

# Task models

A task model is a definition of an instanciable task, ie

- a name
- a list of ports accepting resources from different resource types
- a list of available implementations

```
tg = TaskModel(  
    'tg',  
    outfifos = ['output'] )
```

# Task implementations

## Software Task

A software task implementation is defined by:

- a function to call
- a list of C source files (even if some files are shared between tasks, you must redeclare them in each using task implementation definition)

```
tg = TaskModel(  
    'tg',  
    outfifos = ['output'],  
    impl = [ SwTask( 'tg',  
                    stack_size = 4096,  
                    sources = [ 'src/tg_posix.c' ] )  
            ] )
```

## Hardware Task

An hardware implementation is in fact a coprocessor (which must be implemented in simulator/synthesis context and declared in DSX) doing the same thing as the task.

The following restrictions apply:

- Task must only use MwMr fifos for data exchange
- Task must not be RealTime?

Declaration consists of:

- Coprocessor reference
- Optional coprocessor arguments

```
tg = TaskModel(  
    'tg',  
    outfifos = ['output'],  
    impl = [ SwTask( 'tg',  
                    stack_size = 4096,  
                    sources = [ 'src/tg_posix.c' ] )  
            HwTask( Tg, filename = 'plan.jpg' )  
            ] )
```

## Virtual Task

If a task follows constraints for having an hardware implementation, it may be virtually implemented in hardware, which means it will be run in simulator's context as if it were a coprocessor, actually using C implementation.

This kind of virtualization is only available for SystemC simulation (ie not VHDL), and you must have defined a software implementation. If your task has a `Virtual()` declaration and an `HwTask?` at the same time, the implementation chosen by DSX for simulation purposes will be unpredictable, you should avoid such situations.

```
idct = TaskModel(  
    'idct',  
    infifos = [ 'input' ],  
    outfifos = [ 'output' ],  
    impl = [ SwTask( 'idct',  
                    stack_size = 1024,  
                    sources = [ 'src/idct.c' ] ),  
            Virtual()  
    ] )
```

## Task Synthesis

Using Ugh, a task may be synthesized, this is not supported at this time.