

# Lightgrep Cheat Sheet

## 1 Single Characters

<code>c</code>	the character <code>c</code> *
<code>\a</code>	U+0007 (BEL) bell
<code>\e</code>	U+001B (ESC) escape
<code>\f</code>	U+000C (FF) form feed
<code>\n</code>	U+000A (NL) newline
<code>\r</code>	U+000D (CR) carriage return
<code>\t</code>	U+0009 (TAB) horizontal tab
<code>\ooo</code>	U+ <i>ooo</i> , 1-3 octal digits <i>o</i> , ≤ 0377
<code>\xhh</code>	U+00 <i>hh</i> , 2 hexadecimal digits <i>h</i>
<code>\x{hhhhhh}</code>	U+ <i>hhhhhh</i> , 1-6 hex digits <i>h</i>
<code>\zhh</code>	the byte 0 <i>zhh</i> (not the character!)†
<code>\N{name}</code>	the character called <i>name</i>
<code>\N{U+hhhhhh}</code>	same as <code>\x{hhhhhh}</code>
<code>\c</code>	the character <code>c</code> ‡

\*except U+0000 (NUL) and metacharacters

†Lightgrep extension; not part of PCRE.

‡except any of: adefnprstwdPSW1234567890

## 2 Named Character Classes

<code>.</code>	any character
<code>[0-9]</code>	(= ASCII digits)
<code>\D</code>	[^0-9]
<code>\s</code>	[\t\n\f\r ] (= ASCII whitespace)
<code>\S</code>	[^\t\n\f\r ]
<code>\w</code>	[0-9A-Za-z_] (= ASCII words)
<code>\W</code>	[^0-9A-Za-z_]
<code>\p{property}</code>	any character having <i>property</i>
<code>\P{property}</code>	any character lacking <i>property</i>

## 3 Character Classes

<code>[stuff]</code>	any character in <i>stuff</i>
<code>[^stuff]</code>	any character <b>not</b> in <i>stuff</i>
where <i>stuff</i> is...	
<code>c</code>	a character
<code>a-b</code>	a character range, inclusive
<code>\zhh</code>	a byte
<code>\zhh-\zhh</code>	a byte range, inclusive
<code>[S]</code>	a character class
<code>ST</code>	$S \cup T$ (union)
<code>S&amp;T</code>	$S \cap T$ (intersection)
<code>S-T</code>	$S - T$ (difference)
<code>S~T</code>	$S \Delta T$ (symmetric difference, XOR)

## 8 EnCase GREP Syntax

<code>c</code>	the character <i>c</i> (except metacharacters)
<code>\xhh</code>	U+00 <i>hh</i> , 2 hexadecimal digits <i>h</i>
<code>\whhhh</code>	U+ <i>hhhhh</i> , 4 hexadecimal digits <i>h</i>
<code>\c</code>	the character <i>c</i>
<code>.</code>	any character
<code>#</code>	[0-9] (= ASCII digits)
<code>[a-b]</code>	any character in the range <i>a-b</i>
<code>[S]</code>	any character in <i>S</i>
<code>[^S]</code>	any character not in <i>S</i>
<code>(S)</code>	grouping
<code>S*</code>	repeat <i>S</i> 0 or more times (max 255)
<code>S+</code>	repeat <i>S</i> 1 or more times (max 255)
<code>S?</code>	repeat <i>S</i> 0 or 1 or time
<code>S{n,m}</code>	repeat <i>S</i> <i>n-m</i> times (max 255)
<code>ST</code>	matches <i>S</i> , then matches <i>T</i>
<code>S T</code>	matches <i>S</i> or <i>T</i>

## 9 Importing from EnCase into Lightgrep

<code>\whhhh</code>	→ <code>\xhhhh</code>	<i>S*</i> and <i>S+</i> are limited to 255 repetitions by EnCase;
<code>#</code>	→ <code>\d</code>	Lightgrep preserves this in imported patterns.
<code>S*</code>	→ <code>S{0,255}</code>	
<code>S+</code>	→ <code>S{1,255}</code>	
<code>\w</code> is limited to BMP characters (≤ U+10000) only.		

Some people, when confronted with a problem, think "I know, I'll use regular expressions." Now they have two problems.

—JWZ in alt.religion.emacs, 12 August 1997

## Notes & Examples

Characters:

`.*?\x00` (= null-terminated string)  
`\z50\z4B\z03\z04` (= ZIP signature)  
`\N{EURO SIGN}, \N{NO-BREAK SPACE}`  
`\x{042F}` (= CYRILLIC CAPITAL LETTER YA)  
`\+12\.%` (= escaping metacharacters)

Grouping: Operators bind tightly. Use `(aa)+`, not `aa+`, to match pairs of `a`'s.

Ordered alternation: `a|ab` matches `a` twice in `aab`. Left alternatives preferred to right.

Repetition: Greedy operators match as much as possible. Reluctant operators match as little as possible. `a+a` matches all of `aaaa`; `a?a` matches the first `aa`, then the second `aa`.

`+` will (uselessly) match the **entire** input. Prefer reluctant operators when possible.

Character classes:

`[abc]` = `a`, `b`, or `c`  
`[^a]` = anything but `a`  
`[A-Z]` = `A` to `Z`  
`[A-Z]`  
= `A`, `Z`, or hyphen (!)  
`[A-Zaeiou]` = capitals or lowercase vowels  
`[.+*?\\]`  
= `.`, `+`, `*`, `?`, or `\`  
`[Q\z00-\z7F]`  
= `Q` or 7-bit bytes  
`[[abcd][bce]]`  
= `a`, `b`, `c`, `d`, or `e`  
`[[abcd]&&[bce]]`  
= `b` or `c`  
`[[abcd]--[bce]]`  
= `a` or `d`  
`[[abcd]~~[bce]]`  
= `a`, `d`, or `e`  
`\p{Greek}\d`  
= Greek or digits  
`[^\\p{Greek}7]`  
= neither Greek nor 7  
`\p{Greek}&&\p{Ll}`  
= lowercase Greek

Operators need not be escaped inside character classes.

Lightgrep Search for EnCase®

Fast Search for Forensics

[www.lightgrep.com](http://www.lightgrep.com)

## 4 Grouping

`(S)` makes any pattern *S* atomic

## 5 Concatenation & Alternation

`ST` matches *S*, then matches *T*

`S|T` matches *S* or *T*, preferring *S*

## 6 Repetition

Repeats *S*...

Greedy	<i>S</i> *	0 or more times (= $S\{0,\}$ )
	<i>S</i> +	1 or more times (= $S\{1,\}$ )
	<i>S</i> ?	0 or 1 time (= $S\{0,1\}$ )
	<i>S</i> { <i>n</i> ,}	<i>n</i> or more times
	<i>S</i> { <i>n</i> , <i>m</i> }	<i>n-m</i> times, inclusive
Reluctant	<i>S</i> *?	0 or more times (= $S\{0,\}$ )
	<i>S</i> +?	1 or more times (= $S\{1,\}$ )
	<i>S</i> ??	0 or 1 time (= $S\{0,1\}$ )
	<i>S</i> { <i>n</i> ,}?	<i>n</i> or more times
	<i>S</i> { <i>n</i> , <i>m</i> }?	<i>n-m</i> times, inclusive

## 7 Selected Unicode Properties

Any	Assigned
Alphabetic	White_Space
Uppercase	Lowercase
ASCII	Noncharacter_Code_Point
Name=name	Default_Ignorable_Code_Point
General_Category=category	
L, Letter	P, Punctuation
Lu, Uppercase Letter	Pc, Connector Punctuation
Ll, Lowercase Letter	Pd, Dash Punctuation
Lt, Titlecase Letter	Ps, Open Punctuation
Lm, Modifier Letter	Pe, Close Punctuation
Lo, Other Letter	Pi, Initial Punctuation
M, Mark	Pf, Final Punctuation
Mn, Non-Spacing Mark	Po, Other Punctuation
Me, Enclosing Mark	Z, Separator
N, Number	Zs, Space Separator
Nd, Decimal Digit Number	Zl, Line Separator
Nl, Letter Number	Zp, Paragraph Separator
No, Other Number	C, Other
S, Symbol	Cc, Control
Sm, Math Symbol	Cf, Format
Sc, Currency Symbol	Cs, Surrogate
Sk, Modifier Symbol	Co, Private Use
So, Other Symbol	Cn, Not Assigned

