

*December 1998*

---

## **In this issue**

- 3 Reading the IESCNTRL file and creating a DFHSNT
- 38 VSE/ESA 2.4 announcement review
- 40 Restart prevention revisited
- 41 Intranet download EXEC
- 45 VSE/ESA – what’s in a name?
- 53 IBM’s VSE/ESA Web site
- 60 Program-driven output segmentation – correction
- 61 March 1991 – December 1998 index
- 64 VSE news

update

# VSE Update

---

## Published by

Xephon  
27-35 London Road  
Newbury  
Berkshire RG14 1JL  
England  
Telephone: 01635 38342  
From USA: 01144 1635 38342  
E-mail: xephon@compuserve.com

## North American office

Xephon/QNA  
1301 West Highway 407, Suite 201-405  
Lewisville, TX 75077-2150  
USA  
Telephone: 940 455 7050

## Subscriptions and back-issues

A year's subscription to *VSE Update*, comprising four quarterly issues, costs £100.00 in the UK, \$150.00 in the USA and Canada, £106.00 in Europe, £112.00 in Australasia and Japan, and £110.50 elsewhere. In all cases the price includes postage. Individual issues starting with the March 1991 issue, are available separately to subscribers for £25.00 (\$37.50) each including postage.

## Editorial panel

Articles published in *VSE Update* are reviewed by our panel of experts. Members of the panel include Stanley Stewart (USA), Robert Botsis (USA), and Jesse Joyner (USA).

## Editor

Fiona Hewitt

## 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.

## Contributions

Articles published in *VSE 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*.

## VSE Update on-line

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

---

© Xephon plc 1998. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

*Printed in England.*

## Reading the IESCNTL file and creating a DFHSNT

The program presented here reads the IESCNTL file and creates a DFHSNT including all the JCL necessary to assemble, LNKEDT, and 'new' it. It also produces two reports:

- The first report is sorted by the eight-character userid from the IESCNTL file.
- The second report is sorted by the transaction security code from the DFHCSD file.

When IBM first produced a VSE that supported the Interactive User Interface (IUI), it simplified the adding of new userids and any associated passwords, removing the need to edit a DFHSNT member residing in ICCF, VOLLIE, BIM-EDIT, or another on-line editor, code DFHSNT macro statements, and submit it for assembly and cataloguing. With the IUI, you simply use the IUI dialogs and panels, adding the necessary information to the panels presented.

Unfortunately, however, IBM did not carry its IUI implementation far enough: if ATTACHSEC=IDENTIFY is specified for the AORs in an MRO/ISC environment, it doesn't work, and you have either to go back to maintaining a DFHSNT or continue to use the IUI and also maintain a DFHSNT.

This program was written to resolve this problem, and to enable a DFHSNT to be maintained for the AORs. You can still use the IUI to maintain your userids and passwords, and can then submit a job that executes this program to automatically create DFHSNT macro statements from the IESCNTL file and submit them for assembly and cataloguing.

The program is executed in a batch partition by submitting the following required statements. Note that you will need to insert your own DOS JOB and VSE/POWER JECL statements.

```
// ASSGN SYS001, CUU
// ASSGN SYS002, CUU
// ASSGN SYS003, CUU
// ASSGN SYS009, SYSLST
// DLBL SORTWK1, 'SS1', 0                SS1=SORTWK1 NAME
```

```

// EXTENT SYS001,VVVVVV,1,0,SSSS,NNNN
// DLBL SORTWK2,'SS2',0          SS2=SORTWK2 NAME
// EXTENT SYS002,VVVVVV,1,0,SSSS,NNNN
// DLBL SORTWK3,'SS3',0          SS3=SORTWK3 NAME
// EXTENT SYS003,VVVVVV,1,0,SSSS,NNNN
// DLBL IESCNTRL,'NNNNNNNN',,VSAM NNN=IESCNTRL FILE NAME.
// DLBL DFHCSD,'NNNNNNNN',,VSAM NNN=DFHCSD FILE NAME.
// DLBL OUTFILE,'NNNNNNNN',... NNN=OUTFILE FILE NAME.
// EXTENT SYS040,VVVVVV,1,0,SSSS,NNNN
// ASSGN SYS040,???             ???= CUU OF OUTPUT DISK.
// EXEC DPCSNT,SIZE=(DPCSNT,128K)
...ONE OR MORE THAN ONE OPTIONAL PARAMETER...
/*
/&

