BALTIC ONLINE

GAR
Advanced Topics

OpenCSW Technical Summercamp
Oslo 2009

Dagobert Michelsen
dam@opencsw.org

Topic:

Modulations

„A **Modulation** is the process of varying one waveform in relation to another waveform“

„A **Modulation** is the process of varying one waveform in relation to another waveform“

For software builds that means:

„A **Modulation** is the process of varying a build in relation to certain parameters“

Phases

Phases

* Prerequisites

Phases

* Prerequisites
* Fetch

Phases

* Prerequisites
* Fetch
* Checksum

Phases

* Prerequisites
* Fetch
* Checksum
* Extract

Phases

* Prerequisites
* Fetch
* Checksum
* Extract
* Patch

Phases

* Prerequisites
* Fetch
* Checksum
* Extract
* Patch
* Configure

Phases

* Prerequisites
* Fetch
* Checksum
* Extract
* Patch
* Configure
* Build

Phases

* Prerequisites
* Fetch
* Checksum
* Extract
* Patch
* Configure
* Build
* Test

Phases

* Prerequisites
* Fetch
* Checksum
* Extract
* Patch
* Configure
* Build
* Test
* Install

Phases

* Prerequisites
* Fetch
* Checksum
* Extract
* Patch
* Configure
* Build
* Test
* Install
* Merge

Phases

* Prerequisites
* Fetch
* Checksum
* Extract
* Patch
* Configure
* Build
* Test
* Install
* Merge
* Package

Phases

* Prerequisites
* Fetch
* Checksum
* Extract
* Patch
* Configure
* Build
* Test
* Install
* Merge
* Package

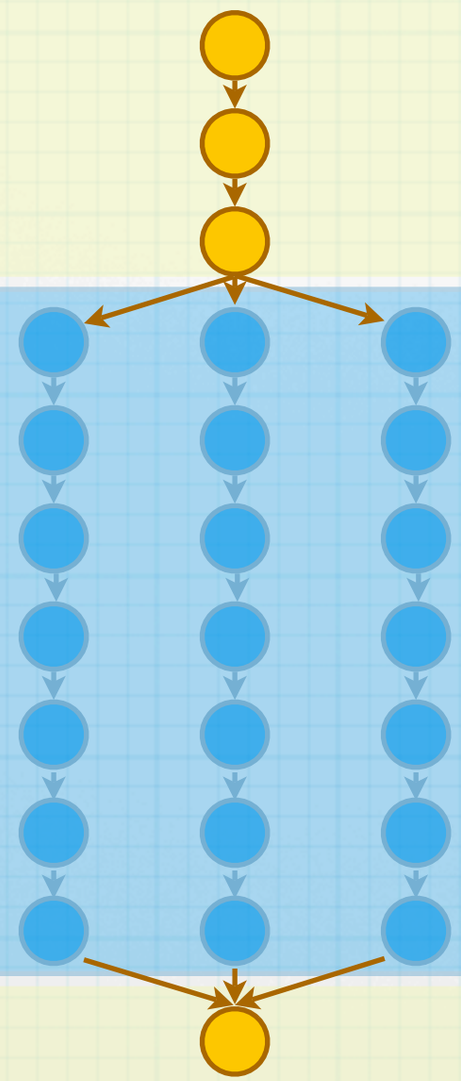
Executed globally

Executed inside modulation

Executed globally

Phases

* Prerequisites
* Fetch
* Checksum
* Extract
* Patch
* Configure
* Build
* Test
* Install
* Merge
* Package



