

NAME

TAP::Object - Base class that provides common functionality to all "TAP::*" modules

VERSION

Version 3.36

SYNOPSIS

```
package TAP::Whatever;

use strict;

use base 'TAP::Object';

# new() implementation by TAP::Object
sub _initialize {
    my ( $self, @args ) = @_;
    # initialize your object
    return $self;
}

# ... later ...
my $obj = TAP::Whatever->new(@args);
```

DESCRIPTION

[TAP::Object](#) provides a default constructor and exception model for all TAP::* classes. Exceptions are raised using Carp.

METHODS**Class Methods**

new

Create a new object. Any arguments passed to *new* will be passed on to the “_initialize” method. Returns a new object.

Instance Methods

_initialize

Initializes a new object. This method is a stub by default, you should override it as appropriate.

Note: “new” expects you to return *\$self* or raise an exception. See “_croak”, and Carp.

_croak

Raise an exception using *croak* from Carp, eg:

```
$self->_croak( 'why me?', 'aaarrgh!' );
```

May also be called as a *class* method.

```
$class->_croak( 'this works too' );
```

_confess

Raise an exception using *confess* from Carp, eg:

```
$self->_confess( 'why me?', 'aaarrgh!' );
```

May also be called as a *class* method.

```
$class->_confess( 'this works too' );
```

_construct

Create a new instance of the specified class.

mk_methods

Create simple getter/setters.

```
__PACKAGE__->mk_methods(@method_names);
```