```

Note that in some cases the // ASSGNs for SYS001 through SYS003 and the associated DLBLS/EXTENTS for the SORT work files may be optional. This depends on whether or not you are using a disk space manager and whether your DLBLS/EXTENTS for the SORT work areas are in standard/partition labels.

When executed, the program opens the IESCNTRL file. It then determines whether a PARM value was entered on the // EXEC statement and whether it contains COPY=Y. If COPY=Y was specified, there were no leading or trailing spaces, and it was the only value specified, an output file (CPYFILE) is opened and the program reads and outputs 'ALL' records in the IESCNTRL file containing 'US' in the first two bytes of the key. At EOF, both files are closed and execution is terminated. This allows existing user records to be copied so that they can be used as input in your own program or a SORT, and copied during a migration. Before using this function, you should check that the user records in the new IESCNTRL file are in the same format as the old records; failure to do this may render your system unusable. The 'CPYFILE' is created as a variable/unblocked formatted file.

If COPY=Y was not specified, the program opens the other output files (REPORT and OUTFILE), an I/O file (NAME-TABLE), and the DFHCSD file as input, and prepares to create/submit a DFHSNT. Parameter cards, a card security table, and JCL necessary to assemble, catalog, and 'new' the DFHSNT are also read as input.

The output files are as follows:

- The report file is used to produce two reports:
  - The first is created by combining information from the IESCNTL file and the NAME-TAB file, if used.
  - The second is created by combining information from the IESCNTL file and the DFHCSD file.
- The OUTFILE is used for whatever purpose you wish. It contains information collected by reading 'ALL' the other input and I/O files. The record layout of this file can be obtained by reviewing the SORT-FILE record description found later in this program's file-section.
- The NAME-TABLE file is optional (see the discussion below on parameter cards). It is used to obtain users' first and last names, keyed by the eight-byte userid which is the key of the IESCNTL file, and users' department and/or location names. This file is not maintained by an on-line transaction/program; it is simply reloaded by reading an ICCF member which contains the data used to reload it. At some point, a CICS program should be written to maintain this file rather than editing an ICCF member whenever new users are added or current users are changed or deleted.
- The DFHCSD file is used to obtain the transaction IDs (PCT entries) of every transaction, and the programs/maps (PPT entries) of every program/map, although the program and map entries are not used. Depending on the parameter card selected (see below), the file is read sequentially or randomly. Because the record layout of the CSD file is release-dependent, changes may have to be made to this program if migrating from one CICS release to another (no changes are needed with CICS 2.2 and 2.3).

If all the transaction ids are read, together with their associated security code values, they can be shown along with a user's security code values, obtained from the IESCNTL file records. This enables you to see who can invoke which transactions.

## PARAMETER CARDS

The parameter cards are used to control some of the program's operation. The security table is used to assign meanings to the security values. Each of the parameter cards has a function code in positions 1-4, an equals sign (=) in position 5, and the function beginning in position 6. If a card doesn't conform to these rules, it's assumed to be a JCL statement used to assemble/catalog/'new' the DFHSNT. The order of the parameter cards is not critical, except for those that deal with the submission of JECL/JCL, for the assembly/cataloguing of the DFHSNT must follow any/all others.

An error in specifying a parameter card will generally cause the program to ignore the offending entry, although in some cases the program will be terminated. In either case, a message will always be issued on the system console.

