

Difficulté



Facile

## Error occurs by browsing the archives

By Fedora 23, an error occurs by browsing the archives:

[https://bugzilla.redhat.com/show\\_bug.cgi?id=1290964](https://bugzilla.redhat.com/show_bug.cgi?id=1290964)

The solution is given [here](#) and [the rdiff](#): edit /usr/share/BackupPC/lib/BackupPC/CGI/Browse.pm and put into:

```
#===== -*-perl-*-
#
# BackupPC::CGI::Browse package
#
# DESCRIPTION
#
#   This module implements the Browse action for the CGI interface.
#
# AUTHOR
#   Craig Barratt <cbarratt@users.sourceforge.net>
#
# COPYRIGHT
#   Copyright (C) 2003-2015  Craig Barratt
#
#   This program 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.
#
#   This program 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 this program; if not, write to the Free Software
#   Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301
USA
#
#=====
#
# Version 3.3.1, released 11 Jan 2015.
#
# See http://backuppc.sourceforge.net.
#
#=====
```

```
package BackupPC::CGI::Browse;

use strict;
use Encode qw/decode_utf8/;
use BackupPC::CGI::Lib qw(:all);
use BackupPC::View;
use BackupPC::Attrib qw(:all);

sub action
{
    my $Privileged = CheckPermission($In{host});
    my($i, $dirStr, $fileStr, $attr);
    my $checkBoxCnt = 0;

    if ( !$Privileged ) {
ErrorExit(eval("qq{$Lang->{Only_privileged_users_can_browse_backup_files}}")
);
    }
    my $host = $In{host};
    my $num = $In{num};
    my $share = $In{share};
    my $dir = $In{dir};

    ErrorExit($Lang->{Empty_host_name}) if ( $host eq "" );
    #
    # Find the requested backup and the previous filled backup
    #
    my @Backups = $bpc->BackupInfoRead($host);

    #
    # default to the newest backup
    #
    #if ( !defined($In{num}) && defined(@Backups) && @Backups > 0 ) {
if ( !$In{num} && @Backups > 0 ) {
    $i = @Backups - 1;
        $num = $Backups[$i]{num};
    }

    for ( $i = 0 ; $i < @Backups ; $i++ ) {
        last if ( $Backups[$i]{num} == $num );
    }
    if ( $i >= @Backups || $num !~ /\^d+$/ ) {
        ErrorExit("Backup number ${EscHTML($num)} for host ${EscHTML($host)}
does"
            . " not exist.");
    }
    my $backupTime = timeStamp2($Backups[$i]{startTime});
    my $backupAge = sprintf("%.1f", (time - $Backups[$i]{startTime})
        / (24 * 3600));
    my $view = BackupPC::View->new($bpc, $host, \@Backups, {nlink => 1});
```

```

if ( $dir eq "" || $dir eq "." || $dir eq ".." ) {
  $attr = $view->dirAttrib($num, "", "");
  if ( keys(%$attr) > 0 ) {
    $share = (sort(keys(%$attr)))[0];
    $dir = '/';
  } else {
    ErrorExit(eval("qq{$Lang->{Directory__EscHTML}}"));
  }
}
$dir = "/$dir" if ( $dir !~ /^\\// );
my $relDir = $dir;
my $currDir = undef;
if ( $dir =~ m{(^|/)\.\.(/|$)} ) {
  ErrorExit($Lang->{Nice_try__but_you_can_t_put});
}

#
# Loop up the directory tree until we hit the top.
#
my(@DirStrPrev);
while ( 1 ) {
  my($fLast, $fLastum, @DirStr);

  $attr = $view->dirAttrib($num, $share, $relDir);
  # if ( !defined($attr) ) {
  if ( !$attr ) {
    $relDir = decode_utf8($relDir);
    ErrorExit(eval("qq{$Lang->{Can_t_browse_bad_directory_name2}}"));
  }

  my $fileCnt = 0;          # file counter
  $fLast = $dirStr = "";

  #
  # Loop over each of the files in this directory
  #
  foreach my $f ( sort {uc($a) cmp uc($b)} keys(%$attr) ) {
    my($dirOpen, $gotDir, $imgStr, $img, $path);
    my $fURI = $f;          # URI escaped $f
    my $shareURI = $share; # URI escaped $share
    if ( $relDir eq "" ) {
      $path = "$f";
    } else {
      ($path = "$relDir/$f") =~ s{//+}{/}g;
    }
    if ( $shareURI eq "" ) {
      $shareURI = $f;
      $path = "/";
    }

    $path =~ s{^/+}{/};
    $path = s/([^\w.\//-])/uc sprintf("%%02X", ord($1))/eg;
  }
}

