



Welcome to the first Virtual IMS *CONNECTION* newsletter. Virtual IMS *CONNECTION* at www.virtualims.com is an independently-operated vendor-neutral site run by and for the IMS user community.

Virtual IMS *CONNECTION* presentation

The very first webinar from Virtual IMS *CONNECTION* was entitled “Why Assembler is a 21st Century Language?”, and was presented by Kristine Harper, a developer from NEON Enterprise Software.

Kristine explained how she got into Assembler programming. It was “Take Your Daughter To Work Day”, she was 12, and her father was an Assembler programmer.

Kristine went on to tell us that 98% of all instructions executed on a mainframe are written in Assembler, including DFSORT, IEBGENER, ISPF, TSO, and IMS itself. She said that anything that is high-volume execution is written in Assembler. Champions of COBOL should note that although their shop may have a lot of COBOL programs, these source code lines are instructions written, not instructions executed. And all a Doubting Thomas needs to do to prove this is look at the SMF records and see what is consuming CPU time!

All the classic reasons for using Assembler still exist, but Kristine explained that there are new arguments for using Assembler as well – such as the possibility of structured programming, HLASM has a powerful Structured Programming Macro (SPM) facility, and there are new tools and techniques that make programs easier to debug. In addition: you can use ISPF to edit and create Assembler programs easily; interfaces with other languages are supported; new instructions are popping up all the time (including new 64-bit instructions); you can use storage above the 2GB bar; HLL concepts such as inheritance, abstraction, polymorphism, and reuse are available; and when programmers take advantage of the new features of Assembler (and HLASM), code is easier to analyze and maintain, and pass on to the next generation!

[Figure 1](#) shows an example of ‘old’ Assembler.

Kristine pointed out the obvious faults of this old-style Assembler. She said: GOTOs and Jumps were as popular as reality TV today; there were labels, labels, labels everywhere; and it is very difficult (impossible) to cut/paste a code fragment filled with labels and GOTOs. In addition: ‘Top-down’ design was rare (most Assembler programs stuck to ‘flat’ design); it is difficult to create a mainline of control flow; there is no nesting; code indentation was rare; the order of complexity of code was larger than necessary;

Contents:

Virtual IMS <i>CONNECTION</i> presentation	1
Steering committee	3
Virtual IMS <i>CONNECTION</i> forums	4
Updating Virtual IMS <i>CONNECTION</i>	5
Virtual IMS <i>CONNECTION</i> meeting dates	5
About Virtual IMS <i>CONNECTION</i>	5

```

154270      CLI    0(R1),VALUE_A
154280      JNE   LABEL_1
154290      :
154300      :      (VALUE_A code)
154301      :
154302      J     LABEL_4
154303 LABEL_1 DC    0H
154304      CLI    0(R1),VALUE_B
154305      JNE   LABEL_2
154306      :
154308      :      (VALUE_B code)
154309      :
154310      J     LABEL_4
154311 LABEL_2 DC    0H
154312      CLI    0(R1),VALUE_C
154313      JNE   LABEL_3
154314      :
154315      :      (VALUE_C code)
154316      :
154317      J     LABEL_4
154318 LABEL_3 DC    0H
154321      :
154322      :      (all other code)
154323      :
154324 LABEL_4 DC    0H

```

Figure 1: 'Old' Assembler

and you had to read someone else's code over-and-over just to understand what was going on.

Over the years, Assembler has seen many new enhancements, new instructions, and upgraded tools. Among these enhancements are new USING statements that can make simultaneous references to multiple instances of a control section, among other things.

Kristine listed many of the new instructions that are available with Assembler these days. These included: BAS/BASR (replacing BAL/BALR); AHI, CHI, LHI, MHI (halfword immediates); MSR; CUSE (Compare Until String Equal); CLST (Compare Logical String); MVST (Move Logical String); SRST (Search String); CVBY, CVBG, CVDY, CVDG (which convert binary and decimal extended); CSP/CSPG: (Compare and Swap Purge); IHH, IHL, IILH, IILL (Insert Immediate); NIxx, OIxx, XIxx (Boolean Immediate); LPQ/STPQ (Load/Store Register Pair); TMH/TML (Test Under Mask High/Low); TRE (Translate Extended); TRTR (Translate and Test in Reverse); FLOGR (Find Leftmost One Bit Grande Register); SLLG, SRLG, SLAG, SRAG; and many others.

There are also a number of Assembler tools that boost productivity that Kristine highlighted. She mentioned the ASMDASM disassembler; the ASMXREF cross-reference facility; ASMPUT (Program Understanding Tool), which provides graphic displays of control flow within and among programs; ASMIDF the Interactive Debug Facility; Structured Programming Macros (SPMs); SuperC, and the file comparison utility.

Kristine then talked about structured programming techniques available with Assembler and showed examples of what 'new' Assembler looks like. One of her examples is shown in [Figure 2](#).

