

Retaining Data Center Skills and Shortage

By Greg Hamlyn

Look around your data center and you will undoubtedly uncover many highly experienced technical staff that have reached that age where they have already started to think about retiring. How will you replace them? All their skills and knowledge will be lost once they leave!

This article discusses how simple coaching and mentoring programs can be implemented that insure this doesn't occur. In this first of **four part series** we look at how data center skills and knowledge requirements have evolved and identify the importance of their retention.

Part One - The Data Center Skills Crisis

You are probably numb from all the articles and hype over the last few years, discussing the aging demographic of data center staff and the cries of "What will we do when Jo the systems programmer leaves?" This year AFCOM's Data Center Institute reported that more than 60% of IT workers with mainframe experience are now at least 50 years old(if you don't believe me, take a look around when you next attend an IT technical conference). Symantec's 2010 Data Center report, which surveyed 1780 data center managers in 26 countries, highlights that over 50% of Data Centers are understaffed, with 23% indicating they had a major problem associated with staffing as a result of employee retirements.

While expertise is being lost to retirement, there are several other data center issues that are resulting in what experts are calling "an IT skills crisis". Many data centers today are highly automated, running a virtualized IT infrastructure and are seeing a greater rate of technical change than ever before. Finding staff with a mixture of skills and knowledge to run and manage these environments is becoming increasingly difficult, especially if you have just built a new data center and looking to staff it from scratch. On top of this technical skill and knowledge requirement is the organization's need for implementation and management of best practice, compliance and auditing practices. Employers are looking for people that have all of the above skills and are pro-active, fully aware of repercussions of work performed.

So, you have an environment where skills and knowledge are leaving the building and limited amount of people with the technical and business skills to take their place. In a best

Inside This Issue

Retaining Data Center Skills and Shortage	1
Upcoming Datatrain Releases	2
Learning Spotlight - SDSF	2
Vendor Briefs	3
Should you be adopting ITIL?	4
Test your Knowledge - JCL Problem Resolution	4
PROGRAMMERS' PLACE - Understanding Re-Entrant Programming	5

case scenario, you could send staff off to training courses where they could learn all that was required. However we all know that in the real world of tight budgetary constraints and flat head counts that it is difficult to release anyone even for short term classroom training.

Another issue up until a few years ago was that no young workers were interested in working in a data center or mainframe environment because it was perceived as an IT area that was going nowhere, and was far from being cool. This has been addressed to some degree by IBM and their System z academic initiative program, which is seeing a range of mainframe IT skills now being taught in schools and colleges around the world. But even then you may end up with graduates that theoretically skilled up, but have no practical experience.

So the bottom line as usual is that you need to do more with what you already have. Before your experts retire or move on, you need to somehow extract all their expertise and pass it on to other staff. If only you could run a dump program on people, like you can with your operating systems!

Still, all is not lost as many data centers have found that this transference of skills and knowledge through methods such as: on the job training, coaching and mentoring can be achieved, with a secondary benefit being that it could also be used to cut training costs. Symantec's 2010 data center report showed that 48% of those surveyed indicated that their greatest cost cutting containment from the previous year was related to the cross training of IT staff.

In the next installment of this newsletter we will look at ways of preventing the drain of your organization's valuable skills and knowledge using relatively simple knowledge transfer programs.



Greg Hamlyn, a senior technical course designer and developer with Datatrain, has over 25 years experience working in the mainframe and z/OS data center environments.

Welcome to the Datatraining Mainframe Quarterly

Datatraining Mainframe Quarterly is an e-Zine published by Datatask Pty Ltd, that provides world-class elearning Mainframe training programs for the Information Communications and Technology industry.

As a user of Datatask's award winning Datatraining online mainframe training, we are sending you the first edition of the Datatraining Mainframe Quarterly newsletter. As a subscriber you will receive an edition every three months which includes z/OS related articles for technical experts and management, and information on upcoming Datatraining releases.

If you wish to unsubscribe to Datatraining Mainframe Quarterly, please reply to info@datatask.com.au with the title 'Unsubscribe'.

