

Coding (Or Script) - 7

```
#!/usr/local/bin/perl5
```

```
##### HEADER BEGIN #####
```

```
#  
# NO PART OF THIS DOCUMENT MAY BE REPRODUCED WITHOUT  
# THE WRITTEN PERMISSION OF TELLABS Operations, INC.,  
# 4951 INDIANA AVENUE, LISLE, IL 60532  
#
```

```
#####
```

```
# Name: removeviews_gen.pl  
# Type: perl5 script  
# Originator: KEJIAN WU, BRAIN FLEMING 2/12/01 - ONG Software Tools Group  
#  
# Version Control Information:  
# /vobs/dstools/titan/clearcase/scripts/removeviews/removeviews_gen.pl  
#  
# Description: This script will generate all the user directories, data and  
# user mail list file for unused MR views.  
#
```

```
##### HEADER END #####
```

```
require "getopts.pl";  
use Getopt::Std;  
use File::Basename;
```

```
# use DSDTools;  
use Getopt::Std;  
use File::Copy;  
use Date::Calc qw(Add_Delta_Days);  
use Date::Calc qw(Delta_Days);
```

```
#  
$CT = "/usr/atria/bin/cleartool";  
$TITANADM_HOME = "/home/tellab8/titanadm";  
$PCMRs = "$TITANADM_HOME/bin/pcmr";  
$DATA_FILE_DIR = "/home/tlabas-6/kwu/rem_temp/ken";  
#
```

```
($DAY, $MONTH, $YEAR) = (localtime)[3,4,5];  
chomp($keep_day = 10);  
($Y, $M, $D) = Add_Delta_Days($YEAR+1900, $MONTH+1, $DAY, $keep_day);
```

```
open(LSVIEW_CT, "$CT lsview -short |") || die "removeviews_gen.pl: Could not execute command! LSVIEW_CT\n";
```

```
# Initialization for hash
```

```
while(<LSVIEW_CT>)
```

```
{  
    chomp($viewtag_ct = $_);  
  
    if($viewtag_ct =~ /(.*?)_mr(\d+)/){  
        {  
            $user_ct = $1;  
            $count{$user_ct} = 0;  
            $file_flag{$user_ct} = 0;  
        }  
    }  
    elsif($viewtag_ct =~ /(.*?)_tmp(\d+)/){  
        {  
            $user_ct = $1;  
            $count{$user_ct} = 0;  
            $file_flag{$user_ct} = 0;  
        }  
    }  
}  
close(LSVIEW_CT);
```

```
# List all the views
```

```
open(LSVIEW, "$CT lsview -short |") || die "removeviews_gen.pl: Could not execute command! LSVIEW\n";
```

```
while(<LSVIEW>)
```

```
{  
    chomp($viewtag = $_);  
  
    if($viewtag =~ /(.*?)_mr(\d+)/){  
        {  
            $user = $1; # get user  
            $mrno = $2; # get mr number  
  
            if(-e "$DATA_FILE_DIR/$user") # if the file exist  
            {  
                if($count{$user} == 0) # To read each existing user file one time only  
                {  
                    $file_flag{$user} = 1; # =1 means the user file exist  
                    open(FILE, "<$DATA_FILE_DIR/$user") || die "removeviews_gen.pl: can not open $DATA_FILE_DIR/$user for reading $! \n";
```

```

        while($line = <FILE>)
        {
            @string = split(/ /, $line);
            chomp $string[0];
            # store each user views for comparison
            $oldview{$user} .= "$string[0] ";
        }
        close(FILE);
    }
    $count{$user}++;

    # Checking MR views has been built into the load and
    # has been put into a terminated state in MRSYS
    if( mr_load($mrno) || mr_resolved($mrno) )
    {
        $current_view{$user} .= "$viewtag ";
    }
}

if($viewtag =~ /(.*?)_tmp(\d+)/)
{
    $user = $1; # get user
    $mrno = $2; # get mr number

    if(-e "$DATA_FILE_DIR/$user") # if the file exist
    {
        if($count{$user} == 0) # To read each existing user file one time only
        {
            $file_flag{$user} = 1; # =1 means the user file exist
            open(FILE, "<$DATA_FILE_DIR/$user") || die "removeviews_gen.pl: can not open $DATA_FILE_DIR/$user for reading $! \n";
            while($line = <FILE>)
            {
                @string = split(/ /, $line);
                chomp $string[0];
                # store each user views for comparison
                $oldview{$user} .= "$string[0] ";
            }
            close(FILE);
        }
        $count{$user}++;

        $current_view{$user} .= "$viewtag ";
    }
}

}
close(LSVIEW);

open(MAIL_LIST, ">$DATA_FILE_DIR/MAIL_LIST") || die "removeviews_gen.pl: can not open $DATA_FILE_DIR/MAIL_LIST $! \n";

# for each users
foreach (keys(%current_view)) {

    print MAIL_LIST $_; # print each users to the file MAIL_LIST
    print MAIL_LIST "\n";

    open(DATA_FILE, ">>$DATA_FILE_DIR/$_") || die "removeviews_gen.pl: can not open $DATA_FILE_DIR/$_ $! \n";

    chop($current_view[$_]); # remove ending space
    @viewlist = split (" ", $current_view[$_]);

    chop($oldview[$_]); # remove ending space
    @oldviewlist = split (" ", $oldview[$_]);

    if($file_flag[$_] == 0) # the user file does not exist
    {
        foreach $view (@viewlist)
        {
            if($view =~ /(.*?)_mr(\d+)/)
            {
                print DATA_FILE "$view ";
                print DATA_FILE "0-0-0\n";
            }
            elsif($view =~ /(.*?)_tmp(\d+)/)
            {
                print DATA_FILE "$view ";
                printf(DATA_FILE "%04d-%02d-%02d\n", $Y, $M, $D);
            }
        }
    }
}

