Learn how to write a `Makefile`

Berend de Boer – eurotex 2001

introduction
targets
prerequisites
commands
Tips and Tricks
big example
close

introduction

Makefiles codify the following knowledge:

1. How to build: the specification how things need to be processed, i.e. the command to turn a `.tex` document into a `.pdf` document.

2. When to build: they have the ability to only build the things that have been changed. That's what makes them different from ordinary shell scripts.

introduction
targets
prerequisites
commands
Tips and Tricks
big example
close

structure of a Makefile

Makefiles are usually stored in a file called `Makefile`. Or they have the extension .mk.

The basic structure of a makefile is:

```
target: prerequisites ...
    command
    ...
    ...
```

intro-duction
targets
prereq-uisites
commands
Tips and Tricks
big example
close

basic ConTeXt makefile

Before delving into the details, first an example of a ConTeXt makefile:

introduction
targets
prerequisites
commands
Tips and Tricks
big example
close

```
PRODUCT=test.pdf

.PHONY: clean

default: $(PRODUCT)

$(PRODUCT): test.tex graphic_a.1 graphic_b.1

%.pdf: %.tex
       texexec $<

%.1: %.mp
       mpost $<

clean:
       texutil --purge
```

intro-duction

targets

prereq-uisites

commands

Tips and Tricks

big example

close

what Make to use

What make tool should I use, or simply, where can I get it?

The answer is: GNU make. Comes with almost any system. It has a free manual. BSD make might do, but is somewhat less powerful. The advanced options are not compatible.

Windooze users should get cygwin from `http://sources.redhat.com/cygwin/index.html`. Don't be tempted to use brain dead makes from Microsoft, Borland, or anyone else.

targets

A target is a file. A target is a file. A target is a file.

```
myfile.pdf: myfile.tex
	texexec myfile.tex
```

Did I already say that a target is a file?

introduction

targets

prerequisites

commands

Tips and Tricks

big example

close

Makefiles can have phony targets (none files).

```
default: all

all: myfile.pdf

...
```

If a phony target happens to be a file, you're out of luck. Really save phony targets are written as:

```
.PHONY: default
.PHONY: all clean

...

clean:
  texutil --purge
```

introduction

targets

prerequisites

commands

Tips and Tricks

big example

close

Make accepts as parameters names of targets:

```
make clean
make all
```

If no target is given, the first target is the default target:

```
.PHONY: all default clean

default: all

all: myfile.pdf

clean:
  texutil --purge
```

intro-duction

targets

prereq-uisites

commands

Tips and Tricks

big example

close

prerequisites

A prerequisite is anything a target depends on:

1. Source file (files that cannot be generated).
2. Other targets.

`make` will build targets in dependency order.

```
  myfile.pdf: myfile.tex graphic.1
    texexec myfile.tex

  graphic.1: graphic.mp
    mpost graphic.mp
```

Wildcards are supported:

introduction
targets
prerequisites
commands
Tips and Tricks
big example
close

```
myfile.pdf: *.tex
  texexec myfile.tex
```

introduction

targets

prerequisites

commands

Tips and Tricks

big example

close

commands

A command is always passed to your shell. A command is therefore anything (or limited too something) your shell understands. You will detect that `command.exe` (or `cmd.exe`) is extremely limited. If you have download Cygwin, you have a complete Unix shell.

WARNING: a command is always preceded by a TAB character.

You can use variables in commands. `$<`is the name of the first prequisite.

```
myfile.pdf: myfile.tex
  texexec $<
```

The target is also available in `$@`

intro-duction
targets
prereq-uisites
commands
Tips and Tricks
big example
close

```
myfile.pdf: myfile.tex
  texexec $< --result=$@
```

intro-duction

targets

prereq-uisites

commands

Tips and Tricks

big example

close

your turn

Help me to create a basic `Makefile`.

variables

You can use variables to make sure things are defined only once:

```
SRC=a.tex b.tex

a.pdf: $(SRC)
  texexec a.tex

b.pdf: $(SRC)
  texexec b.tex
```

Variables can be used anywhere: in targets, prerequites or commands.

introduction
targets
prerequisites
commands
Tips and Tricks
big example
close

implicit rules

It's annyoing to specify you want to run `texexec` for TeX files if you always want to do that. With implicit rules you can specify such things:

```
%.pdf: %.tex
       texexec $<

%.1: %.mp
       mpost $<
```

A '%' is a template character, a kind of wildcard.

