Code installation

- Installation with gmkpack
  - \+ safe
  - – slow, blackbox
- Install using make files
  - \+ fast
  - – unsafe

Install gmkpack

- download and untar (e.g. gmkpack.6.1.tar.gz)
- read `gmkpack.6.1/README`
- update `gmkpack.6.1/arch/ARCHFILE` according to the machine and compiler
- export the following variables in ~/.profile:
  `GMKROOT, ROOTPACK, HOMEPACK, HOMEBIN, GMKFILE, GMKTMP`
- run `build_gmkpack` script
- add `$GMKROOT/util` to your path
  and `$GMKROOT/man` to your man path

INFO:
http://www.cnrm.meteo.fr/gmapdoc/meshtml/Compiler/gmkpack.html

Operator (aladin): proprietor of **main packs**. Compile whole model.
Users: **user packs** built upon main pack. (compiling modifications)

local settings		
GMKROOT	directory of gmkpack	/users/aladin/gmkpack.6.1
ROOTPACK	main packs	/users/aladin/workdir/pack
GMKFILE	config file (compiler options)	$GMKROOT/arch/IBMP690.RS6K
HOMEPACK	user's packs	$HOME/workdir/pack
HOMEBIN	directory of executable	$HOME/workdir/pack
GMKTMP	tmp dir during compilation	$HOME/workdir/tmp
PATH	gmkpack binaries	$PATH:$GMKROOT/util
MANPATH		$MANPATH:$GMKROOT/man

Create main pack

`gmkpack` $\underbrace{-\text{r cy28t3}}_{release}$ $\underbrace{-\text{b aldodb\_main}}_{branch}$ $\underbrace{-\text{a}}_{main}$ $\underbrace{-\text{l regatta}}_{compiler}$ $\underbrace{-\text{o x}}_{exe}$ $\underbrace{-\text{p aldodb}}_{proj}$

Copy source under `src/local/` (according to project tree)

~aladin/workdir/pack/cy28t3_aldodb_main.01.regatta.x/

- ics_aldodb (compilation script)
- **bin/** ALDODB (final executable)
- **lib/** lib *project* .local.a (static archive lib)
- **sys/** odb98.x (to convert sql to c)
- **src/**
  - **unsxref/** (files for undefined symbols)
  - **local/**
    - **.intfb/** *.intfb.h (explicit interfaces)
    - **project/** *.F90, *.F, *.c (source)

Compile main pack

To compile the pack simply run **ics_aldodb** script

- Creates system program (odb98.x: convert sql file to .c)
- Creates explicit interface blocks (*.intfb.h)
- Determines include and module pathes
- Determines ordered list of subroutines → different levels of compilation
- Compiles subroutines
- Creates static archive libraries
- Links, and creates executable

`INTFB_PROJLIST="arp ald"`	projects to handle with expl. interface
`ICS_RECURSIVE_UPDATE=no`	compile everything
`ICS_ICFMODE=yes`	compile according to the compilation list
`ICS_START, ICS_STOP`	select first and last level for compilation
Reduced starting list	add explicitely subroutines to (re)compile
`ICS_UPDLIBS=yes`	create static libraries (lib*.a file)
`Load=yes`	create binary

Create local pack

To compile some modification → create user pack:

gmkpack −r cy28t3 −b aldodb_main −v 01 −u mypack -l regatta -o x -p aldodb

(−r cy28t3 −b aldodb_main −v 01: according to main pack; −u mypack: pack name)

