



- ▶ Multi Threading
- ▶ Starting Threads

## Starting Threads – Spawn Thread

$\{R\} \leftarrow \{X\} f \& Y$

$\&$  is a monadic operator with an ambivalent derived function.  $\&$  spawns a new thread in which  $f$  is applied to its argument  $Y$  (monadic case) or between its arguments  $X$  and  $Y$  (dyadic case). The shy result of this application is the number of the newly created thread.

When function  $f$  terminates, its result (if any), the **thread result**, is returned. If the thread number is the subject of an active  $\square TSYNC$ , the thread result appears as the result of  $\square TSYNC$ . If no  $\square TSYNC$  is in effect, the thread result is displayed in the session in the normal fashion.

Note that  $\&$  can be used in conjunction with the **each** operator:  $\ddot{\cdot}$  to launch many threads in parallel.

### Examples:

```

    ÷&4           A Reciprocal in background
0.25
    □←÷&4       A Show thread number
1
0.25
    FOO&88      A Spawn monadic function.
    2 FOO&3     A dyadic
    {NIL}&0      A niladic
    ♣&'NIL'     A ..
    X.GOO&99    A thread in remote space.
    ♣&'□dl 2'   A Execute async expression.
    'NS' ♣&'FOO' A .. remote .. ..
    PRT&''↓□nl 9 A PRT spaces in parallel.
    
```

**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



- ▶ Multi Threading
- ▶ Starting Threads

## Callback Functions as Threads

A callback function is associated with a particular event via the Event property of the object concerned. A callback function is executed by `□DQ` when the event occurs, or by `□NQ`.

If you append the character `&` to the name of the callback function in the Event specification, the callback function will be executed asynchronously as a thread when the event occurs. If not, it is executed synchronously as before.

For example, the event specification:

```
□WS'Event' 'Select' 'DoIt&'
```

tells `□DQ` to execute the callback function `DoIt` *asynchronously as a thread* when a Select event occurs on the object.

### 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