# Objeck Programming Language

v3.3.8-2 (FCGI); September 21, 2015

## Release notes

Minor updates and bug fixes.

Highlights of this release include the following:

1. Added support for bodiless ‘while’ loops **(new)**
2. Addressed an issue with Vector copy constructors **(bug/high)**
3. Fixed a bug related to unary conditional expressions in loops **(bug/med)**

Please refer to the [project website](http://www.objeck.org/) for additional information about the language. The Programmer’s Guide is located the “doc” directory. Additional examples are available on the [Rosetta Code website](http://rosettacode.org/wiki/Category:Objeck). Please submit bugs to [objeck@gmail.com, bug reports are greatly appreciated!!](mailto:objeck@gmail.com)

## Compiling and executing programs

In order to compile programs outside of the “bin” directory the “OBJECK\_LIB\_PATH” environment variable must be set. When this variable is set all library files must be in the directory specified.

To manually setting the environment paths in Windows:

1. set OBJECK\_LIB\_PATH=C:\Documents and Settings\Administrator\objeck-lang\bin
2. set PATH=%PATH%;C:\Documents and Settings\Administrator\ objeck-lang\bin

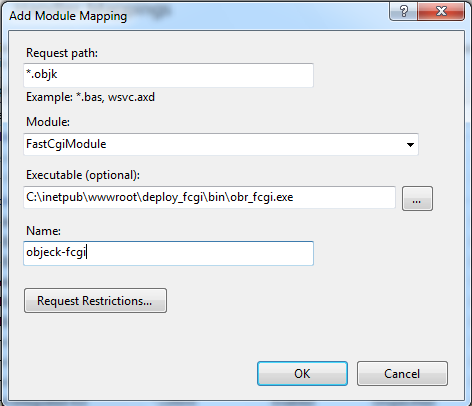
Comping a simple web program:

* Ensure you’ve download the [compiler](https://sourceforge.net/projects/objeck-lang/files/binaries/).
* obc -src ..\examples\hello\_fcgi.obs -lib fcgi.obl,collect.obl -tar web -dest ..\hello\_fcgi.obw

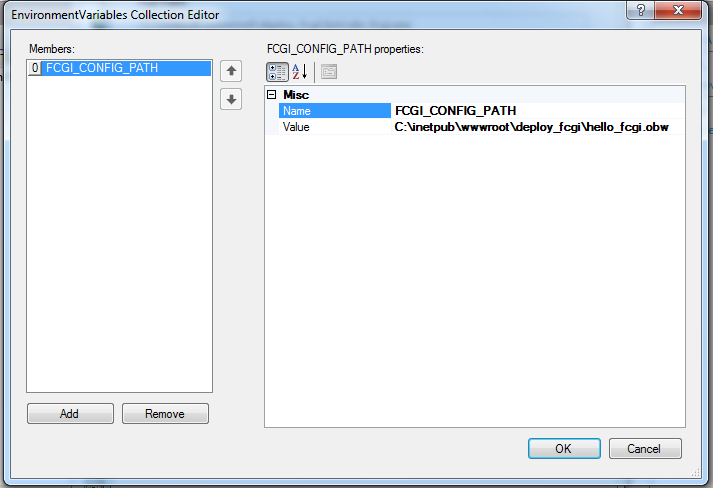
### FastCGI for Windows IIS:

Steps:

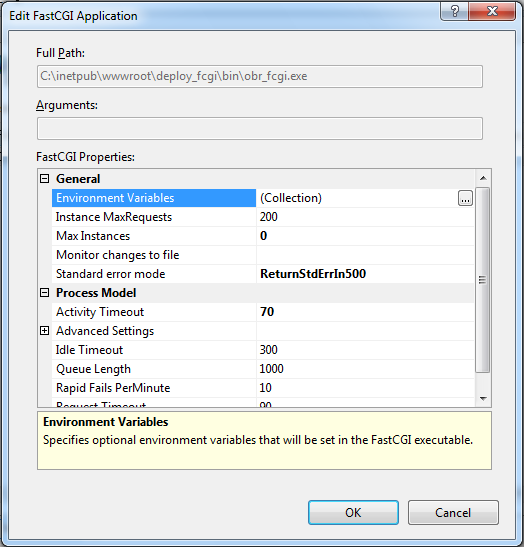
* Install the following IIS modules:
  1. CGI
  2. ISAPI Extensions
  3. ISAPI Filters
  4. IIS Management Console
* Copy the Objeck “deploy\_fcgi” directory to IIS “wwwroot”
* Copy \*.obw to wwwroot\deploy\_fcgi
* Copy or create wwwroot\index.objk
* Navigate to the “default website” and create a module mapping
  1. Adjust the “Request Restrictions”
  2. Mapping: Invoke handler file, check “file”
  3. Verbs: All verbs
  4. Access: None



* Next, go to top home node and select FastCGI Settings
* Select “obr\_fcgi.exe” and add environment variables (see below)



* Set FCGI\_CONFIG\_PATH property to the location of the web app \*.obw



* Restart IIS

### FastCGI for Ubuntu using Apache:

Steps:

* Install Apache and FastCGI:
* sudo apt-get install libapache2-mod-fastcgi apache2-MPM-worker libfcgi-dev uuid-dev
* Add the entry below to the: “/etc/apache2/mods-enabled/fastcgi.conf” file:

|  |
| --- |
| <IfModule mod\_fastcgi.c>  AddHandler objeck-lang .objk  FastCgiServer /var/www/html/deploy\_fcgi/bin/obr\_fcgi -initial-env FCGI\_CONFIG\_PATH=/var/www/html/deploy\_fcgi/hello.obw -idle-timeout 60 -processes 1  ScriptAlias /objk "/var/www/html/deploy\_fcgi/bin/obr\_fcgi"  </IfModule> |

### Tips

Few tips to make FastCGI easier to work with:

* Use the “Sytem.IO.File.Logger” class for debugging
* If you’d like your web app to connect to a database using ODBC setup a 32-bit ODBC data source as admin.

For OS X (10.9 or greater), you’ll need to install OpenSSL in order to use encryption APIs as well as ODBC to enable database support. Please refer to the following [link](http://mac-dev-env.patrickbougie.com/openssl) for OpenSSL directions and this [link](http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/ODBCMacOSX) for about ODBC support.

## Source code

The Objeck source repository is located [here](http://github.com/objeck/objeck-lang) and the source can be fetch using the following command GIT command:

* mkdir objeck
* cd objeck
* git init
* git clone https://github.com/objeck/objeck-lang.git

## Known bugs and limitations

* On Windows, the ODBC driver must be a 32-bit driver for [compatibility](http://stackoverflow.com/questions/1134827/how-do-i-use-a-32-bit-odbc-driver-on-64-bit-server-2008-when-the-installer-doesn).
* Some Redhat based Linux distributions may report errors about heap buffer execution. This is triggered by the JIT compiler’s execution of generated machine code. To resolve this issue please consult the Fedora error console.
* The XML parser class does not support DTDs or DOCTYPE tags
* By design, the regular expression class performs greedy pattern matching.
* All unit tests executed successfully but I'm sure there are some bugs. Send an [e-mail](mailto:objeck@gmail.com) we’ll get them sorted.