January 1999

In this issue

3  Direct console I/O in REXX
13 Mainframe Programming Web site
23 A full screen console interface – part 6
41 Displaying ‘pseudo-graphics’ – part 2
52 VM news

© Xephon plc 1999
Editor
Robert Burgess

Disclaimer
Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, EXECs, and other contents of this journal before making any use of it.

VM Update on-line
Code from VM Update can be downloaded from our Web site at http://www.xephon.com; you will need the user-id shown on your address label.

Contributions
Articles published in VM Update are paid for at the rate of £170 ($250) per 1000 words for original material. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our Notes for Contributors.
Direct console I/O in REXX

GENERAL DESCRIPTION

REXX-only-based terminal-oriented programs require a corresponding built-in REXX tool, giving fast direct console I/O without the use of additional terminal interfaces. ESAY is just such a tool – created as a REXX function as a part of the RXSYSFN function package.

ESAY is based on the faster CMS full screen console services for 3270 display terminals. It provides fast console access, using the CMS CONSOLE macro, to perform the following functions:

• Opening and closing a path.
• Writing REXX data from a position on the terminal defined by row and column number.
• Writing REXX data from a position on the terminal defined by row and column number, and generating an input field on the screen with defined length. ESAY then waits for the input and returns the entered data into the REXX variable.

ESAY is written in Assembler and runs under CMS with VM/SP Release 5.

ESAY USAGE

To the user, ESAY appears to be like an ordinary REXX function. It is always available if the function package RXSYSFN MODULE resides on an accessible mini-disk.

ESAY is invoked as shown below:

    ESAY(row[, col, expression [, length]])

where:

• ‘row’ is the row on the screen, and is between 1 and 24, or zero. If row has value of 0, then the path to the console is closed.
• ‘col’ is the column on the screen and is between 1 and 80.
• ‘expression’ is the data to be displayed.
• ‘length’ is the length of the generated input field.

The current implementation restrictions of ESAY are:
• The truncation length of ‘expression’ is 40 bytes.
• The truncation value of ‘length’ is 30 bytes.

The path must be closed before the end of the REXX EXEC and return to the CMS interactive environment. Because ESAY uses the same path name on every occasion, leaving the REXX environment without explicitly closing the path may cause problems on the next occasion that ESAY is invoked.

When the parameter ‘length’ is specified, ESAY creates an input field with a length of ‘length’ and waits for the console input. The first byte of data, returned to REXX, is the Attention Identifier (AID). Valid values of AID are shown in Figure 1.

Note: the clear key – AID X'6D'(_) – is caught internally by ESAY and will never be passed to REXX.

ESAY only clears the terminal screen when it opens the path to the console. This means that all subsequent output to the screen will be visible, until it is overwritten.

The displayed field may be erased in the following ways:
• Overwrite the field with a field of equal length.
• Close the path to the console. However, this will erase any other fields that are displayed on the terminal.

ESAY USAGE
Examples of ESAY usage are:
• Display help in row 24, prompt in row 7. If the user presses the PF3 key, execution is stopped.
  
  ESAY(24,1,'1-Help 2-Proc 3-Quit')
USER_ACTION = ESAY(7,1,'Select action',1)
IF SUBSTR(USER_ACTION, 1, 1) = '3' THEN /* PF3 selected */
DO
  ESAY(Ø)
  EXIT
END

• Display the title in row 9 and wait for user input; then clear the screen and display two messages in rows 3 and 4:

  USER_ACTION = ESAY(9,9,'System a ready to start',1)

---

**Figure 1: Valid values of AID**

<table>
<thead>
<tr>
<th>Hex</th>
<th>Char</th>
<th>AID</th>
</tr>
</thead>
<tbody>
<tr>
<td>7D</td>
<td>'</td>
<td>Enter</td>
</tr>
<tr>
<td>F1</td>
<td>1</td>
<td>PF 1 key</td>
</tr>
<tr>
<td>F2</td>
<td>2</td>
<td>PF 2 key</td>
</tr>
<tr>
<td>F3</td>
<td>3</td>
<td>PF 3 key</td>
</tr>
<tr>
<td>F4</td>
<td>4</td>
<td>PF 4 key</td>
</tr>
<tr>
<td>F5</td>
<td>5</td>
<td>PF 5 key</td>
</tr>
<tr>
<td>F6</td>
<td>6</td>
<td>PF 6 key</td>
</tr>
<tr>
<td>F7</td>
<td>7</td>
<td>PF 7 key</td>
</tr>
<tr>
<td>F8</td>
<td>8</td>
<td>PF 8 key</td>
</tr>
<tr>
<td>F9</td>
<td>9</td>
<td>PF 9 key</td>
</tr>
<tr>
<td>7A</td>
<td>:</td>
<td>PF 10 key</td>
</tr>
<tr>
<td>7B</td>
<td>#</td>
<td>PF 11 key</td>
</tr>
<tr>
<td>7C</td>
<td>@</td>
<td>PF 12 key</td>
</tr>
<tr>
<td>C1</td>
<td>A</td>
<td>PF 13 key</td>
</tr>
<tr>
<td>C2</td>
<td>B</td>
<td>PF 14 key</td>
</tr>
<tr>
<td>C3</td>
<td>C</td>
<td>PF 15 key</td>
</tr>
<tr>
<td>C4</td>
<td>D</td>
<td>PF 16 key</td>
</tr>
<tr>
<td>C5</td>
<td>E</td>
<td>PF 17 key</td>
</tr>
<tr>
<td>C6</td>
<td>F</td>
<td>PF 18 key</td>
</tr>
<tr>
<td>C7</td>
<td>G</td>
<td>PF 19 key</td>
</tr>
<tr>
<td>C8</td>
<td>H</td>
<td>PF 20 key</td>
</tr>
<tr>
<td>C9</td>
<td>I</td>
<td>PF 21 key</td>
</tr>
<tr>
<td>4A</td>
<td>[</td>
<td>PF 22 key</td>
</tr>
<tr>
<td>4B</td>
<td>.</td>
<td>PF 23 key</td>
</tr>
<tr>
<td>4C</td>
<td>&lt;</td>
<td>PF 24 key</td>
</tr>
<tr>
<td>6C</td>
<td>%</td>
<td>PA 1 key</td>
</tr>
<tr>
<td>6E</td>
<td>&gt;</td>
<td>PA 2 key</td>
</tr>
<tr>
<td>6B</td>
<td>.</td>
<td>PA 3 key</td>
</tr>
<tr>
<td>F0</td>
<td>Ø</td>
<td>TEST</td>
</tr>
</tbody>
</table>
ESAY(0)          /* clear the screen */
ESAY(3,33,'System initialization at' TIME())
ESAY(4,33,'X-subsystem activated at' TIME())

INSTALL EXEC

/* clear the screen */
ESAY(3,33,'System initialization at' TIME())
ESAY(4,33,'X-subsystem activated at' TIME())

INSTALL EXEC

*******************************************************************************
*** INSTALL          generate RXSYSFN MODULE         *** DG'98  
*******************************************************************************
*******************************************************************************
*** SIZE 00043  VER 1.0 MOD 000                                       
*******************************************************************************
*******************************************************************************

CLRSCRN
MESSAGE = 'user request'
SAY ' — Start RXSYSFN MODULE generation - reply Y or N'
PULL REPLY
IF REPLY = 'Y' THEN
SIGNAL ERROR
SET CMSTYPE HT
STATE RXSYSFN MODULE A
SAVE_RC = RC
SET CMSTYPE RT
IF SAVE_RC = 0 THEN
DO
  SAY ' — RXSYSFN MODULE found on disk A'
  SAY ' — Replace RXSYSFN MODULE A - reply Y or N'
PULL REPLY
  IF REPLY = 'Y' THEN
    SIGNAL ERROR
  END
END
SET CMSTYPE HT
SIGNAL ON ERROR
MESSAGE = 'error when assemble' RXSYSFN
ASSEMBLE RXSYSFN
ERASE RXSYSFN LISTING A
MESSAGE = 'error when load' RXSYSFN
LOAD RXSYSFN '(' NOMAP NOLIBE
MESSAGE = 'error when genmod' RXSYSFN
GENMOD
ERASE RXSYSFN TEXT A
SIGNAL OFF ERROR
SET CMSTYPE RT
SAY ' — RXSYSFN MODULE generated successfully'
EXIT
ERROR:
  SET CMSTYPE RT
  SAY ' — RXSYSFN MODULE not generated due to' MESSAGE
SAY 'Start ESAY checker - reply 1/Yes/
PULL ANS.
IF ANS ≠ '1' THEN
EXIT
SAY 'Select number of test - reply 1 or 2'
PULL TEST.
CLRSCRN
ESAY(11,26,'AI ready to communicate')
ESAY(23,1,'AI is activated identification subsystem')
UNKNOWN = ESAY(24,1,'— Enter your identification data', 8)
IF TEST ≠ 2 THEN
LIMIT = 33
ELSE
LIMIT = 99
DO I = 1 TO LIMIT
   IF TEST ≠ 2 THEN
      ESAY(Ø)
      J = RANDOM(1,24)
      K = RANDOM(1,8Ø)
      ESAY(J, K,'AI cannot communicate with 'SUBSTR(UNKNOWN,2))
      IF TEST ≠ 2 THEN
         SLEEP 1 SEC
   END
SLEEP 5 SEC
ESAY(Ø)

RXSYSFN ASSEMBLE

******************************************************************************
**** REXX say instruction extension *** DG'98
****
******************************************************************************
****   SIZE 00253 VER 1.0 MOD 000                                ****
******************************************************************************
RXSYSFN CSECT
USING *,12
LR  11,14
CLC  8(8,1),=CL8'LOAD'
BE  DONUCEXT