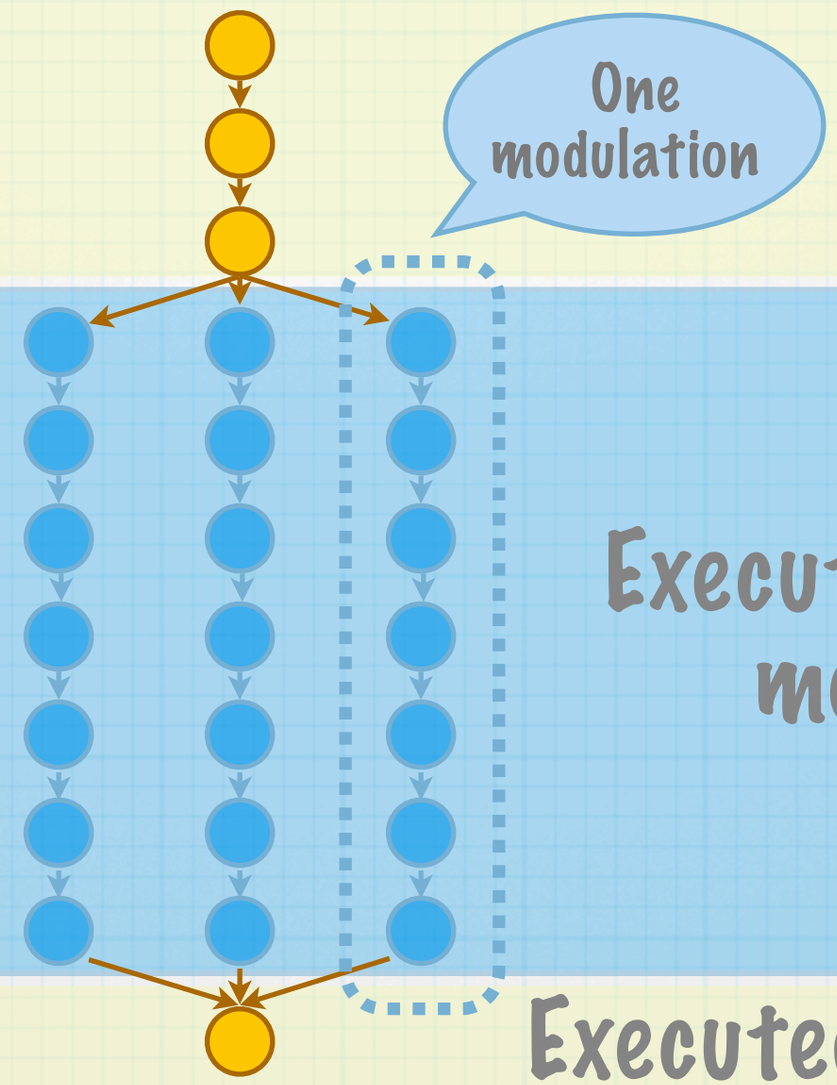
Executed globally

Executed inside modulation

Executed globally

Phases

* Prerequisites
* Fetch
* Checksum
* Extract
* Patch
* Configure
* Build
* Test
* Install
* Merge
* Package



Executed globally

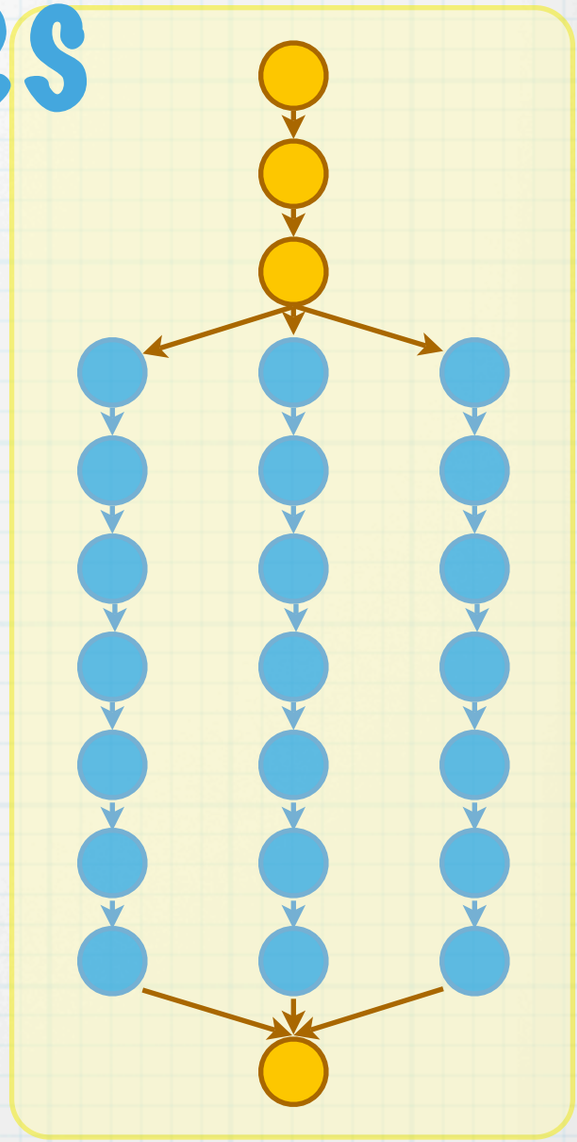
Executed inside modulation

Executed globally

Modulations and pre-/post-phases

* When is `pre-build` called?
  * Technically ran before first build, but after every configure
* Problem: each phase should be runnable independently
* Often sufficient: To be run per modulation
  `pre-build-modulated`
* `pre-/post-<phase>` is likely to be replaced with `pre-/post-<phase>-modulated`