```

```

else # the user file exist
{
    outer: foreach $view (@viewlist)
    {
        foreach $the_oldview (@oldviewlist)
        {
            if($view eq $the_oldview)
            {
                next outer;
            }
        }

        if($view =~ /(.*_mr(\d+)$/)
        {
            print DATA_FILE "$view ";
            print DATA_FILE "0-0-0\n";
        }
        elseif($view =~ /(.*_tmp(\d+)$/)
        {
            print DATA_FILE "$view ";
            printf(DATA_FILE "%04d-%02d-%02d\n", $Y, $M, $D);
        }
    }
}

close(DATA_FILE);
chmod 0666,"$DATA_FILE_DIR/$_";
}
close(MAIL_LIST);

# Confirm The MR is old/built by checking the state in MR system
sub mr_resolved {

    my ($mrno) = @_;

    open(PCMRS, "$PCMRS -c get_field -f state -n -m $mrno |") || die "PCMRS $! \n";

    $remove_resolve = 1;
    undef $resolve;
    while(<PCMRS>)
    {
        chomp($resolve = $_);
    }
    close(PCMRS);

    if($resolve =~ /submitted/)
    {
        $remove_resolve = 0;
    }
    elseif($resolve =~ /open/)
    {
        $remove_resolve = 0;
    }
    elseif($resolve =~ /assigned/)
    {
        $remove_resolve = 0;
    }
    elseif($resolve =~ /deferred/)
    {
        $remove_resolve = 0;
    }
    elseif($resolve =~ /approved/)
    {
        $remove_resolve = 0;
    }
}

return $remove_resolve;
}

```

```

# Checking the state attribute in the brtype in the admin vob to see
# if has been built into the load
sub mr_load {

```

```

    my ($mrno) = @_;

    undef $load_build;
    $load_build = " ";

    open(LOAD, "$CT desc -short -aattr state brtype:mr$mrno@/vobs/titan5500/admin |") || die "LOAD $! \n";

    while(<LOAD>)
    {
        chomp($load_build = $_);
    }

```

```
)
close(LOAD);

$remove_load = 1;
if($load_build =~ /working/) # $load_build = "working", then user can not remove the views
{
    $remove_load = 0;
}

if( $load_build =~ / / ) # The state attribute is an empty string
{
    $remove_load = 0;
}

return $remove_load;
}
```

```
open(MAIL, "<$DATA_FILE_DIR/MAIL_LIST") || die "cant not open\n";
```

```
while($themail = <MAIL>)
{
    chomp($themail);
    open(data, "<$DATA_FILE_DIR/$themail") || die "canot not open \n";
    while($line10 = <data>)
    {
        chomp($line10);
        @string10 = split(/ /, $line10);
        if($string10[0] =~ /(.*?)_mr(\d+)$/)
        {
            $the_file = $1;
            open(TEMP20, ">>$DATA_FILE_DIR/FILE20/$themail") || die "20 \n";
            print TEMP20 "$string10[0] ";
            print TEMP20 "$string10[1]\n";
        }
        close(TEMP20);
    }
    close(data);
}
close(MAIL);
```

```
open(MAIL10, "<$DATA_FILE_DIR/MAIL_LIST") || die "cant not open\n";
```

```
while($themail10 = <MAIL10>)
{
    chomp($themail10);
    open(data10, "<$DATA_FILE_DIR/$themail10") || die "canot not open \n";
    while($line20 = <data10>)
    {
        chomp($line20);
        @string20 = split(/ /, $line20);
        if($string20[0] =~ /(.*?)_tmp(\d+)$/)
        {
            $the_file10 = $1;
            open(TEMP30, ">>$DATA_FILE_DIR/FILE20/$themail10") || die "30 \n";
            print TEMP30 "$string20[0] ";
            print TEMP30 "$string20[1]\n";
        }
        close(TEMP30);
    }
    close(data10);
}
close(MAIL10);
```

```
system("mv /home/tlabas-6/kwu/rem_temp/ken/FILE20/* /home/tlabas-6/kwu/rem_temp/ken");
```