

Kiwi Log Viewer

A Freeware Log Viewer for Windows

by SolarWinds, Inc.

Kiwi Log Viewer displays text based log files in a tabular format. Only a small section of the file is read from disk at a time which saves memory and allows you to view a file that would be too big to fit in memory.

The tail option monitors the specified log file for changes and displays any new data that is added in real time.

Text string matching allows important text to be highlighted by using colors or bold/italic font emphasis. Text matching options include simple sub-string matching or more powerful PERL style Regular Expression matching.

Table of Contents

Foreword	0
Part I Introduction	2
Part II Features in the free version	2
Part III Features in the licensed version	3
Part IV Command line arguments	3
Part V Setup options	5
Part VI Find function	6
Part VII Highlighting	7
1 Sharing Highlights with other users	9
Part VIII Filtering	10
Part IX Tailing	11
Part X Regular Expression Syntax	11
1 RegExp examples	14
Part XI Feedback - Comments or Bugs	14
Part XII Other settings	14
Part XIII Automating the install on Windows	14
Index	16

1 Introduction



A Freeware Log Viewer for Windows

Program copyright 2005-2009 by SolarWinds, Inc.

Latest version available from: www.kiwisyslog.com

Support: www.kiwisyslog.com/support

Community: www.thwack.com

Kiwi Log Viewer displays text based log files in a tabular format. Only a small section of the file is read from disk at a time which saves memory and allows you to view a file that would be too big to fit in memory.

The tail option monitors the specified log file for changes and displays any new data that is added in real time.

Text string matching allows important text to be highlighted by using colors or bold/italic font emphasis. Text matching options include simple sub-string matching or more powerful PERL style Regular Expression matching.

2 Features in the free version

The free version of Kiwi Log Viewer includes the following features:

- ✓ Open recent files (2 files)
- ✓ Color highlighting (up to 2 text matches)
- ✓ Saving of favorite highlights
- ✓ Importing/Exporting of highlights
- ✓ Runs on Windows 2000 onwards
- ✓ Open files up to 700MB in size
- ✓ Returns up to 100 matches of filtered results
- ✓ Monitor log files for new data and display in real time
- ✓ Detailed help file
- ✓ Copy selected rows to the clipboard in tab delimited format
- ✓ Search for a text string within the file
- ✓ Freeware

More features are available in the licensed version.

To purchase a license, please visit: www.kiwisyslog.com/purchase

3 Features in the licensed version

The licensed version of Kiwi Log Viewer includes the following features:

All the features in the free version, plus...

- ✓ Up to 10 entries in the Recent file list
- ✓ Up to 100 text matches per highlight
- ✓ Up to 100 favorite highlights
- ✓ Returns all filtered results
- ✓ Open files of any size*
- ✓ Drag & Drop file support
- ✓ Ability to click-select-copy any part of the row text

To purchase a license, please visit: www.kiwisyslog.com/purchase

* Files up to 14TB can be read (if the operating system supports them)

4 Command line arguments

Kiwi Log Viewer can be controlled from the command line with the following options:

The command line format:

```
KiwiLogViewer [-option ["Option Value"]]
```

Where -option can be:

-t

Enable file tailing mode (no filters applied). No parameters required.

-th

Enable filtered file tailing in Memory Mode showing only the rows that match the highlights. No parameters required.

-tf RegExp InvertMatch IgnoreCase "String to match"

Enable filtered file tailing in Memory Mode showing only the rows that match the specified string.

All 4 parameters are required.

RegExp can be 0 or 1

InvertMatch can be 0 or 1

IgnoreCase can be 0 or 1

String to match must be specified in double quotes

e.g. -tf 0 0 1 "some text to match"
e.g. -tf 1 0 0 "some regular expression"
e.g. -tf 0 1 1 "all rows NOT matching this text"

-f "Path\File name"

Open specified file. If the path or file name contains spaces, you will need to enclose the entire path and file name with quotes.

