

Lemma 1 Ltd.
2nd Floor
31A Chain St.
Reading
Berks
RG1 2HX

Creating Example ProofPower Databases

Abstract

This document contains the makefile, plus its associated files, that produces example databases for ProofPower. The “Interface” section describes the creation and use of this makefile from a user’s point of view.

Version: 1.17
Date: 11 February 2008
Reference: DS/FMU/IED/WRK051
Pages: 21

Prepared by: R.D. Arthan
Tel: +44 118 958 4409
E-Mail: rda@lemma-one.com

Copyright © : Lemma 1 Ltd. 2011

1 DOCUMENT CONTROL

1.1 Contents List

1	DOCUMENT CONTROL	2
1.1	Contents List	2
1.2	Document Cross References	3
1.3	Changes History	3
1.4	Changes Forecast	4
2	GENERAL	5
2.1	Scope	5
2.2	Introduction	5
2.3	Interface	7
3	START OF MAKEFILE	10
4	PRELIMINARIES	10
4.1	Macros	10
4.2	Default	10
4.3	Pattern Matching Rules	11
5	PREPARING CLEAN DATABASES	12
6	MAKING THE TUTORIAL MANUAL (usr004)	13
7	MAKING THE Z TUTORIAL MANUAL (usr011)	14
7.1	Z Tutorial Source Files	14
7.2	Z Tutorial Make Entries	14
8	MAKING THE HOL TUTORIAL MANUAL (usr013)	15
8.1	HOL Tutorial Source Files	15
8.2	HOL Tutorial Make Entries	15
9	MAKING TUTORIAL OVERHEADS (usr022)	16
10	MAKING Z TUTORIAL OVERHEADS (usr023)	17
11	MAKING HOL FORMALISED (spc001)	17
12	MAKING HOL SEMANTICS (spc002)	17
13	MAKING MODAL LOGIC (wrk022)	17

<i>Lemma 1 Ltd.</i>	<i>Creating Example ProofPower Databases</i>	3
14 MAKING RAMSEYS THEOREM (wrk043)		18
15 MAKING FINITENESS THEOREMS (wrk044)		18
16 MAKING FINITENESS THEOREMS (wrk046)		18
17 MAKING CASE STUDY (wrk050)		19
18 CREATE ALL DATABASES		19
19 TIDYING UP THE FILE STORE		20
20 make ENDINGS		20
21 INDEX		21

1.2 Document Cross References

- [1] DS/FMU/IED/USR004. *ProofPower Tutorial Manual.* Lemma 1 Ltd., <http://www.lemma-one.com>.
- [2] DS/FMU/IED/USR007. *ProofPower Installation and Operation.* Lemma 1 Ltd., <http://www.lemma-one.com>.
- [3] DS/FMU/IED/USR011. *ProofPower Z Tutorial.* Lemma 1 Ltd., <http://www.lemma-one.com>.
- [4] DS/FMU/IED/USR013. *ProofPower HOL Tutorial Notes.* Lemma 1 Ltd., <http://www.lemma-one.com>.
- [5] DS/FMU/IED/USR022. *ProofPower HOL Tutorial Transparencies.* Lemma 1 Ltd., <http://www.lemma-one.com>.
- [6] DS/FMU/IED/USR023. *ProofPower Z Tutorial Transparencies.* Lemma 1 Ltd., <http://www.lemma-one.com>.
- [7] *SunOS Release 4.1 Documentation.* Sun Microsystems, Inc.

1.3 Changes History

Issues 1.1-1.3 (To 10th March 1993) Initial versions.

Issue 1.4 (22nd September 1993) Addition of new HOL tutorial, changed to work with new release mechanism.

Issue 1.5 (23rd September 1993) Reordering of dependencies for all_hol, addition of spc001.th, spc002.th

Issue 1.6 (28th September 1993) Change from usr011x.sml, usr013x.sml to usr011x.doc, usr013x.doc

Issue 1.9 (25th August 1994) Changed \$INSTALLDIR to be \$PPPPINSTALLDIR.

