



Dynamic Functions

Tail Calls

## Tail Calls

A novel feature of the implementation of Dynamic Functions is the way in which tail calls are optimised.

When a Dynamic Function calls a sub-function, the result of the call may or may not be modified by the calling function before being returned. A call where the result is passed back immediately without modification is termed a tail call.

For example in the following, the first call on function *fact* is a tail call because the result of *fact* is the result of the whole expression, whereas the second call isn't because the result is subsequently multiplied by *w*.

```
(α×w)fact w-1 A Tail call on fact.
w×fact w-1 A Embedded call on fact.
```

Tail calls occur frequently in Dynamic Functions, and the interpreter optimises them by re-using the current stack frame instead of creating a new one. This gives a significant saving in both time and workspace usage. It is easy to check whether or not a call is a tail call by tracing it. An embedded call will pop up a new trace window for the called function, whereas a tail call will re-use the current one.

### 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