



EUPHORIA

programming language

tutorialspoint

SIMPLYEASYLEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About Tutorial

This tutorial gives you basic understanding of Euphoria programming language. Euphoria is simple, flexible, easy to learn, and interpreted high-level programming language for DOS, Windows, Linux, FreeBSD, and more. This tutorial describes everything a programmer needs to know such as its environment, data types, syntax and operators, file handling, and controlling the flow of program.

Audience

This tutorial is designed for the aspiring students who are keen to learn and understand Euphoria in detail. This tutorial would be of great help for the IT professionals working as programmers. The enthusiastic readers can access this tutorial as a source of additional reading.

Prerequisites

Before proceeding with this tutorial, you need to have a basic knowledge of working on Windows or Linux. You need to be familiar with any programming language such as C, C++. You need to have sound understanding of operating system, memory allocation and de-allocation, and basics of efficient programming and debugging.

Disclaimer & Copyright

© Copyright 2015 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher. We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com.



Table of Contents

| | |
|--|-----------|
| About Tutorial..... | |
| Audience..... | i |
| Prerequisites..... | i |
| Disclaimer & Copyright | i |
| Table of Contents..... | ii |
| 1. EUPHORIA OVERVIEW..... | 1 |
| Euphoria Features..... | 1 |
| Platform Requirements..... | 2 |
| Euphoria Limitations..... | 2 |
| Euphoria Licensing | 2 |
| 2. EUPHORIA ENVIRONMENT..... | 4 |
| Linux, Free BSD Installation..... | 4 |
| WIN32 and DOS Installation..... | 6 |
| Euphoria Interpreters | 7 |
| 3. BASIC SYNTAX | 8 |
| First Euphoria Program | 8 |
| Euphoria Identifiers | 8 |
| Reserved Words..... | 9 |
| Expressions | 10 |
| Blocks of Code | 10 |
| Multi-Line Statements | 10 |
| Escape Characters | 11 |
| Comments in Euphoria..... | 11 |

| | |
|--|----|
| 4. EUPHORIA VARIABLES..... | 13 |
| Variable Declaration | 13 |
| Assigning Values | 13 |
| Identifier Scope..... | 14 |
| 5. EUPHORIA CONSTANTS..... | 16 |
| The <i>enums</i> | 17 |
| 6. EUPHORIA DATATYPES..... | 19 |
| Euphoria Integers..... | 19 |
| Euphoria Atoms | 19 |
| Euphoria Sequences..... | 20 |
| Euphoria Objects..... | 21 |
| 7. EUPHORIA OPERATORS..... | 22 |
| The Arithmetic Operators | 22 |
| The Relational Operators | 23 |
| The Logical Operators | 25 |
| The Assignment Operators..... | 27 |
| Miscellaneous Operators | 29 |
| Precedence of Euphoria Operators | 30 |
| 8. EUPHORIA BRANCHING..... | 32 |
| The <i>if</i> Statement | 32 |
| The <i>if...else</i> Statement | 33 |
| The <i>if...elsif...else</i> Statement..... | 34 |
| The <i>if...label...then</i> Statement | 35 |
| Nested <i>if...else</i> Statement..... | 36 |
| The <i>switch</i> Statement | 38 |

| | |
|--|----|
| The <i>switch...with fallthru</i> Statement..... | 39 |
| The <i>switch...label</i> Statement | 41 |
| The <i>ifdef</i> Statement | 43 |
| The <i>ifdef...elsedef</i> Statement | 45 |
| The <i>ifdef...elsifdef</i> Statement..... | 46 |
| 9. EUPHORIA LOOPS | 48 |
| While Statement..... | 49 |
| The <i>while...with entry</i> Statement | 50 |
| The <i>while...label</i> Statement..... | 51 |
| The <i>loop until</i> Statement | 53 |
| The <i>loop...with entry</i> Statement..... | 54 |
| The <i>loop...label</i> Statement | 55 |
| The <i>for</i> Loop..... | 57 |
| 10. EUPHORIA FLOW CONTROL | 60 |
| The <i>exit</i> Statement..... | 60 |
| The <i>break</i> Statement | 61 |
| The <i>continue</i> Statement..... | 63 |
| The <i>retry</i> Statement..... | 64 |
| The <i>goto</i> Statement | 65 |
| 11. SHORT CIRCUIT EVALUATION..... | 67 |
| 12. EUPHORIA SEQUENCES..... | 69 |
| Character String | 70 |
| String Arrays | 71 |
| Euphoria Structures | 71 |
| Unary Operation..... | 72 |