```

```
$fURI      =~ s/([^\w.\//-])/uc sprintf("%%02X", ord($1))/eg;
$shareURI  =~ s/([^\w.\//-])/uc sprintf("%%02X", ord($1))/eg;
#          $dirOpen = 1 if ( defined($currDir) && $f eq $currDir );
$dirOpen = 1 if ( $currDir && $f eq $currDir );
    if ( $attr->{$f}{type} == BPC_FTYPE_DIR ) {
        #
        # Display directory if it exists in current backup.
        # First find out if there are subdirs
        #
        my $tdStyle;
        my $linkStyle = "fview";
$im� | = 1 << 6;
$im� | = 1 << 5 if ( $attr->{$f}{nlink} > 2 );
if ( $dirOpen ) {
    $linkStyle = "fviewbold";
    $im� | = 1 << 2;
    $im� | = 1 << 3 if ( $attr->{$f}{nlink} > 2 );
}
my $im�FileName = sprintf("%07b.gif", $im�);
$im�Str = "<im� src=\"\$Conf{CgiImageDirURL}/$im�FileName\"
align=\"absmiddle\" width=\"9\" height=\"19\" border=\"0\">";
if ( "$relDir/$f" eq $dir ) {
    $tdStyle = "fviewon";
} else {
    $tdStyle = "fviewoff";
}
my $dirName = $f;
$dirName =~ s/ /&nbsp;/g;
    $dirName = decode_utf8($dirName);
push(@DirStr, {needTick => 1,
                tdArgs => " class=\"\$tdStyle\"",
                link => <<EOF});
<a
href="$MyURL?action=browse&host=${EscURI($host)}&num=$num&share=$shareURI&di
r=$path">$im�Str</a><a
href="$MyURL?action=browse&host=${EscURI($host)}&num=$num&share=$shareURI&di
r=$path" class="$linkStyle">&nbsp;$dirName</a></td></tr>
EOF
    $fileCnt++;
    $gotDir = 1;
if ( $dirOpen ) {
    my($lastTick, $doneLastTick);
    foreach my $d ( @DirStrPrev ) {
        $lastTick = $d if ( $d->{needTick} );
    }
#    $doneLastTick = 1 if ( !defined($lastTick) );
$doneLastTick = 1 if ( !$lastTick );
    foreach my $d ( @DirStrPrev ) {
        $im� = 0;
        if ( $d->{needTick} ) {
```

```

        $img |= 1 << 0;
    }
    if ( $d == $lastTick ) {
        $img |= 1 << 4;
        $doneLastTick = 1;
    } elseif ( !$doneLastTick ) {
        $img |= 1 << 3 | 1 << 4;
    }
    my $imgFileName = sprintf("%07b.gif", $img);
    $imgStr = "<img src=\"\$Conf{CgiImageDirURL}/$imgFileName\"
align=\"absmiddle\" width=\"9\" height=\"19\" border=\"0\">";
    push(@DirStr, {needTick => 0,
                    tdArgs  => $d->{tdArgs},
                    link     => $imgStr . $d->{link}
                });
    }
}

}

if ( $relDir eq $dir ) {
    #
    # This is the selected directory, so display all the files
    #
    my ($attrStr, $iconStr);
#
    if ( defined($a = $attr->{$f}) ) {
if ( $a = $attr->{$f} ) {
        my $mtimeStr = $bpc->timeStamp($a->{mtime});
        # UGH -> fix this
        my $typeStr = BackupPC::Attrib::fileType2Text(undef,
            $a->{type});
        my $modeStr = sprintf("%o", $a->{mode} & 07777);
        $iconStr = <<EOF;

EOF
            $attrStr .= <<EOF;
            <td align="center" class="fviewborder">$typeStr</td>
            <td align="center" class="fviewborder">$modeStr</td>
            <td align="center" class="fviewborder">$a->{backupNum}</td>
            <td align="right" class="fviewborder">$a->{size}</td>
            <td align="right" class="fviewborder">$mtimeStr</td>
</tr>
EOF
                } else {
                    $attrStr .= "<td colspan=\"5\" align=\"center\"
class=\"fviewborder\"> </td>\n";
                }
                (my $fDisp = "${EscHTML($f)}") =~ s/ /&nbsp;/g;
                $fDisp = decode_utf8($fDisp);
                if ( $gotDir ) {
                    $fileStr .= <<EOF;
<tr><td class="fviewborder">
    <input type="checkbox" name="fcb$checkBoxCnt"

