MANEWS 04

==
==

M A News
Mainframe Audit News
February, 2003
Issue Number 04

==
==

Table of Contents

I N T H I S I S S U E
==
==

   (YOU CAN USE the Find function of your email software or Adobe Acrobat to jump directly to a given section. For example, to jump directly to the Websites section, you would Find on "7)" or on "Websites".)
===
===

1) Introducing the Mainframe Audit News

This is the fourth edition of the Mainframe Audit News, a vehicle for sharing information about auditing IBM mainframe computers.

The MA News is a free, email, newsletter for auditors who need (or

suspect that they will need) to be auditing IBM mainframe systems (primarily MVS, OS/390, z/OS, and the system software associated with them). This software includes: CICS, DB2, JES, VTAM, MQSeries, TSO, USS (UNIX System Services), TCP/IP, and others. It also includes the Websphere server software which connects a mainframe to the Internet. (Note, we will expand each of these acronyms and explain how the software works over the course of the next several issues.)

We expect to have a new issue at least every six months.

The MA News is meant for auditors who are new to IBM mainframes, as well as for experienced MVS auditors who want to keep up to date with the latest developments from IBM. We will not make the list of subscribers available to anyone else for any reason.

To Sign Up for the Mainframe Auditors' Newsletter, see section 9).

===
===

2) What Should Be In Standing Files for Mainframes?

Standing files of course are the permanent files kept by auditors to provide basic background information to be used in planning future audits.

For mainframes they should include descriptions of:

? Each data center and the computers it houses
? The communications links to each data center, including **NJE** (Network Job Entry), **RJE** (Remote Job Entry), **TCP/IP** (Transmission Control Protocol/Internet Protocol), **APPC** (Advanced Program to Program Communication).
? The major system software in each data center: versions of OS/390 or z/OS, CICS regions, Websphere webserver software, MQ Series sub-systems, DB2 sub-systems, other database management software, security software (RACF, ACF2, or TopSecret)
? The major applications executing in each data center (Sales, General Ledger, etc)
? For each data center, a list of previous audits, including whether they addressed security, effectiveness, efficiency, management controls (capacity planning, problem

management, response time management, automatic backups, etc.), or other aspects.

Much of this information can be collected by use of the MVS Operator **DISPLAY** command described later in this issue.

===
===

3) What's the Difference Between TCP, IP, and IPv6?

TCP stands for Transmission Control Protocol, the protocol for managing applications over IP. **IP** is the Internet Protocol, the protocol which provides message routing, but not applications It uses **DNS** (Domain Name System) names.such as http://www.stuhenderson.com. It uses IP addresses (similar to telephone numbers) to help computers know where to route the messages over the network. Each IP address is made up of four numbers separated by dots, such as **192.101.14.5**. Each of these numbers can range from zero to 255. You should think of each IP address as representing one computer. IP provides translation between the DNS names and the IP addresses.

IPv6 is the new version of IP, which uses longer addresses to support more addresses (just as adding more digits to phone numbers means we can have more phone numbers). The z/OS system software now supports IPv6, as well as the older IP addressing scheme, which is called IPv4. A useful IBM manual is "IPv6 Network and Application Design Guide", SC31-8885.

With IPv4, addresses consist of 32 bits (four "octets" of 8 bits each, separated by periods, for example: **9.10.11.12**). With IPv6, addresses consist of 128 bits (eight pieces of sixteen bits each, separated by colons, for example**: 261:262:263:264:265:266:267:268**).

Whether IPv4 or IPv6, IP provides message routing. It carries **packets**, which may be TCP packets, UDP packets, ICMP packets, or other type of packets. (**UDP** stands for User Datagram Protocol; **ICMP** stands for Internet Control Message Protocol.). TCP packets each specify a port number. The port number identifies a specific application, for example: Port 25 is almost always email. Port 80 is almost always the webserver (the software that is the other end of the conversation with Netscape or Internet Explorer. Since Port 80 on the mainframe is the web server, then Netscape or Internet Explorer must be the web client.)