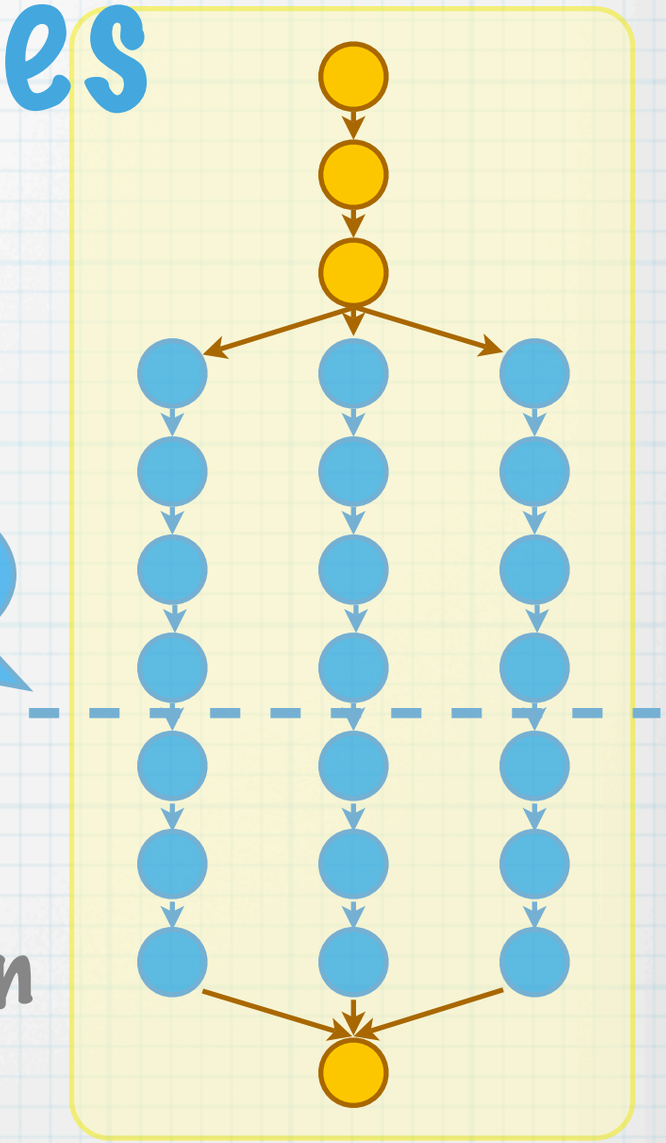
Modulations and pre-/post-phases



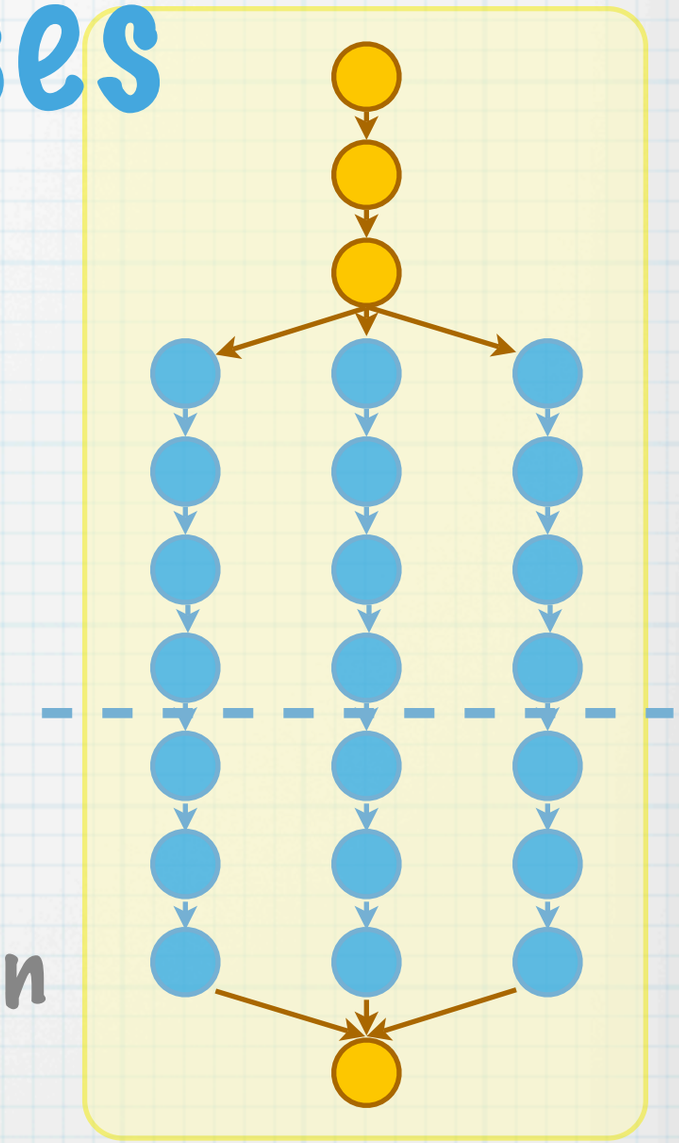
* When is `pre-build` called?
  * Technically ran before first build, but after every configure
* Problem: each phase should be runnable independently
* Often sufficient: To be run per modulation
  `pre-build-modulated`
* `pre-/post-<phase>` is likely to be replaced with `pre-/post-<phase>-modulated`

Modulations and pre-/post-phases

* When is `pre-build` called?
  * Technically ran before first build, but after every configure
* Problem: each phase should be runnable independently
* Often sufficient: To be run per modulation
  `pre-build-modulated`
* `pre-/post-<phase>` is likely to be replaced with `pre-/post-<phase>-modulated`



Modulations and pre-/post-phases



* When is `pre-build` called?
  * Technically ran before first build, but after every configure
* Problem: each phase should be runnable independently
* Often sufficient: To be run per modulation
  `pre-build-modulated`
* `pre-/post-<phase>` is likely to be replaced with `pre-/post-<phase>-modulated`

What can be used as modulator?

