
PhpProject Documentation

Release 0.12.0

The PhpProject Team

August 13, 2014

1	Introduction	3
1.1	Features	3
1.2	File formats	3
1.3	Contributing	4
2	Installing/configuring	5
2.1	Requirements	5
2.2	Installation	5
2.3	Using samples	6
3	General usage	7
3.1	Basic example	7
3.2	Document information	7
4	Task	9
4.1	Allocation	9
5	Resource	11
6	Recipes	13
6.1	Title	13
7	Frequently asked questions	15
8	Credits	17
9	References	19
9.1	GanttProject (GAN)	19
9.2	MSPProjectExchange (MPX)	19
10	Indices and tables	21

PHPProject

PHPProject is a library written in pure PHP that provides a set of classes to write to different project managements file formats, i.e. GanttProject (.gan) and MS-Project (.mpx). PHPProject is an open source project licensed under LGPL.

Introduction

PHPProject is a library written in pure PHP that provides a set of classes to write to different project management file formats, i.e. Microsoft *MSPProjectExchange* <<http://support.microsoft.com/kb/270139>> (.mpx) and *GanttProject* (.gan).

PHPProject is an open source project licensed under the terms of [LGPL version 3](#). PHPProject is aimed to be a high quality software product by incorporating [continuous integration](#) and [unit testing](#). You can learn more about PHPProject by reading this [Developers' Documentation](#) and the [API Documentation](#).

1.1 Features

- Create an in-memory project management representation
- Set file meta data (author, title, description, etc)
- Add resources from scratch or from existing one
- Add tasks from scratch or from existing one
- Output to different file formats: MSPProjectExchange (.mpx), GanttProject (.gan)
- ... and lots of other things!

1.2 File formats

Below are the supported features for each file formats.

1.2.1 Writers

Features		MPX	GAN
Document Properties	Standard		
	Custom		
Document Informations	Task		
Project	Resource		
	Allocation		

1.2.2 Readers

Features		MPX	GAN
Document Properties	Standard Custom		
Document Informations Project	Task Resource Allocation		

1.3 Contributing

We welcome everyone to contribute to PHPPProject. Below are some of the things that you can do to contribute:

- Read [our contributing guide](#)
- [Fork us](#) and [request a pull](#) to the `develop` branch
- Submit [bug reports](#) or [feature requests](#) to GitHub
- Follow [@PHPOffice](#) on Twitter

Installing/configuring

2.1 Requirements

Mandatory:

- PHP 5.3+
- PHP [XML Parser](#) extension

Optional PHP extensions:

- [XMLWriter](#)

2.2 Installation

There are two ways to install PHPProject, i.e. via [Composer](#) or manually by downloading the library.

2.2.1 Using Composer

To install via Composer, add the following lines to your `composer.json`:

```
{
    "require": {
        "phpoffice/phpproject": "dev-master"
    }
}
```

2.2.2 Manual install

To install manually, [download PHPProject package from github](#). Extract the package and put the contents to your machine. To use the library, include `src/PHPProject/Autoloader.php` in your script and invoke `Autoloader::register`.

```
require_once '/path/to/src/PhpProject/Autoloader.php';
\PhpOffice\PhpProject\Autoloader::register();
```

2.3 Using samples

After installation, you can browse and use the samples that we've provided, either by command line or using browser. If you can access your PHPProject library folder using browser, point your browser to the `samples` folder, e.g. `http://localhost/PhpProject/samples/`.

General usage

3.1 Basic example

The following is a basic example of the PHPProject library. More examples are provided in the [samples folder](#).

```
require_once 'src/PhpProject/Autoloader.php';
\PhpOffice\PhpProject\Autoloader::register();

$objPHPProject = new PhpProject();

// Create resource
$objRes1 = $objPHPProject->createResource();
$objRes1->setTitle('UserBoy');

// Create a task
$objTask1 = $objPHPProject->createTask();
$objTask1->setName('Start of the project');
$objTask1->setStartDate('02-01-2012');
$objTask1->setEndDate('03-01-2012');
$objTask1->setProgress(0.5);
$objTask1->addResource($objRes1);

$writerGAN = IOFactory::createWriter($objPHPProject, 'GanttProject');
$writerGAN->save(__DIR__ . "/sample.gan");
```

3.2 Document information

You can set the document information such as title, creator, and company name. Use the following functions:

```
$properties = $objPHPProject->getProperties();
$properties->setCreator('My name');
$properties->setCompany('My factory');
$properties->setTitle('My title');
$properties->setDescription('My description');
$properties->setCategory('My category');
$properties->setLastModifiedBy('My name');
$properties->setCreated(mktime(0, 0, 0, 3, 12, 2014));
$properties->setModified(mktime(0, 0, 0, 3, 14, 2014));
$properties->setSubject('My subject');
$properties->setKeywords('my, key, word');
```

Task

A task is an activity that needs to be accomplished.

- name
- duration in days
- progress in percent
- date start
- date end

Example:

4.1 Allocation

For each task, a resource can be assigned.

Resource

Resource is required to carry out a project task. It can be people or equipment.

- title

Example:

Recipes

6.1 Title

Recipe Text

`$content;`

Frequently asked questions

Credits

References

9.1 GanttProject (GAN)

- [Website](#)

9.2 MSProjectExchange (MPX)

- [MSDN : Description of the MPX Project File Exchange Format](#)

Indices and tables

- *genindex*
- *modindex*
- *search*