Upcoming Datatraining Releases – March 2011

The Datatraining 10.3 Release will contain the following courses:

- The complete z/OS Series updated to V1.11
- The z/OS Assembler Language Series updated to V1.11. The Assembler z/OS Advance course is to be released in 2011
- The complete Linux Series updated to current Linux Kernel level 2.6.35
- A new series on Virtualization that includes z/VM 6.1 and Linux on System z
- A new Introduction to IBM Mainframe Communications Series at V1.11 that replaces the old VTAM Series. A further course on TCP/IP to complete the series is to be released in 2011
- The complete CA OPS/MVS® Event Management and Automation Series updated to V11.8

Learning Spotlight – SDSF

The System Display and Search Facility (SDSF) has always provided batch job users with a way to control jobs and view output from the work they submit. For the operator, SDSF is an invaluable tool used to monitor various aspects of the z/OS system and with each z/OS release new enhancements and options are provided to assist with the monitoring of ever-growing system complexities.

In the last few z/OS releases, the following SDSF improvements have appeared:

- Support for REXX has been added providing an alternative to SDSF batch. As many users already know, REXX is a powerful language and can be used to develop scripts for complex and repetitive tasks.
- Several new checks have been added to the Health Check option
- A number of new columns of data have been added to the majority of SDSF panels.



Two modules from Datatraining's SDSF curriculum are available for you to browse. The [Displaying Job Activity](#) module explains how job processing is undertaken by the system and how the SDSF Input, Display Active and Status panels are used for job tracking. The [Monitoring System Health](#) module describes the purpose of the IBM Health Checker and how checks can be managed using the SDSF Health Checker panel.

Vendor Briefs

This article looks at some of the new mainframe-related products and services released during the last three months.

IBM

September brought with it the release of z/OS 1.12, which heralds in IBM's zEnterprise system, which is able to manage workloads across System z, and select POWER7 and System x servers. IBM has already shipped the first of these systems to global reinsurance company, Swiss Re. As you would expect from a major system release such as this, the processing power and workload capabilities have also been considerably improved making it an attractive proposition for data intensive processing organizations.

z/OS 1.12 itself delivers the usual smattering of improvements to a number of areas and also provides an upgrade to its z/OS Management Facility (z/OSMF) application. This product uses a web interface to perform z/OS system management tasks such as monitoring sysplex and Linux performance, managing WLM service definitions and configuring TCP/IP policy based networking functions. z/OSMF provides users with wizards to help simplify tasks and can be configured to link several tasks. An incident log captures system and user-initiated problem data that can be used by the organization or sent to IBM for analysis.

Watch out for Datatrain's z/OS 1.12 version of elearning courses and a new z/OS Management Facility module to be released in the near future.

The release of DB2 10 for z/OS promises increased performance (IBM quotes a 5 to 10% saving in CPU) and scalability. The vendors mentioned below: CA, BMC, ASG and Compuware have openly supported movement to this version by updating their DB2 product offerings.

CA

Automation seems to be the flavor of the day at CA with several new and updated products focusing on process, configuration, workload, virtual, server and client automation. With the projected mainframe skills shortage, CA is looking at creating tools that will be more powerful, but easier to use by providing linked tasks using a web interface. CA's workload automation suite of tools expands on the traditional operations job scheduling products to provide timely reaction to events occurring on both mainframe and distributed platforms.

BMC

BMC's Control-M workload scheduling product has received a make-over and version 7 now handles policy based workload automation, routing priority work to servers with the highest availability. It also contains compliance and audit capabilities by enforcing the documenting of scheduling-related changes.

Compuware

Compuware also has the retiring workforce in mind in releasing its mainframe workbench that provides the user with an Eclipse-based graphical interface to its mainframe products: Abend-AID, Xpediter and File-AID. The workbench also provides the programmer with ISPF-like functions such as: editing of source code, access to a mainframe compiler, batch job submission and monitoring, and data set allocation and deletion.

ASG Software Solutions

ASG have followed in the footsteps of several other vendors by releasing an update of its automation solution: ASG-Zena. The 3.0 version of this product now provides workload automation for both physical and virtual servers across UNIX, AIX, Solaris, Windows, Linux, AS400 and z/OS environments.

Should you be adopting ITIL?