* ISA for build architecture
  For 32/64 bit versions and optimized binaries.
  See lame
* Package version
  Useful when multiple versions of a library should be included
  See automake
* ...anything that controls the build
  * Static vs. dynamic linked versions
    See wget
  * Parameter size
    See nrpe
  * Wide vs. narrow characters
    See ncurses

Modulation over ISA

* Allows easy building of multiple ISAs
* Built-in special case of modulation
  * No merge needed also
* Simplest case

  `build64 = 1`
* Advanced

  `EXTRA_BUILD_ISAS = sparcv8plus+vis pentium_pro`

Verify modulations over ISA

```
build8s% gmake modenv
          Arch: sparc
        Kernel: sparcv9

Default ISA 32: sparcv8
Default ISA 64: sparcv9

Requested ISAs: sparcv8 sparcv9 i386 amd64 sparcv8plus+vis pentium_pro
   Needed ISAs: sparcv8 sparcv9 sparcv8plus+vis
    Build ISAs: sparcv8 sparcv9 sparcv8plus+vis

  ISAEXEC dirs: /opt/csw/bin /opt/csw/sbin /opt/csw/libexec
 ISAEXEC files: /opt/csw/bin/lame

 Merge include:
 Merge exclude:     /opt/csw/share/info/dir /opt/csw/lib/.*\.la .*\~ /opt/csw/lib/.*\.a

    Modulators: ISA
   Modulations: isa-sparcv8 isa-sparcv9 isa-sparcv8plus+vis

Requested compiler flags:

* Modulation isa-sparcv8: ISA=sparcv8
           PATH = /home/dam/mgar/pkg/lame/trunk/work/install-isa-sparcv8/opt/csw/bin:/home/dam/mgar/pkg/lame/
trunk/work/install-isa-sparcv8/opt/csw/bin:/home/dam/mgar/pkg/lame/trunk/work/install-isa-sparcv8/opt/csw/sbin:/
home/dam/mgar/pkg/lame/trunk/work/install-isa-sparcv8/opt/csw/sbin:/opt/csw/bin:/opt/csw/bin:/opt/csw/sbin:/opt/
csw/sbin:/opt/studio/SOS11/SUNWspro/bin:/home/dam/mgar/pkg/lame/trunk/gar/bin:/usr/bin:/usr/sbin:/usr/java/bin:/
usr/ccs/bin:/usr/openwin/bin
PKG_CONFIG_PATH = /opt/csw/lib/pkgconfig
         CFLAGS = -xO3 -xarch=v8 -I/opt/csw/include
       CXXFLAGS = -xO3 -xarch=v8 -I/opt/csw/include
       CPPFLAGS = -I/opt/csw/include
        LDFLAGS = -xarch=v8 -L/opt/csw/lib
     LD_OPTIONS = -R/opt/csw/lib/$ISALIST -R/opt/csw/lib
        ASFLAGS =
```

Modulation over ISA requirements

* Certain ISAs require specific platforms
  x64 requires Solaris 10, but integrated package for Solaris 8-10
* Parallelizing modulation builds
  may require builds on multiple hosts (work-in-progress)
* Build platform for each modulation must be specified in the package manifest (work-in-progress)
* How does ISAEXEC relate to multi-ISA?

Modulation over version

```
GARNAME = automake
GARVERSION = 1.11
...
EXTRA_MODULATORS = GARVERSION
MODULATIONS_GARVERSION = 1.6.3 1.7.9 1.8.5 1.9.6 1.10.2 1.11
...
SOURCEFILES =   $(foreach VERSION,$(MODULATIONS_GARVERSION),
                  $(GARNAME)-$(VERSION).tar.gz
                 )
DISTFILES = $(SOURCEFILES)
...
NOEXTRACT = $(filter-out $(GARNAME)-$(GARVERSION).tar.gz, $(SOURCEFILES))
...
$(foreach VERSION,$(MODULATIONS_GARVERSION),
  $(eval MERGE_SCRIPTS_isa-$(ISA)-garversion-$(VERSION) = copy-all)
)
```

Modulation over version

```
GARNAME = automake
GARVERSION = 1.11
...
EXTRA_MODULATORS = GARVERSION
MODULATIONS_GARVERSION = 1.6.3 1.7.9 1.8.5 1.9.6 1.10.2 1.11
...
SOURCEFILES =   $(foreach VERSION,$(MODULATIONS_GARVERSION),
                  $(GARNAME)-$(VERSION).tar.gz
                )
DISTFILES = $(SOURCEFILES)
...
NOEXTRACT = $(filter-out $(GARNAME)-$(GARVERSION).tar.gz, $(SOURCEFILES))
...
$(foreach VERSION,$(MODULATIONS_GARVERSION),
  $(eval MERGE_SCRIPTS_isa-$(ISA)-garversion-$(VERSION) = copy-all)
)
```

Adds modulator with this name

Modulation over version

```
GARNAME = automake
GARVERSION = 1.11
...
EXTRA_MODULATORS = GARVERSION
MODULATIONS_GARVERSION = 1.6.3 1.7.9 1.8.5 1.9.6 1.10.2 1.11
...
SOURCEFILES =   $(foreach VERSION,$(MODULATIONS_GARVERSION),
                  $(GARNAME)-$(VERSION).tar.gz
                )
DISTFILES = $(SOURCEFILES)
...
NOEXTRACT = $(filter-out $(GARNAME)-$(GARVERSION).tar.gz, $(SOURCEFILES))
...
$(foreach VERSION,$(MODULATIONS_GARVERSION),
  $(eval MERGE_SCRIPTS_isa-$(ISA)-garversion-$(VERSION) = copy-all)
)
```

