October 4, 2022

1 How to build CMason from sources?

First, download CMason sources from cmason.gabrielmargarido.org.
Install Node.js and NPM (via: apt, dnf, yum, pacman)
or download from www.nodejs.org.
Install Node dependencies as –save-dev: typescript, @types/node, pkg

```
[GNU Make]
cd cmason-build/src && make

[Shell Script]
cd cmason-build/src
sudo chmod +x make.sh
./make.sh

[Windows - Batch]
cd cmason-build\src
start make.bat
```

2 Run CMason - CMasonfile

```
SYNTAX:
cmason-common -t <target>
cmason-save -t <target>
cmason-dev -t <target>

cmason-common -t
cmason-save -t
cmason-dev -t

cmason-common --version
cmason-save --version
cmason-dev --version



EXAMPLE:
cmason-common -t build
cmason-save -t build
cmason-dev -t build

cmason-common -t
cmason-save -t
cmason-dev -t

cmason-common --version
cmason-save --version
cmason-dev --version
```

3 CMason Common Architectures

```
WIN32
WIN64
WIN*ARM32
WIN*ARM64

MAC32
MAC64
MAC*ARM64

LINUX32
LINUX64
LINUX*ARM32
LINUX*ARM64
LINUX*PPC32
LINUX*PPC64
LINUX*S390
LINUX*S390X
LINUX*MIPS
LINUX*MIPSEL

FREEBSD32
FREEBSD64
FREEBSD*ARM32
FREEBSD*ARM64
FREEBSD*PPC32
FREEBSD*PPC64
FREEBSD*S390
FREEBSD*S390X
FREEBSD*MIPS
FREEBSD*MIPSEL

OPENBSD32
OPENBSD64
OPENBSD*ARM32
OPENBSD*ARM64
OPENBSD*PPC32
OPENBSD*PPC64
OPENBSD*S390
OPENBSD*S390X
OPENBSD*MIPS
OPENBSD*MIPSEL
```

```
SUNOS32
SUNOS64
SUNOS*ARM32
SUNOS*ARM64
SUNOS*PPC32
SUNOS*PPC64
SUNOS*S390
SUNOS*S390X
SUNOS*MIPS
SUNOS*MIPSEL

UNIVERSAL
```

4 CMason Common

CMason Common binary inspects CMasonfile archives looking for targets based on: Current running Operating System and CPU Architecture

```
# Microsoft Windows (all)
WIN32_all: "echo 'main target for windows'"
WIN64_all: "echo 'main target for windows (64-bit)'"
WIN*ARM64_all: "echo 'main target for windows (ARM 64-bit)'"


# Apple MacOS X (all)
MAC32_all: "echo 'main target for macos x (old models)'"
MAC64_all: "echo 'main target for macos x (64-bit)'"
MAC*ARM64_all: "echo 'main target for macos x (Apple Silicon)'"



# Microsoft Windows (guess)
WIN32_guess: "echo 'guess target for windows'"
WIN64_guess: "echo 'guess target for windows (64-bit)'"
WIN*ARM64_guess: "echo 'guess target for windows (ARM 64-bit)'"

# Apple MacOS X (guess)
MAC32_guess: "echo 'guess target for macos x (old models)'"
MAC64_guess: "echo 'guess target for macos x (64-bit)'"
MAC*ARM64_guess: "echo 'guess target for macos x (Apple Silicon)'"

# Microsoft Windows (build)
WIN32_build: "echo 'build target for windows'"
WIN64_build: "echo 'build target for windows (64-bit)'"
WIN*ARM64_build: "echo 'build target for windows (ARM 64-bit)'"


# Apple MacOS X (build)
MAC32_build: "echo 'build target for macos x (old models)'"
MAC64_build: "echo 'build target for macos x (64-bit)'"
MAC*ARM64_build: "echo 'guess target for macos x (Apple Silicon)'"
```

5 CMason Save Operating System List

```
WIN    MAC    LINUX   FREEBSD   OPENBSD    SUNOS    UNIVERSAL
```

6 CMason Save

CMason Dev binary inspects CMasonfile archives looking for targets based on: Current running Operating System

```
# Microsoft Windows
WIN_all: "echo 'main target for windows'"

# Apple MacOS X
MAC_all: "echo 'main target for macos x'"
```

7 CMason Dev

CMason Dev binary inspects CMasonfile archives looking for targets based on its name.

```
# Any Operating System
all: "echo 'main target'"
build: "echo 'build target'"
guess: "echo 'guess target'"
```

This documentation is dedicated to:
Phd. Professor Luis Filipe Miranda de Souza Ribeiro - ***In memoriam***
Phd. Professor Elizabete Velloso de Margarido Ribeiro
Severina Silva Velloso de Margarido
Ana Cristina Baptista Miranda de Souza Ribeiro
João Paraízo - C.E.G.
Toseli Matos Paraízo (Fortran Developer and Mechanic Engineer Student) - C.E.G.
Isaías Francisco Ferreira da Silva - M.'. M.'.
Centro C.E.G. - São Gonçalo, RJ
Capítulo Perfeita União Nº 115 - Ordem DeMolay