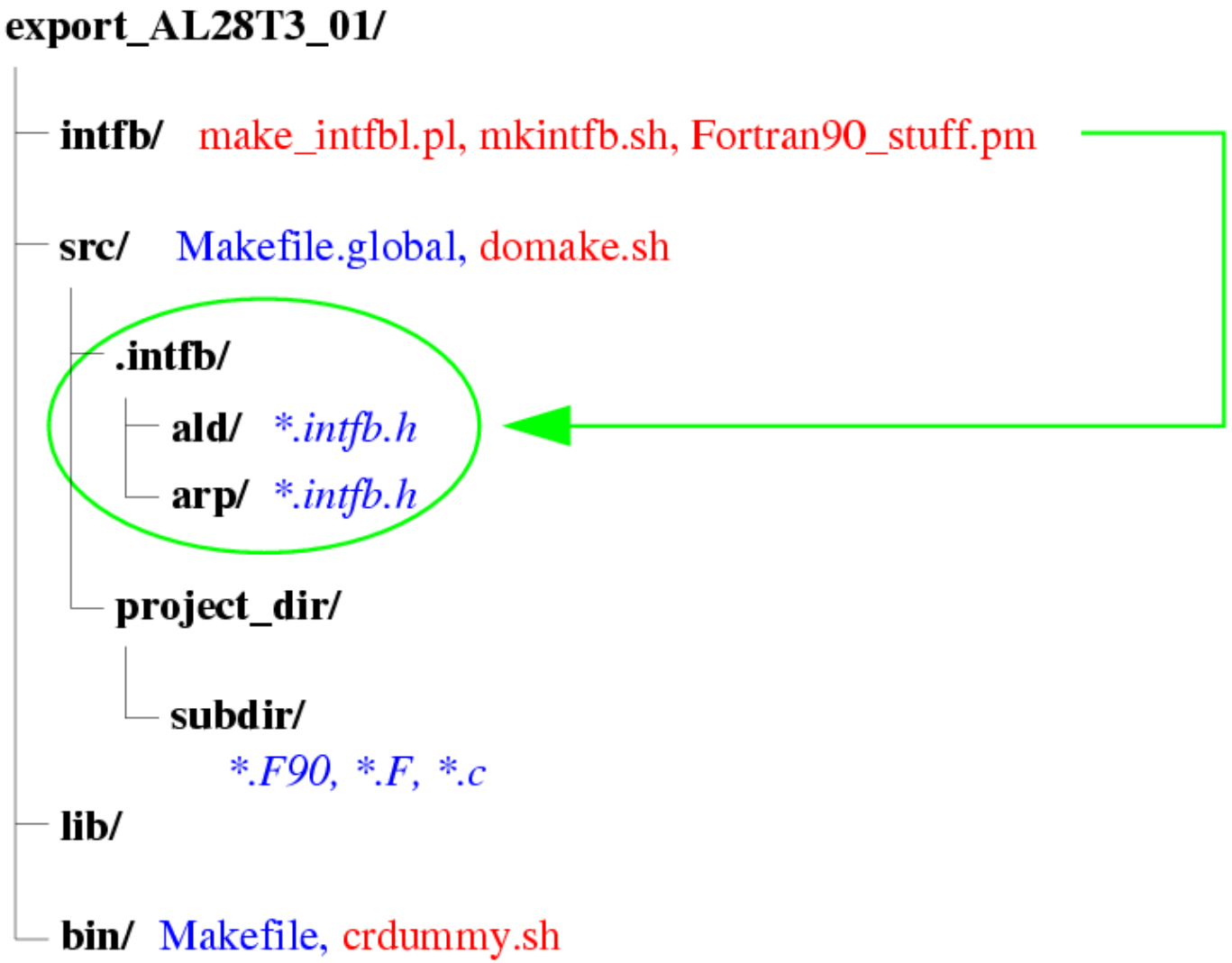
Compilation script: ics_aldodb

