PERL TIE FUNCTION

http://www.tutorialspoint.com/perl/perl_tie.htm

Description

This function ties the VARIABLE to the package class CLASSNAME that provides implementation for the variable type. Any additional arguments in LIST are passed to the constructor for the entire class. Typically used to bind hash variables to DBM databases.

Syntax

Following is the simple syntax for this function −

```
tie VARIABLE, CLASSNAME, LIST
```

Return Value

This function returns reference to tied object.

Example

Following is the example code showing its basic usage, we have only two files in /tmp directory −

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

package MyArray;

sub TIEARRAY {
        print "TYING\n";
        bless [];
}
sub DESTROY {
        print "DESTROYING\n";
}
sub STORE{
        my ($self, $index, $value ) = @_;
        print "STORING $value at index $index\n";
        $self[$index] = $value;
}

sub FETCH {
        my ($self, $index ) = @_;
        print "FETCHING the value at index $index\n";
        return $self[$index];
}

package main;
$object = tie @x, MyArray; #@x is now a MyArray array;

print "object is a ", ref($object), "\n";

$x[0] = 'This is test'; #this will call STORE();
print $x[0], "\n";      #this will call FETCH();
print $object->FETCH(0), "\n";
untie @x      #now @x is a normal array again.
```

When above code is executed, it produces the following result −

```
TYING
object is a MyArray
STORING This is test at index 0
FETCHING the value at index 0
This is test
FETCHING the value at index 0
```

```
This is test
DESTROYING
```

When the tie function is called, what actually happens is that the TIESCALAR method from FileOwner is called, passing '.bash_profile' as the argument to the method. This returns an object, which is associated by tie to the $profile variable.

When *profileisusedintheprintstatements, theFETCHmethodiscalled. Whenyouassignavalueto*profile, the STORE method is called, with 'mcslp' as the argument to the method. If you can follow this, then you can create tied scalars, arrays, and hashes, since they all follow the same basic model. Now let.s examine the details of our new FileOwner class, starting with the TIESCALAR method: