

# ISO/IEC JTC1/SC22/WG5 N1364

(J3/99-171)

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:



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  
execution 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  
polymorphic 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."