RET  EQU  *
LA   15,1
BR   11
DONUCEXT EQU  *
LA   3,EXTS
LA   4,12
LA   5,EXTE
L    0,TOALLOC
LR   10,0
DMSFREE DWORDS=(0),TYPE=NUCLEUS,ERR=RET
LR   0,1
LR   1,10
SLL  1,3
ST   1,NUCXLEN
LA   14,USERBGN
LR   15,1
LR   10,0
SPKA 0
MVCL 0,14
LR   0,10
B    SETXBLK
NEXTEXT EQU  *
XC   NUCXLEN(4),NUCXLEN
L    0,0(3)
AR   0,10
SETXBLK EQU  *
ST   0,NUCXADDR
ST   0,NUCXORG
MVC  NUCXNAME(8),4(3)
LA   1,NUCXDCL
SVC  2Ø2
DC   AL4(1)
LTR  15,15
BNZ  RET
BXLE 3,4,NEXTEXT
BR   11
EXTS DS 0F
USEROFF DC A(Ø)
USERNM DC CL8'RXSYSFN'
ESAYOFF DC A(ESAYBGN-USERBGN)
ESAYNM DC CL8'ESAY'
EXTE EQU  *-12
TOALLOC DC A((REALEND-USERBGN+7)/8)
NUCXDCL DS 0F
DC   CL8'NUCEXT'
NUCXNAME DS CL8
NUCXSM DC X'ØØ'
NUCXCMWP DC X'Ø4'
NUCXFLG DC X'8Ø'
USERFLG DC X'ØØ'
NUCXADDR DS A
USERWORD DC A(Ø)
NUCXORG DS A
NUCXLEN DS A
LTORG
USERBGN DS 0D
LA 15,1
BR 14
DROP 12
ESAYBGN DS 0D
USING *,12
LR 11,14
LR 10,0
USING EPLIST,10
L 1,ARGLIST
LM 2,9,0(1)
LA 1,CXWBUF
CH 3,=H'11'
BH ORDREQ
CLI 0(2),X'F0'
BNE ORDREQ
SR 7,7
SR 8,8
B SETPATH
ORDREQ EQU *
LTR 7,7
BNP ERR
MVC 0(8,1),=X'C3115D7F1311400'
CUTWLEN EQU *
CH 7,=H'40'
BNH OKWLEN
BCT 7,CUTWLEN
OKWLEN EQU *
LA 15,8(7)
BCTR 7,0
STC 7,MOVETXT+1
MOVETXT MVC 8(Ø,1),Ø(6)
LTR 9,9
BNP WRTONLY
BCTR 9,0
LR 6,7
LR 14,8
EX 9,PACK
CVB 9,WORKA
CUTRLEN EQU *
CH 9,=H'30'
BNH OKRLEN
BCT 9,CUTRLEN
OKRLEN EQU *
LA 7,0(1,15)
MVC 0(2,7),=X'1D40'
LA 15,2(9,15)
LA   14,0(1,15)
MVC  Ø(2,14),=X'1DEØ'
LA   15,2(15)
BCTR 9,0
LA   8,2(7)
MVI  Ø(8),C'_'
LTR 9,9
BZ   WRTEXT
BCTR 9,0
STC 9,MOVEPAD+1
MOVEPAD MVC 1(Ø,8),Ø(8)
B   WRTEXT
WRONLY EQU *
SR  8,8
WRTEXT EQU *
BCTR 3,0
LR  14,2
EX  3,PACK
CVB  Ø,WORKA
BCTR 0,0
MH  0,=H'8Ø'
BCTR 5,0
LR  14,4
EX  5,PACK
CVB 2,WORKA
AR  0,2
LTR 0,0
BM   LOWADDR
B   SETADDR
LOWADDR EQU *
SR  0,0
SETADDR EQU *
LTR 8,8
BE  ASIS
AR  6,0
LA   6,2(6)
STC 6,3(1)
SRL  6,6
STC 6,2(1)
NI  3(1),X'3F'
TR  2(2,1),ADDRTAB
ASIS  EQU *
STC 0,7(1)
SRL  0,6
STC 0,6(1)
NI  7(1),X'3F'
TR  6(2,1),ADDRTAB
LR  7,15
LR  6,1
SETPATH EQU *
LA   5,ESAYPATH
ESAY PREPARATION

The INSTALL EXEC should be used to generate the RXSYSFN function package. After the successful creation of the RXSYSFN MODULE, CHECKER EXEC may be started to verify the ESAY function.

The CHECKER EXEC contains two tests, which demonstrate the use of direct console I/O.

Dobrin Goranov  
Information Services Co (Bulgaria)  © Dobrin Goranov 1999
Mainframe Programming Web site

Continuing our series of VM Web site reviews, we visit the Mainframe Programming Web site. The site, subtitled ‘Some useful links for practitioners of the craft’, can be accessed at http://www.geocities.com/~oberoi/mainfrme.html. If you have comments on the Web sites reviewed in this series, or suggestions for relevant sites to review, please feel free to contact the author at gabe@acm.org or Xephon at any of the addresses shown on page 2.

If you watch closely as this Web page loads, you may see the words ‘Think enterprise, think mainframe’ appear at the top left of the screen, before a space station image replaces them. The words, a good motto for the site, are on an <ALT> HTML tag and display if your browser isn’t loading images, or (with some browsers) if you hover the cursor over the image. Resources gathered and linked here by Balbir Oberoi can be valuable to VMers in the current technical and industry environment, where VM is one component – maybe central, maybe less so than in the past – in a complex enterprise computing environment.

Of the nine links grouped next to the space station, eight jump down to sections within the main page. The first, ‘General resources’, is a blend of IBM and non-IBM pages. Its first link, ‘Redbooks’, leads to a page which opens “Redbooks, named for their red covers, are ‘how to’ books, written by very experienced IBM professionals from all over the world”. Redbooks, excellent supplements to more formal product manuals, are often written by joint customer/IBM teams working on ‘residencies’ – intense brief research projects devoted to gathering real-world information and publishing it in time to be useful. The redbook site links to its mission statement and description:

“IBM redbooks are developed and published by IBM’s International Technical Support Organization, the ITSO. With headquarters in Poughkeepsie (New York), the ITSO operates centres associated with key IBM development divisions in Austin, Poughkeepsie, Raleigh, Rochester, and San Jose...

The ITSO develops and delivers skills, technical know-how, and
materials to technical professionals of IBM, business partners, customers, and to the marketplace generally. ITSO’s objective is to extend the understanding of IBM’s products and to accelerate the deployment and exploitation of IBM solutions...

In 1997, ITSO introduced IBM redpieces, which are redbooks under development. The objective of redpieces is to speed up access to books that are not yet published. Consequently they are less polished and have not undergone the formal review that takes place for completed IBM redbooks.”

VM redbooks listed as being in production are *TCP/IP tutorial and technical overview*, *Exploiting recent CMS function: a user’s guide to CMS application multitasking*, and *VM/ESA network computing with Java and NetREXX*. The use of the built-in search frame retrieved many screens of VM-related redbooks. The left-hand frame offers more resources, such as the ‘Additional materials’ link, an FTP site with downloadable material from the publications.

The next ‘General resource’ link finds Internet access to IBMLink, a form much more convenient than the proprietary access through which it was introduced. The first screen offers a choice of language and country; selecting United States access reveals the ‘IBM public information and services main menu’, offering access to diverse resources without logging on (some IBMLink applications still require log-on, available via the toolbar at the top). Information available includes the following categories:

- 10thAnniversary – the IBMLink 10th anniversary celebration.
- NewUser – new user information – for example, how to get started.
- IBMLinkPreview – an introduction and overview of this service.
- InfoLink – sales manuals, announcements, catalogs, etc. For example:
  - iSource – subscribe to IBM announcement information (ListServer).
– SalesManual – IBM hardware and software product descriptions.
– PubsCatalog – a description of IBM publications.
– IBMManuals – on-line viewable IBM public manuals.
– Education – IBM course content and scheduling information.
– OnlineOffering – IBM on-line offerings (free Java introduction).
– InfoLinkCenter – white papers, publications, and contacts.
– INews – the IBM information news facility.
– IBM-conferences – IBM international conferences go on-line!

• OrderLink – configurators, prices, and schedules:
  – OrderInfo – IBM software ordering information.
  – Schedules – product shipment and availability schedules.

• MartLink – information services:
  – TalkLink – TalkLink conferencing services.
  – Education mall – education and training mall.

• DataCenter – data centre services.

For example, clicking ‘PubsCatalog’, then ‘Search words’, then entering the keywords ‘redbook’ and ‘CD-ROM’ locates (as the eighth entry listed) ‘SK2T-2177 CDMAN IBM System/390 redbooks collection September 1998’, a CD-ROM containing all System/390-related redbooks. This offers a simple and relatively economical way to purchase and store accumulated redbook wisdom.

Other IBMLink information categories can be very useful in day-to-day data centre operation, and for researching special projects and topics.

Just below, ‘WSC flashes – Washington Systems Center, performance issues’ provides access to a unique source of information. The page,
entitled ‘Washington Systems Center FLASH home page’, begins:

“ATS flashes are published by the Washington Systems Center to alert IBM customers and personnel of significant new technical developments and provide guidance on the installation, use, and management of IBM products.”

Searching for VM entries yielded two screens of topics, including:
- VM support of the System/390 G4 parallel enterprise server.
- RVA considerations for running VM/ESA and OS/390.

A few links below WSC, the Oberoi page addresses ‘Enterprise connection – various IBM conference call presentations material, in PDF format’. At the time of writing, a call held just four days before was posted on the topic ‘VM customers – creative solutions for your IT challenges!’. This interesting event featured presentations by Mike Conchatre of Ducks Unlimited, Chris Williams of the SAS Institute, and Gretchen Thiele of Princeton University, describing leading-edge mission-critical applications and services provided by VM/ESA.

A few links below, ‘Mainframe jobs – a healthy job market for mainframe skills (Computerworld, October 26)’ describes reassuring short and longer-term prospects for mainframers. It begins:

“When it comes to IT pros with mainframe skills, Renee M Schneider just can’t get enough. “If you lose one in this market, that’s one too many”, says the director of staffing for corporate information systems at Ameritech Corp. The Chicago-based telecommunications giant uses multiple systems platforms, but it relies on mainframes to run a billing system tracking more than 18 million customers. To help maintain the behemoths, Schneider hired at least 350 mainframe pros last year, with no slowdown in sight.”

And it ends by noting that 1 January 2000 will not diminish mainframe employment:

“In fact, there may be more mainframe-related work in the future than today. Some companies are turning to mainframes as database servers, Silver says. Besides, she adds, “We are increasing, globally, our reliance on information technology. Companies will grow and
consolidate, and someone will have to manage and merge their systems.”