-d (decimal number between 0 and 255)

Specifies an alternate column delimiter character value. Normally, Kiwi Log Viewer splits each row of data into columns based on the tab delimiter (ASCII character 9). If your file uses something different to delimit each column, you can specify the ASCII character value to look for with the -d argument.

e.g. If your columns are delimited by pipe (|) characters, use -d 124

e.g. If your columns are delimited by null characters, use -d 0

e.g. If your columns are delimited by colon characters, use -d 58

-h "Name of highlight"

Load and use the specified favorite highlight items. The list of highlight favorite names can be found on the Highlights setup page (Options | Highlighting menu).

-p 1 to 6

Specifies the screen position and sizing.

1=Top third of the screen

2=Middle third of the screen

3=Bottom third of the screen

4=Top half of the screen

5=Bottom half of the screen

6=Fit to screen size (full screen)

Examples:

Just start the program normally:

```
KiwiLogViewer
```

Open a file, go into full screen mode and use the "PIX VPN logins" highlights favorite:

```
KiwiLogViewer -f "C:\Logs\CiscoPixLog.txt" -p 6 -h "PIX VPN logins"
```

Open a file, use the top third of the screen and use the "404 errors" highlights favorite:

```
KiwiLogViewer -f "C:\Logs\ApacheWebLog.txt" -p 1 -h "404 errors"
```

Open a file, use the top half of the screen and tail the file as it is updated:

```
KiwiLogViewer -f "C:\Logs\ActiveLog" -p 4 -t
```

Open and tail a file in memory mode while only showing highlighted rows:

```
KiwiLogViewer -th -f "C:\Logs\ApacheWebLog.txt"
```

Open and tail a file in memory mode while only showing rows that contain the words "link down":

```
KiwiLogViewer -tf 0 0 1 "link down" -f "C:\Logs\CiscoRouter.txt"
```

Open and tail a file in memory mode while only showing rows that match the regular expression:

```
KiwiLogViewer -tf 1 0 0 "(192\.168\.55\.1)\.*(10\.10\.67\.89)" -f  
"C:\Logs\CiscoRouter.txt"
```

Open a pipe delimited text file:

```
KiwiLogViewer -f "C:\Logs\PipeDelimitedFile.txt" -d 124
```

5 Setup options

The setup page allows you to modify the behavior of the program.

Display settings

Horizontal grid lines:

Shows light grey horizontal lines on the main grid display between each row.

Vertical grid lines:

Shows light grey vertical lines on the main grid display between each column.

Alternate background colors:

Alternates a light blue and white background colour for each row. This makes large sections of similar data easier to read. If you have highlights enabled, they will override the alternating background colour.

Row padding:

Increases the amount of white space between each row. Setting this option to +5 pixels will add an extra 5 pixels of white space per row. This can be used to make the text easier to read.

Column padding:

Increases the amount of white space shown after the text in each column. In most cases the columns are auto sized correctly and all the text can be seen. If you find your text is being truncated in each column, increase this value. This can be used to make the text easier to read.

Font name:

Allows you to choose which font is used to display the text in the main grid display. Tahoma, Verdana or Arial are easy to read. Choosing the System font will use the default font for your operating system.

Font size:

Allows you to choose the size of the font on the main grid display. Choosing a size of 0 will use the default font size for your operating system. It is recommended you use a font size of between 9 and 12 points for clear viewing.

Bold and Italic:

This sets the default text emphasis. If you have highlights enabled, they will override the bold and italic

settings for the default text.

Always on top of other windows:

This keeps the main window on top of all other windows and therefore visible at all times.

File Options

Read files less than [nnn] MB into memory:

This options causes the entire file contents to be read into memory when the file is opened. This speeds up the searching, filtering and exporting functions at the expense of a slower initial loading time. The size you choose will depend on your system memory, disk speed and processor speed. Somewhere between 50MB and 200MB will work well on most systems.

Size of history:

This setting only applies when tailing a log file with a filter applied. The last [nnn] rows can be kept in the display so you can scroll back and see historic matches.

Scan last [nn] MB of file for matches:

This setting only applies when tailing a log file with a filter applied. The last [nn] MB of the file are scanned for matches and added to the historic display. If this setting is not enabled, the display will initially be blank until new matching records arrive.