introduction
targets
prerequisites
commands
Tips and Tricks
big example
close

Tips and Tricks

1. Do not directly generate the target with texexec and such: if they fail, `make` thinks they're updated when you do the next run.

```
myfile.pdf: myfile.tex
  texexec $<x --result=temp.pdf
  mv temp.pdf $@
```

2. Spaces in filenames or directories: make sure you don't have them.

3. You can include other Makefiles, for example you can include your TEX implicit rules with:

introduction
targets
prerequisites
commands
Tips and Tricks
big example
close

```
myfile.pdf: myfile.tex
include tex.mk
```

introduction
targets
prerequisites
commands
Tips and Tricks
big example
close

big example

```
#
# Makefile used with sql2context.tex
#
# tests/converts/compiles various files

.PHONY: all clean validate archive
.SUFFIXES:
.SUFfIXES: .xml .csv .sql .ib .ddl .txt .pdf .tex .ibout .db2

# main target

all: sql2context.pdf

# my document

DOCSRC = \
       sql2context.tex \
       example.xml example.dtd \
```

intro-duction
targets
prereq-uisites
commands
Tips and Tricks
big example
close

```
        flintstones.ddl flintstones.csv flintstones.xml \
        csv2xml.pl ib2xml.pl \
        select1.sql select1.ibout \
        select2.sql select2.ibout select2.xml \
        select3.sql select3.ibout select3.tex \
        program.xml program.xsl eurotex.xml \
        flintstones.db2 \
        xsltprocessor.png

sql2context.pdf: $(DOCSRC)

# cleanup
clean:
        #texutil --purge
        rm flintstones.interbase flintstones.ib
        rm flintstones.ibmdb2 flintstones.db2
        rm *.ibout

# validation
validate: all
        SAXCount *.xml
```

introduction

targets

prerequisites

commands

Tips and Tricks

big example

close

```
# packing
archive: sql2context.zip

sql2context.zip: sql2context.pdf $(DOCSRC) Makefile
       -rm sql2context.zip
       zip sql2context $(DOCSRC) Makefile

# file generation

flintstones.xml: flintstones.csv  example.dtd

flintstones.sql: flintstones.ddl

flintstones.ib: flintstones.ddl

flintstones.db2: flintstones.ddl

select2.xml: select2.ibout ib2xml.pl

select3.tex: select3.ibout
       @echo cannot make select3.tex yet
```

intro-duction
targets
prereq-uisites
commands
Tips and Tricks
big example
close

```
       @echo It should be included, so just say:
       @echo   touch select3.tex
       @echo and make again
       exit 1

select1.ibout: select1.sql

select2.ibout: select2.sql

select3.ibout: select3.sql

eurotex.xml: program.xml program.xsl
       testXSLT -in program.xml -XSL program.xsl -out eurotex.x

# db generation

flintstones.interbase: flintstones.ib
       rm -f flintstones.gdb flintstones.out
       echo 'create database "flintstones.gdb";' > createdb.sq
       /opt/interbase/bin/isql -i createdb.sql
       /opt/interbase/bin/isql -i flintstones.ib -o flintstones
```

intro-duction

targets

prereq-uisites

commands

Tips and Tricks

big example

close

```
        echo ’select "surname", "age" from "flintstone" order by
        touch flintstones.interbase
        rm flintstones.out

flintstones.ibmdb2: flintstones.db2
        #db2 start database manager
        -db2 "drop database flint"
        db2 "create database flint"
        db2 -td\; -vf flintstones.db2
        touch flintstones.ibmdb2

# rules

%.xml: %.csv csv2xml.pl
        perl -w csv2xml.pl $< > tmp.tmp
        mv tmp.tmp $@

%.xml: %.ibout ib2xml.pl
        perl -w ib2xml.pl $< > tmp.tmp
        mv tmp.tmp $@
```

introduction
targets
prerequisites
commands
Tips and Tricks
big example
close

```
%.ib: %.ddl
       xplain2sql -interbase $< > tmp.tmp
       mv tmp.tmp $@

%.db2: %.ddl
       xplain2sql -db2 $< > tmp.tmp
       mv tmp.tmp $@

%.sql: %.ddl
       xplain2sql -ansi $< > tmp.tmp
       mv tmp.tmp $@

%.pdf: %.tex
       texexec $<

%.ibout: %.sql flintstones.interbase
       if [ -e $@ ]; then rm $@; fi
       /opt/interbase/bin/isql flintstones.gdb -i $< -o tmp.out
       mv tmp.out $@
```

intro-duction

targets

prereq-uisites

commands

Tips and Tricks

big example

close

your turn again

intro- duction

targets

prereq- uisites

commands

Tips and Tricks

big example

close