Some parameter cards can be specified as a PARM value in a // EXEC statement. In this case, the sum total of all the characters entered must not exceed 76 bytes. If you specify a parameter in the PARM= operand in the // EXEC statement and specify the same parameter in a parameter card, the last value specified (that specified in the parameter card) will override the one specified before.

In the list below, the parameters that can be specified in the PARM= operand are indicated by an asterisk. Parameters that are displayed in the headings of either report are indicated by the letter 'H'. Any defaults for the parameters are indicated as such after the equals sign. If a blank is indicated, that parameter is considered null, as is the case with the tables. If you wish to accept the default value of a parameter, it need not be specified.

- \*H ASNT=Y

This parameter controls whether or not a DFHSNT will be created and submitted for assembly, LNKEDT, and possibly 'newing'. Specify 'Y', without the quotes, if you want a DFHSNT created and submitted; if not, specify 'N' (or any other value).

Note that, if you specify ASNT=Y, are using the IUI for signing on, and are using MRO/ISC, the LIB.SUBLIB of the DFHSNT

must be catalogued into not being the same as the one that contains IBM's original DFHSNT.PHASE. In fact, you should really create a special LIB.SUBLIB for your AOR(s) containing a single .PHASE MEMBERTYPE (DFHSNT.PHASE), and LIBDEF this LIB.SUBLIB in the CICS start-up JCL for your AOR(s). Note that the CICS in which the IUI is running and in which you're maintaining your sign-ons (ie userids and passwords) must not have access to this DFHSNT. If it does, the IUI will prevent you from changing any signed-on user id it finds in the accessed (LIBDEFed) DFHSNT.

Specifying ASNT=N together with RPTS=N will cause the program to terminate.

- \*H DUPS=N

This parameter controls whether or not duplicate transaction ids are printed in the second report. Specify 'Y', without the quotes, if you want duplicate transaction ids printed; if not, specify 'N' (or any other value).

Duplicate transaction ids, together with their transaction security value, are normally dropped from the second report. However, if a transaction id is specified in more than one group name and with a different transaction security value, it will be printed twice, once for each of the transaction security values.

Specifying DUPS=Y and also specifying group list names via GRPL= may also cause a transaction id to be printed more than once if the group(s) it belongs to are attached to multiple group list names and those group list names were specified in the GRPL= parameter. You should normally specify DUPS=N or allow it to default.

- \*H FULL=N

This parameter allows you to print up to two additional lines from the IESCNTL file:

- The first line is printed only if PASS=Y is specified, and contains a user's last twelve passwords.

- The second line contains the password expiry date, in Gregorian format, and the number of days to expiry. This line is printed regardless of whether or not PASS=Y was specified.

Note that neither line is printed if there is no data to be printed – ie all last twelve passwords contain low-values and the password expiry date and the number of days to expiry contain blanks or zeros. Note also that even if the password expiry date contains a value, IBM does not use it unless the number of days to expiry is not zero.

If you wish to print this additional information, specify ‘Y’, without the quotes; if not, specify ‘N’ (or any other value).

- H EXGP=

This parameter allows you to exclude (EXGP=) or include (INGP=) group names contained in the DFHCSD H INGP= FILE. Note that you cannot specify any INGP= parameter statements if you have also specified any EXGP= parameter statements, as the two are mutually exclusive.

A maximum of 99 group names can be specified. You can do this either by using 99 EXGP=/INGP= statements or by specifying each group name in one or more parameter card(s), separating each group name with a comma and terminating the last entry with a space. Group names can be from 1 to 8 characters, and the last group name must end before column 62. Comments can be entered following the only or last group name, as long as there is at least one blank following it/them. Group names can be entered generically by specifying an asterisk in any one of the eight positions where you wish the character to be matched. If you specify a group name that does not exist in your DFHCSD, it will be ignored. The following are some examples.