By Greg Hamlyn

ITIL has been around for quite a while now, but is its implementation too difficult for organizations? This article looks at some of the trials and tribulations associated with ITIL adoption and suggests how you can get started with it.

Before looking at how ITIL is being used, a general description of what ITIL is and isn't needs to be provided. ITIL V3 is a framework of globally recognized best practices for IT service management that covers service strategy, service design, service transition, service operation and continual service improvement. Popular misconceptions are that ITIL is a product that can be plugged in to solve an IT service problem. At the other extreme, is an all encompassing framework under which all IT processes should live.

Recent Gartner statistics show that around less than 10% of organizations over the last few years have completed adoption of ITIL, while another 40% plan some action over the next few years. What do these figures really say? If ITIL is as good as it is touted, then why is there seemingly slow uptake of it? There are many possible inhibitors preventing ITIL adoption:

- Some organizations plan to implement all aspects of ITIL at the same time but the amount of work and resources required for this becomes too daunting when reality sets in.
- Some organizations see it as a certification exercise only, which loses value to those who are looking to implement it to improve IT services.
- The cost of implementing it with its books, training and certification becomes too difficult and may be hard to sell to management without hard ROI data.
- Commitment, both initially and ongoing, from management is often an issue.

So the big question is, should you adopt ITIL? The answer is that it really comes down to the organization, its budget and commitment. If you have an opportunity to test the waters you should look at the pieces of ITIL that you can really benefit from within your budget constraints. It is important to be realistic in the expected outcome targets you want to achieve. If it is a success story, then you might consider other areas. It is much easier to approach management with a success story and obtain additional budget if it can be shown how ITIL can benefit your organization.

You may also come to the conclusion that ITIL just isn't for you! The following are compelling reasons for not implementing ITIL:

- You may already have a solid set of processes covering the IT service areas that ITIL does
- With the global recession, initial or ongoing funding may not be available (even if you have a solid business case)
- You don't have any compliance need that ITIL certification will solve

If you are looking to get started with ITIL then the Office of Government Commerce (OGC) website should be your first stop. The [Information Systems Examinations Board](#) (ISEB) website, which is part of the Chartered Institute for IT, contains information relating to attainable basic and advanced ITIL qualifications, while a quick search of Google will also throw together a number of organizations providing ITIL certification services.



Greg Hamlyn, a senior technical course designer and developer with Datatrain, has over 25 years experience working in the mainframe and z/OS data center environments, having performed in numerous operational roles ranging from computer operator to national mainframe technical training manager for a large Government agency. He has presented at AFCOM and has written a number of articles relating to training methods and trends and everything that is z/OS. You can contact him at g.hamlyn@elearningpeople.com

Test Your Knowledge

With every release of this newsletter a mini online test will be provided of a product that you are likely to be using in the workplace. You may want to challenge your work colleagues or just want to confirm your own knowledge!

The test for this newsletter focuses on resolving JCL problems and consists of 15 questions. Click the link below to start.

[JCL Problem Resolution](#)

PROGRAMMERS' PLACE

Understanding Re-Entrant Programming

By David Stephens, Lead Systems Programmer at Longpela Expertise

Program long enough, and sooner or later you'll need to write a re-entrant program. But what is a re-entrant program, and how do you make one?

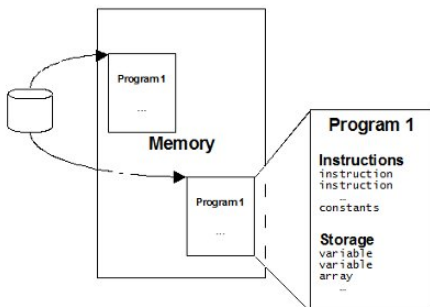
What is a Re-Entrant Program?

A re-entrant program (or a thread-safe program, if you're more familiar with UNIX) is a program that doesn't change itself. I know what you're thinking: "how can a program change itself?" Well there are actually three parts to a computer program:

- Program code – the instructions that are executed when a program runs.
- Constants – constants used in a program. These are set when the program is written, and never change.
- Storage area – or variables. This area changes as the program runs.

