Q: How do I integrate .psw files with C++ files? In my Makefile.preamble, I changed the compiler to cc++. Since **pswrap** generates C code, the C++ compiler will not work with .c files generated by **pswrap**. What can I do?

A: You need to create a **Makefile.postamble** and tell **make** to use **cc** when compiling pswrap'ed files. See the example below:

For NEXTSTEP Release 2.0:

```
 .psw.o:
        $(PSWRAP) $(PSWFLAGS) -a -h $*.h -o $*.c $*.psw
        cc $(CFLAGS) -c $*.c -o $(OFILE_DIR)/$*.o
        rm $*.c
```

For NEXTSTEP Release 3.0/3.1:

```
.psw.o:
        $(PSWRAP) $(PSWFLAGS) -a -h $(SYM_DIR)/$*.h -o $(SYM_DIR)/$*.c $*.psw
        cc $(DEFAULT_CFLAGS) -c $(SYM_DIR)/$*.c -o $(OFILE_DIR)/$*.o
        rm $(SYM_DIR)/$*.c
```

You have to remove the **.c** files generated by **pswrap** after the compilation, so that in subsequent makes, the **c++** compiler will not try to compile these files.

In general, you can override the default **make** rules to suit your environment. For example, if you only have one C++ file (e.g. foo.c), add the following to your Makefile.postamble rather than overriding the CC directive in the Makefile.preamble:

for NEXTSTEP Release 2.0:

```
.c.o:
                cc++ $(CFLAGS) -c $*.c -o $(OFILE_DIR)/$*.o
```

for NEXTSTEP Release 3.0/3.1:

```
     .c.o:
                cc -ObjC++ $(CFLAGS) -c $*.c -o $(OFILE_DIR)/$*.o
```

See also:
/usr/lib/nib/Makefile.common for NeXT's default rules for NEXTSTEP Release 2.0, or
/NextDeveloper/Makefiles/app/basicrules.make for NEXTSTEP Release 3.1,
and the ../Objective_C/integrating_C++_code for how to integrate C++ and Objective-C files.

QA795

Valid for 2.0, 3.0, 3.1