The next link is to Eric Loriaux’s compendium of useful Web resources about MVS, VM, and VSE – a cornucopia well worth exploring. The last ‘General resource’ link is for entertainment – a wealth of variations on Murphy’s legendary law!

The next major category on the main Oberoi page is ‘Programming languages’, offering something to suit all tastes. The first link describes IBM High Level Assembler for MVS and VM and VSE Version 1.2, highlighting features and productivity benefits, and describing supported operating system versions.

The ‘HLASM Docs’ link offers a non-IBM compilation of Assembler-related links, including manuals and reference material.

The next three links offer a COBOL extravaganza:

- IBM COBOL family – covering VisualAge/COBOL, Millennium Language Extensions, etc.
- COBOL manuals – IBM bookshelves for COBOL.
- The COBOL centre – a well arranged set of pointers to COBOL-related information, including COBOL books.

The first COBOL link points to one of IBM’s main COBOL pages, which opens by describing their workstation development environment, and continues:

“IBM COBOL provides a complete offering of compatible, cross platform, cross product compilers which support OS/2, OS/390, MVS, VM, VSE/ESA, AS/400, AIX, and Windows. IBM gives you the tools you need to tackle your COBOL Year 2000 challenge while leveraging your existing applications. IBM COBOL also provides the tools you need to amplify your program development, enabling you to position your enterprise to take advantage of tomorrow’s technologies.”

This indicates that, contrary to some commentaries, COBOL is alive and well, and not just for Year 2000 efforts. The ‘Library’ link provides fact sheets, manuals, resource catalogs, white papers, performance tuning information, standards information, and many
items to warm the hearts of COBOL users. The second link provides – what else? – COBOL manuals, in Web-viewable format; very convenient when a quick question requires documentation not immediately available. The third link reaches an amazingly diverse set of COBOL resources – including a page with yet more links. The COBOL centre’s Webmaster is described:

“David Haertzen has personally coded over 750,000 lines of COBOL for platforms including: IBM mainframe, PCs, mini-computers, and servers. He has contributed to many successful data management projects and can supply services and seminars/training.”

The next language link introduces Language Environment (LE), described on its overview page:

“IBM’s Language Environment for OS/390 and VM provides a single run-time environment for C, C++, COBOL, Fortran, PL/I, and Assembler applications. Language Environment’s common library includes common services such as messages, date and time functions, math functions, application utilities, system services, and subsystem support. All of these services are available through a set of interfaces that are consistent across programming languages. You may either call these interfaces yourself, or use language-specific services that call the interfaces.”

The main LE page puts the software in Year 2000 context:

“Are you ready for the future? If not, Language Environment can help. Language Environment’s date and time services readily return 4-digit years and continue to process 2-digit years to accommodate the Year 2000. These date and time services conform to national language support guidelines, including full DBCS support.”

Near the bottom of the languages group is a link to ‘PL/I – home page of the PL/I product family’. As an erstwhile PL/I programmer, I’m gratified to read this page’s message:

“Whether you are maintaining legacy applications or eager to move to the workstation, IBM PL/I gives you:

• The elegance of a variety of language constructs, I/O capabilities, and selection of tools.
• *The flexibility to design applications on the platforms you use – MVS and VM, VSE, AIX, OS/2, and Windows.*

• *The power to step up to the challenge of leveraging your existing applications while preparing for the Year 2000.*

Other language links from Oberoi include:

• Java for OS/390 – porting of Java virtual machine to the mainframe, now available: JDK 1.1.4.

• SanFrancisco project – creating reusable components for business applications – concepts and facilities.

• Java discussions – forums about Java, all platforms (IBM Hursley).

• REXX – the REXX language page at IBM Hursley.

• NetREXX – a blend of REXX and Java to enable REXX programmers to create Java code.

The next category, ‘CICS, MQSeries,’ mostly doesn’t apply directly to VM systems, although of course numerous CICS systems, hosted by VSE and OS/390, run in virtual machines. However, the last link in this category, ‘MQSeries family – news, case studies, white papers, etc, about MQSeries’ lists VM/ESA as a supported system. The MQSeries pages are a little frustrating, being short of basic product information – if one doesn’t know what MQSeries is, it’s challenging to find a definition. There are, however, many resources for dedicated MQSeries users, including a pointer to IBM’s MQ magazine.

The next Oberoi category is DB2, now the brand name for IBM’s database systems spanning many computing platforms. Clicking the first link, ‘DB Management’, leads to the main DB2 page, where the pop-up menu under ‘Products’ leads to DB2 for VSE/VM, highlighting Version 5.1 and previewing Version 6, including the new QMF feature. The main DB2 page offers seminars, download material, magazines, and more.

The next category, ‘Operating systems’, stints a bit on VM links and perspective, compared with what it offers on OS/390, although it does link to IBM’s VM home page (reviewed in *VM Update*, Issue 139, March 1998). But from a positive viewpoint, this material is valuable.
to VMers, because it addresses topics and resources not necessarily in the VM mainstream. The last link in this category, ‘ESA/390 Bookshelf’, offers general System/390 hardware publications, in viewable format, that might not be immediately handy when needed. For example:

- Channel-to-channel adapter.
- Common I/O-device commands.
- ESA/390 data compression.
- ESA/390 principles of operation.
- ESCON channel-to-channel adapter.
- ESCON I/O interface.
- System/360 and System/370 I/O interface channel to control unit OEMI.
- Vector operations.

The preceding hardware link segues into the System/390 category, ranging from ‘IBM’s System/390 home page’ through ‘G5 parallel enterprise servers’, ‘Parallel sysplex’, and several articles on mainframes.

In the next category, ‘User groups’, the first link, ‘IBM user groups’, is most useful to VMers. This page lists eight major user groups, located around the world, from North America to Japan. These groups have evolved to satisfy local/regional technical and logistical requirements, while giving ‘strength in numbers’ to customers dealing with IBM. Two umbrella organizations unite these regional groups: The International User Group Council (IUGC) and COMMON International. In addition, a relatively recently organized group, WAVV, is represented at http://listserv.lehigh.edu/lists/vse-l/wavv_conference.shtml and a collection of local VM user groups can be seen at http://www.vmers.org:81/. Other local VM user groups around the world are hosted elsewhere.

The last category, ‘Training’, spans geographic locations, platforms, and applications. IBM training information is available for the United
States and elsewhere. Following the link leads again to IBMLink, where searching on VM classes yields 77 hits, including old standbys and new entries such as:

- TCP/IP for MVS/VM implementation.
- VM/ESA CP debugging techniques.
- VM/ESA CP structure and logic.
- Using VMSES/E for installation and service.
- Installing and tailoring your VM/ESA system.
- Learning REXX programming in CMS.
- VM/CMS fundamentals workshop.
- VM/ESA communications and connectivity.
- Productive programming with VM/CMS pipelines.
- DB2 server for VSE and VM application designer and programmer workshop.

The ‘On the campus’ link represents a relatively new IBM initiative, locating and listing colleges and universities offering classes on mainframe subjects in various countries. This effort clearly recognizes the importance of conveying mainframe awareness, skills, and perspectives to new generations of system and application programmers, and hardware/software developers and engineers. The page entitled ‘Higher education and System/390’ lists nearly 30 institutions, and solicits nominations for other sites for listing. It begins:

“Are you a student who wants to increase your System/390-related skills? Are you an employer who wants to hire a person with these skills? In either case, this page is for you!”

The next link, to ‘ProTech’ (Professional Technical Services), offers a relatively rare course: VM/ESA for computer operations staff. Their REXX curriculum is more ambitious, consisting of classes that include:

- REXX programming in a multi-platform environment.
• REXX in an MVS environment.
• REXX in an OS/2 environment.
• REXX in a VM environment.
• Advanced REXX programming for MVS.
• Advanced REXX programming for OS/2.
• Advanced REXX programming for VM.
• Differences in OS/2 REXX.
• Introduction to Watcom’s VX-REXX.
• Hockware’s VisPro/REXX.
• Advanced REXX programming.

In addition, ProTech offers a mailing list for discussing REXX and automated operations issues. The last link in this category is to technical conferences organized or co-sponsored by IBM. The VM/ESA & VSE/ESA Technical Conference is one of only two events being offered twice in 1999, in Orlando, Florida (May 24-27), and Mainz, Germany (June 14-16).

At the very bottom of the page is an almost invisible link unrelated to VM, or even directly to computers, labelled ‘World newspapers’. This provides jumps to newspapers in 12 countries, from Australia to USA. No matter what your location or interests, this page enables you to find international perspectives on technology, business, sports, politics, and more.

Continuing serendipitous discovery, don’t miss the ‘Favourite links’ link at the bottom of the newspapers page, nor the ‘Languages’ link, providing diverse resources for multiple languages. So this page, having delivered a wealth of mainframe resources, ends with neither bang nor whimper, neither VM nor MVS, but with interesting digressions from technology.

Gabe Goldberg
Computers and Publishing (USA) © Xephon 1999
A full screen console interface – part 6

Editor’s note: this month we continue the code for the full screen console interface for Disconnected Service Machines (DSM). This article is an extensive piece of work which will be published over several issues of VM Update. It was felt that readers could benefit from the entire article and from the individual sections. Any comments or recommendations would be welcomed and should be addressed either to Xephon or directly to the author at fernando_duarte@vnet.ibm.com.