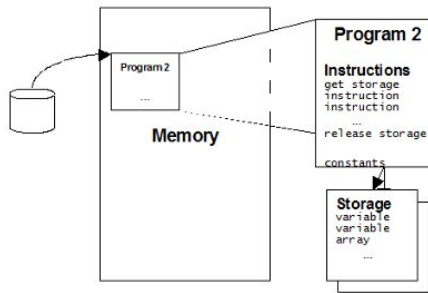
Take a look at the diagram below of Program 1. Program 1 isn't re-entrant so the code, constants and storage area are all together. When a task runs Program 1, it loads the code, constants and storage area into memory from disk, and then executes the code. So when Program 1 runs, it actually changes itself, as it changes its storage area. BMC, ASG and Compuware have openly supported movement to this version by updating their DB2 product offerings.

If a second task wants to run Program 1 at the same time, it can't run the copy that's already in memory. If it did it would overwrite the storage area still being used



by the first task. So a completely new copy must be loaded into memory from disk.

Compare this with Program 2. Program 2 is re-entrant, which means that only the code and constants are together. When a task runs Program 2, it is loaded into memory like Program 1. However Program 2 then gets and uses its own storage area and frees it at the end. This means that many tasks can use Program 2 at the same time (without loading the program from disk every time) - each time it runs it gets a new storage area.



Why Write Re-Entrant Programs?

There are lots of reasons why your program may need to be re-entrant. If you're writing in High Level Assembler (HLASM), your program must be re-entrant if it:

- Will be loaded in LPA
- Is an SVC.
- Is an exit that requires re-entrant code (which is most of them).
- Is loaded in memory and can be used by more than one task (or TCB) at the same time.
- Will be bound as a DLL.
- Is a CICS program or exit that will use the Open Transaction Environment (OTE) – or in other words, run on a TCB other than the CICS Quasi-Reentrant TCB. You can find out more about CICS and thread-safe from the references at the end of this article.

*A re-entrant, or thread-safe, program is...
A program that doesn't change itself*

Continued on page 6

In fact, I write all my HLASM code as re-entrant. This way if my program is later used in a way that I didn't originally consider, there's no chance of any re-entrancy problems. Trying to debug a program with an error caused by a re-entrancy issue can be a real nightmare. I've also found that with a bit of practice, coding a re-entrant routine doesn't take any more work or time than a non-reentrant routine.

But it's not only HLASM programs that may need to be re-entrant. Enterprise COBOL programs must be re-entrant if they:

- Run in CICS
- Are preloaded in IMS
- Will be used as DB2 stored procedures
- Run in UNIX Systems Services
- Need DLL support
- Use object-oriented syntax

Similar issues apply to PL/1 and C programs.

How to Write Re-Entrant Programs in HLLs

If you're programming in a high level language (HLL) like COBOL, the good news is that it's easy to write a re-entrant program. Just compile the program using the RENT option. Nothing simpler.

How to Write Re-Entrant Programs in HLASM

If you're writing a re-entrant program in HLASM, you need to do some extra work yourself. You're going to have to:

- Get some working storage. You'll need to Getmain your storage right at the beginning of the program, and use a DSECT to map it. Don't forget to free this storage at the end of the program.
- Use that working storage. ALL variables have to be in the DSECT that maps to our working storage. If there's any chance that it can change, use your Getmained storage.
- Use List and Execute Forms of macros. Some HLASM macros will have two special formats for re-entrant programmers: a List and Execute form. This is because many macros aren't suitable for re-entrant programs by default. Or in other words, they change their own storage when expanded out. You use the List and Execute forms like this:
 1. You specify the List form in the working storage - it saves storage for the Execute form.
 2. You use the Execute form, pointing to the List form. This forces the macro to use your working storage, rather than

changing storage inline.

Whenever you use a macro in a re-entrant program, check to see if it has separate List and Execute forms. If it does, use them. Our example below shows how.

- Serialize shared resources. You're writing a re-entrant program for a reason: so more than one task can run it at the same time. This means you have to make sure you serialize access to any shared resources.

