

SWC READ ME

SWC is a PERL package that is part of the Structural Biology Extensible Visualization Scripting Language (SBEVSL) project. The SWC framework contains a Director object and a set of input and output converters. Currently, the SWC package allows for the conversion from SBEVSL to PyMol, restricted to the commands “zoom” and “rotate”.

FILES:

The package contains the following files:

SWC::Director # This is the main object which directs all the conversions.

SWC::Exception #The Exception creation and handling within SWC package.

SWC::Converters::PyMol #The PyMol output converter which is used by the Director object.

OBJECT RELATIONSHIP:

The main object is the Director. Converter objects can be utilized by supplying the `set_input_converter` and `set_output_convert` methods on the Director object with the appropriate name (e.g. “pymol”). Note, the converter objects are never created manually, but through the Director.

INSTALL:

To use the package, place the folder SWC in a directory in which you have read and write privileges (Example: ‘/home/bobby/modules’). Make sure the directory has the correct permissions set for the apache cgi_mod to read the files. Permissions should be set to 755 with the command ‘chmod’ in Unix.

In the script in which you want to use the package, add the path into @INC. This can be done by the following command in perl:

```
use lib '/home/bobby/modules/';
```

You should now be able to use the package by including the package name as one would normally in perl:

```
use SWC::Director;
use SWC::Exception;
```

USAGE:

Below is an example script of how to use the SWC package.

```
#!/usr/local/bin/perl -w

use lib '/home/amc8507/modules/builder';

use strict;
use Error qw(:try);

#Include the package and the Exception
use SWC::Director;
use SWC::Exception;

try {
    #Create a Director object, then set the output converter type
    my $converter = SWC::Director->new();
    $converter->set_output_converter("pymol");

    #create some commands, this could be from any source, but it has to be
    #one command per line.
    my $SBEVSL = "zoom true\nzoom 50\nrotate x 90\nrotate z 180\nzoom OFF\n";

    #split commands into array of lines
    my @lines = split(/\n/, $SBEVSL);

    #call convert on the Director, passing the array of commands
    $converter->convert(\@lines);

    #obtain output from Director
    my @output = @ { $converter->get_output() };

    #Print the output
    foreach (@output) {
        print "$_\n";
    }
} catch SWC::Exception with {
    my $ex = shift;
    print("SBEVSL Exception occurred - " . $ex->getMessage() . "\n");
    return 0;
};

exit 0;
```