



Each byte of internal memory is numbered sequentially beginning with 0 for the first byte, 1 for the second byte, and continuing in this way until the end of memory is reached. We will call the number associated with each byte the **address** of the byte. There is a limit to the amount of memory that is supported, since addresses are expressed as binary integers using 31 bits. The smallest address which can be created is 0 and the maximum address is $2^{31} - 1$ which is 2,147,483,647 or 2 gigabytes.

Address fields are created by using the **A**, **Y**, or **V** indicators in a DS or a DC. (Q-constants and S-constants are discussed elsewhere.) If a length indicator is not coded, **A** and **V** indicate fullword fields, while **Y** indicates a halfword address. A length indicator can be used to create A-constants with 1 to 4 bytes, V-constants with 3 or 4 bytes, and Y-constants with 1 or 2 bytes.

The following are formats for creating address constants,

```
name    DC    T( address, ... )
name    DC    TLn( address, ...)
```

where "name" is a field name,

"T" is the address type A,Y, or V,

"address, ..." represents 1 or more addresses separated by commas. Each address is either a relocatable, a complex relocatable, or an absolute expression.

It is difficult to give the technical definition of the terms **relocatable** and **complex relocatable**, but generally, an expression is **relocatable** if it refers to labels in a program and the value of the expression changes if the program is moved from its original location. An expression is **absolute** if it refers to labels in a program, and the value of the expression is constant regardless of the location of the program. An example will help with these ideas,

```
DOG      DS      CL5
CAT      DS      CL7
PIG      DS      CL9
ADR1     DC      A(PIG)          RELOCATABLE
ADR2     DC      A(PIG - DOG)    ABSOLUTE
ADR3     DC      A(PIG + DOG)    COMPLEX RELOCATABLE
```

Notice that "ADR1" is relocatable since its value (address) would change if the program were moved. "ADR2" is absolute since PIG and DOG are 12 bytes apart regardless of where the program is loaded. "ADR3" is complex relocatable since it involves multiple "unpaired" relocatable expressions and its value would change depending on the program's location.

Relocatable expressions are initially created at assembly time and then modified by the linkage editor at link time to contain the correct address for the expression you have created. Absolute expressions are not modified by the linkage editor.

The main difference between A-constants and V-constants is that a V-constant can refer to a label in a different control section. A-constants can only reference labels within the same control section. For V-constants, at assembly time, the assembler creates a fullword containing 0's and leaves an indicator for the linkage editor to supply the correct address in the constant at link time.

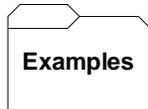
The primary reason for using a V-constant is for program linkage - we want to branch to a separately assembled control section. This can be accomplished with the code below,

```

LA      R1,=A(X,Y,Z)      POINT TO THE PARAMETERS
L       R15,=V(SUBPROG1)  VIRTUAL ADDRESS SUBPROG1
BASR    R14,R15           BRANCH TO THE SUBPROGRAM

```

Notice that the above example also uses A-constants to create 3 consecutive fullword addresses for parameter passing.



Some Typical DS's and DC's:

```

TABSTART EQU *
GOAT      DS    CL8
TABRECLN EQU *-TABSTART
HORSE     DS    CL8
CAT       DS    CL8
TABEND    EQU *
TABLEN    DC    A(TABEND - TABSTART)  Assemble table length
TABLEN2   DC    AL2(TABEND - TABSTART) Halfword table length
NORECS    DC    A((TABEND - TABSTART)/TABRECLN)  Number of entries
P         DS    A                      A fullword field, properly aligned
Q         DS    AL4                    A fullword field, no slack bytes
R         DC    V(SUBPROG1)            The virtual address of SUBPROG1
S         DC    A(CAT)                 A relocatable address
T         DC    A(CAT,HORSE,GOAT)     Three consecutive address constants
U         DC    Y(HORSE)               A halfword address
TRANTAB1 DC    256AL1(0)              A skeleton for a TRT table
TRANTAB2 DC    256AL1(*-TRANTAB2)    A skeleton for a TR table
                                         x'000102030405060708090A...'

```

Tips

1. Read the technical definitions of Relocatable, Complex Relocatable, and Absolute expressions found in IBM's [High Level Assembler Language Reference](#) .