Automatically tail files when opened:

When this setting is checked, the file will automatically be tailed as soon as it is opened. You can stop the file tailing by pressing the tail button in the toolbar.

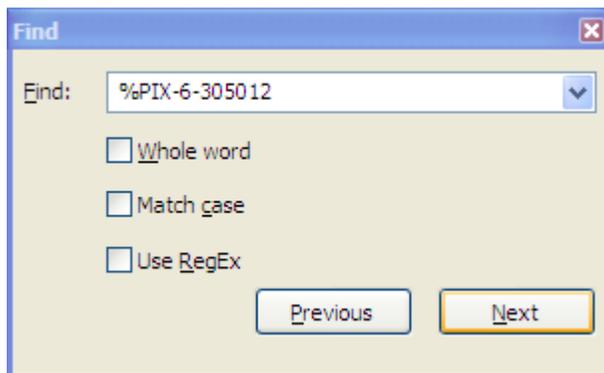
6 Find function

To find a keyword or text in the file, you can use the Find function.

You can show the Find window in the following ways...

- Click the Edit | Find | Find menu
- Click the Find icon in the toolbar which looks like this: 
- Press Ctrl-F

The Find window looks like this:



Type the text you want to locate in your file into the Find text box.

Check "Whole word" to match only against text that is a whole word.

For example, if you search for "cat", it will find a match against the words cat and catalogue. To match only against the word cat, check the "Whole word" option.

Check "Match case" if you only want to match text that has exactly the same case as your search term.

For example, if you search for "cat", it will find a match against the words cat and Cat. To match only against the word cat, check the "Match case" option.

Regular Expression (RegEx) pattern matching is a very flexible and powerful way of matching text. More information on [Regular Expression Syntax](#).

Press the Next or Previous buttons to search the file for the text you have entered. The search begins from the current place in the file. If you want to start the search from the top, press the Home key before you bring up the Find window.

If the text can't be found on the displayed page, the file is searched. A progress dialog will appear on the screen until a match is found.

As a general guide, the Find function will take about one minute to search through a 1GB file.

If a match is not found, a beep will be heard.

If a match is found the row on which the match occurs will be highlighted.

To move to the next match, press the F3 key.

To move to the previous match, press the F2 key.

To find the next highlighted line, press the F8 key

To find the previous highlighted line, press the F7 key

7 Highlighting

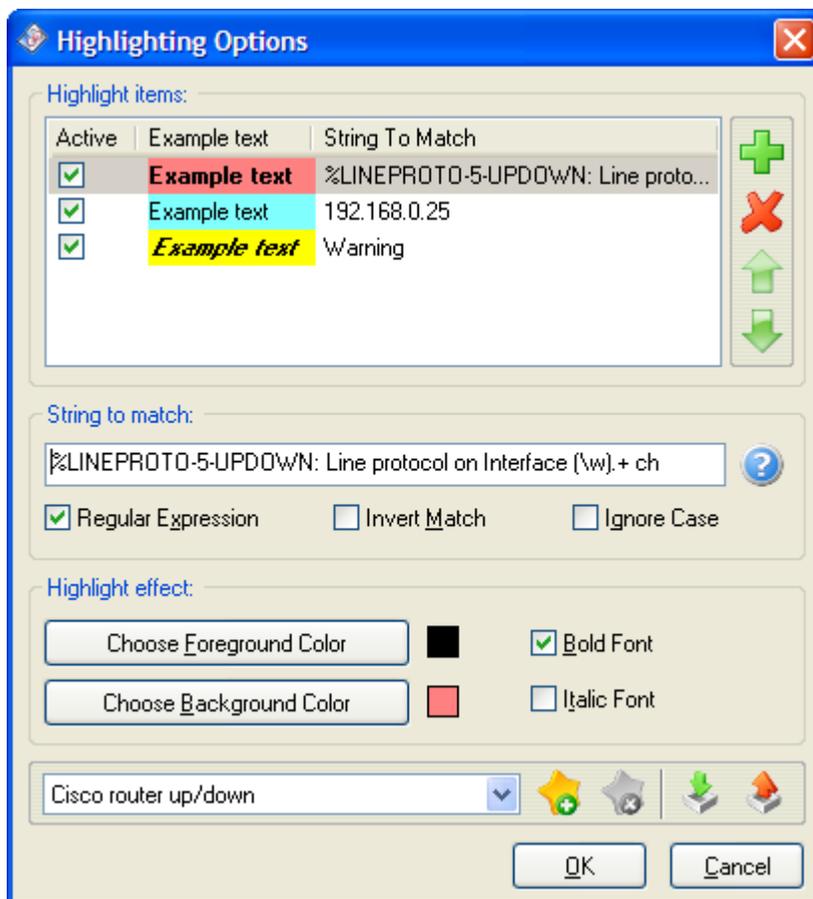
Note: The free version of Kiwi Log Viewer allows up to 2 highlights. The licensed version allows unlimited highlights.

