4. Specific Packaging Rules

Component templates for build systems

Template files for component Makefiles can be found in directory templates, and available for each supported type build system (autotools, cmake).

Software specific make-rules

External software gates

illumos-gate

gfx-drm

Advanced packaging rules

GCC

Standards are defined according to the specification in standards(5).

Software developed on Linux with GCC as main compiler may not adhere strictly to ANSI/ISO or SUS standards and will cause compilation issues on illumos systems unless the chosen standard is explicitly passed to the preprocessor. The main reason is that GCC defaults to its own standard which is a superset of ANSI/ISO and therefore is not in strict compliance.

The set of *feature macros* is used to set the compliance with a standard, and possibly relax the strict compliance to standards, for example to use non-standard GNU or BSD extensions.

Example:

Macros	Use
`_XOPEN_SOURCE`	Compliance with specified **X/Open** standard: • XPG3: < 4 • XPG4: < 500 • XPG4v2: < 500 and _XOPEN_SOURCE_EXTENDED=1 • XPG5: = 500 • XPG6: = 600 • XPG7: = 700
`__EXTENSIONS__`	If set to 1, enable all extensions beyond the chosen standard compliance which are not in conflict with it.

The build system defines a few helper macros in shared-macros.mk:

Macros	Standard	SUS	POSIX	C	C++
CPP_XPG5MODE	XPG5 + EXTENSIONS	SUSv2	POSIX.1b-1993	c89 + GNU	c++98 + GNU
CPP_XPG6MODE	XPG6 + EXTENSIONS	SUSv3	POSIX.1-2001	c99 + GNU	c++98 + GNU
CPP_XPG7MODE	XPG7 + EXTENSIONS	SUSv4	POSIX.1-2008	c99 + GNU	c++11 + GNU

Example: *Setting the C standard to C99*

```
CFLAGS+= -std=c99 $(CPP_XPG6MODE)
```

The logic for handling feature macros can be reviewed in the file feature_tests.h.

In particular, attention should be paid to the use of C99 which requires XPG6 at least:

c99

```
/*
 * It is invalid to compile an XPG3, XPG4, XPG4v2, or XPG5 application
 * using c99.  The same is true for POSIX.1-1990, POSIX.2-1992, POSIX.1b,
 * and POSIX.1c applications. Likewise, it is invalid to compile an XPG6
 * or a POSIX.1-2001 application with anything other than a c99 or later
 * compiler.  Therefore, we force an error in both cases.
 */
#if defined(_STDC_C99) && (defined(__XOPEN_OR_POSIX) && !defined(_XPG6))
#error "Compiler or options invalid for pre-UNIX 03 X/Open applications \
        and pre-2001 POSIX applications"
#elif !defined(_STDC_C99) && \
        (defined(__XOPEN_OR_POSIX) && defined(_XPG6))
#error "Compiler or options invalid; UNIX 03 and POSIX.1-2001 applications \
        require the use of c99"
#endif
```

Software assuming GNU extensions may use C99 features without actually specifying -std=c99 since -std=gnu89 includes a subset of the C99 language.

How to print built-in specs?

Example for the C language:

```
$ echo | gcc -v -x c -E -
Using built-in specs.
COLLECT_GCC=gcc
Target: i386-pc-solaris2.11
Configured with: /jenkins/jobs/oi-userland/workspace/components/developer/gcc49/gcc-4.9.4/configure CC=/usr/gcc
/4.9/bin/gcc CXX=/usr/gcc/4.9/bin/g++ F77=/usr/gcc/4.9/bin/gfortran FC=/usr/gcc/4.9/bin/gfortran CFLAGS='-g -
O2' CXXFLAGS=' ' FFLAGS=' ' FCFLAGS= LDFLAGS=-m32 PKG_CONFIG_PATH=/usr/lib/pkgconfig --prefix=/usr/gcc/4.9 --
mandir=/usr/gcc/4.9/share/man --bindir=/usr/gcc/4.9/bin --libdir=/usr/gcc/4.9/lib --sbindir=/usr/gcc/4.9/sbin --
sbindir=/usr/gcc/4.9/bin --libdir=/usr/gcc/4.9/lib --libexecdir=/usr/gcc/4.9/lib --host i386-pc-solaris2.11 --
build i386-pc-solaris2.11 --target i386-pc-solaris2.11 --with-boot-ldflags=-R/usr/gcc/4.9/lib --enable-plugins
--enable-objc-gc --enable-languages=c,c++,fortran,lto,objc --without-gnu-ld --with-ld=/usr/bin/ld --with-build-
time-tools=/usr/gnu/i386-pc-solaris2.11/bin --disable-libitm --enable-install-libiberty --with-gnu-as --with-as=
/usr/bin/gas LDFLAGS=-R/usr/gcc/4.9/lib
Thread model: posix
gcc version 4.9.4 (GCC)
COLLECT_GCC_OPTIONS='-v' '-E' '-mtune=generic' '-march=pentium4'
 /usr/gcc/4.9/lib/gcc/i386-pc-solaris2.11/4.9.4/cc1 -E -quiet -v - -mtune=generic -march=pentium4
ignoring nonexistent directory "/usr/local/include"
ignoring nonexistent directory "/usr/gcc/4.9/lib/gcc/i386-pc-solaris2.11/4.9.4/include-fixed"
ignoring nonexistent directory "/usr/gcc/4.9/lib/gcc/i386-pc-solaris2.11/4.9.4/../../../../i386-pc-solaris2.11
/include"
#include "..." search starts here:
#include <...> search starts here:
 /usr/gcc/4.9/lib/gcc/i386-pc-solaris2.11/4.9.4/include
 /usr/gcc/4.9/include
 /usr/include
End of search list.
# 1 "<stdin>"
# 1 "<built-in>"
# 1 "<command-line>"
# 1 "<stdin>"
COMPILER_PATH=/usr/gcc/4.9/lib/gcc/i386-pc-solaris2.11/4.9.4/:/usr/gcc/4.9/lib/gcc/i386-pc-solaris2.11/4.9.4/:
/usr/gcc/4.9/lib/gcc/i386-pc-solaris2.11/:/usr/gcc/4.9/lib/gcc/i386-pc-solaris2.11/4.9.4/:/usr/gcc/4.9/lib/gcc
/i386-pc-solaris2.11/:/usr/ccs/bin/
LIBRARY_PATH=/usr/gcc/4.9/lib/gcc/i386-pc-solaris2.11/4.9.4/:/usr/gcc/4.9/lib/gcc/i386-pc-solaris2.11/4.9.4/..
/../../:/lib/:/usr/lib/
COLLECT_GCC_OPTIONS='-v' '-E' '-mtune=generic' '-march=pentium4'
```

Python

Perl

MPI