Issue 1.15 Copyright and banner updates for open source release.

Issue 1.16 PPHol-specific updates for open source release

Issue 1.17 The demo scripts can now be run as tests by the hol and zed make files.

1.4 Changes Forecast

None.

2 GENERAL

2.1 Scope

This document provides documentation, a makefile and auxiliary scripts for the process of producing certain example databases for a **ProofPower** user.

2.2 Introduction

This document contains the literate script of a makefile (*wrk051.mkf*) and a number of auxiliary files which are scripts for other programs. This material allows the creation of example databases containing **ProofPower** tutorial and case study material. The user's interface to this material is given in section 2.3.

To produce the makefile the program **sieve** is used to process this document, with the **sml** option. The result is the creation of a makefile, together with a number of small utility-files that are inputs for **ProofPower**. Most the the makefile entries are concerned with the creation or updating of various example databases.

The commentary upon the makefile assumes some knowledge of the UNIX **make** command. The key item of knowledge is the form and meaning of a standard makefile entry.

Example makefile entry

```
|target1 target2 : dependency1 dependency2 dependency3
|    command1
|    command2
|    command3
|    command4
```

To use such an entry in a makefile the user might type:

csh

```
|make -f makefile target1
```

If so, **make** will first recursively use the rest of the makefile to do what processing is necessary for each of the *dependency_i* (this may be nothing). It will then determine whether *target1* exists and has a “time of last change” after the “time of last change” of each *dependency_i*. If so **make** will halt. If not, then it will execute each of the *command_i* in sequence, in different shells, and then halt.

It is the above **make** algorithm, in combination with a single file (a database) containing all the effect of cumulative loads in **ProofPower**, that leads to the system of “flags” in the makefile entries in section 5 and beyond.

See UNIX documentation[7] for further details, such as a description of macros and pattern matching rules.

2.3 Interface

In the following descriptions the text marked by:

```

csh
| marked text

```

may be cut and pasted without change from the source file of this document (wrk051.doc) into a SunView shell tool or command tool.

To produce the example databases, start in a directory with the necessary files of:

```

(1)   usr004.doc
(2)   usr011A.doc usr011B.doc usr011C.doc usr011D.doc
(3)   usr011E.doc usr011T.doc usr011S.doc usr011X.doc
(4)   usr013A.doc usr013B.doc usr013C.doc usr013D.doc usr013E.doc
(5)   usr013F.doc usr013G.doc usr013H.doc usr013S.doc usr013X.doc
(6)   usr022_slides.doc usr023_slides.doc
(7)   wrk022.doc wrk043.doc wrk044.doc
(8)   wrk051.doc (this file)

```

These should all be present in the subdirectory *docs* of the ProofPower installation directory. However, it is better to work with copies of these files, in a new directory, e.g. *examples*. If using a ProofPower installation made in the manner suggested in [2] then copies may be made by:

```

csh
| cd $PPINSTALLDIR
| mkdir examples
| cd $PPINSTALLDIR/docs
| cp usr004.doc usr011A.doc usr011B.doc usr011C.doc usr011D.doc \
|   usr011E.doc usr011T.doc usr011S.doc usr011X.doc \
|   usr013A.doc usr013B.doc usr013C.doc usr013D.doc usr013E.doc \
|   usr013F.doc usr013G.doc usr013H.doc usr013S.doc usr013X.doc \
|   usr022_slides.doc usr023_slides.doc wrk022.doc wrk043.doc \
|   wrk044.doc wrk051.doc $PPINSTALLDIR/examples

```

Then create the makefile *wrk051.mkf* by :

```

csh
| cd $PPINSTALLDIR/examples
| sieve sml < wrk051.doc

```

Having produced the makefile (`wrk051.mkf`) and utilities, as above, there are then several options for further commands.