The highlighting options available in Kiwi Log Viewer enable the user to specify a set of highlighting rules which will be applied to the text that is displayed in the main window. Highlighting rules are evaluated from the top-down, and any text which matches a given highlight rule will have the specified highlight-effects applied.

You can show the Highlighting window in the following ways...

- Click the Options | Highlighting menu
- Click the Highlighting icon in the toolbar which looks like this: 
- Press Ctrl-H

The Highlighting window looks like this:



Each rule can be activated/deactivated by checking/unchecking the check boxes on each row of the list.

Highlighting rules can be added/deleted by clicking the buttons on the toolbar to the right of the highlights list. Rule precedence can be changed in this toolbar as well, by clicking the up/down arrows.

String to match:

The string pattern that will be searched for in the row of displayed text.

Regular Expression:

If checked, this option specifies if the string to match is a regular expression. (see [Regular Expression Syntax](#))

You can press the RegExp syntax help button  to open the help file to the regular expression syntax topic.

Invert Match:

If checked, this option specifies that the effect will be applied only if a match is NOT found.

Ignore Case:

If checked, the search pattern (string to match) will be treated as case insensitive.

Highlight Effects:

Apply Foreground Color:

If checked, the foreground color selected will be applied, and override any existing foreground color. If unchecked, the current foreground color will be used.

Apply Background Color:

If checked, the background color selected will be applied, and override any existing background color. If unchecked, the current background color will be used.

Bold Font:

If checked, the font weight will be bold.

Italic Font:

If checked, the font style will be italicised.

Favorites:

If you are using different highlight settings for each file you are viewing, you may want to save your highlight settings to disk. The Favorites area allows you to save and load your created highlights by name for quick retrieval at a later time.

You can even import and export your highlights to a text file so you can share them with others.

It is also possible to specify which favorite highlights are loaded when the program starts by using [command line arguments](#).

Jumping to a highlighted line in the main window:

When you have enabled highlighting of matched rows, you can quickly jump to the next or previous highlight using the F8 and F7 keys.

To find the next highlighted line, press the F8 key

To find the previous highlighted line, press the F7 key

7.1 Sharing Highlights with other users

To deploy your highlights from a base system to others using Kiwi Log Viewer, you can use the following steps...

On your base system. Setup your desired highlights, then export them to a file using the Export button in the Highlights setup page. The export button is the one at the bottom right (Red up facing arrow). As an example, you can save your highlights to a file called "HiLite-CiscoPIX.ini"

Copy this file into the Kiwi Log Viewer installation folder on the deployed systems. Usually this is C:\Program Files\Kiwi Log Viewer. You can do this via a login script or any other method you use to deploy common files to remote systems.

On the deployed systems, setup a shortcut on the desktop to Kiwi Log Viewer. Then modify the command line as follows...

```
"C:\Program Files\Kiwi Log Viewer\KiwiLogViewer.exe" -h "HiLite-CiscoPIX"
```

The -h switch allows you to specify the highlight settings to load at startup.