(I,00,03,'PFX',',PREFIX),
(I,00,02,'REMOTE',',REMOTE),
(I,00,01,'ROUTE',',ROUTE),
(I,00,03,'RTE',',ROUTE),
(I,00,01,'TITLE',',TITLE),
(I,00,03,'TTL',',TITLE),
(I,00,01,'USER',',USER),
(I,00,03,'USR',',USER)

SPACE

EVNTABLE CMMD (I,00,01,'NAME',',EVNNAME), Event options
(I,00,01,'COMMAND',',EVNCMD),
(I,00,03,'CMD',',EVNCMD),
(I,00,01,'DATE',',EVNDATE),
(I,00,01,'FROM',',EVNFROM),
(I,00,01,'TO',',EVNTO),
(I,00,02,'TIME',',EVNTIME),
(I,00,02,'FIRST',',EVNFIRST),
(I,00,01,'LAST',',EVNLAST),
(I,00,01,'INTERVAL',',EVNINT),
(I,00,03,'DAYS',',EVNDAYS)

SPACE

LCLTABLE CMMD (B,00,01,'LOCAL',',RNDOLRS), Local options
(B,00,01,'GLOBAL',')

SPACE

MSGTABLE CMMD (I,00,01,'USER',',MSGU), MSG options
(B,00,01,'ALARM',',MSGALARM),
(I,00,01,'EXIT',',MSGE),
(B,00,01,'HOLD',',MSGHOLD),
(I,00,01,'NAME',',MSGN),
(B,00,03,'NOCASE',',MSGCASE),
(B,00,03,'NODISPLAY',',MSGNODISP),
(I,00,02,'RELEASE',',MSGR),
(I,00,02,'ROUTE',',MSGS),
(B,00,02,'UNIQUE',',MSGUNIQ+MSGHOLD),
(I,00,01,'LOCATE',',MSGL)
SPACE
EDSTABLE CMMD (B,ØØ,Ø2,'HIGH ',EDSHIGH), EDS attributes *
  (B,ØØ,Ø3,'BLINK ',EDSBLINK), *
  (B,ØØ,Ø2,'REVVIDEO ',EDSREVV), *
  (B,ØØ,Ø2,'UNDERLINE ',EDSUNDER), *
  (B,ØØ,Ø1,'BLUE ',EDSBLUE), *
  (B,ØØ,Ø3,'RED ',EDSRED), *
  (B,ØØ,Ø1,'PINK ',EDSPINK), *
  (B,ØØ,Ø1,'GREEN ',EDSGREEN), *
  (B,ØØ,Ø1,'TURQUOISE ',EDSTURQ), *
  (B,ØØ,Ø1,'YELLOW ',EDSYELLO), *
  (B,ØØ,Ø1,'WHITE ',EDSWHITE)
SPACE
EDSHIGH EQU B'00100000'
EDSBLINK EQU B'00001000'
EDSREVV EQU B'00010000'
EDSUNDER EQU B'00011000'
EDSBLUE EQU B'00000001'
EDSRED EQU B'00000010'
EDSPINK EQU B'00000011'
EDSGREEN EQU B'00000100'
EDSTURQ EQU B'00000101'
EDSYELLO EQU B'00000110'
EDSWHITE EQU B'00000111'
SPACE
OPTTABLE CMMD (I,ØØ,Ø3,'MSG        ',OPTM), OPTions options *
  (I,ØØ,Ø6,'MSGNOH     ',OPTM), *
  (I,Ø0,Ø5,'PRINT      ',OPTP)
SPACE
PFXTABLE CMMD (B,ØØ,Ø1,'CLASS      ',*) PFX options
SPACE
USRTABLE CMMD (I,ØØ,Ø1,CLASSES,Ø) User options
SPACE 3
*
*   Process Configuration commands
*
*
SPACE 3
*   DFReecs
*   DFSize n
*
DFRECS EQU * DFRECS Define DF file size
USING DFRECS,R9
ST R14,CFGCSV14
SR R0,R0 No table to look up
GO CSCSCN Get value
BNZ DFRE500 Nothing found
GO CSCSCNVN Check for numeric
BNZ DFRE600 No good, forget it
SR R0,R0  No table to look up
GO CSCSCN  Check for something else
BZ DFRE700  Bad news, too much
LR R1,R2  Copy number of records
LA R1,31(R1)  Adjust to full 4K blocks
SRL R1,5  That's 32 records per block
SLL R1,5
C R1,DFSZMIN  Expand if smaller than minimum
BNL DFRE100
L R2,DFSZDFLT  Use default if less than minimum
ST R2,DFNEWTOT
LR R4,R1  Copy value for message
MSG ØØ60  Let somebody know about it
B DFRE900
SPACE
DFRE100 ST R1,DFNEWTOT  Store new value
CR R1,R2  Was it changed?
BE DFRE900
LR R4,R1  Yes, copy for MSG
MSG ØØ61  Display message
B DFRE900
SPACE
DFRE500 MSG ØØ50  Missing operand
B DFRE900
SPACE
DFRE600 MSG ØØ51  Invalid operand
B DFRE900
SPACE
DFRE700 MSG ØØ52  Unexpected operand
B DFRE900
SPACE
DFRE900 L R14,CFGCSV14
BR R14
DROP R9
SPACE
*  Event  Name  name  Date  date  From  from  To  to  Time  time
*  First  first  Last  last  Interval  interval  DAYS  days
*  Command  command
*
EVENT EQU *  EVENT Define Time Based Events
USING EVENT,R9
USING TMRSECT,R5
USING CMDSECT,R2  CMS Commands Table
ST R14,CFGCSV14
LA R0,TMRSIZE  Entry size in double words
LINK OBTAIN  Allocate entry
LR R5,R1  Address new entry
SR R0,R0  Zero option bytes
ST R0,TMROPT1
EVENSCAN LA R0,EVNTABLE
GO CSCSCN Scan next option
BNZ EVENGOOD All done, validate entry
LTR R15,R15
BZ EVEN100 Invalid option, display message
MVC CSCSCN,CMDCNAME Save option name for messages
LR R4,R15 Save processing routine address
SR R0,R0 No table to search
GO CSCSCN Scan option value
BNZ EVEN200 Not found, display message
LR R15,R4 Restore address of routine
GO , Process option
B EVENSCAN Scan all options
SPACE
EVENGOOD LR R1,R5 Copy entry address for CSCTMRVL
GO CSCTMR Validate entry
BZ EVEN900 Good news...
LR R5,R1 Restore entry address to release
EVENBAD LA R0,TMRSIZE Something went wrong
LR R1,R5 Address entry
LINK RELEASE Release it
B EVEN900
SPACE
EVEN100 MSG 0051 Invalid operand
B EVENBAD
SPACE
EVEN200 MSG 0070 Missing option value
B EVENBAD
SPACE
EVEN300 MSG 0053 Operand too long
B EVENBAD
SPACE
EVEN400 MSG 0071 Invalid option value
B EVENBAD
SPACE
EVEN500 MSG 0072 Duplicate option
B EVENSCAN
SPACE 3
EVNNAME EQU * NAME option
LA R0,L'TMRNAME Option length
CR R0,R1 Compare with value
BL EVEN300 Too long
MVC TMRNAME,SCANUPP Move value
TM TMROPT1,TMRNAME Check for first time
BO EVEN500 No, warn duplicate option
OI TMROPT1,TMRNAME Yes, set option
B EVENSCAN
SPACE
EVNCMD EQU * COMMAND option
LA  R0,L’TMRCMND  Option length
L  R1,CSCBUFFE  Address end of data

EVNC100  BCTR  R1,0  Command too long
CLI  0(R1),C’ ’  Remove trailing blanks
BE  EVNC100  Move value
SR  R1,R6  Length of command - 1
CR  R0,R1
BH  EVEN300  Store it
EX  R1,EVNCMVC  Must be last option
LA  R1,1(R1)  Get real command length
STH  R1,TMRCMNDL  Set option
B  EVENGOOD
SPACE

EVNCMVC  MVC  TMRCMND(*-*),0(R6)  Move command
SPACE

EVNDATE  EQU  *  DATE option
LA  R4,TMRDATE  Address field
TM  TMROPT2,TMRODATE  Check for first time
BO  EVNDDUP  No. warn duplicate option
OI  TMROPT2,TMRODATE  Yes. set option
B  EVNDVAL  Validate Date
SPACE

EVNFROM  EQU  *  FROM option
LA  R4,TMRFROM  Address field
TM  TMROPT2,TMROFROM  Check for first time
BO  EVNDDUP  No. warn duplicate option
OI  TMROPT2,TMROFROM  Yes. set option
B  EVNDVAL  Validate Date
SPACE

EVNTO  EQU  *  TO option
LA  R4,TMRTO  Address field
TM  TMROPT2,TMROTO  Check for first time
BO  EVNDDUP  No. warn duplicate option
OI  TMROPT2,TMROTO  Yes. set option
B  EVNDVAL  Validate Date
SPACE

EVNTIME  EQU  *  TIME option
LA  R4,TMRTIME  Address field
TM  TMROPT2,TMROTIME  Check for first time
BO  EVNTDUP  No. warn duplicate option
OI  TMROPT2,TMROTIME  Yes. set option
B  EVNTVAL  Validate Time
SPACE

EVNFIRST  EQU  *  FIRST option
LA  R4,TMRFIRST  Address field
TM  TMROPT2,TMROFRST  Check for first time
BO  EVNTDUP  No. warn duplicate option
OI  TMROPT2,TMROFRST  Yes. set option
B     EVNTVAL                Validate Time
SPACE
EVNLAST EQU *                LAST option
LA    R4,TMR LAST              Address field
TM    TMROPT2,TMR LAST         Check for first time
BO    EVNT DUP                 No, warn duplicate option
OI    TMROPT2,TMR LAST         Yes, set option
B     EVNTVAL                Validate Time
SPACE
EVNINT EQU *                INTERVAL option
LA    R4,TMR INT               Address field
TM    TMROPT2,TMR INT          Check for first time
BO    EVNT DUP                 No, warn duplicate option
OI    TMROPT2,TMR INT          Yes, set option
B     EVNTVAL                Validate Time
SPACE
EVNDAYS EQU *                DAYS option
SPACE 3
EVNDUP EQU *                Date physical validation
LR    R4,R1                   Save word length
MSG   ØØ72                    Duplicate option
LR    R1,R4                   Restore length
EVNDVAL EQU *                Normal entry
LA    RØ,4                    Maximum length for year
CR    RØ,R1                   Compare with value
BNE   EVEN4ØØ                 Invalid year
LA    R2,Ø(R6,R1)             Address first byte after year
CLI   Ø(R2),C'/'              It must be a "/", no spaces
BNE   EVEN4ØØ                 It is not, display error message
GO    CSCSCNVN                 Make sure it is numeric
BNZ   EVEN4ØØ                 It is not, bad news
STH   R2,Ø(R4)                Save year
SR    RØ,RØ                   No table to search
GO    CSCSCN                   Scan next "/"
CLI   1(R6),C' '              No spaces allowed
BE    EVEN4ØØ                 Display error message
L     RØ,ONE                  Force length of one. That's "/"
ST    RØ,SCANLEN               Store for CSCSCN
SR    RØ,RØ                   No table to search
GO    CSCSCN                   Get value for month
BNZ   EVEN4ØØ                 It is not there, bad news
LA    RØ,2                    Maximum length for month
CR    RØ,R1                   Compare with value
BL    EVEN4ØØ                 Invalid month
LA    R2,Ø(R6,R1)             Address first byte after month
CLI   Ø(R2),C'/'              It must be a "/", no spaces
BNE   EVEN4ØØ                 It is not, display error message
GO    CSCSCNVN                 Make sure it is numeric
BNZ   EVEN4ØØ                 It is not, bad news
STC R2,2(,R4)  Save month
SR R0,R0  No table to search
GO CSCSCN  Scan next "/
CLI 1(R6),C' '  No spaces allowed
BE EVEN400  Display error message
L R0,ONE  Force length of one. That's "/"
ST R0,SCANLEN  Store for CSCSCN
SR R0,R0  No table to search
GO CSCSCN  Get value for days
BNZ EVEN400  It is not there, bad news
LA R0,2  Maximum length for days
CR R0,R1  Compare with value
BL EVEN400  Invalid days
GO CSCSCNVN  Make sure it is numeric
BNZ EVEN400  It is not, bad news
STC R2,3(,R4)  Save days
B EVENSCAN

