apachelog.pql
by *Pequel*

sample@youraddress.com

Apache CLF Log Input Example Script

2.2

Table of Contents
Apache CLF Log Input Example Script

SCRIPT NAME

apachelog.pql

DESCRIPTION

Demonstrates reading Apache CLF Log file — split record on space delimiter parse qouted fields and square bracketed fields. This is done by 1) specifying a space delimiter for the 'input_delimiter' and 2) specifying a double qoute (must be escaped) characted and a open square bracket character for the 'input_delimiter_extra' option. This option specifies other characters that may delimit fields. Pequel will match open bracket character specfication with their respective closing bracket. Requires Inline::C and a C compiler to be installed because the 'input_delimiter_extra' option will instruct Pequel to generate C code.

1. PROCESS DETAILS

Input records are read from standard input. The input record contains ***6*** fields. Fields are delimited by the ' ' character.

Output records are written to standard output. The output record contains ***6*** fields. Fields are delimited by the '|' character.

1.1 *IP_ADDRESS*

Output Field

Description
Set to input field ***IP_ADDRESS***

1.2 *TIMESTAMP*

Output Field

Description
Set to input field ***TIMESTAMP***

1.3 *REQUEST*

Output Field

Description
Set to input field ***REQUEST***

1.4 *F4*

Output Field

Description
Set to input field ***F4***

1.5 *F5*

Output Field

Description
Set to input field ***F5***

1.6 *F6*

Output Field

Description

Set to input field ***F6***

2. CONFIGURATION SETTINGS

2.1 *pequeldoc*

generate pod / pdf pequel script Reference Guide.: pdf

2.2 *detail*

Include Pequel Generated Program chapter in Pequeldoc: 1

2.3 *script_name*

script filename: apachelog.pql

2.4 *header*

write header record to output.: 1

2.5 *optimize*

optimize generated code.: 1

2.6 *nulls*

print zero for null numeric/decimal.: 1

2.7 *doc_title*

document title.: Apache CLF Log Input Example Script

2.8 *doc_email*

document email entry.: sample@youraddress.com

2.9 *doc_version*

document version for pequel script.: 2.2

2.10 *input_delimiter*

input file field delimiter:

2.11 *input_delimiter_extra*

Extra input field delimiter(s): "[

2.12 *inline_cc*

Inline: CC: CC

2.13 *inline_optimize*

Inline: OPTIMIZE: -xO5 -xinline=%auto

2.14 *inline_ccflags*

Inline: CCFLAGS: -xchip=ultra3 -DSS_64BIT_SERVER -DBIT64 -DMACHINE64

3. TABLES

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name

5. APACHELOG.PQL

options

```
pequeldoc(pdf)
detail(1)
script_name(apachelog.pql)
header(1)
optimize(1)
nulls(1)
doc_title(Apache CLF Log Input Example Script)
doc_email(sample@youraddress.com)
doc_version(2.2)
input_delimiter( )
input_delimiter_extra("[)
inline_cc(CC)
inline_optimize(-xO5 -xinline=%auto)
inline_ccflags(-xchip=ultra3 -DSS_64BIT_SERVER -DBIT64 -DMACHINE64)
```

description

```
Demonstrates reading Apache CLF Log file -- split record on space delimiter
parse qouted fields and square bracketed fields. This is done by 1) specifying
a space delimiter for the ’input_delimiter’ and 2) specifying a double qoute
(must be escaped) characted and a open square bracket character for the
’input_delimiter_extra’ option. This option specifies other characters that
may delimit fields.  Pequel will match open bracket character specfication with
their respective closing bracket.
Requires Inline::C and a C compiler to be installed because the
’input_delimiter_extra’ option will instruct Pequel to generate C code.
```

input section

```
IP_ADDRESS
TIMESTAMP
REQUEST
F4
F5
F6
```

output section

```
string    IP_ADDRESS IP_ADDRESS
string    TIMESTAMP  TIMESTAMP
string    REQUEST    REQUEST
string    F4         F4
string    F5         F5
string    F6         F6
```

