



# LearnIT Express is on its way!

- The demonstration will start at 12:30 pm.
- To use the Live Chat:
  - Maximize your browser window so you can see the postings.
  - If you refresh the browser page, you'll need to log in again.
  - After the demonstration, staff will remain available in the chat for a few minutes to answer additional questions.

*Today's topic:*  
**Mills HPC Cluster**

# LearnIT Express: Makefiles: Getting it right on Mills



# Objectives

- Learn what is a package.
- Learn where packages are installed.
- Learn how to use VALET for group packages, and with the make utility to build your packages.

# What is a package?

A package usually has a root directory which contains:

- Source files, usually in `src`
- Executable files, usually in `bin`
- Library files, usually in `lib`
- Documentation, in `man`, `info` or `doc`.
- Resources, may be in `lib` too.

# Where are packages installed?

- `/opt/shared/` - VALET packages
  - `vpkg_list`
- `/lustre/work/` - Group packages
- `/lustre/scratch/` - Public package you want to try.
- `$TMPDIR` - Temporary installs!  
(reinstall for each run)



# Make utility

## *Package root directory*

Should have a makefile that describes the relationship among files in your package, and provides rules for updating each file.

See `info make`

# Basic makefile

```
objects = program.o foo.o utils.o
program : $(objects)
    cc -o program $(objects)

$(objects) : defs.h
```

**Variable = value**

**Target : Prerequisites**

**Rules** (Each rule line must begin with a tab)

If there is no rule, the implicit rules apply using the suffix of the file name.



# Implicit Rule Variables

Fortran file .`f`

FC and FFLAGS

C file .`c`

CC and CFLAGS

Both C and Fortran

CPPFLAGS and LDFLAGS

Object files .`o`

LDLIBS



# What does VALET do?

1. Sets executable path; don't need full path names for executables (e.g.)  
setting `$ (FC)` or `$ (CC)` .
2. Sets the library path; don't need full path names to find libraries (e.g.)  
after `-L`
3. Sets the `CPPFLAGS`; uses  
implicit rules. **Only when using**  
`vpkg_devrequire`

# Makefile changes

- Look for full paths in general
- Look for library paths `-L`
- Look for include paths `-I`

and ...

Remove them and use VALET.

# Example 1 - executable

```
F77 = /home/software/intel/composer_xe_2011_sp1.8.273/bin/intel64/ifort
```

Change to

```
F77 = ifort
```

Better to globally change `F77` to use the implicit rule variable `FC`.

**Note:** Valid for `vpkg_require` **or** `vpkg_devrequire`

# Example 2 - lib directory

```
FLAGS = -L/opt/shared/ACML/5.1.0/gfortran64/lib -lacml
```

Change to

```
FLAGS = -lacml
```

Better to globally define all libraries using the implicit rule variables `LDFLAGS` or `LDLIBS`.

**Note:** Valid for `vpkg_require` or `vpkg_devrequire`

# Example 3 - include directory

```
CFLAGS = -o2 -I/opt/shared/ACML/5.1.0/gfortran64/include
```

Change to

```
CFLAGS = -o2 $(CPPFLAGS)
```

Better to use the implicit rule variable `CPPFLAGS` for all include directories.

Note: Valid for `vpkg_devrequire` only.



# Summary

- Using the make utility makes it easy to manage your packages.
- Using VALET with the make utility simplifies your makefiles.



# More information

- **IT Research Computing**

<http://www.it.udel.edu/research-computing>

- - Alerts
  - Announcements
  - Resources: Training

# Contact the IT Support Center

- **Email:** [consult@udel.edu](mailto:consult@udel.edu)

If you make the first line of the e-mail message

Type=Cluster-Mills

your question will be routed more quickly.

- **Phone:** (302) 831-6000
- **Text:** (302) 722-6820