This whole arrangement permits delegation of authority. IP just routes the messages and doesn't care what's in them. When the message reaches its destination, IP hands it to the TCP software (or UDP software, or whatever protocol is involved). TCP looks at the content of the message to see what port is involved. TCP then hands the packet to the software (that is, the <u>server</u>) that handles that port. In this way, TCP doesn't need to worry about message routing; IP does that. TCP also doesn't have to worry about the format of email messages, since the email server does that.

All OS/390 and z/OS systems now come with TCP and IP automatically installed.

===
===

4) How Can I Use CLISTs and REXX with System Symbols?

CLISTs and REXX execs let you determine the value of system symbols and other variables to collect information useful to your audit..

4A) >>>>CLISTs

_____

TSO is the software that lets programmers at terminals edit files, test programs, and do similar functions. It is often called a "programmer's workbench". It can also server as an auditor's workbench. Sometimes a programmer will want to issue the same series of commands over and over again, in some cases, varying one field from execution to execution. CLISTs make this possible. A **CLIST** is a command list, that is a disk file, each record of which is a single TSO command. When the programmer tells TSO to execute the CLIST, TSO executes the commands in the file one after another. CLISTs generally start with a PROC statement to identify them as CLISTs. Here's a sample CLIST named **X** to write a message to the programmer's terminal:

```
PROC 0
WRITE THIS IS A CLIST EXAMPLE
EXIT
```

(In this example, the **PROC** statement and the **EXIT** statement just mark the beginning and end of the CLIST. The "0" in the PROC statement is the number of named variables in the CLIST, which you may safely ignore for now. The **WRITE** statement is the only statement that is actually executed. It writes the following text to the terminal screen. If I then tell TSO to execute X, TSO will read the file and execute the WRITE statement.)

If I wanted to change the CLIST to issue a different message every time I execute it, then I would use a substitution variable named &1 like this:

PROC 0
WRITE   THE WORD FOR TODAY IS &1
EXIT

Now if I tell TSO to execute

X MUGWUMP

TSO will execute the CLIST, substituting **MUGWUMP** for the **&1**. I will then see on my terminal:

THE WORD FOR TODAY IS MUGWUMP

(The "1" in "&1" means "the first item typed after the name of the CLIST". You can have an &2, &3, and so on, but let's keep things simple for now.)

You can submit a CLIST as a batch job, which will result in the output going to the printer instead of to your terminal screen. This gives you hard copy for your workpapers. Ask the system programmer or administrator to provide you with the necessary JCL (Job Control Language).

To execute a CLIST, you need to make it be either a dataset or a member of a partitioned dataset. For example, if it is a member of a dataset named **MY.CLIST.LIB**, then to execute it you would issue the TSO command

EXEC 'MY.CLIST.LIB(X)'.

(The parentheses around the X indicate that it is a member of MY.CLIST.LIB.)

If **MY.CLIST.LIB** is allocated to my TSO session with a **DDNAME** of **SYSPROC**, then I can execute X by typing the TSO command:

X

(Ask the system programmer for this allocation if you want it. DDNAME is the name of the JCL or Job Control Language statement defining the dataset to the TSO session.)

4B) >>>>Control Variables

There are other variables beyond <u>&1</u> called **control variables**, whose names begin with an ampersand and which provide information about a system. Here's a CLIST to illustrate:

```
PROC 0
WRITE  MVS LEVEL IS: &SYSMVS
WRITE SYSPLEX IS: &SYSPLEX
WRITE SYSNAME IS: &SYSNAME
WRITE SYSSMFID IS: &SYSSMFID
WRITE SYSLRACF IS: &SYSLRACF
WRITE SYSRACF IS: &SYSRACF
WRITE SYSTSOE IS:&SYSTSOE
WRITE SYSDATE AND SYSTIME IS: &SYSDATE &SYSTIME
EXIT
```

Here's what they mean, along with a few others not included above:

? **&SYSMVS** (Level of MVS)
? **&SYSPLEX** (sysplex name)
? **&SYSNAME** (system name for the MVS system)
? **&SYSSMFID** (system identifier used by SMF (System Management Facility) logging software, as specified in SYSPRMxx member of parmlib)
? **&SYSLRACF** (Level of RACF security software)
? **&SYSRACF** (whether RACF security software is installed and/or available [if its available, then you can assume that it is also installed])
? **&SYSTSOE** (level of TSO software)
? **&SYSDATE** (current date, to document for working papers)
? **&SYSTIME** (current time)

Some others not included in the example above:

? **&SYSCLONE** (last two characters of system symbol &SYSNAME)
? **&SYSJES** (level of JES (Job Entry Subsystem) software
? **&SYSNODE** (name of the node JES uses to call this location)
? **&SYSOPSYS** (z/OS operating system name and level)
? **&SYSSYMDEF** (system symbols used to name the system and its components)

As you can imagine, this can be a nice tool for collecting data about a system for an audit. . The IBM manual "TSO/E CLISTs" provides more information.

