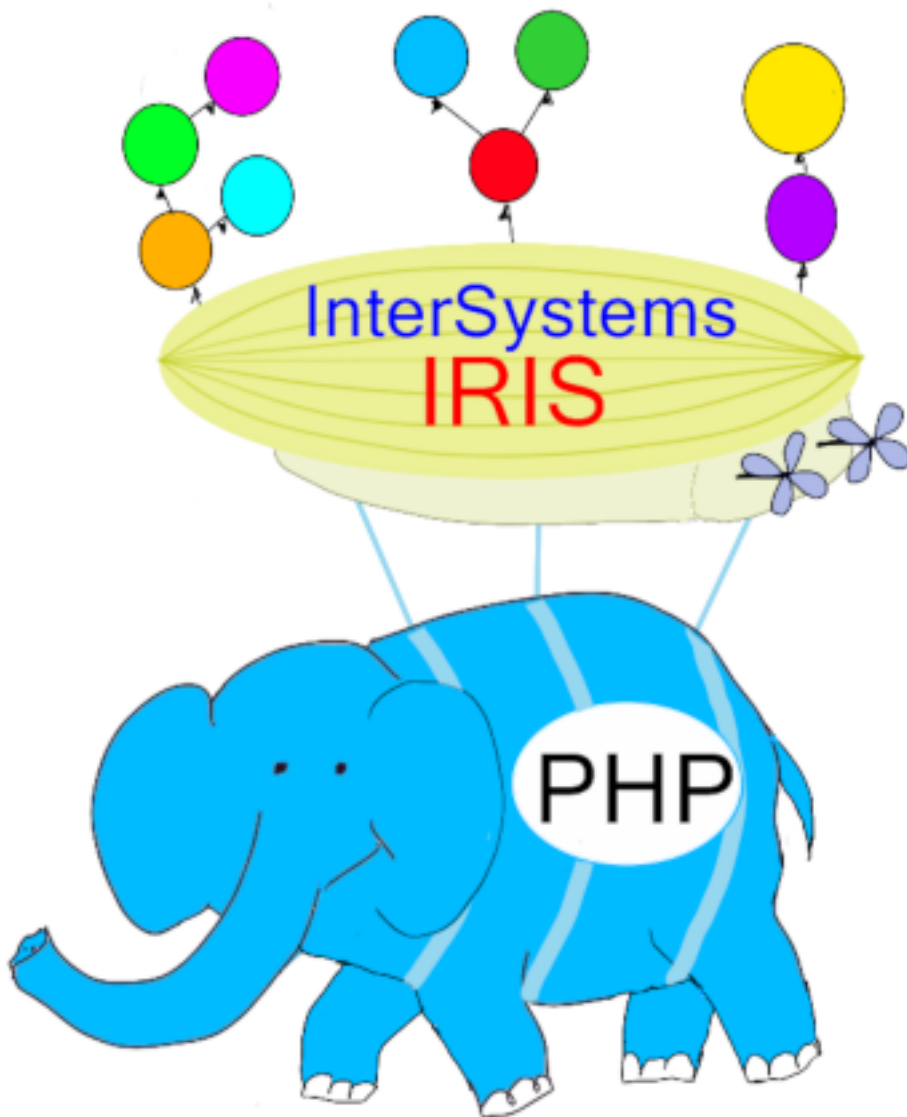


PHP module for working with hierarchical data in InterSystems IRIS



[Sergey Kamenev](#) 23 May 2019

[API](#), [Cloud](#), [Code Snippet](#), [Development Environment](#), [Key Value](#), [Testing](#), [SOAP](#), [Caché](#), [Documentation](#), [InterSystems IRIS](#), [InterSystems IRIS for Health](#), [Open Exchange](#), [Other](#)



PHP, from the beginning of its time, is renowned (and criticized) for supporting integration with a lot of libraries, as well as with almost all the DB existing on the market. However, for some mysterious reasons, it did not support hierarchical databases on the globals.

Globals are structures for storing hierarchical information. They are somewhat similar to key-value database with the only difference being that the key can be multi-level:

```
Set ^inn("1234567890", "city") = "Moscow"  
Set ^inn("1234567890", "city", "street") = "Req Square"  
Set ^inn("1234567890", "city", "street", "house") = 1
```

```
Set ^inn("1234567890", "year") = 1970
Set ^inn("1234567890", "name", "first") = "Vladimir"
Set ^inn("1234567890", "name", "last") = "Ivanov"
```

In this example, multi-level information is saved in the global **^inn** using the built-in ObjectScript language. Global **^inn** is stored on the hard drive (this is indicated by the “^” sign in beginning).

In order to work with globals from PHP, we will need new functions that will be added by the PHP module, which will be discussed below.