Let's say that you have a word in memory that is shared between several different tasks. In this case, you need to do something so two different tasks won't try to update it at the same time – a way to serialize access. You can do this using the Compare and Swap (CS) instruction, or ENQ/DEQ macros (using the List and Execute forms, of course). The example below shows how to use Compare and Swap.

Binding and Running Re-Entrant Programs

When writing a re-entrant program, you must bind it as a re-entrant module using the RENT binder option. Without this option, a fresh copy of the entire module will be loaded into memory every time a programs LOADs (or LINKs) it. Some things to remember:

- RENT is not the same as REUS (re-usable) or REFR (refreshable). A reusable module does not have to be re-entrant, and a re-entrant module does not have to be refreshable.
- RENT applies to the entire module. You cannot mix re-entrant and non-re-entrant programs in one module.

Checking Your Program is Re-Entrant

Ok, so you've finished your re-entrant program. But is it really re-entrant, or are you writing over part of your program's storage without knowing? Using the assembler RENT option or RSECT statement will pick up many, but not all of your re-entrancy problems. There are a couple of ways to really make sure your program is running re-entrant:

- If your program is APF authorized, then this is already done. z/OS will load your module into subpool 252. Any program (that is not running in key 0 or supervisor state - including yours) that tries to change your module's memory will abend with a SOC4.
- Use the (undocumented) CsvRentSp252 option in your DIAGxx parmlib member. This forces all re-entrant programs (APF authorized or not) to be loaded into subpool 252.

Continued on page 7

Warning: Use this with caution, and do NOT set this value in your production environment. Some of your software (including some program products) may not be truly re-entrant. Using this option could abend your environment.

- Use the (undocumented) CsvRentProtect option in your DIAGxx parmlib member. The previous two examples won't detect problems with programs that run in key 0 or supervisor state. Using the CSVRENTPROTECT option will. With this option, any program that attempts to write over your program storage will SOC4 abend.

Warning: Again, use this with caution, and do NOT set this value in your production environment. Some of your software (including some program products) may not be truly re-entrant. Using this option could abend your environment.

- If you're running in CICS, use the RENTPGM=PROTECT SIT option. In this case, your program will get a DFHSR0622 message and protection exception/ SOC4 if it tries to overwrite its storage. Use this option with caution in your production CICS environments.

A Re-Entrant HLASM Program Example

So let's look at an example:

```

*=====
* Main Program
*=====
PGM1 RSECT (note 1)
* --- Save Callers Environment -----
      BAKR R14,0

* --- Setup Program Addressability -----
      LR R12,R15
      USING PGM1,R12

* --- Put parameter passed into R5 -----
      LR R5,0(R1) R5 -> Block

* --- Get and Address Workarea -----
      USING WORK,R13
      LA R2,@WORKLEN
      STORAGE OBTAIN LENGTH=(R2),LOC=BELOW
(note 2)
      LR R13,R1 R13 -> Workarea
      XC WORK(@WORKL),WORK Zero area

* --- Write Message to Console -----
      LA R2,#MSG
      WTO TEXT=(R2),MF=(E,WWTO) (note 3)
    
```

* --- Example of an OPEN and CLOSE Macro ----
MVC WDCB(@MODELL),#DCB Move model over
(note 6)

```

LA R3,WDCB
OPEN (R3),MF=(E,WOPEN) Open DD1
CLOSE (R3),MF=(E,WCLOSE) Close DD1
    
```

* --- Add one to original value in block ---- (note 5)

```

L R1,0(R5) R1 = value
ADDVAL DS 0H
LA R2,1(R1) R2 = value+1
L R1,0(R5) R1 = value
CS R1,R2,0(R5) Insert value+1
BC 2,ADDVAL (unless it changed)
    
```

* --- Free Working Storage and Return -----

```

LA R2,@WORKL
STORAGE RELEASE ADDR=(R13),LENGTH=(R2)
PR Return
    
```

*=====

* Constants

*=====

* --- Message Constants -----

```

#MSG DC AL2(L'#MSGTXT) (note 4)
#MSGTXT DC C'PGM 1 MESSAGE.'
    
```

* --- Macro Model Area -----

```

#DCB DCB DDNAME=DD1,MACRF=GM,MF=L
#OPEN OPEN 0,MF=L
#CLOSE CLOSE 0,MF=L
@MODELL EQU *-#DCB Length model area
    
```