To print this document (assuming `pstex` without arguments sends output to the desired printer):

```
csb
|make -f wrk051.mkf wrk051.prt
```

To create a child ProofPower database containing the effect of loading the tutorial scripts of `usr004` [1]:

```
csb
|make -f wrk051.mkf usr004.tutorial
```

The resulting database can be invoked by:

```
|hol -d example_hol
```

Similarly for `usr011`[3], creating a database suitable for loading the ProofPower-Z exercises and trying solutions for them, by hand:

```
csb
|make -f wrk051.mkf usr011.tutorial
```

The resulting database can be invoked by:

```
|zed -d example_zed
```

Similarly for `usr013`[4], creating a database suitable for loading the ProofPower-HOL exercises and trying solutions for them, by hand:

```
csb
|make -f wrk051.mkf usr013.tutorial
```

The resulting database can be invoked by:

```
|zed -d example_hol
```

Similar entries also exist for `usr022`[5] and `usr023`[6], rather than `usr004`.

```
csb
|make -f wrk051.mkf usr022.tutorial
|make -f wrk051.mkf usr023.tutorial
```


To create a database containing the effect of loading the exercise scripts for `usr011` (which will also run the tutorial scripts for `usr011`, if not already done):

```
csh
|make -f wrk051.mkf usr011.zexercises
```

To create a database containing the effect of loading the solutions scripts for `usr011` (which will also run the exercises, if not already done):

```
csh
|make -f wrk051.mkf usr011.zsolutions
```

To create a database containing the effect of loading the exercise scripts for `usr013` (which will also run the tutorial scripts for `usr013`, if not already done):

```
csh
|make -f wrk051.mkf usr011.exercises
```

To create a database containing the effect of loading the solutions scripts for `usr011` (which will also run the exercises, if not already done):

```
csh
|make -f wrk051.mkf usr011.solutions
```

To tidy up afterwards, deleting all created databases and files, execute:

```
csh
|make -f wrk051.mkf clean_examples
```

To remove all auxiliary files created by “sieving” this one:

```
csh
|rm -f wrk051.mkf \
|   wrk051_delete_theories_usr004.sml \
|   wrk051_delete_theories_usr011.sml \
|   wrk051_delete_theories_zexercises.sml \
|   wrk051_delete_theories_usr013.sml \
|   wrk051_delete_theories_exercises.sml \
|   wrk051_delete_theories_spc001.sml \
|   wrk051_delete_theories_spc002.sml \
|   wrk051_delete_theories_usr022.sml \
|   wrk051_delete_theories_usr023.sml \
|   wrk051_delete_theories_wrk022.sml \
|   wrk051_delete_theories_wrk043.sml \
|   wrk051_delete_theories_wrk044.sml \
|   wrk051_delete_theories_wrk050.sml
```

The file `pp_install`, to be found in the installation directory, copies all the files needed into a new directory and runs all the ProofPower scripts to generate the two databases (`'arch'example_zed.db`) and (`'arch'example_hol.db`). It copies these two databases into the `'arch'bin` directory, and cleans up afterwards. This file needs to be run from the release directory.

It is advised that the `changeParent` command is run on the new databases, the new parents being `sun4pp_hol.db` and `sun4pp_zed`, found in the subdirectory `sun4bin`. The script `pp_install` does this.

3 START OF MAKEFILE

Text dumped to file wrk051.mkf

```
| # wrk051.mkf
| #     A makefile that produces example databases.
| #     Created from: /Users/rda/pp/opp/bld/RCS/wrk051.doc,v 1.17 2008/02/11 11:35:34 rda
```

4 PRELIMINARIES

4.1 Macros

The first two macros below define the base names of the example databases.

Text appended to file wrk051.mkf

```
| EHOLDB = example_hol
| EZEDDB = example_zed
```

The following are the comments returned at the end of a successful or failed make:

Text appended to file wrk051.mkf

```
| SUCCESS_MESSAGE = "make command succeeded"
| FAILURE_MESSAGE = "make command failed"
```

4.2 Default

The default action of the makefile is to print a message. The default is indicated by it being the first target in the makefile.

