IMPLICIT

Statement: Overrides the default implicit typing rules for names. (The default data type is INTEGER for names beginning with the letters I through N, and REAL for names beginning with any other letter.)

The IMPLICIT statement takes one of the following forms:

Syntax

IMPLICIT *type* (*a*[, *a*] ...)[, *type* (*a*[, *a*] ...)] ...
IMPLICIT NONE

type
Is a data type specifier (CHARACTER*(*) is not allowed).

a
Is a single letter, a dollar sign ($), or a range of letters in alphabetical order. The form for a range of letters is a_1-a_2, where the second letter follows the first alphabetically (for example, A-C).

The dollar sign can be used at the end of a range of letters, since IMPLICIT interprets the dollar sign to alphabetically follow the letter Z. For example, a range of X-$ would apply to identifiers beginning with the letters X, Y, Z, or $.

Description

The IMPLICIT statement assigns the specified data type (and kind parameter) to all names that have no explicit data type and begin with the specified letter or range of letters. It has no effect on the default types of intrinsic procedures.

When the data type is CHARACTER*len, *len* is the length for character type. The *len* is an unsigned integer constant or an integer specification expression enclosed in parentheses. The range for *len* is 1 to 2**31-1 on IA-32 architecture; 1 to 2**63-1 on Intel® 64 and IA-64 architectures.

Names beginning with a dollar sign ($) are implicitly INTEGER.

The IMPLICIT NONE statement disables all implicit typing defaults. When IMPLICIT NONE is used, all names in a program unit must be explicitly declared. An IMPLICIT NONE statement must precede any PARAMETER statements, and there must be no other IMPLICIT statements in the scoping unit.

NOTE
To receive diagnostic messages when variables are used but not declared, you can specify compiler option warn declarations instead of using IMPLICIT NONE.

The following IMPLICIT statement represents the default typing as specified by the Fortran Standard for names when they are not explicitly typed:

```
IMPLICIT INTEGER (I-N), REAL (A-H, O-Z)
```

See Also:

Data Types, Constants, and Variables, -warn declarations compiler option

Examples

The following are examples of the IMPLICIT statement:

```
IMPLICIT DOUBLE PRECISION (D)
IMPLICIT COMPLEX (S,Y), LOGICAL(1) (L,A-C)
IMPLICIT CHARACTER*32 (T-V)
IMPLICIT CHARACTER*2 (W)
IMPLICIT TYPE(COLORS) (E-F), INTEGER (G-H)
```

The following shows another example:

```
SUBROUTINE FF (J)
IMPLICIT INTEGER (a-b), CHARACTER*(J+1) (n), TYPE(fried) (c-d)

TYPE fried
INTEGER e, f
REAL g, h
END TYPE
age = 10 ! integer
name = 'Paul' ! character
c%e = 1 ! type fried, integer component
```