Then when the user clicks on the shortcut, the highlights will be read in and used on the files being viewed.

this allows you to create highlights centrally and roll them out to other users of Kiwi Log Viewer.

8 Filtering

Kiwi Log Viewer can operate on files in two modes. Memory mode and Chunk mode.

Memory mode:

The entire file is read into memory. The setup options allow you to choose the maximum file size you want to load into memory at once. If the file you want to open exceeds this size, the program will default to Chunk mode. In Memory mode, the filtering, searching and exporting is all carried out against the data in memory and the original file is not accessed again.

Chunk mode:

Only a small chunk of the file is read at a time and displayed on the grid. This means you can view log files that are too big to fit into memory. The disadvantage is that filtered exports and searching are slower as the entire file must be re-scanned each time.

When operating in Chunk mode, the Filter button allows the following options:

(o) Include only the highlighted rows:

This will include the log entries that match your Highlights list. This is useful when you only want to see the entries that are highlighted and exclude all the clutter.

(o) String to match:

Here you can match against any string or regular expression.

In Chunk mode, the only output option available is to Create a new file. The filter will scan the original file and write any matches found to the new file you specify. Once the entire file has been scanned, the newly created results file will be opened. You can then repeat the process again to further narrow down the results. This is useful when you have very large files that you want to reduce down to smaller subsets of filtered data.

When operating in Memory mode, the Filter button allows the following options:

(o) No filter:

This will remove any existing filter applied to the grid display and show all of the entries in the log file.

(o) Show only the highlighted rows:

This will include the log entries that match your Highlights list. This is useful when you only want to see the entries that are highlighted and exclude all the clutter.

(o) String to match:

Here you can match against any string or regular expression.

In Memory mode, you can choose to apply the filter to the current data in memory or output the results to a new file.

When **creating a new file**, the filter will scan the original file and write any matches found to the new file you specify. Once the entire file has been scanned, the newly created results file will be opened. You can then repeat the process again to further narrow down the results. This is useful when you have very large files that you want to reduce down to smaller subsets of filtered data.

Pressing the **Apply button** will scan the memory for all matches that meet your filter and display them in the grid. An indication of how many matches are shown in the main toolbar area. The freeware version is limited to showing only the first 100 matches. The licensed version will show all the matches. Pressing the filter button again will allow you to choose a different filtered view of the original data. This is useful for quickly trying different filters against the same data to get the results you want.

9 Tailing

File tailing is useful for monitoring new entries to log files in real time.

Various tailing options are available:

(o) No filter

This will open the file in [chunk mode](#) and display new entries as they arrive. Moving the vertical scroll bar will automatically cancel the tailing mode and allow you to view any part of the log file. Pressing the F6 key will allow you to quickly jump back into this tailing mode again.

(o) Show only the Highlighted rows

Operates in [Memory mode](#). Depending on your [setup options](#), the program will scan the last [nn] MB of the file for highlight matches and store them in the tailing buffer. Any new log entries that match the highlights will be added to the display.

(o) String to match

Operates in [Memory mode](#). Depending on your [setup options](#), the program will scan the last [nn] MB of the file for filter matches and store them in the tailing buffer. Any new log entries that match your filter text will be added to the display.

10 Regular Expression Syntax

Kiwi Log Viewer supports the use of Perl Compatible Regular Expression (PCRE) matching.

More information on PCRE syntax can be found at: <http://www.pcre.org>

The following pattern syntax can be used:

Pattern	Description
.	Matches any character except new line.
[a-z0-9]	Matches any single character of set.
[^a-z0-9]	Matches any single character not in set.
\d	Matches a digit. Same as [0-9].
\D	Matches a non-digit. Same as [^0-9].
\w	Matches an alphanumeric (word) character -- [a-zA-Z0-9_].
\W	Matches a non-word character [^a-zA-Z0-9_].
\s	Matches a white space character (space, tab, new line, etc.).
\S	Matches a non-white space character.
\n	Matches a new line (line feed).
\r	Matches a return.
\t	Matches a tab.
\f	Matches a form feed.
\b	Matches a backspace.
\0	Matches a null character.
\000	Also matches a null character because of the following:
\nnn	Matches an ASCII character of that octal value.
\xnn	Matches an ASCII character of that hexadecimal value.
\cX	Matches an ASCII control character.