4C) >>>>REXX Execs

REXX (Restructured eXtended eXecutor)is a language which does everything CLISTs do, but more, and more elegantly. Each REXX program is called an **EXEC** and consists of a disk file containing REXX commands. Here's an example of a REXX EXEC (We wont' teach you the language here, but we'll show you enough to collect audit information.):

```
/* REXX PROGRAM */
/*                    */
/* EVERYTHING BETEEN SLASH-ASTERISK AND ASTERISK-SLASH*/
/* IS A COMMENT  */
SAY 'THIS IS A SAMPLE REXX PROGRAM'
EXIT 0
```

You can see that REXX uses the **SAY** command to write to the terminal screen. It also uses a function called **SYSVAR** to collect information about the system. The SYSVAR function is comparable to the control variables in CLISTs. Furthermore, some information in REXX comes from the SYSVAR function. Other information comes from the **MVSVAR** function. Here's a REXX EXEC to collect information about the system using SYSVAR and MVSVAR:

```
/* REXX PROGRAM  */
SAY 'HERE ARE SOME SYSTEM VARIABLES:'
SAY 'JES LEVEL IS ' SYSVAR(SYSJES)
SAY 'JES NODE NAME IS ' SYSVAR(SYSNODE)
SAY 'RACF AVAILABILITY IS ' SYSVAR(SYSRACF)
SAY 'RACF LEVEL IS ' SYSVAR(SYSLRACF)
SAY 'MVS LEVEL IS ' MVSVAR(SYSMVS)
SAY 'SYSNAME IS ' MVSVAR(SYSNAME)
SAY 'SYSOPSYS IS ' MVSVAR(SYSOPSYS)
SAY ' SYSSMFID IS ' MVSVAR(SYSSMFID)
SAY 'SYSCLONE IS ' MVSVAR(SYSCLONE)
SAY 'SYSPLEX IS ' MVSVAR(SYSPLEX)
SAY 'SYMDEF IS ' MVSVAR(SYMDEF)
```

EXIT 0

For explanations of the meanings of the fields, please see the comparable explanations above for CLISTs. The IBM manual "TSO/E REXX User's Guide" provides more information.

You execute REXX EXECs the same way you do CLISTs, with the TSO command **EXEC,** except that you follow the command with the keyword EXEC to flag it as REXX. For example, if you have a REXX EXEC as a member named **Y** in the dataset **MY.REXX.LIB**, you would execute it with:

EXEC 'MY.REXX.LIB(Y)' EXEC

You can also save keystrokes by storing the REXX EXEC in a dataset which is allocated to your TSO session with the DDNAME of SYSPROC (the same as with CLISTs). In this case, you would just type:

Y

REXX EXECs can also be used to display control blocks in memory. This can be a fine tool for hackers or auditors to use. But we will leave that for another issue.

==
==

5) Question and Answer Column

(Readers are invited to send Questions to stu@stuhenderson.com, along with an indication of whether we should print your name. If you need an immediate reply to your question, please indicate this in your email.)

Q) What are subsystem consoles and how do I address them in my audit?

.
A) Subsystems are programs which get started up automatically under MVS, and which sit in the back of the computer waiting for someone to ask them to perform some function to which they are dedicated. The mainframe term subsystem is in some ways comparable to the UNIX term server or daemon. For example, DB2, the database software is a subsystem, so is MQ Series, so is JES, and so is RACF.

The MVS system programmer specifies the names of the subsystems and the options with which they are to be started in a member of parmlib whose name begins

IEFSSN…. One of the options that can be specified is the name of the operator console where this subsystem's messages are to be written. This is the subsystem console for that subsystem.

