New USINGs

by Ian S. Worthington

Introduction

IBM's High Level Assembler Release 2, originally released in 1995, introduced a number of significant enhancements that make assembler programming rather easier. Amongst them are changes to the USING statement allowing the easy description of more complex structures with less registers and simpler code than was possible hitherto.

By now this release has almost certainly found its way into your company, but you may not have been informed of the changes that accompany it. Even if you don't use them, the new USING formats might be confusing when you first come across them. This article attempts to explain them by looking at the problems they seek to address and the solutions that they provide. Assembler code fragments, and their assembled output, are given as examples.

Further information about all of the changes is available in the IBM Presentation Guide, Redbook SG24-3910, available from http://www.redbooks.ibm.com/, as well as the HLASM Language Reference SC26-4940[1].

Labeled Usings

The problem: Within a block of code we wish to manipulate a number of different records all mapped by the same DSECT.

The old way: We don't use a USING statement at all. We code field displacements as offsets from a base label, and manually code a base register to identify which record we wish to refer to, for example:

```
ST    R03,OFFSET-BASE(,R01)
```

This has the disadvantage that if we wish to change the base register for the record we need to identify and recode each affected instruction. Also, as the assembler has no knowledge of the significance of this register it cannot warn us when we attempt to use it outside of its domain.

The solution: This new assembler release allows the programmer to qualify a DSECT symbol used in an instruction with a name to make explicit which of a number of

[1] Though you may find its description of these new USINGs rather confusing. I know I did.

("labeled") USING statements specifying a DSECT name and an associated register the programmer wishes to use.

This qualifying name is coded both in the label field of the USING statement and used, along with a period, to prefix the DSECT field name in the instruction.

Example: Three records described by the same DSECT are to be linked together using forward and backward chains:

```
BLOCK1   DSECT
         DS    XL8
FCHAIN   DS    A                       FORWARD CHAIN
BCHAIN   DS    A                       BACK CHAIN
         RSECT ,
…
…
…
NEW      USING BLOCK1,R01              ITEM TO BE ADDED TO CHAIN
BEFORE   USING BLOCK1,R02              ITEM BEFORE INSERT POINT
AFTER    USING BLOCK1,R03              ITEM AFTER INSERT POINT
*
         ST    R03,NEW.FCHAIN          LINK NEW TO AFTER ITEM
         ST    R02,NEW.BCHAIN          LINK NEW TO BEFORE ITEM
         ST    R01,BEFORE.FCHAIN       LINK BEFORE ITEM TO NEW
         ST    R01,AFTER.BCHAIN        LINK AFTER ITEM TO NEW
```

```
  Loc  Object Code    Addr1 Addr2  Stmt   Source Statement

000000                00000 00010  3896 BLOCK1   DSECT
000000                             3897          DS    XL8
000008                             3898 FCHAIN   DS    A                       FORWARD CHAIN
00000C                             3899 BCHAIN   DS    A                       BACK CHAIN
                                   3900          RSECT ,
                                        …
                                        …
                                        …
                 R:1  00000        3903 NEW      USING BLOCK1,R01              ITEM TO BE ADDED TO CHAIN

                 R:2  00000        3904 BEFORE   USING BLOCK1,R02              ITEM BEFORE INSERT POINT

                 R:3  00000        3905 AFTER    USING BLOCK1,R03              ITEM AFTER INSERT POINT
                                   3906 *
000020 5030 1008            00008  3907          ST    R03,NEW.FCHAIN          LINK NEW TO AFTER ITEM
000024 5020 100C            0000C  3908          ST    R02,NEW.BCHAIN          LINK NEW TO BEFORE ITEM
000028 5010 2008            00008  3909          ST    R01,BEFORE.FCHAIN       LINK BEFORE ITEM TO NEW
00002C 5010 300C            0000C  3910          ST    R01,AFTER.BCHAIN        LINK AFTER ITEM TO NEW
```

Dependent USINGs

The problem: We wish to manipulate a data structure that contains one or more other data structures, or a storage area that contains a concatenation of data structures.

The old way: Previously we would do this either in a similar fashion to that described above, manually coding the offsets and bases for each field, for example:

```
         ST    R03,OFFSET2-BASE2+OFFSET1-BASE1(,R01)
```

Or by using multiple USING statements and corresponding base registers.

This second method is of course by far the most robust, but is wasteful of registers, and we may not always have enough available to use as a secondary base registers, or the risks of reorganising code to free up sufficient for use in this way may be greater than the benefits we are seeking to achieve by so doing, leading us to favour the previous method with its disadvantages as previously described.

The solution: This new assembler release allows the programmer to address multiple DSECTS with a single base register, basing the secondary ("dependent") dsects on *offsets* from labels within another DSECT, rather than a base register.

