Omf MultiOberon/OFront Quick-Start

Omf is the instance of MultiOberon compiler with OFront backend. Based on Josef Templ's OFront.
Version 0.95 27-Jun-2020

Installation.

On Windows (this color - for Windows):
For BlackBox 1.6

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

For BlackBox 1.7

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

On Linux (this color – for Linux):
Download and install https://blackbox.obertone.ru/download

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

How to Start from Black Box

1 Installation

1. Preconditions.

Omf doesn't use any other services, except BlackBox. Omb is needed also while compiling omfsh on Linux.

2. Omf/Docu/Quick-Start.odc

3. Compile the following modules:

```
^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 Compiling Examples

2.1. Compiling examples for 32-bit:

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

Expected result in ~/Omtest/Cfwe/ directory: OmtestHelloWorld.c OmtestFormats.c OmtestDateTime.c OmtestMkTraps.c OmtestHeap.c

2.2. Compiling examples for 64-bit:

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

Expected result in ~/Omtest/Cfwr/ directory: OmtestHelloWorld.c OmtestFormats.c OmtestDateTime.c OmtestMkTraps.c OmtestHeap.c

3 Preparing Shell Executive

Creating Omf Shell executive.
OmfSh is created by the standard BlackBox DevCompiler.
Further self-compiling shell is planned.

```
^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 Unloading Omf Compiler

```
^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
```

How to Start from Command Line.

1 Installation

1. Preconditions.

Omf doesn't use any other services. Process all the commands below from the Mob-master root dir.

2 Compiling examples

```
Bfwe\omfsh co OmtestHelloWorld
Bfue/omfsh co OmtestHelloWorld
```

A new symbol file is created first, then OmtestHelloWorld.mod is compiled to Omtest/Cfwe/HelloWorld.c. The last : symbol means that the example is compiled to main program, otherwise the .h files are generated too.

```
Bfwe\omfsh co –odc :OmtestHelloWorld :OmtestFormats :OmtestDateTime
:OmtestMkTraps :OmtestHeap
Bfue/omfsh co –odc :OmtestHelloWorld :OmtestFormats :OmtestDateTime
:OmtestMkTraps :OmtestHeap
```

Option –odc takes BlackBox OmtestHelloWorld.odc file instead of mod. The command above compiles all the examples listed

3 Example set

Preconditions.

In order to compile and link to binary executives the C-development environment is needed. I provide no Visual Studio or MinGW or CMake tools. Please, use external tools or modify scripts. In scripts use the following:

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

```
fwe_tomake
fue_tomake.sh
```

Makes all the executives of 32-bit example set

```
fwe_toclean
fue_toclean.sh
```

Cleans all the executives of 32-bit example set

```
fwr_tomake
fur_tomake.sh
```

Makes all the executives of 64-bit example set

```
fwr_toclean
fur_toclean.sh
```

Cleans all the executives of 64-bit example set

4 Running executives

Preconditions.

I suppose that executives are compiled successfully from ‘.c’ and ‘.h’ sources.
The executives for 32-bit are located in Cfwe.
The executives for 64-bit are located in Cfwr.

4.1.The simplest Hello, World example (32-bit)

```
Omtest\Cfwe\OmtestHelloWorld
Omtest/Cfur/OmtestHelloWorld
```

Logging with char, int and real formats

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

4.2 Shows date, time and delay

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

4.3 Traps handling abilities of runtime

Simple Assert

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

Simple Halt

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

Zero divide

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

Nil pointer dereference

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

4.4 Dynamic memory and garbage collector

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

5 Tests set

```
fwe_tests_tomake
fue_tests_tomake.sh
```

And 64-bit

```
fwr_tests_tomake
fur_tests_tomake.sh
```

Makes all the tests prepared: OmtestOmcSimpleTest OmtestOmcStringsTest OmtestOmcSystemTest OmtestOmcImportsTest OmtestOmcExtensionsTest OmtestOmcBoundTest OmtestOmcAdvancedTest

```
fwe_tests_run
fue_tests_run.sh
```

And 64-bit

```
fwr_tests_run
fur_tests_run.sh
```

Runs all the compiler tests

Change log

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! - ЎЪsalos y disfrъtalos! - Bonne utilisation - Приятного использования - Powodzenia - Viel SpaЯ

Dmitry V. Dagaev
dvdagaev@oberon.org