



▶ Version 10.1

▶ Synchronising Threads

▶ Latch Example

Latch Example

A latch holds back a number of threads until the coast is clear. At a signal from another thread, the latch is opened so that all of the threads are released. The latch may (or may not) then be closed again to hold up subsequently arriving threads.

A visual example of a latch might be a ferry terminal, where cars accumulate in the queue until the ferry arrives. The barrier is then opened and all (up to a maximum number) of the cars are allowed through it and on to the ferry. When the last car is through, the barrier is re-closed.

```

      tkt←6  # 6-token: ferry ticket.

      ▽ car ...
[1]   □TGET tkt # await ferry.
[2]   ...

      ▽ ferry ...
[1]   arrives in port
[2]   □TPUT(↑,/□treq □tnums)ntkt # ferry tickets
      for all.
[3]   ...
  
```

Note that it is easy to modify this example to provide a maximum number of ferry places per trip by inserting *max_places*↑ between □TPUT and its argument. If fewer cars than the ferry capacity are waiting, the ↑ will fill with trailing 0s. This will not cause problems because zero tokens are ignored.

Let us replace the car ferry with a new road bridge. Once the bridge is ready for traffic, the barrier could be opened permanently by putting a negative ticket in the pool:

```
□TPUT -tkt # open ferry barrier permanently.
```

Cars could choose to take the last ferry if there are places:

```

      ▽ car ...
[1]   :Select □TGET tkt
[2]   :Case tkt ◊ take the last ferry.
[3]   :Case -tkt ◊ ferry full: take the new
      bridge.
[4]   :End
  
```

The above *:Select* works because by default, □TPUT -tkt puts a value of -tkt into the token.

Dyalog Ltd

South Barn
Minchens Court
Minchens Lane
Bramley
Hampshire
RG26 5BH
United Kingdom

Phone:

+ 44 (0) 1256 830 030

Fax:

+ 44 (0) 1256 830 031

e-mail:

sales@dyalog.com