```perl
#!/usr/bin/env perl
# $Id: Sreview,v 1.8 2012/08/07 22:33:16 murakami Exp murakami $
# See perl pod documentation at the end of this file

use strict;
use warnings;
use Pod::Usage;
use POSIX;
use Time::Local qw(timelocal);
use Getopt::Long;

use constant PROGRAM => "Sreview";
use constant VERSION => "1.3";

##::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
##
## PROTOTYPES
##
##::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

sub parse_args;
sub _error;

##::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
##
## GLOBALS
##
##::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

my @OBJECTS;
my $CMD;
my $ARG;
my $ERRORS = 0;
my $RECURSIVE = 0;
my $STATUS;
my $DEBUG = 0;
my $VERBOSE = 0;

##::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
##
## BEGIN
##
##::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

if (! parse_args())
  {
    _error("failed to parse arguments");
    exit(1);
  }

## Exit if $DEBUG is set before actually modifying files
if ($DEBUG)
  {
    exit(0);
  }

# Set scheme info on all requested objects
```

```
 58 $CMD = 'Sscheme -w -val Admin.Last_Review_Time::[sysdate]';
 59
 60 if ($RECURSIVE)
 61   {
 62     $CMD .= " -R";
 63   }
 64
 65 my $OBJSTR = join(' ', @OBJECTS);
 66 if ($OBJSTR =~ m/\*/)                          # If wildcard
 67   {
 68     printf("DEBUG: %s\n", $OBJSTR) if ($VERBOSE);
 69     $OBJSTR =~ s/\*/\\\*/g;                     # Escape wildcard
 70     printf("DEBUG: %s\n", $OBJSTR) if ($VERBOSE);
 71   }
 72
 73 $CMD .= sprintf(" %s", $OBJSTR);
 74
 75 printf("DEBUG: %s\n", $CMD) if ($VERBOSE);
 76
 77 $STATUS = system($CMD);
 78
 79 if (($STATUS >> 8) != 0)
 80   {
 81     $ERRORS++;
 82   }
 83
 84 if ($ERRORS)
 85   {
 86     exit(1);
 87   }
 88
 89 exit(0);
 90
 91 ##::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 92 ##
 93 ## SUBROUTINES
 94 ##
 95 ##::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 96
 97 #::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 98 #
 99 # NAME
100 #      _debug -- print debug information
101 #
102 # SYNOPSIS
103 #      _debug($fmt, $arg1[, $arg2]);
104 #
105 # DESCRIPTION
106 #      If $VERBOSE is set to 1, prints $fmt and optional $args using printf to
107 #      STDOUT.
108 #
109 # RETURN VALUES
110 #      None.
111 #
112 #::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
113 sub _debug
114 {
```

```
115   my $fmt = shift(@_);
116   my @args = @_;
117   my $msg;
118
119   return if (! $VERBOSE);
120
121   $fmt = sprintf("DEBUG: %s%s", $fmt, ($fmt !~ /\n$/ ? "\n" : ""));
122
123   $msg = sprintf($fmt, @args);
124
125   printf($msg);
126 }
127
128 #:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
129 #
130 # NAME
131 #       _error -- print error information
132 #
133 # SYNOPSIS
134 #       _error($fmt, $arg1[, $arg2]);
135 #
136 # DESCRIPTION
137 #       Prints $fmt and options $args using printf to STDERR.
138 #
139 # RETURN VALUES
140 #       None.
141 #
142 #:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
143 sub _error
144 {
145   my $fmt = shift(@_);
146   my @args = @_;
147   my $msg;
148
149   $fmt = sprintf("Sreview ERROR: %s%s", $fmt, ($fmt !~ /\n$/ ? "\n" : ""));
150
151   $msg = sprintf($fmt, @args);
152
153   printf(STDERR $msg);
154 }
155
156 #:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
157 #
158 # NAME
159 #       parse_args -- parse command line arguments
160 #
161 # SYNOPSIS
162 #       $status = parse_args();
163 #
164 # DESCRIPTION
165 #       The parse_args subroutine parses command line arguments provided by the
166 #       @ARGV array.
167 #
168 # RETURN VALUES
169 #       Upon successful completion a value of 1 is returned.  Otherwise, a value
170 #       of 0 is returned.
