



examples/macro_pack.pql

by *Pequel*

sample@youraddress.com

Macro Pack and Unpack Example Script

2.3

Table of Contents

Macro Pack and Unpack Example Script

SCRIPT NAME	1
DESCRIPTION	1
1. PROCESS DETAILS	1
1.1 location	1
Description	1
1.2 product_code	1
Description	1
1.3 salesman_list	1
Description	1
1.4 p_salesman_list	1
Description	1
Derived Input Field Evaluation	1
1.5 up_salesman_list	1
Description	1
Derived Input Field Evaluation	1
2. CONFIGURATION SETTINGS	2
2.1 pequeldoc	2
2.2 detail	2
2.3 noverbose	2
2.4 prefix	2
2.5 script_name	2
2.6 header	2
2.7 optimize	2
2.8 nulls	2
2.9 doc_title	2
2.10 doc_email	2
2.11 doc_version	2
3. TABLES	3
4. TABLE INFORMATION SUMMARY	4
4.1 Table List Sorted By Table Name	4
5. EXAMPLES/MACRO_PACK.PQL	5
options	5
description	5
input section	5
output section	5
6. PEQUEL GENERATED PROGRAM	6
7. ABOUT PEQUEL	8
COPYRIGHT	8

SCRIPT NAME

examples/macro_pack.pql

DESCRIPTION

Demonstrates the use of &pack and &unpack macros. These macros require a format first argument. Please refer to the perlpacktut document (on Perl.Org) for a full description of formats. The &unpack() macro will produce an array field.

1. PROCESS DETAILS

Input records are read from standard input. The input record contains **9** fields. Fields are delimited by the '|' character.

Output records are written to standard output. The output record contains **5** fields. Fields are delimited by the '|' character.

1.1 *location*

Output Field

Description

Set to input field *location*

1.2 *product_code*

Output Field

Description

Set to input field *product_code*

1.3 *salesman_list*

Output Field

Description

Set to input field *salesman_list*

1.4 *p_salesman_list*

Output Field

Description

Set to input field *p_salesman_list*

Derived Input Field Evaluation

```
=> &pack('A3/Z*' x &arr_size(salesman_list),&to_array(salesman_list))
```

1.5 *up_salesman_list*

Output Field

Description

Set to input field *up_salesman_list*

Derived Input Field Evaluation

```
=> &unpack('A3/Z*' x &arr_size(salesman_list),p_salesman_list)
```

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 *noverbose*

do not progress counter: 1

2.4 *prefix*

directory pathname prefix.: examples

2.5 *script_name*

script filename: examples/macro_pack.pql

2.6 *header*

write header record to output.: 1

2.7 *optimize*

optimize generated code.: 1

2.8 *nulls*

print zero for null numeric/decimal.: 1

2.9 *doc_title*

document title.: Macro Pack and Unpack Example Script

2.10 *doc_email*

document email entry.: sample@youraddress.com

2.11 *doc_version*

document version for pequel script.: 2.3

3. TABLES

4. TABLE INFORMATION SUMMARY

4.1 Table List Sorted By Table Name

5. EXAMPLES/MACRO_PACK.PQL

options

```
pequeldoc(pdf)
detail(1)
noverbose(1)
prefix(examples)
script_name(examples/macro_pack.pql)
header(1)
optimize(1)
nulls(1)
doc_title(Macro Pack and Unpack Example Script)
doc_email(sample@youraddress.com)
doc_version(2.3)
```

description

Demonstrates the use of &pack and &unpack macros. These macros require a format first argument. Please refer to the perlpacktut document (on Perl.Org) for a full description of formats. The &unpack() macro will produce an array field.

input section

```
product_code
cost_price
description
sales_code
sales_price
sales_qty
sales_date
location
salesman_list
p_salesman_list => &pack('A3/Z*' x &arr_size(salesman_list),&to_array(salesman_list))

up_salesman_list => &unpack('A3/Z*' x &arr_size(salesman_list),p_salesman_list)
```

output section

```
string    location      location
string    product_code  product_code
string    salesman_list  salesman_list
string    p_salesman_list p_salesman_list
string    up_salesman_list up_salesman_list
```

6. PEQUEL GENERATED PROGRAM

```
#!/usr/bin/perl
#-----
# vim: syntax=perl ts=4 sw=4
#-----
#Generated By: pequel Version 2.3-6, Build: Wednesday October 26 23:16:49 BST 2005
# : http://sourceforge.net/projects/pequel/
#Script Name : examples/macro_pack.pql
#Created On : Wed Oct 26 13:15:36 2005
#Perl Version: /usr/bin/perl 5.6.1 on solaris
#For :
#-----
#Options:
#pequeldoc(pdf) generate pod / pdf pequel script Reference Guide.
#detail(1) Include Pequel Generated Program chapter in Pequeldoc
#noverbose(1) do not progress counter
#prefix(examples) directory pathname prefix.
#script_name(examples/macro_pack.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(Macro Pack and Unpack Example Script) document title.
#doc_email(sample@youraddress.com) document email entry.
#doc_version(2.3) document version for pequel script.
#-----
use strict;
use constant _I_product_code      => int    0;
use constant _I_cost_price        => int    1;
use constant _I_description       => int    2;
use constant _I_sales_code        => int    3;
use constant _I_sales_price       => int    4;
use constant _I_sales_qty         => int    5;
use constant _I_sales_date        => int    6;
use constant _I_location          => int    7;
use constant _I_salesman_list     => int    8;
use constant _I_p_salesman_list   => int    9;
use constant _I_up_salesman_list  => int   10;
use constant _O_location          => int    1;
use constant _O_product_code      => int    2;
use constant _O_salesman_list     => int    3;
use constant _O_p_salesman_list   => int    4;
use constant _O_up_salesman_list  => int    5;
local $\="\n";
local $,="|";
use constant VERBOSE => int 10000;
use constant LAST_ICELL => int 10;
my @I_VAL;
my @O_VAL;
foreach my $f (1..5) { $O_VAL[$f] = undef; }
&PrintHeader();
while (<STDIN>)
{
    chomp;
    @I_VAL = split("[|]", $_);
    $O_VAL[_O_location] = $I_VAL[_I_location];
    $O_VAL[_O_product_code] = $I_VAL[_I_product_code];
    $O_VAL[_O_salesman_list] = $I_VAL[_I_salesman_list];
    $I_VAL[_I_p_salesman_list] = pack('A3/Z*' x int(split(/\s*,\s*/, $I_VAL[_I_salesman_list], -1)), split(/\s*,\s*/, $I_VAL[_I_salesman_list], -1));
    $O_VAL[_O_p_salesman_list] = $I_VAL[_I_p_salesman_list];
    $I_VAL[_I_up_salesman_list] = join(',', unpack('A3/Z*' x int(split(/\s*,\s*/, $I_VAL[_I_salesman_list], -1)), $I_VAL[_I_p_salesman_list]));
    $O_VAL[_O_up_salesman_list] = $I_VAL[_I_up_salesman_list];
    print STDOUT
        $O_VAL[_O_location],
        $O_VAL[_O_product_code],
        $O_VAL[_O_salesman_list],
        $O_VAL[_O_p_salesman_list],
        $O_VAL[_O_up_salesman_list]
    ;
}

#-----
sub PrintHeader
{
    local $\="\n";
    local $,="|";
    print
        'location',
        'product_code',
        'salesman_list',
```

```
    'p_salesman_list',  
    'up_salesman_list'  
    ;  
}
```

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