6. PEQUEL GENERATED PROGRAM

```
# vim: syntax=perl ts=4 sw=4
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#Generated By: pequel Version 2.2-9, Build: Tuesday September 13 08:43:08 BST 2005
#            : https://sourceforge.net/projects/pequel/
#Script Name : apachelog.pql
#Created On  : Tue Sep 13 10:17:35 2005
#For         :
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#Options:
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#script_name(apachelog.pql) script filename
#header(1) write header record to output.
#optimize(1) optimize generated code.
#nulls(1) print zero for null numeric/decimal.
#doc_title(Apache CLF Log Input Example Script) document title.
#doc_email(sample@youraddress.com) document email entry.
#doc_version(2.2) document version for pequel script.
#input_delimiter( ) input file field delimiter
#input_delimiter_extra("[) Extra input field delimiter(s)
#inline_cc(CC) Inline: CC
#inline_optimize(-xO5 -xinline=%auto) Inline: OPTIMIZE
#inline_ccflags(-xchip=ultra3 -DSS_64BIT_SERVER -DBIT64 -DMACHINE64) Inline: CCFLAGS
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
use strict;
local $\="\n"; local $,="|";
print STDERR '[apachelog.pql ' . localtime() . "] Init";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 5;
my @I_VAL;
my @O_VAL;
foreach my $f (1..6) { $O_VAL[$f] = undef; }
use constant _I_IP_ADDRESS    => int    0;
use constant _I_TIMESTAMP     => int    1;
use constant _I_REQUEST       => int    2;
use constant _I_F4            => int    3;
use constant _I_F5            => int    4;
use constant _I_F6            => int    5;
use constant _O_IP_ADDRESS    => int    1;
use constant _O_TIMESTAMP     => int    2;
use constant _O_REQUEST       => int    3;
use constant _O_F4            => int    4;
use constant _O_F5            => int    5;
use constant _O_F6            => int    6;
&PrintHeader();
print STDERR '[apachelog.pql ' . localtime() . "] Start";
use Benchmark;
my $benchmark_start = new Benchmark;
print STDERR '[apachelog.pql ' . localtime() . "] Tables opened.";
my $i;
while (readsplit(\@I_VAL))
{
    ++$i;
    print STDERR '[apachelog.pql ' . localtime() . "] $i records." if ($i % VERBOSE == 0);
    $O_VAL[_O_IP_ADDRESS] = $I_VAL[_I_IP_ADDRESS];
    $O_VAL[_O_TIMESTAMP] = $I_VAL[_I_TIMESTAMP];
    $O_VAL[_O_REQUEST] = $I_VAL[_I_REQUEST];
    $O_VAL[_O_F4] = $I_VAL[_I_F4];
    $O_VAL[_O_F5] = $I_VAL[_I_F5];
    $O_VAL[_O_F6] = $I_VAL[_I_F6];
    print
        $O_VAL[_O_IP_ADDRESS],
        $O_VAL[_O_TIMESTAMP],
        $O_VAL[_O_REQUEST],
        $O_VAL[_O_F4],
        $O_VAL[_O_F5],
        $O_VAL[_O_F6]
    ;
}

print STDERR '[apachelog.pql ' . localtime() . "] $i records.";
my $benchmark_end = new Benchmark;
my $benchmark_timediff = timediff($benchmark_start, $benchmark_end);
print STDERR '[apachelog.pql ' . localtime() . "] Code statistics: @{[timestr($benchmark_timediff)]}";
#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
sub PrintHeader
{
    local $\="\n";
    local $,="|";
    print
```

```
        ’IP_ADDRESS’,
        ’TIMESTAMP’,
        ’REQUEST’,
        ’F4’,
        ’F5’,
        ’F6’
        ;
}

#-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
#***** I N L I N E *****
use Inline
    C => Config =>
    NAME => ’apachelog’,
    CC => ’CC’,
    CLEAN_AFTER_BUILD => ’1’,
    CLEAN_BUILD_AREA => ’1’,
    PRINT_INFO => ’0’,
    BUILD_NOISY => ’0’,
    BUILD_TIMERS => ’0’,
    FORCE_BUILD => ’0’,
    LIBS => ’ ’,
    INC => ’ ’,
    CCFLAGS => ’-xchip=ultra3 -DSS_64BIT_SERVER -DBIT64 -DMACHINE64 ’,
    OPTIMIZE => ’-xO5 -xinline=%auto ’
    ;

use Inline C => <<’END_OF_C_CODE’

#define GFMAXPIPBUFFER      1536
#define GFMAXPIPFLDS        7

#define _I_IP_ADDRESS       0
#define _I_TIMESTAMP        1
#define _I_REQUEST          2
#define _I_F4               3
#define _I_F5               4
#define _I_F6               5

static const char *fields[GFMAXPIPFLDS];

int readsplit (SV* I_VAL_ref)
{
    char sql[4096];
    int ret;
    char *pzErrMsg = 0;
    register char *p;
    static char inp[GFMAXPIPBUFFER];
    register AV* I_VAL;
    register int i=0;
    int inside_quotes=0;
    int inside_sqb=0;

    if (!gets(inp) ) return 0;
    if (!SvROK(I_VAL_ref)) croak("I_VAL_ref is not a reference");

    I_VAL = (AV*)SvRV(I_VAL_ref);
    av_clear(I_VAL);

    memset(fields, 0, sizeof(fields));

    p = inp;
    fields[0] = p;
    while (*p)
    {
        if (!inside_quotes && (*p == ’[’ || *p == ’]’)) { inside_sqb = !inside_sqb; }
        if (!inside_sqb && *p == ’"’) { inside_quotes = !inside_quotes; }
        if (!inside_quotes && !inside_sqb && *p == ’ ’)
        {
            *p = ’\0’;
            av_store(I_VAL, i, newSVpvn(fields[i], strlen(fields[i])));
            fields[++i] = p + 1;
        }

        p++;
    }

    av_store(I_VAL, i, newSVpvn(fields[i], strlen(fields[i])));
    while (++i < GFMAXPIPFLDS)
        av_store(I_VAL, i, newSVpvn("", 0));
    int pN;
    return 1;
}

END_OF_C_CODE
```

7. ABOUT PEQUEL

This document was generated by Pequel.

https://sourceforge.net/projects/pequel/

COPYRIGHT

Copyright ©1999-2005, Mario Gaffiero. All Rights Reserved.
‘Pequel’ TM Copyright ©1999-2005, Mario Gaffiero. All Rights Reserved.

This program and all its component contents is copyrighted free software by Mario Gaffiero and is released under the GNU General Public License (GPL), Version 2, a copy of which may be found at http://www.opensource.org/licenses/gpl-license.html

Pequel is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Pequel is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Pequel; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA