

# Omf МультиОберон для OFront Быстрый Старт

Copyright © 2019-2020, by [Dmitry Dagaev](#)

Omf это реализация компилятора МультиОберон с бэкендом OFront. Основана на бэкенде OFront автора Josef Templ, адаптированного Ominc как CpFront.

Версия 0.95 27-Jun-2020

## Инсталляция.

Для Windows (цвет - для Windows):

For BlackBox 1.6

```
win_toinstall.vbs 16 <path-to-blackbox>
```

For BlackBox 1.7

```
win_toinstall.vbs 17 <path-to-blackbox>
```

Для Linux (цвет – для Linux):

Загрузите и установите <https://blackbox.obertone.ru/download>

```
tclsh lin_toinstall.tcl 17 <path-to-blackbox>
```

## Как начать работу из Black Box

### 1 Инсталляция

1. Предусловия.

Omf не использует другие программные средства, кроме BlackBox. Omb также требуется для компиляции omfsh на Linux.

2. Omf/Docu/Quick-Start.odc

3. Скомпилируйте модули ниже:

```
^Q DevCompiler.CompileThis OmcCfgfile OmcTarget OmcCRuntime OmcHooks OmcDialog  
OmcOPM OmcOPT OmcOPU OmcOPB OmcOPS OmcOPP OmcDump OmcTester OmcParams  
OmcCommandParams OmcOdcSource OmcTxtSource OmcRuntimeStd OmcDialogStd  
OmcDialogConsole OmcCompiler OmcTimesDialog OmcConsole OmcShell OmcLoader OmfOPG  
OmfOPC OmfOPV OmfParams OmfBackEnd OmfCompiler OmfLinker OmfSh
```

### 2 Компиляция примеров

2.1. Скомпилируйте примеры для 32-bit:

```
^Q OmfCompiler.CompileThis OmtestHelloWorld OmtestFormats OmtestDateTime  
OmtestMkTraps OmtestHeap
```

Ожидаемый результат в каталоге ~/Omtest/Cfwe/: OmtestHelloWorld.c OmtestFormats.c  
OmtestDateTime.c OmtestMkTraps.c OmtestHeap.c

2.2. Скомпилируйте примеры для 64-bit:

```
^Q OmfCompiler.CompileThis -64 OmtestHelloWorld OmtestFormats OmtestDateTime  
OmtestMkTraps OmtestHeap
```

Ожидаемый результат в каталоге ~/Omtest/Cfwr/: OmtestHelloWorld.c OmtestFormats.c  
OmtestDateTime.c OmtestMkTraps.c OmtestHeap.c

### 3 Подготовка бинарника OmfSh

Создание бинарника оболочки Omf Shell.

OmfSh создается стандартным компилятором BlackBox DevCompiler.

А дальнейшем планируется реализация авто-компиляции.

```
^Q OmbLinker.LinkExe -o "Bfwe/omfsh" $+Kernel Files Log Math Strings OStrings  
OLog Dates Times HostTimes HostFiles Dialog Stores Sequencers Models Services  
Fonts Meta Converters Ports Views Controllers Properties Mechanisms Containers
```

```

Printers Printing OmcCfgfile OmcTimesDialog Documents TextModels TextRulers
TextSetters TextViews HostConLog Runner OmcTarget OmcCRuntime OmcHooks OmcDialog
OmcOPM OmcOPT OmcOPB OmcOPU OmcOPS OmcOPP OmcDump OmcTester OmcParams
OmcOdcSource OmcRuntimeStd OmcDialogConsole OmcConsole OmcShell OmfOPG OmfOPC
OmfOPV OmfBackEnd OmfSh
^Q OmbLinker.LinkExe -o "Bfue/omfsh" $+Kernel Files Log Math Strings OStrings
OLog Dates Times HostTimes HostFiles Dialog Stores Sequencers Models Services
Fonts Meta Converters Ports Views Controllers Properties Mechanisms Containers
Printers Printing OmcCfgfile OmcTimesDialog Documents TextModels TextRulers
TextSetters TextViews HostConLog Runner OmcTarget OmcCRuntime OmcHooks OmcDialog
OmcOPM OmcOPT OmcOPB OmcOPU OmcOPS OmcOPP OmcDump OmcTester OmcParams
OmcOdcSource OmcRuntimeStd OmcDialogConsole OmcConsole OmcShell OmfOPG OmfOPC
OmfOPV OmfBackEnd OmfSh

```

## 4 Выгрузка компилятора Omf

```

^Q DevDebug.UnloadThis OmfCompiler OmfLinker OmfBackEnd OmfParams OmfOPV OmfOPC
OmfOPG OmcCoffLoader OmcLoaderRoutines OmcLoader OmcCompiler OmcDialogStd
OmcRuntimeStd OmcLogStd OmcOdcSource OmcCommandParams OmcParams OmcTester
OmcDump OmcOPP OmcOPS OmcOPU OmcOPB OmcOPT OmcOPM OmcDialog OmcHooks OmcCRuntime
OmcTarget Runner

```

# Как начать работу из Командной Строки.

## 1 Инсталляция

### 1. Предусловия.

Omf не использует другие программные средства. Выполняйте все вышеперечисленные команды из корневого каталога Mob-master.

## 2 Компиляция примеров

```
Bfwe\omfsh co OmtestHelloWorld
```

```
Bfue/omfsh co OmtestHelloWorld
```

Co[mpile] – Компиляция. Новый символьный файл создается с соответствующим сообщением, затем OmtestHelloWorld.mod компилируется в Omtest/Cfwe/HelloWorld.c. Последний символ : означает, что пример компилируется в программу с main, в противном случае также генерируются заголовочные файлы '.h'. Может быть задан не один, а список файлов для компиляции.

```
Bfwe\omfsh co -odc :OmtestHelloWorld :OmtestFormats :OmtestDateTime  
:OmtestMkTraps :OmtestHeap
```

```
Bfue/omfsh co -odc :OmtestHelloWorld :OmtestFormats :OmtestDateTime  
:OmtestMkTraps :OmtestHeap
```

Опция -odc в качестве входного берет файл OmtestHelloWorld.odc формата BlackBox вместо текстового mod. Компиляция всех перечисленных примеров выполняется командой выше.

## 3 Набор примеров

### Предусловия.

Для компиляции и линковки в бинарники требуются средства разработки для языка C. В данном пакете эти средства не поставляются и представлены не будут: ни Visual Studio, ни MinGW, ни CMake. Вы можете устанавливать средства разработки C или модифицировать скрипты сборки под установленные уже средства на Вашем компьютере. В скриптах используется следующее:

- gcc, ar – для fwe (oFront-Windows-32bit),
- clang – для fwr (oFront-Windows-64bit).

```
fwe_tomake
```

```
fue_tomake.sh
```

Создает все исполняемые файлы набора примеров для 32-бит

```
fwe_toclean
```

```
fue_toclean.sh
```

Удаляет все исполняемые файлы набора примеров для 32-бит

```
fwr_tomake
```

```
fur_tomake.sh
```

Создает все исполняемые файлы набора примеров для 64-бит

```
fwr_toclean
```

```
fur_toclean.sh
```

Удаляет все исполняемые файлы набора примеров для 64-бит

## 4 Выполнение примеров

### Предусловия.

Предполагается, что исполняемые файлы успешно откомпилированы из исходников '.c' и '.h'. Исполняемые файлы для 32-bit расположены в Cfwe. Исполняемые файлы для 64-bit расположены в Cfwr.

### 4.1. Самый простой пример - Hello, World (32-bit)

```
Omtest\Cfwe\OmtestHelloWorld
```

```
Omtest\Cfur\OmtestHelloWorld
```

Печать строковых, целых и действительных чисел

```
Omtest\Cfwe\OmtestFormats
Omtest/Cfur/OmtestFormats
```

#### 4.2 Печать даты, времени и реализация задержек

```
Omtest\Cfwe\OmtestDateTime
Omtest/Cfur/OmtestDateTime
```

#### 4.3 Травы – обработка нештатных ситуаций

Простой Assert

```
Omtest\Cfwe\OmtestMkTraps -trap a
Omtest/Cfur/OmtestMkTraps -trap a
Halt
```

```
Omtest\Cfwe\OmtestMkTraps -trap h
Omtest/Cfur/OmtestMkTraps -trap h
```

Деление на ноль

```
Omtest\Cfwe\OmtestMkTraps -trap z
Omtest/Cfur/OmtestMkTraps -trap z
```

Обращение к памяти по нулевому указателю

```
Omtest\Cfwe\OmtestMkTraps -trap p
Omtest/Cfur/OmtestMkTraps -trap p
```

#### 4.4 Работа с динамической памятью и сборка мусора

```
Omtest\Cfwe\OmtestHeap
Omtest/Cfur/OmtestHeap
```

## 5 Набор тестов

```
fwe_tests_tomake
fue_tests_tomake.sh
И 64-bit
```

```
fwr_tests_tomake
fur_tests_tomake.sh
```

Подготовка всех тестовых файлов: OmtestOmcSimpleTest OmtestOmcStringsTest  
OmtestOmcSystemTest OmtestOmcImportsTest OmtestOmcExtensionsTest OmtestOmcBoundTest  
OmtestOmcAdvancedTest

```
fwe_tests_run
fue_tests_run.sh
```

И 64-bit

```
fwr_tests_run
fur_tests_run.sh
```

Запуск тестов компилятора

## Журнал изменений

may 2019 original MultiOberon pre-version 0.8 released

nov 2019 MultiOberon pre-version 0.9 released

jun 2020 MultiOberon pre-version 0.95 released

Use it and enjoy! - Ўзsalos y disfrútalos! - Bonne utilisation - Приятного использования - Powodzenia - Viel Spaß

Дагаев Дмитрий Викторович

[dvdagaev@oberon.org](mailto:dvdagaev@oberon.org)