Globals support many functions for working with hierarchies: traversal tree on fixed level and in depth, deleting, copying and pasting entire trees and individual nodes. And also ACID transactions - as is done in any quality database. All this happens extremely quickly (about 10^5 - 10^6 inserts per second on regular PC) for two reasons:

1. Globals are a lower level abstraction when compared to SQL,
2. The bases have been in production on the globals for decades, and during this time they were polished and their code was thoroughly optimized.

Learn more about globals in the series of articles titled "Globals Are Magic Swords For Managing Data.":

[Part 1.](#)

[Trees. Part 2.](#)

[Sparse arrays. Part 3.](#)

In this world, globals are primarily used in storage systems for unstructured and sparse information, such as: medical, personal data, banking, etc.

I love PHP (and I use it in my development work), and I wanted to play around with globals. There was no PHP module for IRIS and Caché. I contacted InterSystems and asked them to create it. InterSystems sponsored the development as part of an educational grant and my graduate student and I created the module.

Generally speaking, InterSystems IRIS is a multi-model DBMS, and that's why from PHP you can work with it via ODBC using SQL, but I was interested in globals, and there was no such connector.

So, the module is available for PHP 7.x (was tested for 7.0-7.2). Currently it can only work with InterSystems IRIS and Caché installed on the same host.

[Module page on OpenExchange](#) (a directory of projects and add-ons for developers at InterSystems IRIS and Caché).

There is a useful DISCUSS section where people share their related experiences.

Download here:

https://github.com/intersystems-community/php_ext_iris

Download the repository from the command line:

```
git clone https://github.com/intersystems-community/php_ext_iris
```

[Installation instructions for the module in English and Russian.](#)

PHP module for working with hierarchical data in InterSystems IRIS

Published on InterSystems Developer Community (<https://community.intersystems.com>)

Module Functions:

PHP function	Description
Working with data	
<code>iris_set(\$node, value)</code>	Setting a node value.
<code>iris_get(\$node)</code>	Getting a node value.
<code>iris_zkill(\$node)</code>	Deleting a node value.
<code>iris_kill(\$node)</code>	Deleting a node and all descendant branches.
<code>iris_order(\$node)</code>	Traversal the branches of the global on a given level
<code>iris_order_rev(\$node)</code>	Traversal the branches of the global on a given level in reverse order
<code>iris_query(\$CmdLine)</code>	Traversal of the global in depth
Service functions	
<code>iris_set_dir(\$FullPath)</code>	Setting up a directory with a database
<code>iris_exec(\$CmdLine)</code>	Execute database command
<code>iris_connect(\$login, \$pass)</code>	Connect to database
<code>iris_quit()</code>	Close connection with DB
<code>iris_errno()</code>	Get error code
<code>iris_error()</code>	Get text description of error

If you want to play around with the module, check e.g. docker container implementation

Especially for DC and those who wants to use we run a virtual machine with a Caché php-module was set up.

For self-installation of the module for InterSystems Caché

Just out of interest, I ran two primitive tests to check the speed of inserting new values into the database in the docker container on my PC (AMD FX-9370@4700Mhz 32GB, LVM, SATA SSD).

- Insertion of 1 million new nodes into the global took 1.81 seconds or 552K inserts per second.
- Updating a value in the same global 1,000,000 times took 1.98 seconds or 505K updates per second. An interesting fact is that the insertion occurs faster than the update. Apparently this is a consequence of the initial optimization of the database aimed at quick insertion.

Obviously, these tests cannot be considered 100% accurate or useful, since they are primitive and are done in the container. On more powerful hardware with a disk system on a PCIe SSD, tens of millions of inserts per second can be achieved.

What else can be completed and the current state

1. Useful functions for working with transactions can be added (you can still use them with `iris_exec`).
2. The function of returning the whole global structure is not implemented, so as not to traverse the global from PHP.
3. The function of saving a PHP array as a subtree is not implemented.
4. Access to local database variables is not implemented. Only using `iris_exec`, although it's better with `iris_set`.
5. Global traversal in depth in the opposite direction is not implemented.
6. Access to the database via an object using methods (similar to current functions) is not implemented.

The current module is not quite yet ready for production: not tested for high loads and memory leaks. However, should someone need it, please feel free to contact me at any time (Sergey Kamenev sukamenev@gmail.com).

Bottom line

For a long time, the worlds of PHP and hierarchical databases on globals practically did not overlap, although globals provide strong and fast functionality for specific data types (medical, personal).

I hope that this module will motivate PHP programmers to experiment with globals and ObjectScript programmers for simple development of web interfaces in PHP.

P.S. Thank you for your time!

[Check the related application on InterSystems Open Exchange.](#)

- + 2
- 4
- 2
- 354
- 1



[Sergey Kamenev](#) 23 May 2019

Reply

Source URL: <https://community.intersystems.com/post/php-module-working-hierarchical-data-intersystems-iris>