

PHP-FF

Beta 0.4

The file *phpff.php5* and its dependencies have been created for PHP version 5.x. They contain a class named **FFHashMap** and another one named **FFTable (with FFTableRow)**.

Summary

FFHashMap	2
FFTable	3

FFHashMap

The **FFHashMap** class must be instantiated with one parameter, the file name (absolute or relative access path).

For exemple:

```
$ff = new FFHashMap("myfile.dat");
```

Then you can use some functions on the **FFHashMap** class to manage your data.

Functions list:

- `Set($key, $value)` : Create or update a `$value` in the database using a `$key`.
- `Remove($key)` : Remove a value from the database by giving its `$key`.
- `Clear()` : Clear all values from the database.
- `ContainsKey($key)` : Returns true if the given `$key` exists, false else.
- `GetKeys()` : Get all keys.
- `GetValue($key)` : Get a value by giving its associated `$key`.
- `Save()` : Save all data to the file used to create the FlatFile object.

For exemple:

```
$hm = new FFHashMap("myfile.dat");  
$hm->Set("toto", 5);  
$hm->Set("titi", 9);  
$hm->Set("tata", 6);  
$hm->Remove("tata");  
$hm->Save();
```

The file *myfile.dat* will now contain 2 lines:

```
toto=5  
titi=9
```

All functions manage data into memory and not on disk. So you can do all your operations on data and save them at the end.

FFTable

The **FFTable** class is used to store data table into flat file. Values in **FFTable** can contain “|” char or “\n” special char.

For example, the data stored into a **FFTable** generated file, look like this:

```
ID|Name|HireDate|ManagerID|Comment
1|Greg|12/05/2001||The boss#[phpff-LF]on 2 lines#[phpff-LF]or more...
2|Charly|14/04/09 12:34:12|1|
4|Jean-Jacques|14/01/2008|2|
```

To create an **FFTable** object, you must specify on first constructor parameter, the filename:

```
$tf = new FFTable("mytable.dat");
```

To create columns:

```
$tf->Columns = array("ID", "Name", "HireDate", "ManagerID",
"Comment");
```

The create an **FFTableRow**:

```
$rtm = $tf->CreateRow(array(2, "Charly", "13/07/2002", 1));
```

To modify a value in a row:

```
$rtm->Cells[2] = date("d/m/y h:i:s");
```

Or

```
$rtm->SetValue("HireDate", date("d/m/y h:i:s"));
```

Or

```
$tf->SetValue(0, 2, date("d/m/y h:i:s"));
// This statement will set the value of row at index 0, and column
at index 2 to date("d/m/y h:i:s")
```

To get a value from the table:

```
$value = $tf->GetValue(0, 2)
```

Or

```
$value = $rtm->GetValue("HireDate");
```

To save in-memory data to the file used to construct the **FFTable** object:

```
$tf->Save();
```

Or

```
$tf->SaveAs("mytable.csv", FFTable::CSV);
```

To dump the **FFTable** content to a simple <table> html object:

```
$tf->DumpToHtmlTable();
```

To clear columns and rows from the **FFTable**:

```
$tf->Clear();
```

To clear only rows:

```
$tf->ClearRows();
```

To get a column index from its name:

```
$col_index = $tf->GetColumnIndex("HireDate");
```

To delete one or more rows with condition:

```
$tf->DeleteRowsWhere("HireDate", "13/07/2002");
```

To get a row with condition:

```
$row = $tf->SelectRowWhere("HireDate", "13/07/2002");
```

To get one or more rows with condition:

PHP-FF v0.4

```
$rows = $tf->SelectRowsWhere("HireDate", "13/07/2002");  
foreach ($rows as $row)  
    ...
```