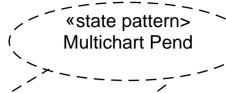
## marshal

```
entry /
 archer.scribble("halt horse")
 archer.scribble("identify next marshal point")
 archer.scribble("field wrap wounds on self and horse")
 archer.scribble("drink water")
 archer.arrows = HorseArcher.MAXIMUM_ARROW_CAPACITY
 chart.post fifo(
  Event(signal=signals.READY),
   times=1.
   period=archer.to_time(60),
   deferred=True)
Ready /
 readv = True
 for name, other in archer.others.items():
  if other.dead() is not True:
   ready &= other.state_name == 'waiting'
  else:
   archer.snoop_scribble(
     "{} thinks {} is dead".
    format(archer.name, name)
 if ready:
  archer.post_fifo(Advance_War_Cry)
```

## waiting\_to\_advance

```
entry /
archer.yell(Event(
  signal=signals.Other Ready War Cry
  payload=archer.name))
ready = True
archer.snoop_scribbel("{} has {} arrows". \
  format(archer.name, archer.arrows)
 time to wait = random.randint(130,300)
 for name, other in archer.others.items():
  if other.dead() is not True:
   ready &= other.waiting()
  else:
   archer.snoop_scribble(
     "{} thinks {} is dead".
    format(archer.name, name)
if ready is False:
  archer.snoop_scribble(
   "{} is impatient he will attack in {} seconds".
   format(archer.name, time_to_wait)
  archer.post_fifo(
   Event(
    signal=signals.Advance War Cry),
   times=1.
   period=random.randint(time_to_wait),
   deferred=True)
 else:
  archer.snoop scribble(
   "{} thinks unit is ready to attack". \
   format(archer.name))
  archer.post fifo(
   Event(signal=signals.Advance War Cry))
exit /
 archer.cancel_events(
  Event(signal=signals.Advance_War_Cry))
```



# Outer state hook:
Other\_Ready\_War\_Cry
archer.dispatch\_to\_empathy(e)