



Dynamic Functions

Recurions

Recurions

A Recursive Dynamic Function can refer to itself using its name explicitly, but because we allow unnamed functions, we also need a special symbol for implicit self-reference: '▽'. For example:

```
fact←{
  ω≤1:1      A Small ω, finished,
  ω×▽ ω-1    A Otherwise recur.
}
```

Implicit self-reference using '▽' has the further advantage that it incurs less interpretative overhead and is therefore quicker. Tail calls (see [Tail Calls pdf](#)) '▽' are particularly efficient.

Recursive Dynamic Operators refer to their derived functions, that is the operator bound with its operand(s) using ▽ or the operator itself using the compound symbol: ▽▽. The first form of self reference is by far the more frequently used.

```
pow←{
  α=0:ω      A Function power.
  (α-1)▽ α α ω  A Apply function operand α times.
}
```

The following example shows a rather contrived use of the second form of (operator) self reference. The *exp* operator composes its function operand with itself on each recursive call. This gives the effect of an exponential application of the original operand function:

```
exp←{
  α=0:α α ω  A Exponential fn application.
  (α-1)α α α ▽ ▽ ω  A Apply operand 2α times.
}
```

```
succ←{1+ω} A Successor (increment).
10 succ exp 0
```

1024

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