SPACE 3

* EVNDUP EQU *  Time physical validation
LR R4,R1  Save word length
MSG ØØ72  Duplicate option
LR R1,R4  Restore length

* EVNVAL EQU *  Normal entry
LA R0,4  Maximum length for hours
CR R0,R1  Compare with value
BL EVEN400  Invalid hours
LA R2,0(R6,R1)  Address first byte after hours
CLI Ø(R2),C':'  It must be a ":", no spaces
BNE EVEN400  It is not, display error message
GO CSCSCNVN  Make sure it is numeric
BNZ EVEN400  It is not, bad news

* CVD R2,CFGWORK  Convert to decimal
* OI CFGWORK+L'CFGWORK'-1,X'ØF' Remove sign
* UNPK Ø(2,R4),CFGWORK  Add leading zeros if required
STH R2,0(,R4)  Save hours
SR R0,R0  No table to search
GO CSCSCN  Scan next ":"
CLI 1(R6),C' '  No spaces allowed
BE EVEN400  Display error message
L R0,ONE  Force length of one. That's ":"
ST R0,SCANLEN  Store for CSCSCN
SR R0,R0  No table to search
GO CSCSCN  Get value for minutes
BNZ EVEN400  It is not there, bad news
LA R0,2  Maximum length for minutes
CR R0,R1  Compare with value
BL EVEN400  Invalid month
LA R2,0(R6,R1)  Address first byte after minutes
CLI Ø(R2),C':'  It must be a ":", no spaces

BNE EVEN4ØØ   It is not, display error message
GO CSCSCN VN   Make sure it is numeric
BNZ EVEN4ØØ   It is not, bad news
*CVD R2,CFGWORK   Convert to decimal
*OI CFGWORK+L'CFGWORK-1,X'OF' Remove sign
*UNPK 2(2,R4),CFGWORK   Add leading zeros if required
STC R2,2(,R4)   Save minutes
SR R0,R0   No table to search
GO CSCSCN   Scan next ":"
CLI 1(R6),C' '   No spaces allowed
BE EVEN4ØØ   Display error message
L R0,ONE   Force length of one. That's "":"
ST R0,SCANLEN   Store for CSCSCN
SR R0,R0   No table to search
GO CSCSCN   Get value for seconds
BNZ EVEN4ØØ   It is not there, bad news
LA R0,2   Maximum length for seconds
CR R0,R1   Compare with value
BL EVEN4ØØ   Invalid days
GO CSCSCN VN   Make sure it is numeric
BNZ EVEN4ØØ   It is not, bad news
*CVD R2,CFGWORK   Convert to decimal
*OI CFGWORK+L'CFGWORK-1,X'OF' Remove sign
*UNPK 4(2,R4),CFGWORK   Add leading zeros if required
STC R2,3(,R4)   Save seconds
B EVENSCAN

EVEN9ØØ   L R14,CFGCSV14
BR R14
DROP R9,R5,R2
SPACE 3
* Local node resource Local Global
*
LOCAL EQU *   LOCAL Define Local Resource name
USING LOCAL,R9
USING RNDSECT,R1
ST R14,CFGCSV14
TM CFGOPTS,CFGLOCAL   Is it the first one
BO LOC4ØØ   No, too bad
BAS R14,ADDRSRCE   Add entry to RND Table
BNZ LOC9ØØ   Something went wrong
LA R0,LCLTABLE   Address table to search
GO CSCSCN   Do it
BNZ LOC2ØØ   Nothing found, that's OK
LTR R15,R15   Is it a valid option
BNZ LOC1ØØ
SR R6,R1   No, back-up last word
B LOC2ØØ   Validate will detect it
SPACE
LOC1ØØ   LA R0,RNDOLRS   Check for Local option
CR R0,R15
BNE LOC200 It is Global...
L R1,RNDPTR Address new (first) entry
OI RNDOPT1,RNDOLRS Set Local resource option
LOC200 BAS R14,VALSRCE Validate entry created
BNZ LOC900 Something went wrong
OI RNDOPT1,RNDOLCL Remember this a local entry
OI CFGOPTS,CFGLOCAL Set option
MVC CSCLOCAL,RNDNODE Copy local APPC/VM Node name
B LOC900

SPACE

LOC400 MSG ØØ8Ø Repeated Local statement
*
B LOC900

SPACE

LOC900 L R14,CFGCSV14
BR R14
DROP R9,R1

SPACE 3

* MSG
* Message Attributes Hold NoDisplay NoCase Unique Alarm
* User user-id Name name Release name Route user-id
* Exit name Locate mask
*

MSG EQU * MSG Build Message Table
USING MSG,R9
ST R14,CFGCSV14
USING MSGSECT,R5 MSG Message Table
USING CMDSECT,R2 CMS Commands Table
LA R0,MSGSIZE Entry size in double words
LINK OBTAINT Allocate entry
LR R5,R1 Address new entry
XC MSGSECT(MSGSIZEB),MSGSECT Clear to zeros
MVI MSGARBCH,CFGARBCH Move default arbitrary character
MVI MSGANYCH,CFGANYCH Move also default any character
MSGSCAN LA R0,MSGTABLE Address options table
GO CSCSCN Scan next option
BNZ MSGGOOD All done, validate entry
LTR R15,R15
BZ MSG100 Not found, check EDS attributes
CLI CMDTYPE,C'B' Is it a "B" type entry
BNE 2Ø,R15 No, execute routine
IC R0,MSGOPTS Yes, load option byte
OR R0,R15 Add new option
STC R0,MSGOPTS Store it back
B MSGSCAN Scan all options and attributes

SPACE

MSG100 LA R0,EDSTABLE Address attributes table
GO CSCSCNSC Check EDS attributes
LTR R15,R15
BZ MSG200 Invalid option, display message
IC RØ,MSGATTR
OR RØ,R15
STC RØ,MSGATTR
B MSGSCAN

MESSAGE

MSGGOOD CLI MSGUSER,X'ØØ'
BE MSG3ØØ
L RØ,MSGMASKE
LTR RØ,RØ
BZ MSG4ØØ
L R1,MSGPTR
ST R5,MSGPTR
ST R1,MSGFWD
B MSG9ØØ

SPACE

MSGBAD LA RØ,MSGSIZE
LR R1,R5
LINK RELEASE
B MSG9ØØ

SPACE

MSG2ØØ MSG ØØ51
B MSGBAD

SPACE

MSG3ØØ MSG ØØ9Ø
B MSGBAD

SPACE

MSG4ØØ MSG ØØ91
B MSGBAD

SPACE 3

MSGU EQU *
LA R4,MSGUSER
B MSGMOVE

MSGS EQU *
OI MSGOPTS,MSGORTE
LA R4,MSGROUTE
B MSGMOVE

SPACE

MSGN EQU *
LA R4,MSGNAME
B MSGMOVE

SPACE

MSGR EQU *
OI MSGOPTS,MSGORLS
LA R4,MSGRLSE
B MSGMOVE

SPACE

MSGE EQU *
OI MSGOPTS,MSGOEXT
LA R4,MSGEXIT
B MSGMOVE

*
SPACE

MSGMOVE SR R0,R0 No table to search
MVC CSCCOMM,CMDNAME Save option name for messages
GO CSCSCN Scan data
BNZ MSGV100 Not found
LA R0,8 Check length
CR R0,R1
BL MSGV200 Greater than 8, invalid
MVC @(R0,R4),SCANUPP Move data into required field
B MSGSCAN

SPACE

MSGV100 MSG Ø92 Missing user value
B MSGBAD

MSGV200 MSG Ø93 Value greater than 8 bytes
B MSGBAD

SPACE

MSGL EQU * LOCATE option. Must be LAST
MVC CSCCOMM,CMDNAME Save option name for messages
LA R6,1(R1,R6) Mask starts one byte after key
C R6,CSCBUFFE Anything left
BH MSGL500 No, it is missing
L R1,CSCBUFFE Address end of mask
SR R1,R6 Calculate length
LA R0,L'MSGMASK Maximum accepted
CR R0,R1
BL MSGL600 Too long
BCTR R1,0 Prepare to EXecute
EX R1,MSGLMVC Move mask
TM MSGOPTS,MSGCASE NoCase option?
BZ MSGL100
EX R1,MSGLTR Yes, translate mask to uppercase
MSGL100 LA R0,MSGMASK+1(R1) End address
ST R0,MSGMASKE Store
B MSGGOOD

SPACE

MSGL500 MSG Ø92 Missing Locate value
B MSGBAD

SPACE

MSGL600 MSG Ø94 Locate Mask too long
B MSGBAD

SPACE

MSGLMVC MVC MSGMASK(*-),O(R6) Move mask
MSGLTR TR MSGMASK(*-),CSCUPP Translate to uppercase
DROP R5,R2