Text appended to file wrk051.mkf

```
default :
    @echo Use:
    @echo "      make -f wrk051.mkf all"
    @echo to create example ProofPower databases for HOL and Z.
    @echo See document DS/FMU/IED/WRK051 for further details.
```

4.3 Pattern Matching Rules

See `make` documentation for how the following work. They provide the “standard” processing rules for:

- .doc** - documents,
- .sml** - Standard ML files, usually derived from documents,
- .tex** - L^AT_EX files derived from documents,
- .dvi** - files displayable by `dvipage` or printable by `pstex`,
- .prt** - dummy files, used to invoke printing.

Text appended to file wrk051.mkf

```
.SUFFIXES: .doc .sml .tex .dvi .prt
%.tex: %.doc
    doctex $*

%.dvi: %.tex
    texdvi $*
    - bibtex $*
    texdvi $*
    texdvi $*

%.sml: %.doc
    docsmml $*

%.prt: %.dvi
    pstex $*
```

The following .doc files are supplied in a release of ProofPower.

Text appended to file wrk051.mkf

```
usr011A.doc \  
usr011B.doc \  
usr011C.doc \  
usr011D.doc \  
usr011E.doc \  
usr011T.doc \  
usr013A.doc \  
usr013B.doc \  
usr013C.doc \  
usr013D.doc \  
usr013E.doc \  
usr013F.doc \  
usr013G.doc \  
usr013H.doc:  
    @echo Missing $@, which should have been present in release  
    exit 1
```

5 PREPARING CLEAN DATABASES

The two flags, *wrk051_hol.flag* and *wrk051_zed.flag* are two empty files, whose “time of last change” is the time of creation of the example databases. Various items later in this makefile have these flags amongst their dependencies. This means that, amongst other reasons for reloading, such items are reloaded if their previous load was made prior to the creation of the appropriate example database. If the dependencies of these items were the database filenames themselves, then a reload would be needed on any change to the databases, such as another, orthogonal, load. However, the databases themselves will still be recreated if the flags are used as dependencies when they are out of step with their parents (or are not there at all!).

Text appended to file wrk051.mkf

```
| HOLPARENT=$(shell echo $${PPHOLPARENT:-hol})
| wrk051.hol.flag $(EHOLDB) :
|     pp_make_database -f -p $(HOLPARENT) $(EHOLDB)
|     touch wrk051.hol.flag
|
| ZEDPARENT=$(shell echo $${PPZEDPARENT:-zed})
| wrk051.zed.flag $(EZEDDB) :
|     pp_make_database -f -p $(ZEDPARENT) $(EZEDDB)
|     touch wrk051.zed.flag
```

6 MAKING THE TUTORIAL MANUAL (*usr004*)

The following pattern of file creation and makefile entry is used in most of the following sections, though we only comment upon it in this one.

The following command creates a file: *wrk051_delete_theories_usr004.sml* that is used to delete, if present, the theories that would be created by loading *usr004.sml*. If the theories were present the load of *usr004.sml* would fail. A similar file is created for each of the items that may be added into the example databases.

Text dumped to file wrk051_delete_theories_usr004.sml

```
| (* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
| (force_delete_theory "Peanissimo" handle Fail _ => ());
| (force_delete_theory "ordinals" handle Fail _ => ());
```

The following creates, if necessary, an example HOL database (via *wrk051.hol.flag*). It then creates an *.sml* file from *usr004.doc*, loads the above theory-deleting file into the example database, and then loads this formal text of *usr004.doc*. It then saves the database with the results of the above loads. Finally, it touches a flag, *usr004.tutorial*, so that the “time of last change” for the flag is the time of last loading *usr004.sml*.