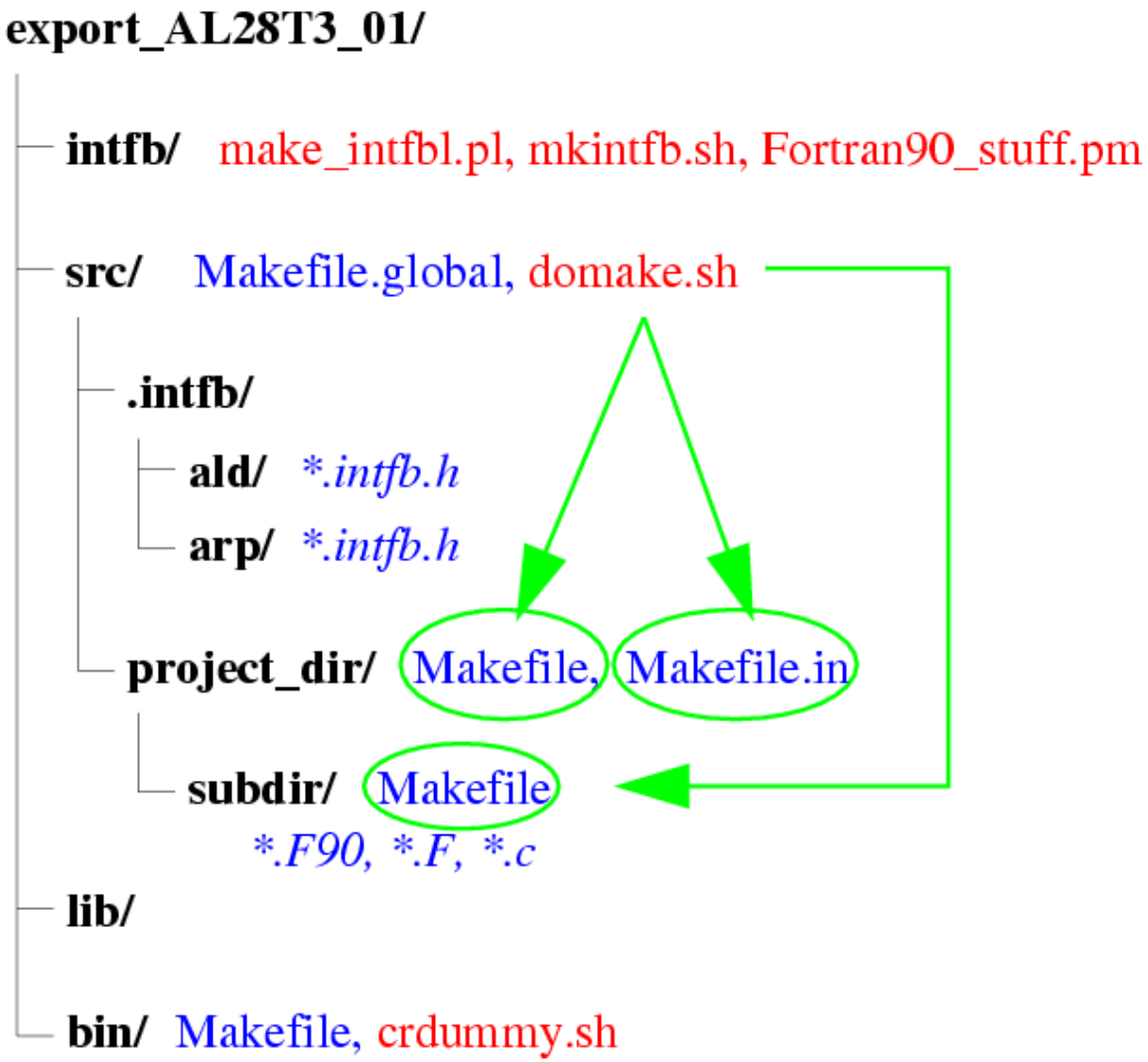
some parameters:

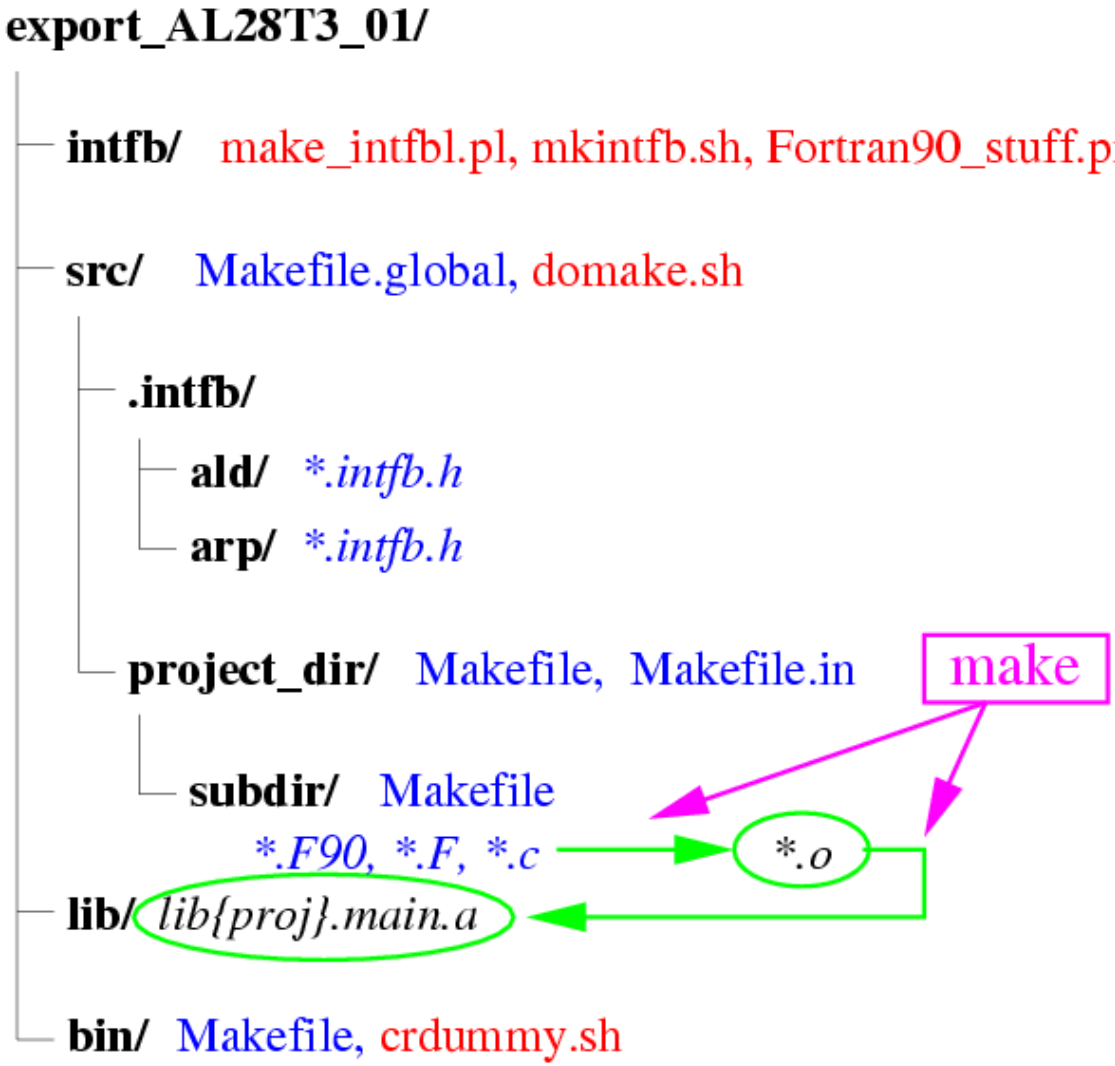
Ofrt=3	optimization level
Dep=yes	compile dependencies (i.e. when modifying a module)
ICS_RECURSIVE_UPDATE=yes	recursive update (only modified routines are compiled)
ICS_ICFMODE=no	do not specify which level to compile
ICS_UPDLIBS=yes	create libraries (lib*.a file)
Load=yes	create a binary