LTORG

*=====

* Map Working storage

*=====

```

WORK DSECT
SAVE1 DS 18F Savearea
    
```

```

WWTO WTO TEXT=,MF=L List form
    
```

```

WDCB DCB DDNAME=DD1,MACRF=GM,MF=L
WOPEN OPEN 0,MF=L
WCLOSE CLOSE 0,MF=L
    
```

```

@WORKL EQU *-WORK Length
    
```

Notes:

1. I've used an RSECT instead of a CSECT, which tells the assembler that this is a re-entrant module. The assembler will check for some (but not all) re-entrancy errors. Using the RENT assembler option does the same thing.

2. The STORAGE macro doesn't have a List and Execute form - it is re-entrant in itself. How do we know this? Because there's no separate List and Execute form (check it out for yourself in the z/OS Assembler Services Reference manual). In our program, we get storage below the 24 bit line, as our storage includes a DCB.
3. I use the List and Execute form of the WTO macro, simply because it has them.
4. The #MSG label is a constant, so it can be left in the program code. However if it was a variable (ie. updated by the program in any way), it would have to be moved into the WORK DSECT.
5. This program shows how to use the Compare and Swap instruction to serialize access to a control block. The address of a field is passed to the program by the caller. Our program increments this value like this:
 - o It gets the original value from the control block, and puts it into R1.
 - o It increments this value and puts it in R2.
 - o It gets the original value from the control block again, and puts it in R1.
 - o In one instruction (Compare and Swap), it inserts R2 into the control block if the original value hasn't changed. Otherwise it puts the changed original value into R1, and sets the condition code to branch.
6. Some older z/OS macros include constants in their List form. In my example program, DCB, OPEN and CLOSE fall into this category.

Using the List form of these macros in my WORK DSECT loses these constants. I've worked around this by defining a model area in the program for these macros, and copied this over an equivalent area in my storage. This is a very standard way of dealing with this problem.

Let's look at my WOPEN label. If you look at the macro expansion (from the assembler output), you would see something like:

```
WOPEN OPEN 0,MF=L
+WOPEN DC 0F'0'
+ DC AL1(128)
+ DC AL3(0)
+ DC A(0)
```

You can see z/OS defining the byte constant '128.' However this doesn't work with my storage - I reset it to zeroes at the beginning.

So I define a second OPEN macro in my constants area (#OPEN - with the '128' constant). I then copy this over the WOPEN in my storage so I have that constant.

So the big question is "How do you know if you need to do this?" The only way is to look at the assembler expansion of the List form of the macro. If you see a constant, then you'll need a macro model.

There are many times when you need to write a re-entrant program. But the good news is that it isn't hard. High level language programmers just need the RENT option, Assembler programmers need to do a bit more.

If there's any chance that more than one task can execute a common copy of your program at the same time, you must make your program re-entrant. In fact many programmers make all their programs re-entrant as a matter of course, including me.

David Stephens

References

- [Writing Re-Entrant Programs](#) - Trainer's Friend article on re-entrant programming.
- [Threadsafty and the CICS Open Transaction Environment](#) - zJournal article on threadsafty and CICS.
- [Threadsafe Considerations for CICS](#) - IBM Redbook that provides excellent insight on all aspects of CICS threadsafty

Source:

Longpela Expertise, *LongEx Mainframe Quarterly* - August 2010, Retrieved from <http://www.longpelaexpertise.com.au/ezine/ReentrantBeginners.html>

In the next edition of Datatrain Mainframe Quarterly:

Retaining Data Center Skills and Shortage - Part Two - Preventing the drain of skills and knowledge using knowledge transfer programs

Learning Spotlight - Linux on System z - We look at Linux on System z and provide you with access to our new modules

Vendor Briefs - What have the major mainframe vendors on offer?

z/OS 1.12 - What's in it for me? An article aimed at both Operations and Programmers

Test Your Knowledge - Problems with Errant JCL Is This The End of Tape? Improvements in disk and virtual tape technologies offer ways to run a tapeless mainframe. Is this the end of tape?