Adds modulator with this name

One file per modulated version

Modulation over version

```
GARNAME = automake
GARVERSION = 1.11
...
EXTRA_MODULATORS = GARVERSION
MODULATIONS_GARVERSION = 1.6.3 1.7.9 1.8.5 1.9.6 1.10.2 1.11
...
SOURCEFILES =   $(foreach VERSION,$(MODULATIONS_GARVERSION),
                  $(GARNAME)-$(VERSION).tar.gz
                )
DISTFILES = $(SOURCEFILES)
...
NOEXTRACT = $(filter-out $(GARNAME)-$(GARVERSION).tar.gz, $(SOURCEFILES))
...
$(foreach VERSION,$(MODULATIONS_GARVERSION),
  $(eval MERGE_SCRIPTS_isa-$(ISA)-garversion-$(VERSION) = copy-all)
)
```

Adds modulator with this name

One file per modulated version

Make sure only the needed file is extracted in each modulation

Modulation over version

```
GARNAME = automake
GARVERSION = 1.11
...
EXTRA_MODULATORS = GARVERSION
MODULATIONS_GARVERSION = 1.6.3 1.7.9 1.8.5 1.9.6 1.10.2 1.11
...
SOURCEFILES =   $(foreach VERSION,$(MODULATIONS_GARVERSION),
                  $(GARNAME)-$(VERSION).tar.gz
                )
DISTFILES = $(SOURCEFILES)
...
NOEXTRACT = $(filter-out $(GARNAME)-$(GARVERSION).tar.gz, $(SOURCEFILES))
...
$(foreach VERSION,$(MODULATIONS_GARVERSION),
  $(eval MERGE_SCRIPTS_isa-$(ISA)-garversion-$(VERSION) = copy-all)
)
```

Adds modulator with this name

One file per modulated version

Make sure only the needed file is extracted in each modulation

Merge-rules needed!
We'll see about this later

Modulation over static/dynamic linking

```
GARNAME = wget
...
EXTRA_MODULATORS = STATIC
MODULATIONS_STATIC = disable enable

EXTRA_CONFIGURE_STATIC-disable = --with-ssl
EXTRA_CONFIGURE_STATIC-enable = --disable-shared
EXTRA_CONFIGURE_STATIC-enable += --enable-static
EXTRA_CONFIGURE_STATIC-enable += --without-ssl
...
CONFIGURE_ARGS += $(EXTRA_CONFIGURE_STATIC-$(STATIC))

MERGE_SCRIPTS_isa-sparcv8-static-disable = copy-all
 MERGE_SCRIPTS_isa-sparcv8-static-enable = rename-wget copy-only
    MERGE_DIRS_isa-sparcv8-static-enable = $(bindir)

    MERGE_SCRIPTS_isa-i386-static-disable = copy-all
     MERGE_SCRIPTS_isa-i386-static-enable = rename-wget copy-only
        MERGE_DIRS_isa-i386-static-enable = $(bindir)

merge-rename-wget:
        @mv $(DESTDIR)$(bindir)/wget $(DESTDIR)$(bindir)/wget.static
...
include gar/category.mk
```

Modulation over static/dynamic linking

Adds modulator with this name

```
GARNAME = wget
...
EXTRA_MODULATORS = STATIC
MODULATIONS_STATIC = disable enable

EXTRA_CONFIGURE_STATIC-disable = --with-ssl
EXTRA_CONFIGURE_STATIC-enable = --disable-shared
EXTRA_CONFIGURE_STATIC-enable += --enable-static
EXTRA_CONFIGURE_STATIC-enable += --without-ssl
...
CONFIGURE_ARGS += $(EXTRA_CONFIGURE_STATIC-$(STATIC))

MERGE_SCRIPTS_isa-sparcv8-static-disable = copy-all
 MERGE_SCRIPTS_isa-sparcv8-static-enable = rename-wget copy-only
    MERGE_DIRS_isa-sparcv8-static-enable = $(bindir)

    MERGE_SCRIPTS_isa-i386-static-disable = copy-all
     MERGE_SCRIPTS_isa-i386-static-enable = rename-wget copy-only
        MERGE_DIRS_isa-i386-static-enable = $(bindir)

merge-rename-wget:
        @mv $(DESTDIR)$(bindir)/wget $(DESTDIR)$(bindir)/wget.static
...
include gar/category.mk
```