<code>\metachar</code>	Matches the meta-character (e.g., <code>\</code> , <code>.</code>)
<code>(abc)</code>	Used to create sub expressions. Remembers the match for later back references.
<code>\1, \2, ...</code>	Referred by replacement patterns that use <code>\1</code> , <code>\2</code> , etc. Matches whatever first (second, and so on) of parens matched.
<code>x?</code>	Matches 0 or 1 x's, where x is any of above.
<code>x*</code>	Matches 0 or more x's.
<code>x+</code>	Matches 1 or more x's.
<code>x{m,n}</code>	Matches at least m x's, but no more than n.
<code>abc</code>	Matches all of a, b, and c in order.
<code>\b</code>	Matches a word boundary (outside <code>[]</code> only).
<code>\B</code>	Matches a non-word boundary.
<code>^</code>	Anchors match to the beginning of a line or string.
<code>\$</code>	Anchors match to the end of a line or string.

Wildcards

Some special characters are used to match a class of characters:

Wildcard Matches

`.` Any single character except a line break, including a space.

The following wildcards match by position in a line:

Wildcard Matches Example

<code>^</code>	Beginning of a line (unless used in a character class; see below)
<code>^Phone:</code>	Finds lines that begin with "Phone:"
<code>\$</code>	End of a line (unless used in a character class)
<code>character in</code>	the current line. <code>\$:</code> Finds the last character in

Character Classes

A character class allows you to specify a set or range of characters. You can choose to either match or ignore the character class. The set of characters is enclosed in brackets. If you want to ignore the character class instead of match it, precede it by a caret (`^`). Here are some examples:

Character Class Matches

<code>[aeiou]</code>	Any one of the characters a, e, i, o, u.
<code>[^aeiou]</code>	Any character except a, e, i, o, u.
<code>[a-e]</code>	Any character in the range a-e, inclusive
<code>[a-zA-Z0-9]</code>	Any alphanumeric character.
<code>[[]</code>	Finds a <code>[</code> .
<code>[]]</code>	Finds a <code>]</code> . To find a closing bracket, place it immediately after the opening bracket.
<code>[a-e^]</code>	Finds a character in the range a-e or the caret character. To find the caret character after the opening character, place it anywhere except as the first character after the opening bracket.
<code>[a-c-]</code>	Finds a character in the range a-c or the <code>-</code> sign. To match a <code>-</code> , place it at the beginning or end of the set.

Non-printing Characters

You can use the following notation to find non-printing characters:

Special Character Matches

<code>\r</code>	Line break (return)
<code>\n</code>	New line (line feed)

```

\t          Tab
\f          Form feed (page break)
\xNN       Hex code NN.

```

Other Special Characters

The following patterns are wildcards for the following special characters:

Special Character Matches

```

\s         Any white space character (space, tab, return, linefeed, form feed)
\S         Any non-white space character.
\w         Any "word" character (a-z, A-Z, 0-9, and _)
\W         Any "non-word" character (All characters not included by \w).
\d         Any digit [0-9].
\D         Any non-digit character.

```

Repetition Characters

Repetition characters are modifiers that allow you to repeat a specified pattern.

Repetition Character Matches Examples

```

*         Zero or more characters.  d* finds no characters, or one or more consecutive
"d"s. .*  finds an entire line of text, up to but not including the return
character.
+         One or more characters.  d+ finds one or more consecutive "d"s. [0-9]+
finds a string of one or more consecutive numbers, such as "90404", "1938", the
"32" in "Win32", etc.
?         Zero or one characters.  d? finds no characters or one "d".

```

Please note that, since * and ? match zero instances of the pattern, they always succeed but may not select any text. You can use them to specify an optional character, as in the examples in the following section.

