Accessing Other Partitions

BY LEO J. LANGEVIN

In previous columns I have written about the use of access registers for defining and using data spaces. However, one thing that you might not be aware of is that you can also use access registers to access other partitions as well. That's right. Access registers will allow you to browse the contents of CICS storage from a batch program.

Now there are some interesting uses for this. For example, you might run a database update program that takes hours and hours to run. Rather than having the program output messages every thousand records, you can just put an eye-catcher in storage, such as "CURRENT COUNT", and when you are interested in the total, you can display the amount. Or let's say that you want to query the control blocks of any subsystem, such as CICS, Power, VTAM, or TCP/IP. If you know what you are looking for, you can easily get to it. And yes, you can even make modifications.

How scary can you make this? Well, you could code a web page CGI (Common Gateway Interface) that can display, browse, and modify partition storage (Now that's scary!). But the important thing is that you understand the use of this process. What you do with it is up to you.

First of all, remember that we have several different kinds of registers. The two that we are interested in are the general registers and access registers. You're probably used to using general registers whereas access registers have the same numbering, but different op-codes are used for manipulating them. When you activate the use of access registers and load them, they work in conjunction with the general registers. Not only do you now point to a specific address, but you also point to a specific address space.

OBTAINING AUTHORITY

Before you can use an access register, you need to set up your program to be authorized to do this. If you are running under CICS, then you don't need to do anything more. If not, you will need to use the PRODID macro as shown in Figure 1. This will allow you to be defined as a product to the VSE system. If you then issue a "SIR VENDOR" command, you can see the entry that you have. In fact, you can scan through the PRODID control blocks to produce your own display. (Why IBM chose to suppress the display of version information is beyond me.)

Now that your product is defined to your ESA system, you need to be authorized. You will then issue the following:

```
LM   R1,R2,IJBTOKEN
PRODID AUTH=YES
```

You are now authorized to use access registers. But first you need an ALET, a token that will point to a specific partition. Let's say that you have the PIK (Partition Identifier Key) for F2 loaded in Register 3. You could get the ALET by using:

```
GETFLD FIELD=ALET,PART=(3)
LR   R7,R1
```

OK, so you are now authorized and you have the partition token, but you still need to turn on access register mode. You can do this using the code shown in Figure 2.

This will activate access register mode, put the ALET that you saved from before into access register 2, perform some process, and then get you out of access register mode. Remember, if you load access register 2 with an ALET, you will need to use general register 2 to do all of the pointing to the CICS partition (or wherever you are trying to go). Since the other access registers, by default,will be set to zero,they will not be used, so you can have some register use the current partition pointer while others use alternate partition pointers.

Figure 1: The PRODID Macro

```
LA R4,PRODAREA
USING PRODSECT,R4
MVC IJBCOMPN(14),=CL14'COMPANY NAME'
MVC IJBPRODN(12),=CL12'ALET RTN'
MVC IJBVRM(6),=CL6'V1.0'
MVC IJBVIDL(2),=H'50'
    PRODID DEFINE,AREA=(4)
B NEXT
PRODAREA DS CL52
PRODSECT PRODID DSECT
```

Figure 2: Code to Turn on Access Register Mode

```
SAC 512
SAR 2,7
… perform some process [MVC CLI 0(R2),0]
SAC 0
```

Finally, make sure you DELETE the entry when your program terminates or subsequent DEFINE requests will fail because of duplicate information being found. That's right. Just because the program terminates doesn't mean that VSE will clear out the table entry. You would perform the DELETE with the following:

```
PRODID DELETE,TOKEN=IJBTOKEN
```

And that's all there is to it. It's easy, it's fun, and it's exciting. Amaze your friends! Impress your colleagues!

NaSPA member Leo J. Langevin has been involved in the VSE community since its inception. He is employed by Connectivity Systems and is the lead developer of NFS for VSE. He can be reached at leo@tcpip4vse.com.