Text appended to file wrk051.mkf

```
| usr004.tutorial: wrk051.hol.flag usr004.doc usr004.sml
|     hol -d $(EHOLDB) -f wrk051_delete_theories_usr004 -f usr004
|     touch usr004.tutorial
```

7 MAKING THE Z TUTORIAL MANUAL (usr011)

7.1 Z Tutorial Source Files

usr011.*	top level document
usr011A.*	Ch. 2 - Z Language
usr011B.*	Ch. 3 - Z Predicate Calculus
usr011C.*	Ch. 4 - Z Expressions
usr011D.*	Ch. 5 - Z Paragraphs
usr011E.*	Ch. 6 - An Example Specification and Proof
usr011T.*	Ch. 7 - The Z ToolKit
usr011X.*	Exercises
usr011S.*	Solutions

7.2 Z Tutorial Make Entries

Text dumped to file wrk051_delete_theories_usr011.sml

```
(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory "usr011" handle Fail _ => ());
```

Text dumped to file wrk051_delete_theories_zexercises.sml

```
(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory "z_exercises_1" handle Fail _ => ());
(force_delete_theory "z_exercises_2" handle Fail _ => ());
(force_delete_theory "z_exercises_3" handle Fail _ => ());
(force_delete_theory "z_exercises_4" handle Fail _ => ());
```

Text appended to file wrk051.mkf

```
usr011.tutorial: wrk051_zed.flag usr011A.sml usr011B.sml \
    usr011C.sml usr011D.sml usr011E.sml usr011T.sml \
    usr023_slides.sml
zed -d $(EZEDDB) -f wrk051_delete_theories_usr011.sml -f usr011A.sml \
    -f usr011B.sml -f usr011C.sml -f usr011D.sml \
    -f usr011T.sml -f usr011E.sml -f usr023_slides.sml
touch usr011.tutorial

usr011.zexercises: usr011.tutorial usr011X.doc usr011X.sml
zed -d $(EZEDDB) -f wrk051_delete_theories_zexercises.sml -f usr011X.sml
touch usr011.zexercises

usr011.zsolutions: usr011.zexercises usr011S.doc usr011S.sml
zed -d $(EZEDDB) -f usr011S.sml
touch usr011.zsolutions
```

8 MAKING THE HOL TUTORIAL MANUAL (usr013)

8.1 HOL Tutorial Source Files

usr013.*	top level document
usr013A.*	Ch. 2 - Introduction
usr013B.*	Ch. 3 - HOL type system
usr013C.*	Ch. 4 - HOL terms
usr013D.*	Ch. 5 - Theories
usr013E.*	Ch. 6 - Forward proof
usr013F.*	Ch. 7 - Goal Oriented proof
usr013G.*	Ch. 8 - Predicate calculus
usr013H.*	Ch. 9 - Induction
usr013X.*	Exercises
usr013S.*	Solutions

8.2 HOL Tutorial Make Entries

Text dumped to file wrk051_delete_theories_usr013.sml

```
(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory "usr013" handle Fail _ => ());
```

Text dumped to file wrk051_delete_theories_exercises.sml

```
|(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
|(force_delete_theory "usr013X" handle Fail _ => ());
```

Text dumped to file wrk051_delete_theories_solutions.sml

```
|(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
|(force_delete_theory "Shankar–Rushby–International" handle Fail _ => ());
```

Text appended to file wrk051.mkf

```
usr013.tutorial: wrk051_hol.flag usr013A.sml usr013B.sml \
                   usr013C.sml usr013D.sml usr013E.sml \
                   usr013F.sml usr013G.sml usr013H.sml
hol -d $(EHOLDB) -f wrk051_delete_theories_usr013.sml \
      -f usr013A.sml -f usr013B.sml -f usr013C.sml \
      -f usr013D.sml -f usr013E.sml -f usr013F.sml \
      -f usr013G.sml -f usr013H.sml
touch usr013.tutorial

usr013.exercises: usr013.tutorial usr013X.doc usr013S.doc usr013X.sml usr013S.sml
hol -d $(EHOLDB) -f wrk051_delete_theories_exercises.sml -f usr013X.sml
touch usr013.exercises

usr013.solutions: usr013.exercises usr013S.doc usr013X.sml usr013S.sml
hol -d $(EHOLDB) -f wrk051_delete_theories_solutions.sml -f usr013S.sml
discgarb $(EHOLDB)
touch usr013.solutions
```