Modulation over static/dynamic linking

Adds modulator with this name

Cool trick:
Use dynamically expanded variables to set per-modulation

```
GARNAME = wget
...
EXTRA_MODULATORS = STATIC
MODULATIONS_STATIC = disable enable

EXTRA_CONFIGURE_STATIC-disable = --with-ssl
EXTRA_CONFIGURE_STATIC-enable = --disable-shared
EXTRA_CONFIGURE_STATIC-enable += --enable-static
EXTRA_CONFIGURE_STATIC-enable += --without-ssl
...
CONFIGURE_ARGS += $(EXTRA_CONFIGURE_STATIC-$(STATIC))

MERGE_SCRIPTS_isa-sparcv8-static-disable = copy-all
 MERGE_SCRIPTS_isa-sparcv8-static-enable = rename-wget copy-only
    MERGE_DIRS_isa-sparcv8-static-enable = $(bindir)

    MERGE_SCRIPTS_isa-i386-static-disable = copy-all
     MERGE_SCRIPTS_isa-i386-static-enable = rename-wget copy-only
        MERGE_DIRS_isa-i386-static-enable = $(bindir)

merge-rename-wget:
        @mv $(DESTDIR)$(bindir)/wget $(DESTDIR)$(bindir)/wget.static
...
include gar/category.mk
```

Modulation over static/dynamic linking

```
GARNAME = wget
...
EXTRA_MODULATORS = STATIC
MODULATIONS_STATIC = disable enable
```

Adds modulator with this name

Cool trick:
Use dynamically expanded variables to set per-modulation

```
EXTRA_CONFIGURE_STATIC-disable = --with-ssl
EXTRA_CONFIGURE_STATIC-enable = --disable-shared
EXTRA_CONFIGURE_STATIC-enable += --enable-static
EXTRA_CONFIGURE_STATIC-enable += --without-ssl
...
CONFIGURE_ARGS += $(EXTRA_CONFIGURE_STATIC-$(STATIC))
```

```
MERGE_SCRIPTS_isa-sparcv8-static-disable = copy-all
 MERGE_SCRIPTS_isa-sparcv8-static-enable = rename-wget copy-only
    MERGE_DIRS_isa-sparcv8-static-enable = $(bindir)

    MERGE_SCRIPTS_isa-i386-static-disabl
     MERGE_SCRIPTS_isa-i386-static-en
        MERGE_DIRS_isa-i386-static-ena

merge-rename-wget:
        @mv $(DESTDIR)$(bindir)/wget $(DESTDIR)$(bindir)/wget.static
...
include gar/category.mk
```

Why is this a bad idea?
Think `gmake remerge`

Modulation over static/dynamic linking

Modulation over static/dynamic linking

Modulation over static/dynamic linking

```
GARNAME = wget
...
EXTRA_MODULATORS = STATIC
MODULATIONS_STATIC = disable enable

EXTRA_CONFIGURE_STATIC-disable = --with-ssl
EXTRA_CONFIGURE_STATIC-enable = --disable-shared
EXTRA_CONFIGURE_STATIC-enable += --enable-static
EXTRA_CONFIGURE_STATIC-enable += --without-ssl
...
CONFIGURE_ARGS += $(EXTRA_CONFIGURE_STATIC-$(STATIC))
...
MERGE_SCRIPTS_isa-sparcv8-static-disable = copy-all
 MERGE_SCRIPTS_isa-sparcv8-static-enable = copy-static-wget-only

   MERGE_SCRIPTS_isa-i386-static-disable = copy-all
    MERGE_SCRIPTS_isa-i386-static-enable = copy-static-

merge-copy-static-wget-only:
        ginstall $(INSTALLISADIR)$(bindir)/wget $(PKGROOT)$(bindir)/wget.static
            ...
include gar/category.mk
```

Always copy from one phase to another

Modulation over parameter size

```
GARNAME = nrpe
...
EXTRA_MODULATORS = SIZE
MODULATIONS_SIZE = 1k 8k
...
PATCHFILES_isa-sparcv8-size-8k = common.h.diff
PATCHFILES_isa-i386-size-8k = common.h.diff
...
MERGE_SCRIPTS_isa-sparcv8-size-1k = copy-all
MERGE_SCRIPTS_isa-sparcv8-size-8k = copy-nrpe-only

MERGE_SCRIPTS_isa-i386-size-1k = copy-all
MERGE_SCRIPTS_isa-i386-size-8k = copy-nrpe-only

include gar/category.mk
...
merge-copy-nrpe-only:
        cp $(INSTALLISADIR)$(bindir)/nrpe_1k $(PKGROOT)$(bindir)/nrpe_8k
        cp $(INSTALLISADIR)$(LIBEXECDIR)/check_nrpe $(PKGROOT)$(LIBEXECDIR)/check_nrpe_8k
        @$(MAKECOOKIE)
```

Modulation over parameter size

Ok, I guess you understood how this works

```
GARNAME = nrpe
...
EXTRA_MODULATORS = SIZE
MODULATIONS_SIZE = 1k 8k
...
PATCHFILES_isa-sparcv8-size-8k = common.h.diff
PATCHFILES_isa-i386-size-8k = common.h.diff
...
MERGE_SCRIPTS_isa-sparcv8-size-1k = copy-all
MERGE_SCRIPTS_isa-sparcv8-size-8k = copy-nrpe-only

MERGE_SCRIPTS_isa-i386-size-1k = copy-all
MERGE_SCRIPTS_isa-i386-size-8k = copy-nrpe-only

include gar/category.mk
...
merge-copy-nrpe-only:
        cp $(INSTALLISADIR)$(bindir)/nrpe_1k $(PKGROOT)$(bindir)/nrpe_8k
        cp $(INSTALLISADIR)$(LIBEXECDIR)/check_nrpe $(PKGROOT)$(LIBEXECDIR)/check_nrpe_8k
        @$(MAKECOOKIE)
```