| | |
|--|----|
| Arithmetic Operations | 72 |
| Command Line Options..... | 74 |
| 13. EUPHORIA DATE AND TIME..... | 75 |
| The <i>date()</i> Method..... | 75 |
| The <i>time()</i> Method..... | 77 |
| Date & Time Related Methods..... | 78 |
| 14. EUPHORIA PROCEDURES..... | 79 |
| Procedure Definition..... | 79 |
| Calling a Procedure | 80 |
| Procedure Parameters | 80 |
| 15. EUPHORIA FUNCTIONS | 82 |
| Function Definition | 82 |
| Calling a Function..... | 82 |
| Function Parameters..... | 83 |
| The <i>return</i> Statement | 84 |
| 16. EUPHORIA FILE I/O..... | 85 |
| Displaying on the Screen..... | 85 |
| Opening and Closing Files | 85 |
| The <i>open</i> Method | 86 |
| The <i>close()</i> Method..... | 87 |
| Reading and Writing Files | 89 |
| The <i>printf()</i> Method..... | 89 |
| The <i>gets()</i> Method..... | 90 |
| File & Directory Related Methods | 92 |

1. EUPHORIA OVERVIEW

Euphoria stands for **E**nd-**U**ser **P**rogramming with **H**ierarchical **O**bjects for **R**obust **I**nterpreted **A**pplications. Euphoria's first incarnation was created by Robert Craig on an Atari Mega-ST and it was first released in 1993. It is now maintained by Rapid Deployment Software.

It is a free, simple, flexible, easy to learn, and interpreted but extremely fast 32-bit high-level programming language for DOS, Windows, Linux, FreeBSD and more.

Euphoria is being used to develop Windows GUI programs, high-speed DOS games, and Linux/FreeBSD X Windows programs. Euphoria can also be used for CGI (Web-based) programming.

Euphoria Features

Here is the list of major features of Euphoria:

- It is a simple, flexible, powerful language definition that is easy to learn and use.
-
- It supports dynamic storage allocation which means variables grow or shrink without the programmer having to worry about allocating and freeing the memory. It takes care of garbage collection automatically.
-
- It is extremely faster than conventional interpreters such as Perl and Python.
-
- Euphoria programs run under Linux, FreeBSD, 32-bit Windows, and any DOS environment.
-
- Euphoria programs are not subject to any 640K memory limitations.
-
- It provides an optimizing Euphoria-To-C translator which you can use to translate your Euphoria program into C and then compile it with a C compiler to get an executable (.exe) file. This can boost your program speed by 2 to 5 times.
-
- Underlying hardware are completely hidden which means programs are not aware of word-lengths, underlying bit-level representation of values, byte-order etc.
-



- Euphoria installation comes along with a full-screen source debugger, an execution profiler, and a full-screen multi-file editor.
-
- It supports run-time error-handling, subscript, and type checking.
- It is an open source language and comes completely free of cost.

Platform Requirements

Euphoria is available on Windows, Linux, FreeBSD, and OSX. Here is the bare minimum version required with the following platforms –

- **WIN32 version:** You need Windows 95 or any later version of Windows. It runs fine on XP and Vista.
-
- **Linux version:** You need any reasonably up-to-date Linux distribution, that has libc6 or later. For example, Red Hat 5.2 or later works fine.
-
- **FreeBSD version:** You need any reasonably up-to-date FreeBSD distribution.
-
- **Mac OS X version:** You need any reasonably up-to-date Intel based Mac.

Euphoria Limitations

- Here are some prominent limitations of Euphoria:
-
- Even though Euphoria is simple, fast, and flexible enough for the programmers; it does not provide call support for many important functionalities. For example, network programming.
-
- Euphoria was invented in 1993, and still you would not find any book written on this language. There is also not much documentation available for the language.

But these days, the language is getting popular very fast and you can hope to have nice utilities and books available for the language very soon.

Euphoria Licensing

This product is free and open source, and has benefited from the contributions of many people. You have complete royalty-free rights to distribute any Euphoria programs that you develop.



Euphoria

Icon files, such as euphoria.ico and binaries available in euphoria\bin, may be distributed with or without your changes.

You can **shroud** or **bind** your program and distribute the resulting files royalty-free. Some additional 3rd party legal restrictions might apply when you use the Euphoria-To-C translator.

The generous [Open Source License](#) allows Euphoria to use for both personal and commercial purposes. Unlike many other open source licenses, your changes do not have to be made open source.



2. EUPHORIA ENVIRONMENT

This chapter describes about the installation of Euphoria on various platforms. You can follow the steps to install Euphoria on Linux, FreeBSD, and 32-bit Windows. So you can choose the steps based on your working environment.