9 MAKING TUTORIAL OVERHEADS (usr022)

Text dumped to file wrk051_delete_theories_usr022.sml

```
|(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
|(force_delete_theory "tutorial" handle Fail _ => ());
```

Text appended to file wrk051.mkf

```
usr022.tutorial: wrk051_hol.flag usr022_slides.doc usr022_slides.sml
hol -d $(EHOLDB) -f wrk051_delete_theories_usr022.sml -f usr022_slides
touch usr022.tutorial
```


10 MAKING Z TUTORIAL OVERHEADS (usr023)

Text dumped to file wrk051_delete_theories_usr023.sml

```
| (* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
| (force_delete_theory "usr023" handle Fail _ => ());
```

Text appended to file wrk051.mkf

```
| usr023.tutorial: wrk051_zed.flag usr023_slides.doc usr023_slides.sml
| zed -d $(EZEDDB) -f wrk051_delete_theories_usr023.sml -f usr023_slides
| touch usr023.tutorial
```

11 MAKING HOL FORMALISED (spc001)

Text dumped to file wrk051_delete_theories_spc001.sml

```
| (* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
| (force_delete_theory "spc001" handle Fail _ => ());
```

Text appended to file wrk051.mkf

```
| spc001.th: wrk051_hol.flag spc001.doc spc001.sml
| hol -d $(EHOLDB) -f wrk051_delete_theories_spc001.sml -f spc001.sml
| touch spc001.th
```

12 MAKING HOL SEMANTICS (spc002)

Text dumped to file wrk051_delete_theories_spc002.sml

```
| (* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
| (force_delete_theory "spc002" handle Fail _ => ());
```

Text appended to file wrk051.mkf

```
| spc002.th: wrk051_hol.flag spc002.doc spc002.sml
| hol -d $(EHOLDB) -f wrk051_delete_theories_spc002.sml -f spc002.sml
| touch spc002.th
```

13 MAKING MODAL LOGIC (wrk022)

Text dumped to file wrk051_delete_theories_wrk022.sml

```
| (* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
| (force_delete_theory "wrk022" handle Fail _ => ());
```

Text appended to file wrk051.mkf

```
wrk022.th: wrk051_hol.flag wrk022.doc wrk022.sml
             hol -d $(EHOLDB) -f wrk051_delete_theories_wrk022.sml -f wrk022.sml
             touch wrk022.th
```

14 MAKING RAMSEYS THEOREM (wrk043)

Text dumped to file wrk051_delete_theories_wrk043.sml

```
(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory "ramsey" handle Fail - => ());
```

Text appended to file wrk051.mkf

```
wrk043.th: wrk051_hol.flag wrk043.doc wrk043.sml
             hol -d $(EHOLDB) -f wrk051_delete_theories_wrk043.sml -f wrk043.sml
             touch wrk043.th
```

15 MAKING FINITENESS THEOREMS (wrk044)

Text dumped to file wrk051_delete_theories_wrk044.sml

```
(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory "fin_thms" handle Fail - => ());
```

Text appended to file wrk051.mkf

```
wrk044.th: wrk051_hol.flag wrk044.doc wrk044.sml
             hol -d $(EHOLDB) -f wrk051_delete_theories_wrk044.sml -f wrk044.sml
             touch wrk044.th
```

16 MAKING FINITENESS THEOREMS (wrk046)

Text dumped to file wrk051_delete_theories_wrk046.sml

```
(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory "lib_thms" handle Fail - => ());
```

Text appended to file wrk051.mkf

```
wrk046.th: wrk051_hol.flag wrk046.doc wrk046.sml
             hol -d $(EHOLDB) -f wrk051_delete_theories_wrk046.sml -f wrk046.sml
             touch wrk046.th
```

17 MAKING CASE STUDY (wrk050)

Text dumped to file wrk051_delete_theories_wrk050.sml

```
(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory "wrk050" handle Fail - => ());
```

Text appended to file wrk051.mkf

```
wrk050.th: wrk051_zed.flag wrk050.doc wrk050.sml
             zed -d $(EZEDDB) -f wrk051_delete_theories_wrk050.sml -f wrk050.sml
             touch wrk050.th
```

wrk050.dvi is created by the standard rule for *.dvi* files.