(1) EXGP=AP,PAYROLL	<-- EXCLUDE THESE
(2) INGP=AP	<-- INCLUDE THIS
(2) INGP=PAYROLL	<-- INCLUDE THIS
(3) EXGP=**D,**E	



- (1) illustrates the entering of multiple group names on a single statement and the use of comments. In this case, the ‘AP’ and ‘PAYROLL’ group names will be excluded.
  - (2) illustrates the entering of group names on multiple INGP= statements and the use of comments. In this case, the ‘AP’ and ‘PAYROLL’ group names will be included. Note that, if no other INGP= statements are entered, the only groups processed will be ‘AP’ and ‘PAYROLL’.
  - (3) illustrates the entering of multiple generic group names in a single statement. In this case, any group name containing any characters in positions 1-2, the character ‘D’ or ‘E’ in position 3, and any characters in positions 4-8 will be excluded. All other group names will be processed.
- H GRPL=

This parameter allows you to include certain group list names. A maximum of seven group list names can be specified, either in 7 GRPL= statements or by specifying each group list name in one or more parameter card(s), separating each group list name with a comma and terminating the last entry with a space. Group list names can be from one to eight characters. Comments can be entered following the only or last group list name as long as there is at least one blank following it/them. Group list names cannot be entered generically.

Specifying a group list name that does not exist in your DFHCSD file will cause an appropriate error message to be displayed on the system console, and that entry will be ignored. If none of the specified group list names exist in your DFHCSD file, the program will display an error message on the system console and then be terminated.

Note that this parameter must not be used if you usually attach only very few or no groups to a group list in your DFHCSD file – as, for example, in a test CICS system. This parameter causes the DFHCSD file to be read randomly, looking for group names that are attached to the specified group list name/s. If there are

none or very few, you will end up with few or no transaction ids/programs/maps in the second report. Specifying this parameter causes only those transaction ids/maps/programs found in the group names attached to the specified group list names to be processed. The following are some examples.

```
(1) GRPL=PROD,TEST      INCLUDE THESE GRP LIST NAMES
(2) GRPL=DEVM          INCLUDE THIS GROUP LIST NAME
(2) GRPL=TEST          ..AND THIS GROUP LIST NAME..
(3) GRPL=VSELIST
```

- (1) illustrates the entering of multiple group list names in a single statement and the use of comments. In this case, only those groups attached to group list names of ‘PROD’ and ‘TEST’ will be processed.
  - (2) illustrates the entering of group list names in multiple GRPL= statements and the use of comments. In this case, only those groups attached to group list names of ‘DEVM’ and ‘TEST’ will be processed.
  - (3) illustrates the entering of a group list name in a single GRPL= statement. In this case, only those groups attached to group list name ‘VSE-LIST’ will be processed.
- \*H NTAB=N

This parameter allows you to bypass the NAME-TABLE file. Specify ‘Y’, without the quotes, if you do not wish to bypass the use of this file (OPEN/READ/REWRITE/CLOSE); if not, specify ‘N’ (or any other value).

Specifying ‘N’, or allowing it to default, removes the headings associated with the printing of information from this file (ie department or location and first or last name) from the first report.

Specifying ‘Y’ requires the name table file to be predefined and loaded before the program is executed. It also requires you to maintain a member in ICCF or some other on-line editor, from which the file can be maintained and loaded, as there is no other means provided to do this. If you wish to maintain a member in your on-line editor, the following is the record layout (note that all fields are in character format).

FIELD NAME	POSITION	LEN
USERID	01-08	08
FILLER	09-19	11
DEPARTMENT/LOCATION	20-39	20
USERS FIRST NAME	40-59	20
USERS LAST NAME	60-79	20
FLAG-BYTE	80-80	01

The flag-byte is set to 'F' whenever a record is found in both this file and the IESCNTL file. Any records not containing an 'F' are in the IESCNTL file but not in the NAME-TABLE file. Records that are not in the IESCNTL file but are in the NAME-TABLE file are noted in the first report with a string of question marks.

The layout of the above record is identical to the NTAB-RECORD layout defined in this program. You can, of course, change it as you see fit.

