



▶ Dynamic Functions

▶ Static Name Scope

## Static Name Scope

When an inner (nested) Dynamic Function refers to a name, the interpreter searches for it by looking outwards through enclosing Dynamic Functions, rather than searching back along the execution stack. This regime, which is more appropriate for nested functions, is said to employ **static scope** instead of APL's usual **dynamic scope**. This distinction becomes apparent only if a call is made to a function defined at an outer level. For the more usual inward calls, the two systems are indistinguishable.

For example, in the following function, variable *type* is defined both within *which* itself and within the inner function *fn1*. When *fn1* calls outward to *fn2* and *fn2* refers to *type*, it finds the outer one (with value '*static*') rather than the one defined in *fn1*:

```
which←{
    type←'static'
    fn1←{
        type←'dynamic'
        fn2 ω
    }
    fn2←{
        type ω
    }
    fn1 ω
}
    which'scope'
static scope
```

### 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