SPACE

MSG900 L R14,CFGCSV14
BR R14
DROP R9

SPACE 3
* Options  MSG  MSGNOH
*
OPTIONS EQU * OPTIONS Define Global processing options
USING OPTIONS, R9
ST R14, CFGCSV14
LA R0, OPTTABLE Address options table
GO CSCSCN Scan option
BNZ OPT500 Nothing found, need at least one
OPT100 LTR R15, R15 Check for valid option
BZ OPT600 Not valid
GO , Execute processing routine
OPT200 LA R0, OPTTABLE Address options table again
GO CSCSCN Scan next option
BZ OPT100 Found it, process it
B OPT900 Nothing left, all done
SPACE
OPTM EQU * MSG MSGNOH options
MVC CSCMSGC, SCANUPP Move it
BR R14
SPACE
OPTP EQU * PRINT option
OI CSCFLG01, MSGPRINT Set Print option
BR R14
SPACE
OPT500 MSG Ø50 Missing operand
B OPT900
SPACE
OPT600 MSG Ø54 Invalid operand
B OPT200 Check all operands
SPACE
OPT900 L R14, CFGCSV14
BR R14
DROP R9
SPACE 3
*
* Prefix  k  User-id  Class nn  Attributes
*
PREFIX EQU * PREFIX Build Prefix Table
USING PREFIX, R9
USING PFXSECT, R1 PFX Prefix Table
ST R14, CFGCSV14
SR R0, R0 No table to look up
GO CSCSCN Get prefix
BNZ PREF400 Not there
LA R0, 1 Check prefix length
CR R0, R1
BNE PREF500 Not one, invalid
MVC CFGPREF, SCANUPP Save prefix for now
SR R0, R0
GO CSCSCN Get User-id
BNZ  PREF600  Not there, display message  
LA  R0,8  Maximum length for user-id  
CR  R0,R1  
BL  PREF610  Too long  
MVC  CFGUSER,SCANUPP  Save user-id  
MVI  CFGATTR,X'00'  Clear attribute byte  
MVI  CFGCLASS,X'00'  Clear class byte  
PREF100  LA  R0,PFXTABLE  Address options table  
GO  CSCSCN  Search for class  
BNZ  PREF300  Nothing, done  
LTR  R15,R15  Only class for now  
BZ  PREF200  No good, try EDS attributes  
SR  R0,R0  
GO  CSCSCN  Get class number  
BNZ  PREF700  Class value is missing  
GO  CSCSCNVN  Make sure it is numeric  
BNZ  PREF710  Non-numeric class  
LA  R0,PFXCLMIN  Compare with minimum  
CR  R0,R2  
BH  PREF720  Out of valid range  
LA  R0,PFXCLMAX  Compare with maximum  
CR  R0,R2  
BL  PREF720  
LA  R0,1  Set last bit of R0  
SR  R1,R1  Clear R1  
SRDL  R0,Ø(R2)  Set correct bit of R1  
STC  R1,CFGCLASS  Save class for now  
B  PREF100  Keep on scanning statement  

PREF200  LA  R0,EDSTABLE  Address table to search  
GO  CSCSCNSC  
LTR  R15,R15  Is it valid?  
BZ  PREF800  No, display error message  
IC  R0,CFGATTR  Get attribute byte  
OR  R0,R15  Merge new attribute  
STC  R0,CFGATTR  Store it back  
B  PREF100  Check all attributes  

PREF300  LA  R0,PFXSIZE  Prefix table entry length  
LINK  OBTAIN  Allocate storage  
L  R2,PFXPTR  Add it to list  
ST  R1,PFXPTR  Make it first entry  
ST  R2,PFXFWD  Chain it with previous first  
MVC  PFXPREF,CFGPREF  Move Prefix  
MVC  PFXATTR,CFGATTR  Move Attribute  
MVC  PFXCLASS,CFGCLASS  Move Class  
MVC  PFXUSER,CFGUSER  Move User-id  
B  PREF900  Done  

PREF400 MSG 0050 Missing prefix
B PREF900
SPACE
PREF500 MVC 2(L‘DOTS,R6),DOTS Prefix not one byte long
MSG 0100
B PREF900
SPACE
PREF600 LA R6,CFGPREF Missing user-id
MSG 0101
B PREF900
SPACE
PREF610 MSG 0053 User-id is too long
B PREF900
SPACE
PREF700 MSG 0102 Class value is missing
B PREF900
SPACE
PREF710 MSG 0051 Non-numeric class
B PREF900
SPACE
PREF720 MSG 0103 Class is out of valid range
B PREF900
SPACE
PREF800 L R2,SCANLEN Invalid attribute
MSG 0051
* B PREF900
SPACE
PREF900 L R14,CFGCSV14 REMOTE Define Remote Resource name
BR R14
DROP R9,R1
SPACE 3
* Remote node resource
*
REMOTE EQU * USING REMOTE,R9
ST R14,CFGCSV14
BAS R14,ADDRSROCE Add entry to RND Table
BNZ REM900 Something went wrong
BAS R14,VALRSRCE Validate entry created
BNZ REM900 Something went wrong
LR R2,R1 Save RND entry address
LA R0,RNDBUF SZ Length of Send/Receive buffers
LINK OBTAINP Allocate Send buffer
ST R1,RNDSBUFF-RNDSECT(.R2) Save Send buffer address
LA R0,RNDBUF SZ
LINK OBTAINP Allocate Receive buffer
ST R1,RNDRBUFF-RNDSECT(.R2) Save Receive buffer address
OI CFGOPTS,CFGMRTE Set option
* B REM900

© 1999. Xephyx UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
SPACE

REM9ØØ   L     R14,CFGCSV14
BR    R14
DROP  R9
SPACE 3

* RTE
* Route name node1 user1 node2 user2...
*
ROUTE   EQU   *
USING ROUTE,R9
USING RTESECT,R5
ST    R14,CFGCSV14
SR    RØ,RØ                  No table to look up
GO    CSCSCN                  Get Route name
BNZ   ROUT6ØØ                 Nothing found, display error
LA    RØ,8                    Check user length
CR    RØ,R1                   No table to look up
BL    ROUT9ØØ                 Too long, display error message
MVC   CFGNAME,SCANUPP         Save name for now
SR    R4,R4                   Counter for Node / User pairs
ROUT1ØØ  SR    RØ,RØ
GO    CSCSCCN                  Get Node name
BNZ   ROUT9ØØ                 Not found
LA    RØ,8                    Maximum node length is 8 bytes
CR    RØ,R1                   No table to look up
BL    ROUT9ØØ                 Too long, bad news
MVC   CFGNODE,SCANUPP         Save Node name
SR    RØ,RØ
GO    CSCSCCN                  Get User name
BNZ   ROUT9ØØ                 Not found, node without user
LA    RØ,8                    Check user length
CR    RØ,R1                   No table to look up
BL    ROUT9ØØ                 Too long, display error message
MVC   CFGUSER,SCANUPP         Save user name
LA    R4,1(,R4)               Increment counter
L     R1,RTEPTR               Search Route table
ROUT2ØØ  LTR   R5,R1
BZ    ROUT3ØØ                 End of table, allocate new entry
L     R1,RTEFWD               Address next entry
CLC   RTENAME,CFGNAME         Check name
BNE   ROUT2ØØ                 No good, check next entry
CLI   RTECNT,RTEMAX           Is entry full?
BNL   ROUT2ØØ                 Yes, check another one
B     ROUT4ØØ                 Add Node / User to this entry
SPACE

ROUT3ØØ  LA    RØ,RTESIZE     Entry length in double words
LINK   OBTAIN                Allocate entry
LR    R5,R1                  Address new entry
MVI   RTECNT,X'ØØ'           Current number of Node / User
MVC   RTENAME,CFGNAME        Copy Route name

L R1,RTEPTR Add new allocated entry to list
ST R5,RTEPTR Make it the first
ST R1,RTEFWD Chain with old first

ROUT4ØØ SR R1,R1 Required by next IC
IC R1,RTECNT Number of current Node / Users
LA R0,1(R1) Increment
STC R0,RTECNT Store new value
SLL R1,4 Calculate offset (16 bytes each)
LA R1,RTENODE(R1) Address Node / User pair
MVC L'RTENODE(R1),CFGNODE Copy Node and User names
MVC L'RTENODE(L'RTEUSER,R1),CFGUSER
B ROUT1ØØ

ROUTGOOD LTR R4,R4 Anything found?
BN! ROUT9ØØ Yes, everything is fine

ROUT6ØØ MSG ØØ5Ø No, missing operand
B ROUT9ØØ

ROUT7ØØ MSG ØØ53 Operand too long
B ROUT9ØØ

ROUT8ØØ MSG Ø11Ø Node without corresponding User
* B ROUT9ØØ

ROUT9ØØ L R14,CFGCSV14 BR R14 DROP R9,R5

TITLE EQU * TITLE Build Title line
US! TITLE,R9 USING TITLE,R9
ST R14,CFGCSV14
SR R0,R0 No table to search
GO CSCSCN Address first non blank
BNZ TITLØØØ Nothing found, clear title
L R1,CSCBUFFE Address end of data (title)