The VSAM DEFINE statements for the file are shown below. Note that you need to enter your own VOLSER names, indicated by the VVVVVV, and your own user catalog name indicated by the CCCCCCCC. The cluster/data/index names are arbitrary, as are the FREESPACE and space allocations (ie CYL), and can be changed if you wish. The share options (ie SHR(2)) should not be changed, except in the following two situations:

- you plan to define this file to CICS and write a program to maintain it on-line.
- you don't have a batch-to-CICS product, which would mean that the file had to be closed manually each time the program is executed, and you don't wish to do this.

```

DEFINE CLUSTER(NAME(DPNKTAB.NAME.TABLE) -
  SPEED REUSE CYL(1 1) KEYS(8 0) SHR(2 3)
  FREESPACE(0 0) RECORDSIZE(80 80)) -
DATA(NAME(DPNKTAB.NAME.TABLE.DATA) -
  CISZ(4096) VOL(VVVVVV)) -
INDEX(NAME(DPNKTAB.NAME.TABLE.INDEX) -
  CISZ(512) VOL(VVVVVV)) -
CATALOG(CCCCCCCC)

```

- \*H OPNM=N

This parameter allows the OPNAME = and USERID= operands to be inserted when you create the SNT macro statements. Specify 'Y', without the quotes, if you wish the OPNAME= operand to be included in the DFHSNT; if not, specify 'N' (or any other value).

If 'Y' is specified, the OPNAME= operand is filled in by using the eight-byte userid taken from the IESCNTL file. The remaining twelve bytes are padded with spaces. There is no known reason for you to specify 'Y', as the IUI does not use the OPNAME= operand.

- \* PASS=N

This parameter controls whether or not the six-byte password contained in the IESCNTL file is printed on the first report. Specify 'Y', without the quotes, if you want the users' passwords printed; if not, specify 'N' (or any other value).

Note that specifying PASS=N or allowing it to default has no effect if ASNT=Y is also specified, as the password must be used when assembling the DFHSNT. Note also that the heading that is normally printed above the password is removed if PASS=Y is specified. This is done in order to prevent a possible security exposure should someone see the report.

- \*H POWP=Y

This parameter controls whether the program will use the DPPOWP subroutine or a punch file (CARDOT) to submit the JECL/JCL for assembly/LNKEDT of the DFHSNT directly to the VSE/POWER RDR queue. Specify 'N', without the quotes, if you do not wish to use the DPPOWP subroutine.

If you specify 'Y', the DPPOWP subroutine must be made available by previously assembling and cataloguing it as a .PHASE MEMBERTYPE in a LIBDEFed library accessible to this program at the time of execution.

Specifying 'N' requires a PUN JECL statement, with the DISP=I operand, to be added to the execution JECL before the program is executed. It is recommended that the following PUN statement be used:

```
* $$ PUN DISP=I,CLASS=Ø,JNM=DFHSNT
```

If POWP=N is specified, you should not add a \* \$\$ job statement to your JECL/JCL, as POWER will create one for you. If you do, the job will be submitted with two \* \$\$ job statements. Also, POWP=N requires that ASNT=N be specified if you do not wish a DFHSNT to be assembled or LNKEDTed. If POWP=N and ASNT=Y are both specified and you enter a \* \$\$ PUN JECL statement with DISP=I, the DFHSNT will be assembled or LNKEDTed whether you want it to be or not. This is because the only way to stop the punched output from being written is to use ASNT=N with DISP=I, as the output will automatically be sent to the RDR queue from the PUN queue unless the DISP=I operand or the entire PUN queue statement are removed. You could also add 'CLASS=?' to the PUN statement and replace the question mark with the POWER reader class that is never started. Because using DPPOWP removes this problem, you can safely specify POWP=Y and ASNT=N.

- \*H REMT=N

This parameter controls whether or not transaction IDs that are specified as remote in the DFHCSD file are reported. Specify 'Y', without the quotes, if you wish to include these transaction ids in the second report; if not, specify 'N' (or any other value). Regardless of what you specify, transaction id records are written to the output file (OUTFILE) to be dealt with at your discretion.

