NAME

Exporter::Declare::Meta − The meta object which stores meta−data for all exporters.

DESCRIPTION

All classes that use Exporter::Declare have an associated Meta object. Meta objects track available exports, tags, and options.

METHODS

`$class`−>new(`$package`)
: Created a meta object for the specified package. Also injects the *export_meta()* sub into the package namespace that returns the generated meta object.

`$class`−>new_from_exporter(`$package`)
: Create a meta object for a package that already uses Exporter.pm. This will not turn the class into an Exporter::Declare package, but it will create a meta object and *export_meta()* method on it. This si primarily used for reexport purposes.

`$package` = `$meta`−>*package()*
: Get the name of the package with which the meta object is associated.

`$meta`−>*add_alias()*
: Usually called at construction to add a package alias function to the exports.

`$meta`−>add_export(`$name`, `$ref`)
: Add an export, name should be the item name with sigil (assumed to be sub if there is no sigil). `$ref` should be a ref blessed as an Exporter::Declare::Export subclass.

`$meta`−>get_export(`$name`)
: Retrieve the Exporter::Declare::Export object by name. Name should be the item name with sigil, assumed to be sub when sigil is missing.

`$meta`−>export_tags_push(`$name`, `@items`)
: Add `@items` to the specified tag. Tag will be created if it does not already exist. `$name` should be the tag name **WITHOUT** −/: prefix.

`@list` = `$meta`−>export_tags_get(`$name`)
: Get the list of items associated with the specified tag. `$name` should be the tag name **WITHOUT** −/: prefix.

`@list` = `$meta`−>*export_tags_list()*
: Get a list of all export tags.

`$bool` = `$meta`−>is_tag(`$name`)
: Check if a tag with the given name exists. `$name` should be the tag name **WITHOUT** −/: prefix.

`$meta`−>options_add(`$name`)
: Add import options by name. These will be boolean options that take no arguments.

my `@list` = `$meta`−>*options_list()*

`$meta`−>arguments_add(`$name`)
: Add import options that slurp in the next argument as a value.

`$bool` = `$meta`−>is_option(`$name`)
: Check if the specified name is an option.

`$bool` = `$meta`−>is_argument(`$name`)
: Check if the specified name is an option that takes an argument.

`$meta`−>add_parser(`$name`, sub { ... })
: Add a parser sub that should be associated with exports via Devel::Declare

`$meta`−>get_parser(`$name`)
: Get a parser by name.

`$ref` = `$meta`−>get_ref_from_package(`$item`)
: Returns a reference to a specific package variable or sub.

`$meta`−>reexport(`$package`)
: Re-export the exports in the provided package. Package may be an Exporter::Declare based package or an Exporter based package.

`$meta->merge( $meta2 )`
Merge-in the exports and tags of the second meta object.

AUTHORS

Chad Granum exodist7 AT gmail DOT com

COPYRIGHT