```

```
value="$path">&nbsp;$iconStr&nbsp;<a
href="$MyURL?action=browse&host=${EscURI($host)}&num=$num&share=$shareURI&di
r=$path">$fDisp</a>
</td>
$attrStr
</tr>
EOF
        } else {
            $fileStr .= <<EOF;
<tr><td class="fviewborder">
    <input type="checkbox" name="fcb$checkBoxCnt"
value="$path">&nbsp;$iconStr&nbsp;<a
href="$MyURL?action=RestoreFile&host=${EscURI($host)}&num=$num&share=$shareU
RI&dir=$path">$fDisp</a>
</td>
$attrStr
</tr>
EOF
        }
        $checkBoxCnt++;
    }
}
@DirStrPrev = @DirStr;
last if ( $relDir eq "" && $share eq "" );
#
# Prune the last directory off $relDir, or at the very end
# do the top-level directory.
#
if ( $relDir eq "" || $relDir eq "/" || $relDir !~ /(.*)\/(.*)/ ) {
    $currDir = $share;
    $share = "";
    $relDir = "";
} else {
    $relDir = $1;
    $currDir = $2;
}
}
$share = $currDir;
my $shareURI = $share;
$shareURI =~ s/([^\w.\//-])/uc sprintf("%%02x", ord($1))/eg;

#
# allow each level of the directory path to be navigated to
#
my($thisPath, $dirDisplay);
my $dirClean = $dir;
$dirClean =~ s{//+}{/}g;
$dirClean =~ s{/+}$};
my @dirElts = split(/\//, $dirClean);
@dirElts = ("/") if ( !@dirElts );
```

```
foreach my $d ( @dirElts ) {
    my($thisDir);

    if ( $thisPath eq "" ) {
        $thisDir = decode_utf8($share);
        $thisPath = "/";
    } else {
        $thisPath .= "/" if ( $thisPath ne "/" );
        $thisPath .= "$d";
        $thisDir = decode_utf8($d);
    }
    my $thisPathURI = $thisPath;
    $thisPathURI =~ s/([\w.\//-])/uc sprintf("%%02x", ord($1))/eg;
    $dirDisplay .= "/" if ( $dirDisplay ne "" );
    $dirDisplay .= "<a
href=\"\$MyURL?action=browse&host=${EscURI($host)}&num=$num&share=$shareURI&dir=$thisPathURI\">${EscHTML($thisDir)}</a>";
}

my $filledBackup;

if ( (my @mergeNums = @{$view->mergeNums}) > 1 ) {
    shift(@mergeNums);
    my $numF = join(", #", @mergeNums);
    $filledBackup =
eval("qq{$Lang->{This_display_is_merged_with_backup}}");
}

foreach my $d ( @DirStrPrev ) {
    $dirStr .= "<tr><td$d->{tdArgs}>$d->{link}\\n";
}

### hide checkall button if there are no files
my ($stopCheckAll, $checkAll, $fileHeader);
if ( $fileStr ) {
    $fileHeader = eval("qq{$Lang->{fileHeader}}");

    $checkAll = $Lang->{checkAll};

    # and put a checkall box on top if there are at least 20 files
    if ( $checkBoxCnt >= 20 ) {
        $stopCheckAll = $checkAll;
        $stopCheckAll =~ s{allFiles}{allFilestop}g;
    }
} else {
    $fileStr = eval("qq{$Lang->{The_directory_is_empty}}");
}
my $pathURI = $dir;
$pathURI =~ s/([\w.\//-])/uc sprintf("%%02x", ord($1))/eg;
if ( my @otherDirs = $view->backupList($share, $dir) ) {
    my $otherDirs;
```

```
    foreach my $i ( @otherDirs ) {
        my $selected;
        my $showDate = timeStamp2($Backups[$i]{startTime});
        my $backupNum = $Backups[$i]{num};
        $selected = " selected" if ( $backupNum == $num );
        $otherDirs .= "<option
value=\"\$MyURL?action=browse&host=${EscURI($host)}&num=$backupNum&share=$shareURI&dir=$pathURI\"$selected>#\$backupNum - ($showDate)</option>\n";
    }
    $filledBackup .=
eval("qq{\$Lang->{Visit_this_directory_in_backup}}");
}
$dir = decode_utf8($dir);
$share = decode_utf8($share);

my $content = eval("qq{\$Lang->{Backup_browse_for__host}}");
Header(eval("qq{\$Lang->{Browse_backup__num_for__host}}"), $content);
Trailer();
}

1;
```

From:  
<https://wiki.guedel.eu/> - **Wiki-Guedel**

Permanent link:  
[https://wiki.guedel.eu/doku.php?id=welcome:fedora:f23\\_bug\\_into\\_backuppc](https://wiki.guedel.eu/doku.php?id=welcome:fedora:f23_bug_into_backuppc)

Last update: **2016/02/09 22:35**