Modulation over parameter size

Ok, I guess you understood how this works

Use patchfiles on a per-modulation basis

```
GARNAME = nrpe
...
EXTRA_MODULATORS = SIZE
MODULATIONS_SIZE = 1k 8k
...
PATCHFILES_isa-sparcv8-size-8k = common.h.diff
PATCHFILES_isa-i386-size-8k = common.h.diff
...
MERGE_SCRIPTS_isa-sparcv8-size-1k = copy-all
MERGE_SCRIPTS_isa-sparcv8-size-8k = copy-nrpe-only

MERGE_SCRIPTS_isa-i386-size-1k = copy-all
MERGE_SCRIPTS_isa-i386-size-8k = copy-nrpe-only

include gar/category.mk
...
merge-copy-nrpe-only:
        cp $(INSTALLISADIR)$(bindir)/nrpe_1k $(PKGROOT)$(bindir)/nrpe_8k
        cp $(INSTALLISADIR)$(LIBEXECDIR)/check_nrpe $(PKGROOT)$(LIBEXECDIR)/check_nrpe_8k
        @$(MAKECOOKIE)
```

Modulations and their names

* The name of a modulation is

```
<modulator1>-<value1>-<modulator2>-<value2>-...
```

* Call each phase from a modulation manually with

```
<phase>-<modulation>
```

Example:

```
gmake build-isa-sparcv8
```

* This results in a sub-make-invocation

```
<phase>-modulated  MODULATION=<MODULATION>
                   <MODULATOR1>=<value1> ...
```

Example:

```
gmake build-modulated MODULATION=isa-sparcv8
                      ISA=sparcv8
```

Topic:

Merging

Merging
Why is this needed?

* Libraries may be located elsewhere in the package than they were configured
* 32 and 64 bit need different install locations to have sane pkgconfig-files

```
build-isa-sparcv8/opt/csw/bin/myapp
build-isa-sparcv9/opt/csw/bin/64/myapp
```

Install

```
pkgroot/opt/csw/bin/myapp
pkgroot/opt/csw/bin/sparcv9/myapp
```

Package

```
/opt/csw/bin/myapp -> isaexec
/opt/csw/bin/sparcv8/myapp
/opt/csw/bin/sparcv9/myapp
```

Merging default rules

* `copy-all`
  Copy all files and put them in standard locations (e. g. `/csw/bin`)

* `copy-only`
  Copy only the files to be relocated and also put them in standard locations

* `copy-relocate`
  Copy all files and relocate the locations to the ISA-subdirs
  (e. g. `/csw/bin/sparcv9/`)

* `copy-relocated-only`
  Copy only the files to be relocated and also relocate the locations to the ISA-subdirs

Merging default rules

	Copy all	Copy only relocated
Do not relocate	`copy-all`	`copy-only`
Do relocate	`copy-relocate`	`copy-relocated-only`

Merging and renaming

* Rename files on-the-fly during merge
  See curl

```
EXTRA_PAX_ARGS_32 = -s ",^\.$(includedir)/curl/curlbuild.h$$,
                        .$(includedir)/curl/curlbuild-32.h,p"

EXTRA_PAX_ARGS_64 = -s ",^\.$(includedir)/curl/curlbuild.h$$,
                        .$(includedir)/curl/curlbuild-64.h,p"

EXTRA_PAX_ARGS_isa-sparcv8 = $(EXTRA_PAX_ARGS_32)

EXTRA_PAX_ARGS_isa-sparcv9 = $(EXTRA_PAX_ARGS_64)

EXTRA_PAX_ARGS_isa-i386    = $(EXTRA_PAX_ARGS_32)

EXTRA_PAX_ARGS_isa-amd64   = $(EXTRA_PAX_ARGS_64)
```

Merging and renaming

* Rename files on-the-fly during merge

  See curl

From here

```
EXTRA_PAX_ARGS_32 = -s ",^\.$(includedir)/curl/curlbuild.h$$,
                        .$(includedir)/curl/curlbuild-32.h,p"

EXTRA_PAX_ARGS_64 = -s ",^\.$(includedir)/curl/curlbuild.h$$,
                        .$(includedir)/curl/curlbuild-64.h,p"

EXTRA_PAX_ARGS_isa-sparcv8 = $(EXTRA_PAX_ARGS_32)

EXTRA_PAX_ARGS_isa-sparcv9 = $(EXTRA_PAX_ARGS_64)

EXTRA_PAX_ARGS_isa-i386    = $(EXTRA_PAX_ARGS_32)

EXTRA_PAX_ARGS_isa-amd64   = $(EXTRA_PAX_ARGS_64)
```