An MRO/ISC environment is typically set up with at least one TOR and with one or more AORs. Transaction ids specified in the TOR do not normally have a program name specified, but rather have the remote attributes specified (ie dynamic, remote system, remote name, etc). The groups that hold these remote transaction ids are or should be specified in special group names; these special group names would be attached to the group list for the TOR.

Because these transaction ids have no program name specified, there is no point in printing them in the second report, and the default is therefore to drop these transactions. Nevertheless, the option is provided for you to include them. However, even if you specify REMT=Y, these transaction ids may be dropped because they are or should be duplicated in other group names, unless DUPS=Y was specified. Also, if you included any EXGP= parameters and the group names specified were special group names (see above), specifying REMT=Y would be useless. This is because the group names would contain only remote transaction ids and would therefore never be seen by this program since they would have been bypassed by the EXGP= parameter.

- \*H RPTS=B

This parameter controls whether or not the first, second, or both reports are printed. Specify '1', without the quotes, to print the first report only. Specify '2' to print the second report only. Specify 'B' to print both reports. Specify 'N' (or any other value, except '1', '2', or 'B') to not print either report. Note that specifying '1', '2', or 'N' simply prevents one or both reports from being printed, and has no effect on any other output. Specifying RPTS=N together with ASNT=N will cause the program to terminate.

- SECT=

This parameter, actually table value, identifies a specific security code and its description. There is room to specify two security codes/descriptions per statement. The first entry begins in position six, immediately following the equals sign; the second entry begins in position 33. There is no space between the security code and its description. The following is an example.

```
SECT=55TRAFFIC INQ AND TICP      56TRAFFIC UPD
```

- \*H TRAN=4

This parameter controls whether or not transaction ids are printed in the second report, with or without the program name they invoke, and whether or not any PF key value associated with a

transaction id is also printed. Specify '4', without the quotes, if you do not want to print the program name the transaction id invokes and the PF key that can be used to invoke the transaction id (ie TASKREQ. Specify '8' if you do wish to print this information. Specifying TRAN=4 causes 25 transaction ids to be printed within one line, each four bytes long with one space between each transaction id, as in the following example.

```
ACT1 ACT2 ACT3 ACT4 PAY1 PAY2 TRN1 TRN2 TRN3 TRN4
```

Specifying TRAN=8 causes approximately eight transaction ids to be printed within one line, each thirteen bytes long (or eighteen bytes long if a PF key is associated with the transaction id), with one space between each transaction id/PF key/program or transaction id/program:

```
ACT1-ACTPGM1 PAY1(PF23)PAYPGM1 PF3 (PF03)IESXXXXX
```

If, in the above example, transaction id 'ACT1' was specified as 'remote' and there was no program specified (which is the recommended method) and REMT=Y was also specified, the above line would appear as in the following example:

```
ACT1-          PAY1(PF23)PAYPGM1 PF3 (PF03)IESXXXXX
```

- \*H USID=4

This parameter controls the way in which userids are printed in the second report. Specify '4', without the quotes, if you have not chosen to use userids of more than four characters. Specify '8' if you have chosen to use eight-character userids. Specify 'F' if you wish to have each userid 'float', depending on its length.

With the release of VSE/ESA, IBM changed the number of characters that could be used for a userid from four to eight (except for ICCF userids, which must still be four characters). Specifying USID=4 causes 25 userids to be printed within one line, each four bytes long with one space between each userid, as in the following example. Note that if a userid is longer than four characters, it is truncated to four characters, and an asterisk is printed to the right of the userid, as in FORS\* (ie FORSEC) below.

```
FORS*SYSA USR1 USR2 USR3 USR4 USR5 USR6 USR7 USR8
```

Specifying USID=8 causes thirteen userids to be printed within one line, each eight bytes long with one space between each, as in the following example.

```
ACCTUSR1 ACCTUSR2 FORSEC SYSA TESTUSR1
```