Extension Mechanism

```

(?#text)  Comment
(?:pattern) For grouping without creating back references
(?=pattern) A zero-width positive look-ahead assertion. For example, \w+(?= )
matches a word followed by a tab, without including the tab in $&.
(?!pattern) A zero-width negative look-ahead assertion. For example
foo(?!bar)/matches any occurrence of "foo" that isn't followed by "bar".

(?<=pattern) A zero-width positive look-behind assertion. For example, (?<= )\w+
matches a word that follows a tab, without including the tab in $&.
Works only for fixed-width look-behind.
(?<!pattern) A zero-width negative look-behind assertion. For example (?<!bar)foo
matches any occurrence of "foo" that does not follow "bar". Works only for
fixed-width look-behind.

```

The Alternation Operator

The alternation operator of pipe (|) allows you to match any of a number of patterns using the logical "or" operator. Place it between two existing patterns to match either pattern. You can use more than one alternation operator in a pattern. eg. cat|dog|fish will match cat or dog or fish.

10.1 RegExp examples

Here are some examples of regular expressions you can use to match text...

To match Address1 OR Address2, use: `(126\.34\.55\.1)|(123\.45\.67\.89)`

To match Address1 AND Address2, use: `(126\.34\.55\.1).* (123\.45\.67\.89)`

11 Feedback - Comments or Bugs

If you have any comments about this program or improvements you would like to see in the next version please feel free to contact us via our feedback page at:

<http://www.kiwisyslog.com/support>

12 Other settings

Some of the more advanced program options can be set via the ini file rather than from the setup interface.

The Kiwi Log Viewer settings file can be located at:

`C:\Documents and Settings\Username\Application Data\KiwiLogFileViewer.ini`

GridRowAltColor=232,235,255

Sets the color of the alternating grid rows. Requires 3 RGB colors to be provided separated by commas.

MaxCharactersPerCell=4096

Sets the maximum number of characters shown per grid cell. Making this number smaller can reduce the amount of text shown in the grid. The purpose of this setting is mainly to avoid the issue of invalid file data crashing the program.

FileColumnDelimiter=9

Specifies an alternate column delimiter character value. Normally, Kiwi Log Viewer splits each row of data into columns based on the tab delimiter (ASCII character 9). If your file uses something different to delimit each column, you can specify the ASCII character value to use here. Any decimal number between 0 and 255 can be specified. A value of 9 is the default.

If your columns are delimited by pipe (|) characters, use a value of 124

If your columns are delimited by null characters, use a value of 0

If your columns are delimited by colon characters, use a value of 58

13 Automating the install on Windows

It is possible to automate the installation of Kiwi Log Viewer on Windows systems by using the /S

(silent) command line switch.

To silently install Kiwi Log Viewer into the default install folder use the following command line...

```
Kiwi_LogViewer_X.X.X_Win32.setup.exe /S
```

This will install Kiwi Log Viewer into the "C:\Program Files\Kiwi Log Viewer" folder.

To silently install Kiwi Log Viewer into a different folder use the following command line...

```
Kiwi_LogViewer_X.X.X_Win32.setup.exe /S /D=C:\Program Files\YourFolder
```

This will install Kiwi Log Viewer into the "C:\Program Files\YourFolder" folder.

As with a regular install, all the Start Menu, Desktop shortcuts and help files are also installed as well.

Normally, the installer will add start menu and desktop shortcuts for the current user. To have the start menu and desktop shortcuts applied to all users on the system, use the /ALLUSERS switch.

```
e.g: Kiwi_LogViewer_X.X.X_Win32.setup.exe /ALLUSERS /S
```

Note:

The license information is stored in a central folder that is available to all users of the Windows system.
C:\Documents and Settings\All Users\Application Data\KiwiLogFileViewer.ini

If you have a site license and want to install the licensed product on many machines, you can copy this license file onto each machine.

All other settings are stored in a location specific to the current user. See [Other settings](#) for more information.

Index

- C -

Command line arguments 3

- F -

Features in the free version 2

Features in the licensed version 3

Feedback - Comments or Bugs 14

- I -

Introduction 2

- R -

Regular Expression Syntax 11