Date: 10th June 1999
To: J3
From: Malcolm Cohen
Subject: Baroque spelling of 'WITH' functionality

1. Introduction

The editor has called to our attention that people who are not using extensible types will not think of looking in SELECT TYPE to find the ASSOCIATE facility. The syntax is also weird and inconvenient - "SELECT TYPE" and "TYPE DEFAULT" are misleading.

Therefore we propose that a separate (but very similar) construct should be used for accessing the ASSOCIATE facility when type-selection is not required.

2. Syntax

ASSOCIATE (<associate-name> => <expr>[, <associate-name> => <expr>]...) ... 
... 
END ASSOCIATE

Each <associate-name> is a construct entity that has the scope of the ASSOCIATE block, just like the <associate-name> in a SELECT TYPE.

For consistency, we should change our existing

SELECT TYPE ( <expr> ) [ ASSOCIATE( <name> ) ]
to

SELECT TYPE ( [ <name> => ] <expr> )

3. Examples

Before these changes, an example might look like this:

TYPE(MYTYPE) AX,AY
...
SELECT TYPE ( AX%B(I,J)%D ) ASSOCIATE ( ZX )
TYPE DEFAULT
SELECT TYPE ( AY%B(I,J)%D ) ASSOCIATE ( ZY )
TYPE DEFAULT
W = ZX*X + ZY*Y
END SELECT
END SELECT

After these changes, it would look like this:
TYPE(MYTYPE) AX,AY
...
ASSOCIATE ( Z1 => AX%B(I,J)%D, Z2 => AY%B(I,J)%D )
    W = ZX*X + ZY*Y
END ASSOCIATE

4. Discussion

Given this construct, it no longer makes sense to allow SELECT TYPE on non-polymorphic expressions; therefore we limit SELECT TYPE to polymorphic exprs.

5. Edits

(Add the ASSOCIATE construct to the list of executable constructs)
[11:2+] Add new line "<<or>> <associate-construct>".

-Allow the blank between END and ASSOCIATE to be omitted
[27:33+] Add new line "END ASSOCIATE".

(Change SELECT TYPE syntax to be consistent with new ASSOCIATE)
[176:24-25] Change "(( <type-selector>) [ ASSOCIATE ( <associate-name> ) ])
    To "([ <associate-name> => ] <type-selector> )

(Require the SELECT TYPE type-selector to be polymorphic.)
[176:30+] Add new constraint
"Constraint: <selector> shall be polymorphic."

(Since we only allow polymorphic expressions in SELECT TYPE, TYPE IS should require an extensible type name. This also fixes the <type-name> problem.)
[176:31] Change "<type-name>" to "<extensible-type-name>"

(SELECT TYPE type-selector is polymorphic, so the associate-name in the TYPE DEFAULT block is polymorphic)
[177:29] After "name" insert "is polymorphic and"
[177:30-31] Delete "; it is ... polymorphic"

(Delete old example)

(Add new syntax)
[178:33+] Insert new section and renumber existing 8.1.5
"8.1.5 ASSOCIATE construct

The ASSOCIATE construct associates a name with an expression during the execution of its block.

8.1.5.1 Form of the ASSOCIATE construct

R820a <associate-construct> <<is>> <associate-stmt>
    <block>
        <end-associate-stmt>
R820b <associate-stmt> <<is>> [ <associate-construct-name> : ] & 
ASSOCIATE ( <association> & 
[ , <association> ]... )

R820c <association> <<is>> <associate-name> => <selector>
Constraint: If <selector> is not a <variable>, <associate-name> shall not 
appear in a variable definition context (14.7.7).

R820d <selector> <<is>> <expr>
<<or>> <variable>

R820e <end-associate-stmt> <<is>> END ASSOCIATE [<associate-construct-name>]
Constraint: If the <associate-stmt> of an <associate-construct> specifies an 
<associate-construct-name>, the corresponding 
<end-associate-stmt> shall specify the same 
<associate-construct-name>. If the <associate-stmt> of an 
<associate-construct> does not specify an 
<associate-construct-name>, the corresponding 
<end-associate-stmt> shall not specify an 
<associate-construct-name>.

8.1.5.2 Execution of the ASSOCIATE construct

Execution of an ASSOCIATE construct causes execution of its block. During 
exection of that block each associate name is associated (14.6.1.4) with the 
corresponding selector. The associate name assumes the type, type 
parameters, rank, and bounds of the selector. If the selector is not 
definable, its associate name is not definable. The associate name is 
polyorphic if and only if the selector is polymorphic.

Execution of an ASSOCIATE construct with a selector that is not a <variable> 
causes that selector expression to be evaluated prior to execution of the 
block.

It is permissible to branch to an END ASSOCIATE statement only from within 
the ASSOCIATE construct.

8.1.5.3 Example of the ASSOCIATE construct

Note 8.13
ASSOCIATE ( Z ) WITH ( EXP(-(X**2+Y**2)) * COS(THETA) ) 
PRINT *, A+Z, A-Z
END ASSOCIATE
".

{Add association information to ch14}
[372:2+] Insert 
"Execution of an ASSOCIATE statement establishes an association between each 
selector and the corresponding associate name of the ASSOCIATE construct. 
The associate name remains associated to its selector throughout the 
execution of the construct. Within the ASSOCIATE construct, each selector is 
known by and may be accessed by its associate name. Upon termination of the 
ASSOCIATE construct, the association is terminated.".