Kristine made a number of interesting concluding remarks. She pointed out that Assembler was definitely a 21st century language and remarked that there have been five releases of HLASM since 1992.

```

079300      ST      R14,BLDCMA_CPA5000      A(where to go back to)
079400
079500      DO      UNTIL=(CR,R1,GT,R9)      Examine statement
079600      LR      R14,R9                      A(end)
079700      SR      R14,R1                      L(to scan)
079800      LA      R2,0                        Initialize
079900      LR      R15,R1                      A(Start of text scan)
080000      EX      R14,CPATRT1              Scan statement
080100      IF      (LTR,R2,R2,NZ)           If something found
080200      CASEENTRY R2,VECTOR=B,POWER=2   Act on result of TRT
080300
080400      CASE 4                               A quote was found
080500      BRAS R14,BLDCMA_6000             Analyze & copy data
080600
080700      CASE 8                               A slash was found
080800      BRAS R14,BLDCMA_6300             Analyze & copy data
080900
081000      CASE 12                            A plus or minus was found
081100      BRAS R14,BLDCMA_6600             Analyze & copy data
081200
081300      ENDCASE
081400
081500      ELSE
081600      LR      R1,R9                      A(last byte of data)
081700      BRAS R14,BLDCMA_6900             Copy remaining data
081800      ENDIF
081900      LA      R1,1(.R1)                  Point past byte examined
082000      ENDDO
082100
082200      L      R14,BLDCMA_CPA5000         A(where to go back to)
082300      BR      R14                        Return

```

Figure 2: 'New' Assembler

She also noted the problems facing all mainframe sites in that while the language is robust and successful, the people who use it all the time are in the main moving towards retirement, and the younger people at college are more interested in Java than Assembler. Hopefully, initiatives such as IBM's Academic Initiative will overcome this. Kristine felt that Assembler had lots of appeal – and a bright future.

Steering committee

Like all User Groups, the Virtual IMS *CONNECTION* User Group needs a steering committee to ensure that members are getting exactly what they want from the user group. The steering committee's role will include deciding what topics would make interesting presentations at the bi-monthly meetings and finding speakers to give those presentations. It will also suggest content for the newsletter and perhaps even provide it. In addition, the steering committee will be able suggest what content the membership would like to see available on the Web site and perhaps go some way towards making that happen. Lastly, the steering committee will make decisions about its own role and composition.

The first meeting of the new steering committee will be before the end of February, and we hope to have monthly meeting after that. We will use Webex on the computer and a telephone to conduct the meetings. Agendas for the first meeting will be sent out soon to everyone interested in being a member.

I have had a number of people contact me already and offer their help on the steering committee. If anyone else would like to be a member of the steering committee please contact me soon at trevor@itech-ed.com or via the Contact Us page of the virtualims.com Web site.

Thank you to everyone who volunteers for this.

Virtual IMS *CONNECTION* forums

I am hoping that the forums will take off in a big way and people will use them fully. However, so far, people are being a bit shy about leaving a comment or asking a question.

I am hoping that once the forums start to be used I will be able to divide them up into general topic areas. I was thinking of something along the lines of:

- **General IMS discussion** – just anything people want to say or ask about IMS usage.
- **Application development** – Assembler, COBOL, PL/1, Java, REXX, etc.
- **Code swap** – a way to share clever bits of code, and also a place to ask for solutions to specific problems.
- **Migration and configuration** – a discussion of migration and/or configuration issues and solutions.
- **How tos** – a place where people can ask how to do something, and other IMS experts can explain it or explain where to find the answer
- **Hints and tips** – general IMS advice from people who wished they'd known the advice themselves when they started their project(!) and anything else people pick up along the way that's worth sharing.

Let me know what you think, and please, start using the Virtual IMS *CONNECTION* forums.

Stop Talking about IMS SOA...
And do it...

IMS and Ivory®:
A Perfect Combination for Mainframe SOA

Get this new White Paper at www.gtsoftware.com



Updating Virtual IMS CONNECTION

The Virtual IMS *CONNECTION* Web site tries to stay up-to-date in terms of IMS news and identifying IMS-related articles that have been published. The IMS Tools page also tries to reflect the latest IMS-related products that are available from IBM and all third-party vendors. We also try to stay current with information about IMS resources and IMS-related events around the world.

However, if you know better, if you spot a mistake, or know of more recent information, then please tell us straight away. You can contact us using the Contact Us page of the Web site or e-mail me at trevor@itech-ed.com.

Virtual IMS CONNECTION meeting dates

The following meeting dates have been arranged for the Virtual IMS *CONNECTION* user group:

- **February 5, 2008** – Bill Keene IMS disaster recovery preparation – part 1.
- *April 8, 2008* – Bill Keene IMS disaster recovery preparation – part 2.
- *June 10, 2008* – to be announced.

All meetings start at 10:30 CST (4:30 GMT). All members will be e-mailed the Webex access code and password for each meeting. Members need to log in about five minutes before the meetings starts in order for the Webex connection to complete and to dial in so they can hear and join in with the audio component of the meeting.

About Virtual IMS CONNECTION

Virtual IMS *CONNECTION* is a user group established as a way for individuals using IBM's IMS hierarchical database and transaction processing systems to exchange information, learn new techniques, and advance their skills with the product.

The Web site at www.virtualims.com provides a central point for coordinating periodic meetings (which will be technically-oriented topics presented in a webinar format), and will provide articles, discussions, links, and other resources of interest to IBM IMS practitioners. Anyone with an interest in IMS is welcome to join Virtual IMS *CONNECTION* and share in the knowledge exchange.

To share ideas, and for further information, contact trevor@itech-ed.com.

The Virtual IMS *CONNECTION* user group is free to its members. Various sponsorship opportunities are available to vendors of IMS-related products. Full details of these opportunities and can be found *here*.