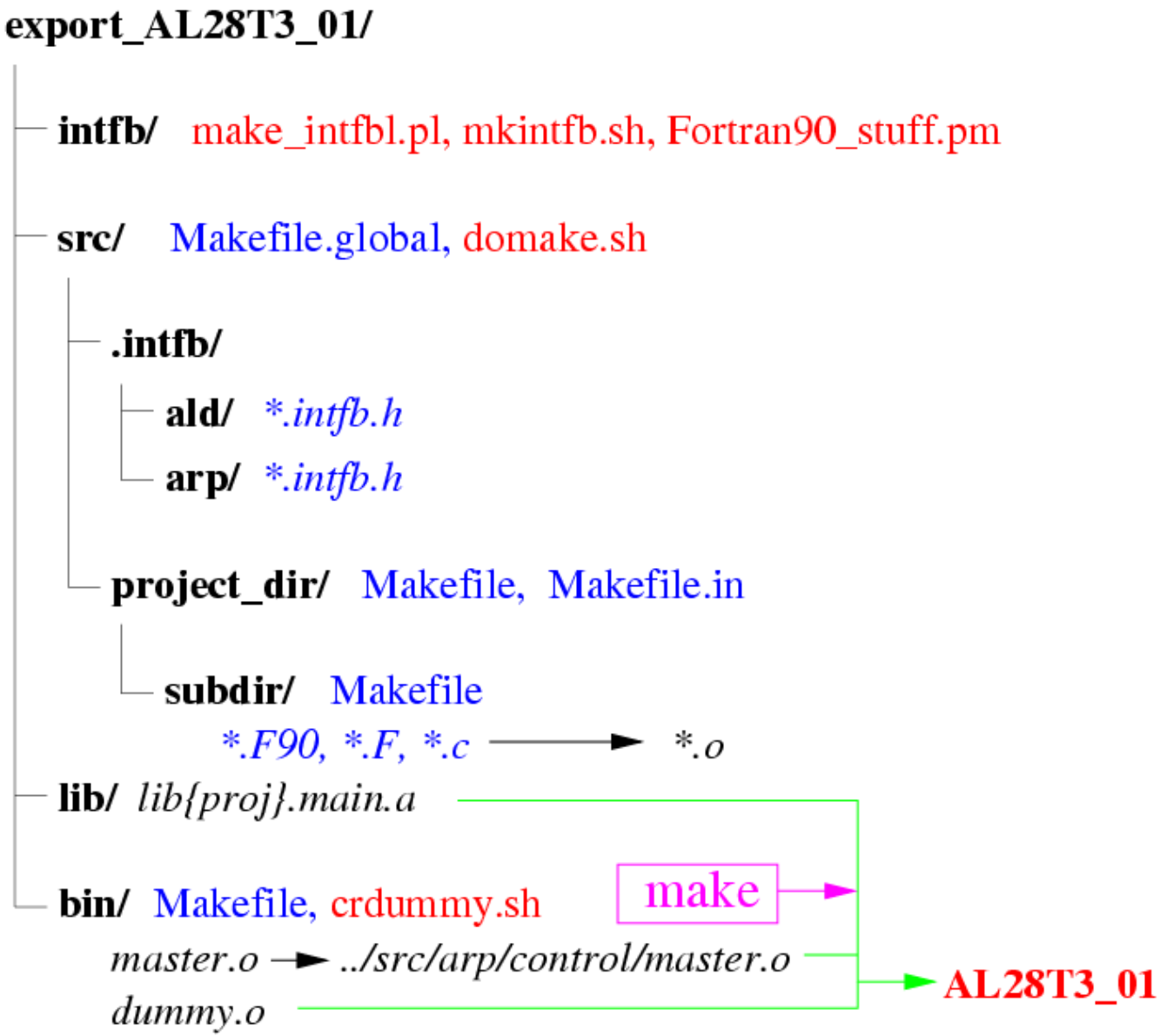
Installation using make files

- Create **explicit interfaces** with perl script in gmkpack: make_intfbl.pl
- Create **Makefiles** for every project and their subdirectories (domake.sh)
  - Modules are sorted according their dependencies (no *make -i* needed)
  - Run make for every project $\Rightarrow$ *lib*{*proj*}*.main.a* archive
- Creating dummy routines for undefined symbols (crdummy.sh)
  - Link without dummy routines $\Rightarrow$ error file containing undefined symbols
  - run crsummy.sh $\Rightarrow$ *dummy.f* file containing dummy subroutines
  - compile *dummy.f* + link $\Rightarrow$ executable
- Compilation of odb in separate step (using odb/build/build_odb script)


export_AL28T3_01/
intfb/ make_intfbl.pl, mkintfb.sh, Fortran90_stuff.pm
src/ Makefile.global, domake.sh
.intfb/
ald/ *.intfb.h
arp/ *.intfb.h
project_dir/
subdir/
*.F90, *.F, *.c
lib/
bin/ Makefile, crdummy.sh


export_AL28T3_01/
intfb/ make_intfbl.pl, mkintfb.sh, Fortran90_stuff.pm
src/ Makefile.global, domake.sh
.intfb/
ald/ *.intfb.h
arp/ *.intfb.h
project_dir/ Makefile, Makefile.in
subdir/ Makefile
*.F90, *.F, *.c
lib/
bin/ Makefile, crdummy.sh


export_AL28T3_01/
intfb/ make_intfbl.pl, mkintfb.sh, Fortran90_stuff.pm
src/ Makefile.global, domake.sh
.intfb/
ald/ *.intfb.h
arp/ *.intfb.h
project_dir/ Makefile, Makefile.in
make
subdir/ Makefile
*.F90, *.F, *.c
*.o
lib/ lib{proj}.main.a
bin/ Makefile, crdummy.sh


export_AL28T3_01/
intfb/ make_intfbl.pl, mkintfb.sh, Fortran90_stuff.pm
src/ Makefile.global, domake.sh
.intfb/
ald/ *.intfb.h
arp/ *.intfb.h
project_dir/ Makefile, Makefile.in
subdir/ Makefile
*.F90, *.F, *.c
*.o
lib/ lib{proj}.main.a
bin/ Makefile, crdummy.sh
make
master.o
../src/arp/control/master.o
dummy.o
AL28T3_01