Subsystem consoles will likely not be an important part of your audit. More to the point, functional subsystems represent one of the "back doors" that let a program execute with the privileges of the operating system. You will want to concentrate on evaluating the controls over the back door to make sure that it doesn't introduce a security exposure.

===
===

6) How Can Auditors Use the Operator DISPLAY Command?

The operator **DISPLAY** command (abbreviated **D**) is used by the operator to display information about the system and its workload. We often picture an operator at a system console typing in operator commands and viewing the responses. But operator commands can be entered from other sources, including TSO sessions. A TSO session can be turned into a virtual console by issuing the TSO command **CONSOLE**. (You need to have permission from the security software to do this. Also the security software should be used to control which operator commands a user is permitted to issue.) Another technique is to issue the TSO command **OPERATOR** (which also requires permission from the security software.) The TSO OPERATOR command allows you to issue the following OPERATOR sub-commands from within TSO: **CANCEL**, **DISPLAY**, **END**, **HELP**, **MONITOR**, **SEND**, **SLIP**, and **STOPMN**. We will discuss here just the END, HELP, and DISPLAY commands.

Use the **END** command to finish OPERATOR command processing and to get back to normal TSO processing.

Use **HELP** to learn the possible commands. Use **HELP commandname** to learn the syntax of one of the commands.

The **DISPLAY** sub-command supports many of the sub-operands of the regular operator DISPLAY command. So for example, if you are in TSO with the READY prompt on your screen, and you want to display information about: what subcommands you can issue under OPERATOR; and then about how to issue the DISPLAY command; and then about device 567 to learn what disk pack lives on it, you might issue:

    OPERATOR
    HELP

HELP DISPLAY
D U ,,,567
END

Before your audit begins, you will want to request a TSO userid with AUDITOR privileges in the security software, and with the ability to issue the DISPLAY operator command.

6A) >>>>What Info Can You Collect With DISPLAY for an Audit?
_____

You can issue the **DISPLAY** command with the following operands (not a complete list, but a list we think useful to auditors):

? **CF** (for info about Coupling Facilities, the boxes that connect mainframes together to form a sysplex
? **CONSOLES** (for info about what consoles exist on the system, but not including TSO sessions turned into virtual consoles or EMCS consoles. For those, use EMCS INFO)
? **EMCS INFO** (for info about EMCS or virtual consoles)
? **IPLINFO** (for IPL information, including start-up options such as parmlibs and system symbols)
? **LOGGER** (for info on the system logger)
? **M** (for info on the system hardware configuration)
? **OMVS** (for info on OMVS, which is now called USS, UNIX under MVS)
? **OPDATA** (for info on functional subsystems and the control characters used to direct commands to each one)
? **PARMLIB** (for info about parmlibs)
? **PROD** (for info on registered software products)
? **PROG,APF** (for info about APF authorized libraries)
? **PROG,EXIT** (for info about dynamic exits)
? **PROG,LNKLST** (for information about LNKLST libraries)
? **SMF** (for information about SMF [System Management Facility])
? **SSI** (for information about subsystems)
? **SYMBOLS** (for information about system symbols)
? **U** (for information about devices)
? **XCF** (for info on cross system coupling about a sysplex)

The IBM manual "MVS System Commands Reference" provides more information, as do earlier issues of this newsletter, Issue 2 in particular.

MANEWS 04

(We recognize that while many of the acronyms in this list have been explained in earlier issues, some have not. This violates the policy we set for this newsletter of explaining every acronym as we use it. However, to explain all the acronyms associated with the DISPLAY command would more than fill this issue. To make up for this, we promise in subsequent issues to explain all of these acronyms.)