Example: We wish to access a field with a record that consists of a fixed header followed by a variable section identified by means of a flag byte:

```
BLOCK1   DSECT
         DS    XL16                    HEADER
DATA1    DS    X                       FIXED DATA ELEMENT
         DS    XL3                     SPARE
VARDATA  DS    XL200                   VARIABLE DATA AREA
         RSECT ,
*
VDATA27  DSECT
FIELD1   DS    F
FIELD2   DS    F                       SOME DATA
         RSECT ,
…
…
…
         USING BLOCK1,R01
*
         #IF   DATA1,EQ,27
               USING VDATA27,VARDATA
*
               L     R00,FIELD2        PICK UP DATA ELEMENT
         #EIF
```

```
  Loc  Object Code    Addr1 Addr2  Stmt   Source Statement
000000                00000 000DC  3896 BLOCK1   DSECT
000000                             3897          DS    XL16                    HEADER
000010                             3898 DATA1    DS    X                       FIXED DATA ELEMENT
000011                             3899          DS    XL3                     SPARE
000014                             3900 VARDATA  DS    XL200                   VARIABLE DATA AREA
                                   3901          RSECT ,
                                        …
                                        …
                                        …
000000                00000 00008  3904 VDATA27  DSECT
000000                             3905 FIELD1   DS    F
000004                             3906 FIELD2   DS    F                       SOME DATA
                                   3907          RSECT ,
                                   3909 *
                  R:1  00000       3910          USING BLOCK1,R01
                                   3911 *
000020 951B 1010      00010        3912          #IF   DATA1,EQ,27
                                   3935+*,1
               1 014  00000 00014  3937                USING VDATA27,VARDATA
                                   3938 *
000028 5800 1018            00004  3939                L     R00,FIELD2        PICK UP DATA ELEMENT
                                   3940          #EIF
```

Labeled Dependent USINGs

This new release also allows us to combine both new USING formats for those cases in which we wish to work with multiple copies of data structures containing multiple substructures.

These "labeled dependent" USINGs combine the advantages of both new types of USINGs.

Example: A combination of the previous two examples. We wish to move a field in the variable section of one record into a field in variable section of a second record.

```
BLOCK    DSECT
         DS    XL8
FCHAIN   DS    A                        FORWARD CHAIN
BCHAIN   DS    A                        BACK CHAIN
VARDATA  DS    XL200                    VARIABLE DATA AREA
*
VDATA    DSECT
FIELD1   DS    F
FIELD2   DS    F                        SOME DATA
         RSECT ,
*
BLOCK2   USING BLOCK,R02
…
…
…
         USING BLOCK,R01
B1VD     USING VDATA,VARDATA
*
B2VD     USING VDATA,BLOCK2.VARDATA
*
         MVC   B1VD.FIELD2,B2VD.FIELD2   MOVE SOME DATA
```

```
  Loc  Object Code    Addr1 Addr2  Stmt   Source Statement
000000                00000 000D8  3896 BLOCK    DSECT
000000                             3897          DS    XL8
000008                             3898 FCHAIN   DS    A                        FORWARD CHAIN
00000C                             3899 BCHAIN   DS    A                        BACK CHAIN
000010                             3900 VARDATA  DS    XL200                    VARIABLE DATA AREA
                                        …
                                        …
                                        …
000000                00000 00008  3902 VDATA    DSECT
000000                             3903 FIELD1   DS    F
000004                             3904 FIELD2   DS    F                        SOME DATA
                                   3905          RSECT ,
                                   3907 *
                 R:2  00000        3908 BLOCK2   USING BLOCK,R02
                                   3909 *
                                   3910 *
                                   3911 *
                 R:1  00000        3912          USING BLOCK,R01
               1 010  00000 00010  3913 B1VD     USING VDATA,VARDATA
                                   3914 *
               2 010  00000 00010  3915 B2VD     USING VDATA,BLOCK2.VARDATA
                                   3916 *
000020 D203 1014 2014 00004 00004  3917          MVC   B1VD.FIELD2,B2VD.FIELD2   MOVE SOME DATA
                                   3918 *
                                   3919 *
000006 00                          3920          FINIS ,
                                   3944          END   ,
```

Terminating the domain of a new USING

Along with the new USING formats comes a new DROP format. DROP now supports either a register or a label of an active USING as a parameter:

DROP register

The DROPing of a USING register terminates the domain of *all* USINGs that reference, directly or indirectly, that register. The only way to terminate the domain of a dependent USING is by terminating the domain of the corresponding ordinary USING.

DROP label

The DROPing of a USING label terminates the domain of a labelled or labelled dependent USING.

Thanks to...

My thanks to Aer Lingus in general, and Paul Keogh in particular, whose requirement was the original inspiration for this article.

Ian S. Worthington is an independent ALCS and MVS systems consultant, currently looking for project work. If you would like to offer him any, or contact him for any other reason, you may do so via email at IanWorthington@usa.net, or via fax through his virtual offices in New York (+1.212.208.4463) and London (+44.20.7900.2916).

Copies of this article may be obtained from the System Technology International Website at http://247sti.tripod.com/documents