Specifying USID=F causes *n* userids to be printed within one line, each *n* bytes long, depending on each userid's length, with one space between each userid, as in the following example.

```
$SRV ACCTUSR1 FORSEC SYSA TESTUSR1 USR1 USR2
```

Any value other than those above is assumed to be JECL/JCL for assembling, cataloguing, and 'newing' a DFHSNT. You must include all JCL required to perform these tasks, although statements to 'NEW' the DFHSNT aren't required, especially if you don't have a batch-to-CICS product – in which case, you will have to perform this step manually, possibly by putting a // PAUSE statement in the job.

All statements must contain the digits '3', '5', or '9' in position one. Columns 73 to 80 are ignored. You may have multiple '3' statements, but only one '5' and one '9' statement are allowed. Type '3' statements are used for all JECL/JCL inserted before and after the DFHSNT macro statements. The type '5' statement informs the program when and where to insert the DFHSNT macro statements. The type '9' statement informs the program that it is the last statement to be submitted, and it must be a VSE/POWER EOJ (ie \* \$\$ EOJ) statement. There is no limit on the number of type '3' statements that can be submitted, and they may be blank after the '3'.

Note that there is no checking to ensure that the JECL/JCL contained in each type '3' statement is correct, or whether the first statement contains a valid POWER or VSE job statement. If POWP=Y is specified and a POWER job statement is not the first statement submitted, POWER will flush the job and the program will cancel.



Continuation is supported by placing an 'X' or any other non-blank character in column 72 in each type '3' statement where it is needed. You should leave column 72 blank unless you are specifying continuation. You must be very careful that each type '3' statement contains valid JECL/JCL, otherwise, when the job to assemble and LNKEDT the DFHSNT is executed, invalid statements or messages may occur on the system console, as the contents of these statements are not validated. If you change ASNT=Y to ASNT=N, you need not remove any of these statements as they will be ignored.

Below is a simple example of some possible JECL/JCL that can be used. Note that if POWP=N was specified, you should not include a POWER job statement, as POWER will create one for you.

```

3* $$ JOB JNM=DFHSNT,DISP=D,CLASS=0,...
3* $$ LST CLASS=...
3// JOB DFHSNT DP00 SNT SIGN ON TABLE FOR AOR'S.
3// ON $ABEND OR $CANCEL GOTO SKIP1
3// LIBDEF PHASE,CATALOG=USR1.DFHSNT
3* $$ SLI ICCF=(DPVLBL),LIB=(02) VDISK LABELS
3// OPTION CATAL
3// EXEC ASMA90,PARM='SIZE(MAX-200K,ABOVE)'
3      TITLE 'SIGN ON TABLE FOR AOR'S.'
3      PRINT NOGEN
3SNT   DFHSNT TYPE=INITIAL
5 >> STATEMENTS FROM IESCNTL FILE INSERTED HERE <<
3      DFHSNT TYPE=FINAL
3      END
3/*
3// IF $RC = 0 THEN
3// EXEC LNKEDT,SIZE=256K
3/&
9* $$ E0J

```

Any statement containing an asterisk in the first position is treated as a comment and ignored.

## NOTES

- 1 A large proportion of this program contains COBOL/VSE statements (ie inline performs, scope terminators, and reference

modification). If you do not yet have the COBOL/VSE compiler (or at least the COBOL II compiler), a large number of changes will be required.