6B) >>>>Examples of DISPLAY Commands and Their Output
_____

D IPLINFO will get you output that looks like this:

IEE254I …
SYSTEM IPLED AT 23.59.59 ON 12/31/2003
RELEASE OS/390 02.10.00
USED LOADXY IN PARMLIB ON 0567
MTLSHARE = Y
IEASYM LIST = AB, CD
IEASYS LIST = AB
IODF DEVICE 0345
IPL DEVICE 0567 VOLUME IPL001

This tells you when the system was IPLd, and what release of OS/390 was used. It also tells you that in determining which parmlibs to use (the datasets that tell MVS its options), MVS used the member **LOADAB** in the parmlib on disk drive 0567. This resulted in using the symbols defined in **IEASYMAB** and **IEASYMCD**, as well as startup parameters as defined in member **IEASYSAB**. You might want to know what disk is on disk drive 0567. In this case you could figure it out, since the last line tells you that the IPL device was **0567** and that it holds disk **IPL001**. If you didn't know that, you might use a **D U** command to ask about disk drive 0567:

D U,,,/567 will get tell you the volser of the disk pack on disk drive 0567.

D PARMLIB will get you output that looks like this:

PARMLIB DATASETS SPECIFIED AT IPL
ENTRY FLAGS VOLUME DATA SET
1   S   DISK17 STU.TEST.PARMLIB
2   S   DISK18 STU.PROD.PARMLIB
3   D   DISK03 PARMLIB

==
==

7) Websites and Other InfoSources for Mainframe Auditors, Seminar Information, and the Proverb of the Day

7A) >>>>Websites and Other InfoSources for Mainframe Auditors

_____

Here are more websites useful to mainframe auditors. We'll let you
judge them for yourself, rather than trying to explain what each one does. We
do not endorse (nor disparage) any of their products, nor the quality of their
websites, since we have not had an opportunity to evaluate them. However, we
find them interesting, and hope you will too.

  These websites are of interest to auditors interested in information security
certification:

  www.cccure.org/ (info on the CISSP, Certified Information Systems Security
Professional)
  www.isaca.org/cism.htm (info on the CISM, Certified Information Security
Manager certification from ISACA)

  These websites (suggested by DelCreo, Inc.) provide information on HIPAA, the
Health Information Portability and Accountability Act).   Rebecca Herold, Chief Privacy
Officer of DelCreo, Inc. points out that final security standards for HIPAA have been
adopted. The security standards work in concert with the final privacy standards
adopted by HHS last year and scheduled to take effect for most covered entities on
April 14, so many organizations may be faced with audits for "HIPAA compliance" in the
near future

  http://www.cms.hhs.gov/hipaa/hipaa2
  http://www.aspe.hhs.gov/admnsimp/
  http://www.hhs.gov/news/press/2002pres/hipaa.html.

  Gerald Mackenzie offers these websites he has found useful:

  http://publib.boulder.ibm.com/cgi-bin/bookmgr/library
  http://www.auditnet.org/

http://www.theiia.org/itaudit/
http://www.itgovernance.org/index2.htm
http://www.isaca.org/gir/girMenu.cfm

7B) >>>>Seminar Information

_____

This issue we highlight two companies which offer seminars useful to mainframe auditors:

    The Henderson Group and

    DelCreo, Inc.

    The Henderson Group : Offers these "How to Audit..." courses :

?   How to Audit **Cross-Platform Applications** (April 2-3, 2003 in MD)
?   How to Audit **Mainframe/Internet Connections** (March 4-5, 2003 in FL)
?   How to Audit **TCP/IP** (April 9, 2003 in MD)
?   How to Audit **CICS** (March 3, 2003 in FL)
?   How to Audit **RACF** (April 7-8, 2003 in MD)
?   How to Audit **MVS** (April 4, 2003 in MD)

    To learn more about them, please go to

    http://www.stuhenderson.com/XAUDTTXT.HTM

    DelCreo, Inc.: Offers these courses

?   **HIPAA Security & Privacy Workshop** (March 10, 2003) in FL)
?   **Continuity Planning Seminar: Understanding ROI, Developing Strategy, & Creating Shareholder Value** (March 11-12, 2003 in FL)
?   **CISSP Prep Workshop** (April 8-10th. 2003) in Atlanta, GA

    Please note that the first two of these classes are offered in the same site in Florida as the Henderson Group seminars, and that a 10 percent discount is available to anyone taking one seminar each that week from the Henderson Group and from DelCreo. For more information on the discount, and for descriptions of other DelCreo classes, please see:

    http://www.delcreo.com/training_courses.htm

7C) >>>>This Issue's Proverb of the Day

_____

(Paraphrased from Peter Drucker) "*The greatest challenge facing managers is to determine what information we need, then to make sure that we get it, to determine what information we have that others need, then to make sure that they get it.*"

