



```
#!/usr/bin/perl -w
# Copyright 2000-2002 Raphaël Hertzog, Fabien Fulhaber
# This file is distributed under the terms of the General Public License
# version 2 or (at your option) any later version.
use lib '..';
use strict;
use vars qw($db);
use GT::Prices;
use GT::Portfolio;
use GT::PortfolioManager;
use GT::Calculator;
use GT::Report;
use GT::BackTest;
use GT::Eval;
use Getopt::Long;
use GT::Conf;
use GT::DateTime;
use GT::Tools qw(:conf :timeframe);
use GT::Graphics::DataSource::PortfolioEvaluation;
use GT::Graphics::Driver;
use GT::Graphics::Object;
use GT::Graphics::Graphic;
use GT::Graphics::Tools qw(:axis :color);
use Pod::Usage;
GT::Conf::load();

# Manage options
my ($full, $nb_item, $start, $end, $timeframe, $max_loaded_items) =
(0, 0, "", "", 'day', -1);
my $man = 0;
my @options;
my ($verbose, $html, $display_trades, $template, $graph_file, $ofname, $broker, $system,
$store_file, $outputdir, $set) =
(0, 0, 0, "", "", "", "", "", "", "");
my (@mmname, @tfname, @csname);
$outputdir = GT::Conf::get("BackTest::Directory") || './';
```

```

GetOptions('full!' => \$full, 'nb-item=i' => \$nb_item,
"start=s" => \$start, "end=s" => \$end,
"max-loaded-items=s" => \$max_loaded_items,
"timeframe=s" => \$timeframe,
'verbose!' => \$verbose, 'html!' => \$html,
'template=s' => \$template, 'display-trades!' => \$display_trades,
'output-directory=s' => \$outputdir, 'set=s' => \$set,
'money-management=s' => \@mmname, 'graph=s' => \$graph_file,
'trade-filter=s' => \@tfname, 'order-factory=s' => \$ofname,
'close-strategy=s' => \@csname, 'broker=s' => \$broker,
'system=s' => \$system, "store=s" => \$store_file,
"option=s" => \@options, "help!" => \$man);
foreach (@options) {
my ($key, $value) = split (/=/, $_);
GT::Conf::set($key, $value);
}
pod2usage( -verbose => 2) if ($man);
if (! scalar(@mmname))
{
@mmname = ("Basic");
}
if ($system && ! scalar(@csname)) {
die "You must give at least one --close-strategy argument !\n";
}
if (! -d $outputdir)
{
die "The directory '$outputdir' doesn't exist !\n";
}
# Create the entire framework
my $pf_manager = GT::PortfolioManager->new;
my $sys_manager = GT::SystemManager->new;

if ($system) {
$sys_manager->set_system(
create_standard_object(split (/s+/, "Systems::$system")));
} else {
my $sysname = shift;
# Check for alias
if (! defined($sysname)) {

```

```
die "You must give either a --system parameter or an alias name.";
}
if ($sysname !~ /\|/)
{
my $alias = resolve_alias($sysname);
die "Alias unknown '$alias'" if (! $alias);
$sys_manager->set_alias_name($sysname);
$sysname = $alias;
}
if (defined($sysname) && $sysname)
{
$sys_manager->setup_from_name($sysname);
$pf_manager->setup_from_name($sysname);
} else {
die "You must give either a --system parameter or an alias name.";
}
}
```