having some mTAIL shortcut on the desktop : launching several time the same shortcut will NOT spawn several mTAIL instance, but recycle the same !
- Use /f= to set mTail filter string to xxxx
- Use /a= to set mTail alarm string to yyyy
- Use /ini= to force mTail ini file *[new v1.7]*
- Use /ccini= (or /inicc=) to force mTail Color Coding ini file *[new v1.8.2]*

Example:

```
mTAIL "c:\*.log" /start
mTAIL "c:\*.log" /start /recycle
mtail #
```

Remarks:

mTAIL tries to find ini files using the following priority :

1. Forced path by command line
2. Same path as mTAIL.exe (new v1.8.2)
3. Windows Application Data/mTail

Note:

[new in v1.5] : the Filename can be in any position in the command line.

If your path included space, use double quote only on the path, not on the whole option:

```
mTAIL "c:\*.log" "/ini=c:\my ini.ini" ← wrong
mTAIL "c:\*.log" /ini="c:\my ini.ini" ← good
```

End.