Script=script

Common Latin Greek Cyrillic Armenian Hebrew Arabic Syraic Thaana Devanagari Bengali Gurmukhi Gujarati Oriya Tamil Telugu Kannada Malayalam Sinhala Thai Lao Tibetan Myanmar Georgian Hangul Ethiopic Cherokee Ogham Runic Khmer Mongolian Hiragana Katakana Bopomofo Han Yi Old\_Italic Gothic Inherited Tagalog Hanunoo Buhid Tagbanwa Limbu Tai\_Le Linear\_B Ugaritic Shavian Osmanya Cypriot Buginese Coptic New\_Tai\_Lue Glagolitic Tifinagh Syloti\_Nagri Old\_Persian Kharoshthi Balinese Cuneiform Phoenician Phags\_Pa Nko Sudanese Lepcha ...  
See Unicode Standard for more.

Email addresses: `[a-z\d!#$%&'*/+=?^_'\{\}\~\-\ ]{0,63}`  
@`[a-z\d\-\ ]{1,253}`.\code[a-z\d\-\ ]{2,22}  
Hostnames: `([a-z\d]([a-z\d\-\ ]{0,61}[a-z\d])?\.)\{2,5\}[a-z\d]([a-z\d\-\ ]{1,22})`  
N. American phone numbers: `\(?\d{3}[ ]\.\.\{0,2\}\d{3}[ ]\.\{0,2\}\d{4}\d`  
Visa, MasterCard: `\d{4}([ ]-\d{4})\{3\}`  
American Express: `3[47]\d{2}([ ]-\d{6}[ ]-\d{5})`  
Diners Club: `3[08]\d{2}([ ]-\d{6}[ ]-\d{4})`  
EMF header: `\z01\z00\z00\z00.\{36\}\z20EMF`  
JPEG: `\zFF\zD8\zFF[\zC4\zDB\zE0-\zEF\zFE]` Footer: `\zFF\zD9`  
GIF: `GIF8[79]` Footer: `\z00\z3B` BMP: `BM.\{4\}\z00\z00\z00\z00.\{4\}\z28`  
PNG: `\z89\z50\z4E\z47` Footer: `\z49\z45\z4E\z44\zAE\z42\z60\z82`  
ZIP: `\z50\z4B\z03\z04` Footer: `\z50\z4B\z05\z06`  
RAR: `\z52\z61\z72\z21\z1a\z07\z00...\{ \z00-\z7F`  
Footer: `\z88\zC4\z3D\z7B\z00\z40\z07\z00`  
GZIP: `\z1F\z8B\z08` MS Office 97-03: `\zD0\zCF\z11\zE0\zA1\zB1\z1A\zE1`  
LNK: `\z4C\z00\z00\z00\z01\z14\z02\z00`  
PDF: `\z25\z50\z44\z46\z2D\z31` Footer: `\z25\z45\z4F\z46`

# Lightgrep™ Search

## Fast Search for Forensics

- Fast searching saves hours
- Full grep syntax
- Thousands of keywords
- Microsoft Office®, Adobe PDF®, and more
- Simultaneous file carving
- Intuitive reports
- Advanced Unicode support
- EnCase® integration

[www.lightgrep.com](http://www.lightgrep.com)

Copyright © 2013, Lightbox Technologies, Inc

Office® is a registered trademark of Microsoft | PDF® is a registered trademark of Adobe Systems Inc | EnCase® is a registered trademark of Guidance Software