TITLØØ BCTR R1,0 Remove trailing blanks
CLI Ø(R1),C' ' Unnecessary test, anything left?
BE TITLØØ
CR R6,R1 No, should never happen
BH TITL8ØØ Length of title - 1
SR R1,R6
LA R0,L'CFGTTL-1 Compare with maximum
CR R0,R1
BNL TITL2ØØ It is valid
MSG Ø12Ø It is too long, display warning
LA R1,L'CFGTTL-1 Trim title
TITL200  EX   R1,TITLMVC       Move title to work area
LA    R1,1(,R1)               Get real length
ST    R1,SCRTTL                Store it for CSCBLD
LA    R0,CFGTTTL              Address title work area
ST    R0,SCRTTL                Store it
B     TITL900
SPACE
TITL800  SR   R0,R0                 Nothing found on Title statement
ST    R0,SCRTTL                Zero title length field
*   B     TITL900
SPACE
TITL900  L   R14,CFGCSV14        MVC   CFGTTL(*-*),Ø(R6)       Move title to work area
                                 SPACE 3
                                 *   USR      User-id   Classes n1 n2 n3 n4 n5 n6...
                                 *   User
                                 *
USER    EQU   *                  USER Build User Table
USING USER,R9
USING USRSECT,R1
ST    R14,CFGCSV14
SR    R0,R0                  No table to look up
GO    CSCSCN                 Get user-id
BNZ   USER500               Not found, bad news
LA    R0,8                   Check length, maximum is 8
CR    R0,R1                  If it is valid
BL    USER600                Too long
MVC   CFGUSER,SCANUPP        Save user-id
LA    R0,USRTABLE           Address User Options table
SR    R4,R4                   Clear work register for classes
GO    CSCSCN                Get user-id
BNZ   USER300               Nothing, default is class ØØ
LTR   R15,R15                Is it valid
BZ    USER650                No, display error message
USER100  SR   R0,R0               No more tables to search
GO    CSCSCN                Get next class
BNZ   USER300               Nothing, all done
GO    CSCSCNVN            Check if numeric
BZ    USER200               It is not, check if a single "**"
LA    R0,1                  Start with length one
CR    R0,R1                No, display error message
BNZ   USER700               Length is good, now the contents
CLI   SCANUPP,C'**'        Length is good, now the contents
BNE   USER700               Forget it, it is bad
X    R4,FFFFFFFF            Reverse all classes
B     USER100              Keep validating everything
SPACE
USER2ØØ  LA  RØ,32          Check if in the range Ø1-32
   CR  RØ,R2
   BL  USER8ØØ            No, too high
   LTR R2,R2
   BNP USER8ØØ            Too low
   LA  RØ,1               Set last bit from RØ
   SR  R1,R1              Clear R1
   SRDL RØ,Ø(R2)          Now shift to class position
   XR  R4,R1              Set or reset class
   B   USER1ØØ            Get next class

SPACE

USER3ØØ  LA  RØ,USRSIZE       USR entry length in double words
   LINK OBTAIN            Allocate storage
   L    R2,USRPTR        Get current USR pointer
   ST   R1,USRPTR        Store new value
   ST   R2,USRFWD        Chain new entry
   ST   R4,USRCLASS      Store classes
   MVC USRNAME,CFGUSER   Move user id
   B   USER9ØØ

SPACE

USER5ØØ  MSG   ØØ5Ø          Missing user-id
   B   USER9ØØ

SPACE

USER6ØØ  MSG   ØØ53          Operand too long
   B   USER9ØØ

SPACE

USER65Ø  MSG   ØØ51          Invalid option
   B   USER9ØØ

SPACE

USER7ØØ  MSG   Ø13Ø          Invalid non numeric class
   B   USER1ØØ

SPACE

USER8ØØ  MSG   Ø131          Invalid out of range class
   B   USER1ØØ

SPACE

USER9ØØ  L    R14,CFGCSV14
   BR   R14
   DROP R9,R1
   SPACE 3
   CSCDATA
   CSCDS (CCH,RDF,CMD,PFX,USR,MSG,RTE,RND,TMR)
   FSCBD
   FSTD
   REGEQU
   END

Editor’s note: this article will be continued next month.

Fernando Duarte
Analyst (Canada) © F Duarte 1999
Displaying ‘pseudo-graphics’ – part 2

This month we conclude the article to display data with graphics in the same way as on a PC.

GRAPH2 EXEC

This is the program to display two Y-axis.

/* GRAPH2: display data in graphical format using 2 y-coord. */
maximum1 = Ø; /* max. datavalue for Y1 */
maximum2 = Ø; /* max. datavalue for Y2 */
minimum1 = Ø; /* min. datavalue for Y1 */
minimum2 = Ø; /* min. datavalue for Y2 */
average1 = Ø; /* average for Y1 */
average2 = Ø; /* average for Y2 */
count1 = Ø; /* number of values in Y1 */
count2 = Ø; /* number of values in Y2 */
y1ctrl = '('; /* Control character for Y1 data (ea=1) */
y1char = 'X'; /* Character for Y1 data         (ea=Ø) */
y2ctrl = ')'; /* Control character for Y2 data (ea=1) */
y2char = '+'; /* Character for Y2 data         (ea=Ø) */
ovctrl = '/'; /* Control character for overlay (ea=1) */
ovchar = '*'; /* Character for overlay         (ea=Ø) */
/* does the terminal support extended attributes? yes: ea=1 */
ea = bitand(substr(diag('8C'),1,1),'4Ø'x) = '4Ø'x;
if c2d(substr(diag('8C'),5,2)) < 32
then
    do;
        iosk = 'PFØ3';
        say 'Screen is too small. Must be at least 32 lines.';
        signal exit;
    end;
/************ End of GRAPH2 EXEC *************/

* Get data from calling EXEC: *
* X. data for x-axis *
* Y1. first data area for y-axis *
* Y2. second data area for y-axis *
* Y1TITLE title for first data area *
* Y2TITLE title for second data area *
* NBR number of first displayed data item of Y1. and Y2.*
* CLEAR CLEAR or NOCLEAR screen before output (IOS327Ø) *
* TITLE panel title *
* PF which key definitions to include from PF IOS327Ø *
* MESSAGE Display this text in the message line *
* MAXY1 YES = Use maximum Y1 value of complete stem to calculate Y1 axis and not only current page.
* MAXY2 YES = Use maximum Y2 value of complete stem to calculate Y2 axis and not only current page.
* HIGHY1 Highest value to be displayed on the Y1 axis.
* HIGHY2 Highest value to be displayed on the Y2 axis.
* LOWY1 Lowest value to be displayed on the Y1 axis.
* LOWY2 Lowest value to be displayed on the Y2 axis.
* ATTRY1 Colour attributes for the Y1 data values
* ATTRY2 Colour attributes for the Y2 data values
* ATTROV Colour attributes for overlaying data values
* ATTRAY1 Colour attributes for the Y1 average line
* ATTRAY2 Colour attributes for the Y2 average line
* ALINEY1 'NO' = don't display average line for Y1.
* ALINEY2 'NO' = don't display average line for Y2.

**********************************************************

'VMFE2E GET X. Y1. Y2. Y1TITLE Y2TITLE NBR CLEAR TITLE PF'.
'MESSAGE MAXY1 MAXY2 HIGHY1 HIGHY2 LOWY1 LOWY2'.
'ATTRY1 ATTRY2 ATTROV ATTRAY1 ATTRAY2 ALINEY1 ALINEY2';
upper maxy1 maxy2 aliney1 aliney2:
/* Set default or user defined attributes for data values */
if attry1 ≠ 'ATTRY1' & attry1 ≠ ' ' then attry1 = '.jx Set Ctl (' attry1;
else attry1 = '.jx Set Ctl ( Hig=reverse Col=blu';
if attry2 ≠ 'ATTRY2' & attry2 ≠ ' ' then attry2 = '.jx Set Ctl )' attry2;
else attry2 = '.jx Set Ctl ) Hig=reverse Col=yel';
if attrov ≠ 'ATTROV' & attrov ≠ ' ' then attrov = '.jx Set Ctl /' attrov;
else attrov = '.jx Set Ctl / Hig=reverse Col=gre';
if attray1 ≠ 'ATTRAY1' & attray1 ≠ ' ' then attray1 = '.jx Set Ctl <' attray1;
else attray1 = '.jx Set Ctl < Hig=default Col=blu';
if attray2 ≠ 'ATTRAY2' & attray2 ≠ ' ' then attray2 = '.jx Set Ctl >' attray2;
else attray2 = '.jx Set Ctl > Hig=default Col=yel';
/* create header line */
if ea then do;
    header = 'FF'x substr(y1title,1,37);
    if y2title ≠ ' ' & y2title ≠ 'Y2TITLE'
      then header = header right(strip(y2title),37) 'FF'x;
end;
else do;
    header = y1char substr(y1title,1,37);
    if y2title ≠ ' ' & y2title ≠ 'Y2TITLE'
      then header = header right(strip(y2title),37) y2char

© 1999. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
/* calculate maximum and average values for Y1. */
if maxy1 = 'YES'
then
  do;
    from = 1;
    count = y1.Ø;
  end;
else
  do;
    from = nbr;
    count = 59;
  end;
do i = from for count;
  if datatype(y1.i) = 'NUM'
  then
    do;
      maximum1 = max(maximum1,y1.i);
      minimum1 = min(minimum1,y1.i);
      average1 = average1 + y1.i;
      count1 = count1 + 1;
    end;
  end;
/* calculate maximum and average values for Y2. */
if maxy2 = 'YES'
then
  do;
    from = 1;
    count = y2.Ø;
  end;
else
  do;
    from = nbr;
    count = 59;
  end;
do i = from for count;
  if datatype(y2.i) = 'NUM'
  then
    do;
      maximum2 = max(maximum2,y2.i);
      minimum2 = min(minimum2,y2.i);
      average2 = average2 + y2.i;
      count2 = count2 + 1;
    end;
  end;
/* Calculate upper and lower limit for the Y1 axis */
if datatype(lowy1) ¬= 'NUM'
then lowy1 = Ø;
if datatype(highy1) ¬= 'NUM'
then highy1 = maximum1;
/* Calculate upper and lower limit for the Y2 axis */
if datatype(lowy2) ¬= 'NUM'
    then lowy2 = Ø;
if datatype(highy2) ¬= 'NUM'
    then highy2 = maximum2;
/* calculate stepwidth for the Y1 axis */
step1 = (highy1-lowy1) / 23;
if count1 > Ø
    then
        do;
            average1 = average1/count1;
            avgline1 = trunc(average1/step1+.999);
        end;
/* calculate stepwidth for the Y2 axis */
step2 = (highy2-lowy2) / 23;
if count2 > Ø
    then
        do;
            average2 = average2/count2;
            avgline2 = trunc(average2/step2+.999);
        end;
c. = ' ';
/* draw y-axis and description                      */
/* Y-values <= 99999 are displayed with 2 decimals */
/* Y-values >  99999 are displayed without decimals */
v. = ' |'copies(' ',59)'|';
do i = 1 to 23;
    /* description Y1 axis */
    if highy1 > 99999
        then v.i = format(step1*i+lowy1,8,Ø)v.i;
        else v.i = format(step1*i+lowy1,5,2)v.i;
    /* description Y2 axis */
    if highy2 > 99999
        then v.i = v.i format(step2*i+lowy2,8,Ø);
        else v.i = v.i format(step2*i+lowy2,5,2);
end;
/* + on the upper corner of the y1 axis means that */
/* values have been truncated due to highy1. */
if maximum1 > highy1
    then
        do;
            v.23 = overlay('+',v.23,9);
            c.23 = overlay('#',c.23,9);
        end;
/* + on the upper corner of the y2 axis means that */
/* values have been truncated due to highy2. */
if maximum2 > highy2
    then
        do;
            v.23 = overlay('+',v.23,71);
c.23 = overlay('#',c.23,71);
end;