171 #
```

```
172 #::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
173 sub parse_args
174 {
175   my %args;
176   my $arg;
177   my $status;
178 
179   GetOptions
180     (
181       \%args,
182       "debug",
183       "verbose",
184       "recursive|r",
185       "version|v" => sub { printf("%s %s\n", PROGRAM, VERSION); exit(0); },
186       "help|h" => sub { pod2usage(-exitval => 0, -verbose => 1); },
187     ) or pod2usage(2);
188 
189   # Args are objects/collections
190   if ($#ARGV == -1)
191     {
192       _error("must specifiy one or more objects or collections");
193       return(0);
194       pod2usage(2);
195     }
196 
197   push(@OBJECTS, @ARGV);
198 
199   # Parse remaining arguments
200   if (exists($args{verbose}))
201     {
202       $VERBOSE = 1;
203     }
204 
205   if (exists($args{debug}))
206     {
207       $DEBUG = 1;
208     }
209 
210   if (exists($args{recursive}))
211     {
212       $RECURSIVE = 1;
213     }
214 
215   return(1);
216 }
217 
218 
219 __END__
220 
221 ##::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
222 ##
223 ## POD DOCUMENTATION
224 ##
225 ## perldoc Sreview
226 ##
227 ##::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
228 
```

```
229 =pod
230
231 =head1 NAME
232
233 B<Sreview> - set Admin.Last_Review_Time attribute.
234
235 =head1 SYNOPSIS
236
237 S<B<Sreview [options] objects|collections>>
238
239 =head1 DESCRIPTION
240
241 B<Sreview> is the command line interface to set I<"Last_Review_Time"> on
242 Storage Lifecycle Management (SLM) objects. The B<Sreview> command updates the
243 I<"Admin"> Scheme which is a System Schmeme that is automatically applied during
244 ingestion of objects into SLM. The B<Sreview> command requires B<objects> or
245 B<collections> to act upon.
246
247 =head1 objects|collections
248
249 The B<objects|collections> parameter is a required parameter. More than
250 one B<object> and/or B<collection> can be specified. The
251 B<objects|collections> parameter specifies the SLM objects and/or
252 collections to act upon.
253
254 =head1 OPTIONS
255
256 The B<Sreview> command allows a number of options when setting I<"Admin"> Scheme
257 attribute I<"Last_Review_Time"> and includes the B<-r> option
258 which causes the attributes to be set recursively on objects within the
259 collection[s].
260
261 The options are as follows:
262
263 =over 4
264
265 =item B<-R>, B<-recursive>
266
267 Specifies that the operation be applied recursively on objects within the
268 collection[s]. The B<-R> option only works when collections are specified.
269
270 =item B<-help>
271
272 Print this help message and exit.
273
274 =item B<-version>
275
276 Print the program version and exit.
277
278 =item B<-verbose>
279
280 Print more verbose output during the execution of B<Sreview>.
281
282 =back
283
284 =head1 EXAMPLES
285
```

```
286 =over 4
287
288 =item Example 1 Setting Last_Review_Time for a single object.
289
290 % S<Sreview MyObj>
291
292 This command will set the Last_Review_Time for the object MyObj in the current
293 collection to the current date and time.
294
295 The equivalent native Sscheme command is:
296
297 % S<Sscheme -w -val 'Admin.Last_Review_Time::[sysdate]' MyObj>
298
299 =item Example 2 Setting Last_Review_Time for multiple objects.
300
301 % S<Sreview MyObj MyObj2 ...>
302
303 This command will set the Last_Review_Time for the objects listed in the current
304 collection to the current date and time.
305
306 The equivalent native Sscheme command is:
307
308 % S<Sscheme -w -val 'Admin.Last_Review_Time::[sysdate]' MyObj MyObj2 ...>
309
310 =item Example 3 Setting Last_Review_Time recursively on a collection
311
312 % S<Sreview -R MyCol>
313
314 This command will set the Last_Review_Time for the collection MyCol and all
315 objects in the collection recursively to the current date and time.
316
317 The equivalent native Sscheme command is:
318
319 % S<Sscheme -w -val 'Admin.Last_Review_Time::[sysdate]' MyObj>
320
321 =back
322
323 =head1 SEE ALSO
324
325 B<Sscheme(1)>, B<Sretain(1)>
326
327 =head1 RELEASE NOTES
328
329 =over 4
330
331 =item Version 1.1 - June 28, 2012 - LAM@HTL
332
333 Initial release.
334
335 Stripped down version of Sretain to allow setting just the
    Admin.Last_Review_Time.
336
337 =item Version 1.2 - June 28, 2012 - LAM@HTL
338
339 Modified script to submit one 'Sscheme -w' command for all objects specified.
340
341 =item Version 1.3 - August 7, 2012 - LAM@HTL
```

```
342
343 Modified help and man page to specify "-R" instead of "-r" for recursion to
    match other Scommands.
344 Modified to always escape any asterisk wildcard characters in the
    objects|collections argument.
345
346 =back
347
348 =cut
349
```