NAME

ExtUtils::Typemaps::OutputMap - Entry in the OUTPUT section of a typemap

SYNOPSIS

```
use ExtUtils::Typemaps;
...
my $output = $typemap->get_output_map('T_NV');
my $code = $output->code();
$output->code("...");
```

DESCRIPTION

Refer to ExtUtils::Typemaps for details.

METHODS

new

Requires `xstype` and `code` parameters.

code

Returns or sets the OUTPUT mapping code for this entry.

xstype

Returns the name of the XS type of the OUTPUT map.

cleaned_code

Returns a cleaned-up copy of the code to which certain transformations have been applied to make it more ANSI compliant.

targetable

This is an obscure but effective optimization that used to live in `ExtUtils::ParseXS` directly. Not implementing it should never result in incorrect use of typemaps, just less efficient code.

In a nutshell, this will check whether the output code involves calling `sv_setiv`, `sv_setuv`, `sv_setnv`, `sv_setpv` or `sv_setpvn` to set the special `$arg` placeholder to a new value **AT THE END OF THE OUTPUT CODE**. If that is the case, the code is eligible for using the `TARG`-related macros to optimize this. Thus the name of the method: `targetable`.

If this optimization is applicable, `ExtUtils::ParseXS` will emit a `dXSTARG;` definition at the start of the generate XSUB code, and type (see below) dependent code to set `TARG` and push it on the stack at the end of the generated XSUB code.

If the optimization can not be applied, this returns undef. If it can be applied, this method returns a hash reference containing the following information:

```
type: Any of the characters i, u, n, p
with_size: Bool indicating whether this is the sv_setpvn variant
what: The code that actually evaluates to the output scalar
what_size: If "with_size", this has the string length (as code,
not constant, including leading comma)
```

SEE ALSO

ExtUtils::Typemaps

AUTHOR

Steffen Mueller <smueller@cpan.org>

COPYRIGHT & LICENSE

Copyright 2009, 2010, 2011, 2012 Steffen Mueller

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.