/* draw average line for Y1. */
if average1 > lowy1 & average1 < highy1 & aliney1 = 'NO'
then
do;
  v.avgline1 = overlay('A'repeat('-',58),v.avgline1,1
  c.avgline1 = overlay(repeat('<',59),c.avgline1,1
end;

/* draw average line for Y2. */
if average2 > lowy2 & average2 < highy2 & aliney2 = 'NO'
then
do;
  v.avgline2 = overlay(repeat('-',58)'A',v.avgline2,1
  c.avgline2 = overlay(repeat('>',59),c.avgline2,1
end;

/* draw Y1 data values */
k = Ø;
do i = nbr for 59;
k = k + 1;
if datatype(y1.i) = 'NUM' & y1.i > lowy1
then
do;
  j = trunc((min(y1.i,highy1)-lowy1)/step1+ .999);
doi = 1 to j;
  /* Display character or colour */
  if ea
    then c.i = overlay(y1ctrl,c.i,k+1
    else v.i = overlay(y1char,v.i,k+1
  end;
end;

/* draw Y2 data values */
k = Ø;
do i = nbr for 59;
k = k + 1;
if datatype(y2.i) = 'NUM' & y2.i > lowy2
then
do;
  j = trunc((min(y2.i,highy2)-lowy2)/step2+ .999);
doi = 1 to j;
  if (ea = 1 & substr(c.i,k+1) = y1ctrl) |,
      (ea = Ø & substr(v.i,k+1) = y1char)
    then
      /* Display overlay character or colour */
      if ea
        then c.i = overlay(ovctrl,c.i,k+1
        else v.i = overlay(ovchar,v.i,k+1
      else
        /* Display character or color */
if ea
    then c.ii = overlay(y2ctrl,c.ii,k+10,1);
    else v.ii = overlay(y2char,v.ii,k+10,1);
end;
end;
end;
/* draw X axis */
b1 = '         'copies('|----+----',6)'|';
/* + on the lower corner of the y1 axis means that */
/* values have been truncated due to lowy1. */
if lowy1 > minimum1
    then
doi;
    v.1 = overlay('+',v.1,9);
    c.1 = overlay('#',c.1,9);
end;
/* + on the lower corner of the y2 axis means that */
/* values have been truncated due to lowy2. */
if lowy2 > minimum2
    then
doi;
    v.1 = overlay('+',v.1,71);
    c.1 = overlay('#',c.1,71);
end;
zf1 = nbr+9;
zf2 = nbr+19;
zf3 = nbr+29;
zf4 = nbr+39;
zf5 = nbr+49;
zf6 = nbr+58;
b2 = '       ';
if nbr <= x.Ø
    then b2 = b2||center(x.nbr,7)'  ';
if zf1 <= x.Ø
    then b2 = b2||center(x.zf1,7)'   ';
if zf2 <= x.Ø
    then b2 = b2||center(x.zf2,7)'   ';
if zf3 <= x.Ø
    then b2 = b2||center(x.zf3,7)'   ';
if zf4 <= x.Ø
    then b2 = b2||center(x.zf4,7)'   ';
if zf5 <= x.Ø
    then b2 = b2||center(x.zf5,7)'  ';
if zf6 <= x.Ø
    then b2 = b2||center(x.zf6,7);
if ea
    then pname = 'GRAPHE';
    else pname = 'GRAPHM';
/* Display panel */
call ios pname '*';
------

/*****************************************************
* Return values to calling EXEC:               *
*   IOSK   pressed key (IOS327Ø)               *
*   IOSC   Cursor position rrccc               *
*   ZINPUT last entered command in commandline *
*   CLEAR NOCLEAR as set by IOS subroutine    *
*****************************************************

exit:    'VMFE2E SET IOSK IOSC ZINPUT CLEAR';
return;

/*************************
* IOS - Show the Panel  *
*************************/

ios:
    'NUCXLOAD IOS327Ø';
    parse upper arg i1 i2 .;
    wer   = userid();
    date  = date('E');
    time  = time();
    pname = i1
    'IOS327Ø' i1 ' ( PA2 SUBSET' cursor clear ' )'
    clear = 'NOCLEAR';
    if rc = 1 | rc = 2 | rc = 3 | rc = 5
        then
            do;
                say 'The panel' i1 ' is not available.';
                say 'Please press the ENTER key';
                'CP SLEEP';
                exit;
            end;
        cursor = 'ØØØ1';
        message = ' '; 
        if IOSK  = 'PFØ3'
            then
                if i2 = '*'
                    then return;
                else signal value strip(i2);
            if IOSK  = 'PFØ1'
                then
                    do;
                        'IOS327Ø' i1 'IOSHELP (' clear;
                        signal 'IOS';
                    end;
                if input = ''
                    then return;
                input = strip(input,'T');
                upper input;
                interpret 'input';
                message = 'Returncode' rc ' from' input;
                zinput = input;

------
input = '';  
signal 'IOS';

GRAPHE IOS3270
Panel for screens with extended attributes.

.jx Set Ctl ¬ Col=red Typ=unp  
.jx Set Ctl # Col=red Typ=(unp skip)  
.jx Set Normal Col=blu  
.TCJ _  
Company Name %&TITLE.  
&PNAME

____________________________________________%&DATE-%&TIME ———
.TC
.&pf
.mcj ?????????  
.ch26  
.&attryl  
.&attryl2  
.&attryl3  
.&attryl2  
.c.jx set mask  
(  
)  
.&header
.mcj _?  
.c.jx set mask  
.&c.23  
.&v.23  
.c.jx set mask  
.&c.22  
.&v.22  
.c.jx set mask  
.&c.21  
.&v.21  
.c.jx set mask  
.&c.20  
.&v.20  
.c.jx set mask  
.&c.19  
.&v.19  
.c.jx set mask  
.&c.18  
.&v.18  
.c.jx set mask  
.&c.17  
.&v.17
GRAPHM IOS3270
Panel for screens without extended attributes.

.companyName
&title.
&name
&date &time
PF IOS3270

Definition of PF keys.

******************************************************************************
* DEFINITION OF PF-KEYS:                                              *
* PF3 = END, PF7 = PREVIOUS PAGE, PF8 = NEXT PAGE                      *
******************************************************************************
;EONLY
.YBHCF % % QUIT
PF1=  2=   3=END 4=  5=   6=    7=  8=   9=  10=   11=    12=
;EFIRST
.YBHCF % % QUIT % % % % <
PF1=  2=   3=END 4=  5=   6=    7=  8=NXT 9=  10=   11=    12=
;EMIDDLE
.YBHCF % % QUIT % % % <
PF1=  2=   3=END 4=  5=   6=    7=PRV 8=NXT 9=  10=   11=    12=
;ELAST
.YBHCF % % QUIT % % % <
PF1=  2=   3=END 4=  5=   6=    7=PRV 8=    9=  10=   11=    12=
;EEND

Thomas Rupp
Vorarlberger Illwerke AG (Austria) © Xephon 1999

Why not share your expertise and earn money at the same time? VM Update is looking for REXX EXECs, macros, program code, etc, that experienced VMers have written to make their life, or the lives of their users, easier. We will publish it (after vetting by our expert panel) and send you a cheque when the article is published. Articles can be of any length and can be sent or e-mailed to Robert Burgess at any of the addresses shown on page 2. Why not call now for a free copy of our Notes for contributors?
IBM has announced Release 3 of DITTO/ESA, its storage media and data maintenance utility solution for the VM, VSE, and MVS environments. Enhancements include additional tape support, more POWER spool services, and VSE External Security Manager.

For further information contact your local IBM representative.

* * *

Sterling Software has announced enhancements to its VM:Webgateway for Web-enabling mainframe applications, with new releases of each of the VM:Webgateway components.

New features of VM:Webgateway Web server Release 2.2 include expired password handling, SSL client certificate support, and mediamap support.

The new version also creates an accounting record that documents resource consumption for access to the legacy application via a browser. Any accounting package can process this record and include it for tracking the use of Web site resources.

There is also enhanced publishing tool support, allowing the Webmaster to use desktop publishing tools to manage content. BFS support enables Webmasters to publish content, including facilities such as FrontPage themes and wizards, which must reside in BFS.

VM:Webgateway CGI Extension Release 1.3 includes extended attribute support and new chapters of the tutorial, providing step-by-step instructions on Web-enhancing a full-screen application.

VM:Webgateway OfficeVision Interface Release 1.4 includes enhancements to the management of nicknames and distribution lists, the display of Internet message headers, and the faster display of a user’s inbox. There is also a 3270-like OV/VM interface, enabling a user to work with all of OV/VM through a 3270-like interface, using a browser.

For further information contact:
Sterling Software, 1800 Alexander Bell Drive, Reston, VA 22091, USA.
Tel: (703) 264 8000.
Sterling Software, 64 London Road, Reading, Berkshire, RG1 5AS, UK.
Tel: (0118) 975 0055.

* * *

Microsoft has announced Exchange Connector for OfficeVision/VM, providing a messaging connection between Microsoft Exchange and IBM OfficeVision/VM and PROFS. In addition, the connector also supports sending both files and notes to Exchange from CMS.

For further information contact:
Microsoft, One Microsoft Way, Redmond, WA 98052-6399, USA.
Tel: (206) 882 8080.
Microsoft, Microsoft Place, Winnersh Triangle, Wokingham, Berks, RG11 5TP, UK.
Tel: (01734) 270001.