Subject: More control over intrinsic assignment
From: Van Snyder
Reference: 03-258r1, section 2.2.3, and references in section 8.2 below

1 Number

TBD

2 Title

More control over intrinsic assignment.

3 Submitted By

J3

4 Status

For consideration.

5 Basic Functionality

Provide a means to turn off intrinsic assignment outwith the module where a type is defined. Provide a means to override defined assignment with intrinsic assignment within a construct within a subprogram within the module where a type is defined.

6 Rationale

There are some types for which no two objects should have the same value. For example the "keys" in a resource-management package should not be duplicated. It is better to prevent assignment with a syntactic device than to catch attempted assignments at run-time by overriding intrinsic assignment with a defined assignment that prints an error message and stops. Detecting assignment at run-time may reveal a mistake years after the software is deployed, while a syntactic device will detect it during development.

Within procedures that implement defined assignment, it would sometimes be convenient to start out by doing intrinsic assignment, and then add a few jots and tittles.

7 Estimated Impact

Minor.

8 Detailed Specification

Provide a type attribute in subclause 4.5 that specifies that intrinsic assignment is not available outwith the module in which the type is defined. Ada calls this attribute **limited**. Stipulate in subclause 7.4 that intrinsic assignment is not available for types with this attribute except within the module where the type is defined.

Provide a type attribute in subclause 4.5 that specifies that intrinsic assignment may be overridden by a construct within a subprogram within the module where the type is defined. Provide a construct that specifies for which types intrinsic assignment overrides defined assignment within the construct.

8.1 Examples

A type for which intrinsic assignment does not exist outwith the module where the type is defined — even if there is no defined assignment:

```
  TYPE, LIMITED :: RESOURCE_KEY
    PRIVATE
    INTEGER :: KEY_INDEX
  END TYPE RESOURCE_KEY
```

A type for which intrinsic assignment can be made to override defined assignment within a construct within a procedure within the module where the type is defined:

```
  TYPE, ASSIGNABLE :: MY_MATRIX_T(RK)
    PRIVATE
    INTEGER, KIND :: RK
    INTEGER :: MATRIX_KIND = M_ABSENT ! or M_FULL or M_BANDED or M_COLUMN_SPARSE
    INTEGER, POINTER :: R1(:) => NULL(), R2(:) => NULL() ! DEPENDS ON MATRIX_KIND
    REAL(RK), POINTER :: VALUES(:,:) => NULL()
  END TYPE MY_MATRIX_T
```

A construct in which intrinsic assignment overrides defined assignment, which is only allowed within a subprogram within the module where the types mentioned in it are defined:

```
  INTRINSIC ASSIGNMENT ( MY_MATRIX_T, ... )
  ...
  END INTRINSIC ASSIGNMENT
```

8.2 References

1. J. G. P. Barnes, **Programming in Ada 95**, Addison-Wesley (1995) ISBN 0-201-87700-7, pp 223-229.

2. Norman H, Cohen, **Ada as a Second Language**, 2nd Ed., McGraw-Hill (1996), ISBN 0-07-011607-5, pp 478-484.

9 History