When we listen to all the "analysis" about whether the FBI and CIA should have prevented and predicted the attacks of 9/11, and all the re-organizations to "fix" the problem, we should remember to include information flow in our improvement suggestions. As auditors, we should consider "information effectiveness" to address the questions of whether our auditees' operations are receiving the information they need and using it effectively.
===
===

8) Tell Us What You Think

We'd love to hear from you, in particular on these topics:

?   What do you like/not like about the MANEWS?
?   What websites do you know that you want to share with other auditors?
?   What topics and/or columns would you like to see in future issues?
?   Is your mainframe connected to the Internet and do you plan to audit the security implications of this connection?

Please email your comments to stu@stuhenderson.com. Thanks.
===
===

9) How to Subscribe/Unsubscribe

This section shows you how to:

9A) Subscribe, Unsubscribe, Request Back Issues
9B) Take advantage of our free technical support for mainframe auditors.

9A) >>>>To Subscribe, Unsubscribe, or Request Back Issues for the Mainframe Auditors' Newsletter (MA News)

_____

Send an email to: **stu@stuhenderson.com** with the subject field set to: **MA News**
and in the body of the email just the phrase you want: **SUBSCRIBE** or **UNSUBSCRIBE**
or **BACK ISSUES: 1, 2**

_____

**9B) >>>>To Get Questions Answered from the Mainframe Auditors' Newsletter
(MA News)**

_____

Send an email to: **stu@stuhenderson.com** with the subject field set to: **MA News**
and in the body of the email the word: **Question**: (followed by
your question, and tell us whether you want your name included or not if we
decide to publish your question and answer)
Whether we print your question and answer or not, we will try to email you
back an answer to your question within five business days. If you need a
faster answer, please phone your question to (301) 229-7187, leaving the
question on the machine. Please repeat your phone number slowly and clearly.
===
===

10) Feature Article: How Can I Collect Basic System Information?

You will want to collect basic information about your mainframe computers for
your standing files, in order to be able to plan what audits should be scheduled in the
upcoming year. In addition, at the beginning of a mainframe audit, you will want some
basic information about what software and hardware is in place, what options have
been set, and what the workload consists of. Here are some basic data-gathering steps
to take:

? Get the ability to collect the information before the audit begins
? Get information about the IPL process, parmlibs, and system symbols
? Get information about hardware and software and workload

10A) >>>>Ability to Collect Info
_____
Before the audit begins, request:

? A <u>TSO userid</u> with **AUDITOR** privileges in the security software and with the ability to issue the operator **DISPLAY** command
? READ permission for that userid for all system datasets
? Use of available tools such as **CA-EXAMINE**, **CONSUL**, and **JANUS ICU**.

10B) >>>>IPL Process, Parmlibs, and System Symbols
_____

The **IPL** (<u>Initial Program Load</u>) process is the process of starting up the MVS operating system. This is when most of its options are initially set, primarily by reading specifications in a dataset called parmlib. This dataset used always to have the name **SYS1.PARMLIB**. More recently it can have other names, and can consist of several datasets concatenated or stuck together. One important set of options is the system symbols, substitution values for names identifying MVS images and other items. You need to collect for your working papers basic information on what options are set. To do this:

Issue the operator command **DISPLAY IPLINFO**

Issue the operator command **DISPLAY U,,,/567** (replacing 567 with the device number obtained from the IPLINFO output)

Use CLISTs or REXX execs in TSO to collect the information described in section 4) above

10C) >>>>Hardware and Software and Workload
_____

You will need to know what hardware is on the system, what other systems it is connected to, what system software is used, and what subsystems are in use. To collect this information, consider using the following operator commands:

DISPLAY CF
DISPLAY CONSOLES
DISPLAY EMCS INFO
DISPLAY M
DISPLAY OMVS
DISPLAY PARMLIB
DISPLAY PROD
DISPLAY PROG,APF
DISPLAY PROG,EXIT
DISPLAY PROG,LNKLST
DISPLAY SMF
DISPLAY SSI
DISPLAY SYMBOLS
DISPLAY XCF