18 CREATE ALL DATABASES

The following create and populates the example HOL and Z databases.

Text appended to file wrk051.mkf

```
all_hol: spc001.th spc002.th usr004.tutorial usr013.exercises \
          wrk022.th wrk044.th wrk043.th wrk046.th
          @echo All module tests passed.

all_zed: usr011.zexercises wrk050.th
          @echo All module tests passed.

all : all_hol all_zed
```

all can be “undone” by *clean_examples*.

19 TIDYING UP THE FILE STORE

Clean out all example database material that might be created during an execution of this makefile:

Text appended to file wrk051.mkf

```
clean_examples :
    rm -f $(EHOLDB) $(EZEDDB)
    rm -f wrk051_hol.flag wrk051_zed.flag
    rm -f usr004.tutorial usr022.tutorial usr023.tutorial
    rm -f usr011.tutorial usr011.zexercises usr011.zsolutions
    rm -f usr013.tutorial usr013.exercises usr013.solutions
    rm -f spc001.th spc002.th
    rm -f wrk022.th wrk043.th wrk044.th wrk046.th wrk050.th
    rm -f usr022_slides.sml usr023_slides.sml usr004.sml
    rm -f usr011X.sml usr011S.sml
    rm -f usr013X.sml usr013S.sml wrk050.sml
```

To remove all auxiliary files created by “sieving” `wrk051.doc` (we refrain from having a makefile entry delete its own makefile!):

csh

```
rm -f wrk051.mkf wrk051_*.*
```

20 make ENDINGS

The following text causes `make` to output the appropriate message upon finished processing with either a successful or failed “make”.

Text appended to file wrk051.mkf

```
.FAILED :
    @ echo "*** $(FAILURE_MESSAGE) ***" 1>&2
    exit 1

.DONE :
    @ echo "*** $(SUCCESS_MESSAGE) ***" 1>&2
```

21 INDEX

<i>.DONE</i>	20
<i>.FAILED</i>	20
<i>all_hol</i>	19
<i>all_zed</i>	19
<i>all</i>	19
<i>clean_examples</i>	20
<i>spc001.th</i>	17
<i>spc002.th</i>	17
<i>usr004.tutorial</i>	13
<i>usr011.tutorial</i>	15
<i>usr011.zexercises</i>	15
<i>usr011.zsolutions</i>	15
<i>usr011A.doc</i>	12
<i>usr011B.doc</i>	12
<i>usr011C.doc</i>	12
<i>usr011D.doc</i>	12
<i>usr011E.doc</i>	12
<i>usr011T.doc</i>	12
<i>usr013.exercises</i>	16
<i>usr013.solutions</i>	16
<i>usr013.tutorial</i>	16
<i>usr013A.doc</i>	12
<i>usr013B.doc</i>	12
<i>usr013C.doc</i>	12
<i>usr013D.doc</i>	12
<i>usr013E.doc</i>	12
<i>usr013F.doc</i>	12
<i>usr013G.doc</i>	12
<i>usr013H.doc</i>	12
<i>usr022.tutorial</i>	16
<i>usr023.tutorial</i>	17
<i>wrk022.th</i>	18
<i>wrk043.th</i>	18
<i>wrk044.th</i>	18
<i>wrk046.th</i>	19
<i>wrk050.th</i>	19
<i>wrk051_hol.flag</i>	13
<i>wrk051_zed.flag</i>	13