Merging and renaming

* Rename files on-the-fly during merge

  See curl

From here

To here

```
EXTRA_PAX_ARGS_32 = -s ",^\.$(includedir)/curl/curlbuild.h$$,
                        .$(includedir)/curl/curlbuild-32.h,p"

EXTRA_PAX_ARGS_64 = -s ",^\.$(includedir)/curl/curlbuild.h$$,
                        .$(includedir)/curl/curlbuild-64.h,p"

EXTRA_PAX_ARGS_isa-sparcv8 = $(EXTRA_PAX_ARGS_32)

EXTRA_PAX_ARGS_isa-sparcv9 = $(EXTRA_PAX_ARGS_64)

EXTRA_PAX_ARGS_isa-i386    = $(EXTRA_PAX_ARGS_32)

EXTRA_PAX_ARGS_isa-amd64   = $(EXTRA_PAX_ARGS_64)
```

Merging and renaming

* Rename files on-the-fly during merge

  See curl

```
EXTRA_PAX_ARGS_32 = -s ",^\.$(includedir)/curl/curlbuild.h$$,
                        .$(includedir)/curl/curlbuild-32.h,p"

EXTRA_PAX_ARGS_64 = -s ",^\.$(includedir)/curl/curlbuild.h$$,
                        .$(includedir)/curl/curlbuild-64.h,p"

EXTRA_PAX_ARGS_isa-sparcv8 = $(EXTRA_PAX_ARGS_32)

EXTRA_PAX_ARGS_isa-sparcv9 = $(EXTRA_PAX_ARGS_64)

EXTRA_PAX_ARGS_isa-i386    = $(EXTRA_PAX_ARGS_32)

EXTRA_PAX_ARGS_isa-amd64   = $(EXTRA_PAX_ARGS_64)
```

This is the modulation where the PAX args should be applied

Merging with custom rules

* Only merge specific files
  See nrpe

```
GARNAME = nrpe
...
MERGE_SCRIPTS_isa-sparcv8-size-1k = copy-all
MERGE_SCRIPTS_isa-sparcv8-size-8k = copy-nrpe-only

MERGE_SCRIPTS_isa-i386-size-1k = copy-all
MERGE_SCRIPTS_isa-i386-size-8k = copy-nrpe-only

include gar/category.mk
...
merge-copy-nrpe-only:
        cp $(INSTALLISADIR)$(bindir)/nrpe_1k $(PKGROOT)$(bindir)/nrpe_8k
        cp $(INSTALLISADIR)$(LIBEXECDIR)/check_nrpe $(PKGROOT)$(LIBEXECDIR)/check_nrpe_8k
        @$(MAKECOOKIE)
```

Merging with custom rules

* Only merge specific files
  See nrpe

```
GARNAME = nrpe
...
MERGE_SCRIPTS_isa-sparcv8-size-1k = copy-all
MERGE_SCRIPTS_isa-sparcv8-size-8k = copy-nrpe-only

MERGE_SCRIPTS_isa-i386-size-1k = copy-all
MERGE_SCRIPTS_isa-i386-size-8k = copy-nrpe-only

include gar/category.mk
...
merge-copy-nrpe-only:
        cp $(INSTALLISADIR)$(bindir)/nrpe_1k $(PKGROOT)$(bindir)/nrpe_8k
        cp $(INSTALLISADIR)$(LIBEXECDIR)/check_nrpe $(PKGROOT)$(LIBEXECDIR)/check_nrpe_8k
        @$(MAKECOOKIE)
```

This is the default build
Just copy everything over

Merging with custom rules

* Only merge specific files
  See nrpe

```
GARNAME = nrpe
...
MERGE_SCRIPTS_isa-sparcv8-size-1k = copy-all
MERGE_SCRIPTS_isa-sparcv8-size-8k = copy-nrpe-only

MERGE_SCRIPTS_isa-i386-size-1k = copy-all
MERGE_SCRIPTS_isa-i386-size-8k = copy-nrpe-only

include gar/category.mk
...
merge-copy-nrpe-only:
        cp $(INSTALLISADIR)$(bindir)/nrpe_1k $(PKGROOT)$(bindir)/nrpe_8k
        cp $(INSTALLISADIR)$(LIBEXECDIR)/check_nrpe $(PKGROOT)$(LIBEXECDIR)/check_nrpe_8k
        @$(MAKECOOKIE)
```

This is the default build
Just copy everything over

We want to control exactly what is used
from the additional modulation

Merging with custom rules

* Only merge specific files
  See nrpe

```
GARNAME = nrpe
...
MERGE_SCRIPTS_isa-sparcv8-size-1k = copy-all
MERGE_SCRIPTS_isa-sparcv8-size-8k = copy-nrpe-only

MERGE_SCRIPTS_isa-i386-size-1k = copy-all
MERGE_SCRIPTS_isa-i386-size-8k = copy-nrpe-only

include gar/category.mk
...
merge-copy-nrpe-only:
        cp $(INSTALLISADIR)$(bindir)/nrpe_1k $(PKGROOT)$(bindir)/nrpe_8k
        cp $(INSTALLISADIR)$(LIBEXECDIR)/check_nrpe $(PKGROOT)$(LIBEXECDIR)/check_nrpe_8k
        @$(MAKECOOKIE)
```

Merging copies from the install location to PKGROOT

Questions?
Bueller? Anybody?