

## NAME

ExtUtils::ParseXS::Eval - Clean package to evaluate code in

## SYNOPSIS

```
use ExtUtils::ParseXS::Eval;
my $rv = ExtUtils::ParseXS::Eval::eval_typemap_code(
    $parsexs_obj, "some Perl code"
);
```

## SUBROUTINES

### `$pxs->eval_output_typemap_code($typemapcode, $other_hashref)`

Sets up various bits of previously global state (formerly [ExtUtils::ParseXS](#) package variables) for eval'ing output typemap code that may refer to these variables.

Warns the contents of `$@` if any.

Not all these variables are necessarily considered “public” wrt. use in typemaps, so beware. Variables set up from the [ExtUtils::ParseXS](#) object:

```
$Package $Alias $func_name $Full_func_name $pname
```

Variables set up from `$other_hashref`:

```
$var $type $ntype $subtype $arg
```

### `$pxs->eval_input_typemap_code($typemapcode, $other_hashref)`

Sets up various bits of previously global state (formerly [ExtUtils::ParseXS](#) package variables) for eval'ing output typemap code that may refer to these variables.

Warns the contents of `$@` if any.

Not all these variables are necessarily considered “public” wrt. use in typemaps, so beware. Variables set up from the [ExtUtils::ParseXS](#) object:

```
$Package $Alias $func_name $Full_func_name $pname
```

Variables set up from `$other_hashref`:

```
$var $type $ntype $subtype $num $init $printed_name $arg $argoff
```

## TODO

Eventually, with better documentation and possible some cleanup, this could be part of [ExtUtils::Typemaps](#)