Linux, Free BSD Installation

Official website provides **.tar.gz** file to install Euphoria on your Linux or BSD OS. You can download your latest version of Euphoria from its official website [Download Euphoria](#).

Once you have .tar.gz file, here are three simple steps to be performed to install Euphoria on your Linux or Free BSD machine:

Step 1: Installing Files

Untar the downloaded file **euphoria-4.0b2.tar.gz** in a directory where you want to install Euphoria. If you want to install it in /home directory as follows, then:

```
$cp euphoria-4.0b2.tar.gz /home
$cd /home
$gunzip euphoria-4.0b2.tar.gz
$tar -xvf euphoria-4.0b2.tar
```

This creates a directory hierarchy inside **/home/euphoria-4.0b2** directory as follows:

```
$ls -l
-rw-r--r--  1 1001 1001 2485 Aug 17 06:15 Jamfile
-rw-r--r--  1 1001 1001 5172 Aug 20 12:37 Jamrules
-rw-r--r--  1 1001 1001 1185 Aug 13 06:21 License.txt
drwxr-xr-x  2 1001 1001 4096 Aug 31 10:07 bin
drwxr-xr-x  7 1001 1001 4096 Aug 31 10:07 demo
```



```
-rw-r--r--  1 1001 1001  366 Mar 18 09:02 file_id.diz
drwxr-xr-x  4 1001 1001 4096 Aug 31 10:07 include
-rw-r--r--  1 1001 1001 1161 Mar 18 09:02 installu.doc
drwxr-xr-x  4 1001 1001 4096 Aug 31 10:07 source
drwxr-xr-x 19 1001 1001 4096 Sep  7 12:09 tests
drwxr-xr-x  2 1001 1001 4096 Aug 31 10:07 tutorial
```

Note: File name euphoria-4.0b2.tar.gz depends on latest version available. We are using 4.0b2 version of the language for this tutorial.

Step 2: Setting Up the Path

After installing Euphoria, you need to set proper paths so that your shell can find required Euphoria binaries and utilities. Before proceeding, there are following three important environment variables you need to set up:

1. Set PATH environment variable to point /home/euphoria-4.0b2/bin directory.
2. Set EUDIR environment variable to point to /home/euphoria-4.0b2.
3. Set EUINC environment variable to point to /home/euphoria-4.0b2/include.

These variables can be set as follows –

```
$export PATH=$PATH:/home/euphoria-4.0b2/bin
$export EUDIR=/home/euphoria-4.0b2
$export EUINC=/home/euphoria-4.0b2/include
```

Note: The above commands used to set environment variables may differ depending on your Shell. We used *bash* shell for executing these commands to set the variables.

Step 3: Confirming Installation

Confirm if you installed Euphoria successfully or not.

Execute the following command:

```
$eui -version
```

If you get following result, then it means you have installed Euphoria successfully; otherwise you have to go back and check all the steps again.



```
$eui -version  
  
Euphoria Interpreter 4.0.0 beta 2 (r2670) for Linux  
  
Using System Memory  
  
$
```

That is it, Euphoria Programming Environment is ready on your UNIX machine, and you can start writing complex programs in easy steps.

WIN32 and DOS Installation

Official website provides **.exe** file to install Euphoria on your WIN32 or DOS OS. You can download your latest version of Euphoria from its official website [Download Euphoria](#).

Once you have .exe file, here are three simple steps to follow for installing Euphoria Programming language on your WIN32 or DOS machine:

Step 1: Installing Files

Double click on the downloaded **.exe** setup program to install all the files. We downloaded euphoria-40b2.exe file for installation.

The filename euphoria-40b2.exe depends on latest version available. We use version 4 beta 2 of the language.

By default Euphoria would be installed in *C:\euphoria-40b2* directory but you can also select a desired location.

Step 2: Rebooting the Machine

Re-boot your machine to complete the installation.

Step 3: Confirming Installation

Confirm if you installed Euphoria successfully or not.

Execute the following command:

```
c:\>eui -version
```

If you get following result, then it means you have installed Euphoria successfully; otherwise you have to go back and check all the steps again.



```
c:\>eui -version  
  
Euphoria Interpreter 4.0.0 beta 2 (r2670) for Windows  
  
Using Managed Memory  
  
c:\>
```

That is it, Euphoria Programming Environment is ready on your WIN32 machine, and you can start writing complex programs in easy steps.

Euphoria Interpreters

- Depending on the platform you are using, Euphoria has multiple interpreters:
- The main interpreter is **eui**.
- On windows platforms, you have two choices. If you run **eui** then a console window is created. If you run **euiw** then no console is created, making it suitable for GUI applications.
-
- Euphoria does not care about your choice of file extensions. By convention however; the console-based applications come with **.ex** extension.
-
- GUI-based applications have **.exw** extension and the include files have **.e** extension.

3. BASIC SYNTAX

The Euphoria language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages. This chapter is designed to quickly get you up to speed on the syntax that is expected in Euphoria.

This tutorial assumes you are working with Linux and all the examples have been written on Linux platform. But it is observed that there is no any prominent difference in program syntax on Linux and WIN32. Hence you can follow the same steps on WIN32.

First Euphoria Program

Let us write a simple Euphoria program in a script. Type the following source code in test.ex file and save it.

```
#!/home/euphoria-4.0b2/bin/eui  
  
puts(1, "Hello, Euphoria!\n")
```

Let us say, Euphoria interpreter is available in */home/euphoria-4.0b2/bin/* directory. Now run this program as follows:

```
$ chmod +x test.ex    # This is to make file executable  
  
$ ./test.ex
```

This would produce the following result:

```
Hello, Euphoria!
```

This script used a built-in function **puts()** which takes two arguments. First argument indicates file name or device number, and second argument indicates a string which you want to print. Here 1 indicates STDOUT device.



Euphoria Identifiers

A Euphoria identifier is a name used to identify a variable, function, class, module, or other object. An identifier starts with a letter A to Z or a to z and then followed by letters, digits, or underscores.

Euphoria does not allow punctuation characters such as @, \$, and % within identifiers.

Euphoria is a case sensitive programming language. Thus **Manpower** and **manpower** are two different identifiers in Euphoria. For example, the valid identifiers are:

- n
- color26
- ShellSort
- quick_sort
- a_very_long_identifier

Reserved Words

The following list shows the reserved words in Euphoria. These reserved words may not be used as constant or variable or any other identifier names. Euphoria keywords contain lowercase letters only.

| | | |
|----------|----------|-----------|
| and | exit | override |
| as | export | procedure |
| break | fallthru | public |
| by | for | retry |
| case | function | return |
| constant | global | routine |

| | | |
|-----------|-----------|---------|
| continue | goto | switch |
| do | if | then |
| else | ifdef | to |
| elseif | include | type |
| elseifdef | label | until |
| end | loop | while |
| entry | namespace | with |
| enum | not | without |
| | or | xor |

Expressions

Euphoria lets you calculate results by forming expressions. However, in Euphoria you can perform calculations on entire sequences of data with one expression.

You can handle a sequence much as you would handle a single number. It can be copied, passed to a subroutine, or calculated upon as a unit. For example:

```
{1,2,3} + 5
```

This is an expression that adds the sequence {1, 2, 3} and the atom 5 to get the resulting sequence {6, 7, 8}. You would learn sequences in subsequent chapters.

Blocks of Code

One of the first caveats programmers encounter when learning Euphoria is the fact that there are no braces to indicate blocks of code for procedure and function definitions or flow control. Blocks of code are denoted by associated keywords.



The following example shows **if...then...end if** block:

```
if condition then
    code block comes here
end if
```

Multi-Line Statements

Statements in Euphoria typically end with a new line. Euphoria does however, allow to write a single statement in multiple lines. For example:

```
total = item_one +
        item_two +
        item_three
```

Escape Characters

Escape characters may be entered using a back-slash. For example:

The following table is a list of escape or non-printable characters that can be represented with backslash notation.

| Backslash notation | Description |
|--------------------|-----------------|
| \n | Newline |
| \r | Carriage return |
| \t | Tab |
| \\ | Backslash |
| \" | Double quote |
| \' | Single quote |

Comments in Euphoria

Any comments are ignored by the compiler and have no effect on execution speed. It is advisable to use more comments in your program to make it more readable.

There are three forms of comment text:

1. Comments start by two dashes and extend to the end of the current line.
2. The multi-line format comment is kept inside `/*...*/`, even if that occurs on a different line.
- 3.
4. You can use a special comment beginning with the two character sequence `"#!"` only on the first line of the program.

Example

```
#!/home/euphoria-4.0b2/bin/eui

-- First comment

puts(1, "Hello, Euphoria!\n") -- second comment

/* This is a comment which extends over a number
of text lines and has no impact on the program
*/
```

This will produce the following result:

```
Hello, Euphoria!
```

Note: On the first line (only) of your program, you can use a special comment beginning with `"#!"`. This informs the Linux shell that your file should be executed by the Euphoria interpreter.

End of ebook preview
If you liked what you saw...
Buy it from our store @ <https://store.tutorialspoint.com>