- 2 This program dynamically calls five subroutines, or four if POWP=N is specified. These subroutines must be assembled and catalogued as .PHASE MEMBERTYPES in a library that is LIBDEFed (temporary or permanent) before execution. If you don't wish to dynamically call these subroutines, you must change every call statement in the program similar to the following:

```
CALL 'DP????' USING ????????
```

You can remove or comment out the 'MOVE DP???? TO SUB-NAME' statements that immediately precede the CALL statements. You must also assemble and catalog the subroutines as .OBJ MEMBERTYPES, and the LIB.SUBLIB into which they are catalogued must be LIBDEFed before link editing the program. If you do this, be advised that you will receive an 'UNRESOLVED EXTERNAL REFERENCE' for DPPOWP if you don't have this subroutine or did not assemble/catalog it. As long as you don't specify POWP=Y, this can be ignored. The 'UNRESOLVED EXTERNAL REFERENCE' for BOMBER can also be ignored. The subroutines can of course be used in your own programs if you wish.

- 3 The use of this program requires that certain criteria be met:
  - a You must be using the IUI to maintain userids, irrespective of whether or not you are using any other IUI function. This is because the program reads the user records contained in the IESCNTL file to create the DFHSNT macro statements and to produce the first report, and the security keys to produce the second report. There is no provision to read DFHSNT macro statements.

Note that switching from maintaining DFHSNT macro statements to using the IUI will consume large amounts of CICS DSA, even if only one or two people are maintaining the userids. This is because the IUI issues CICS shared GETMAINS for each signed-on user. This means that all the

other signed-on users will be consuming CICS DSA even though they aren't maintaining userids. If you have available DSA, this is no problem, but you may want to consider using DPSIGN, previously published in *VSE Update* (Issue 19, pp 3-51). This program replaces the IESIES01 IUI sign-on program, displays the same sign-on screen as the IUI, and does much, much more without the DSA overhead. It also allows you to select whether or not you wish to verify userids and passwords from the IESCNTL file or from a DFHSNT. This feature eliminates the need to maintain userids in an IESCNTL file on another machine if not using shared DASD. There is also a user-written program, available on Nascom's Web site, that allows userids to be maintained in a native CICS environment outside the IUI.

- b You must be using resource definition on-line (ie RDO, also known as DFHCSD) as this program uses the transaction records contained within it to produce the second report, as well as writing these records, and any program and map records, to the OUTFILE. There is no provision to extract PCT/PPT information from the macro equivalents. You should abandon the macro versions and switch to RDO, as, sooner or later, IBM will stop supporting them.
- c If you are not using CICS's built-in security (ie TRANSEC= in the transaction id definitions contained in the RDO file, and equivalent security values for each userid in the IESCNTL file (ie SCTYKEY= in the DFHSNT macro definitions), the second report will be of little or no value since all security codes will probably be specified as '01', causing all users to be listed under '01'.
- d If you are using MRO/ISC and have not specified 'ATTACHSEC=IDENTIFY' in your sessions connection, you do not need any security, and therefore don't need a DFHSNT. However, be aware that you cannot use certain CICS command-level commands, such as 'EXEC CICS ASSIGN OPERKEYS(...)', without having specified 'IDENTIFY'. If you are not using MRO/ISC, you will not

need to automatically create a DFHSNT, as long as you maintain userids through the IUI, or manually through DFHSNT macro statements.

- 4 Although the process statements at the beginning of this source can be changed, doing so may affect the operation of the program. Those currently specified work. This program is currently set up to execute as a 24-bit program, although it can be executed as a 31-bit program. The only reason it wasn't set up for 31-bit is that it was assumed that most installations will execute batch programs in 24-bit (although you should be executing them in 31-bit, especially for programs that process VSAM files in partitions that exceed 16M). If you wish to execute the program in 31-bit mode, you will probably have to make the following change to the process statement at the beginning of this source:

```
DATA(24)
  -T0-
DATA(31)
```

You will also probably have to change your LE/VSE run-time options from/to the following:

```
ALL31(OFF),STACK(128K,128K,BELOW,KEEP)
  -T0-
ALL31(ON),STACK(128K,128K,ANYWHERE,KEEP)
```

Instead of changing your LE/VSE run-time options, you can override them by entering the following, as a PARM= value on the // EXEC DPCSNT,SIZE=(DPCSNT,128K),PARM='????????' statement.

```
WHERE ?????????? IS 'ALL31(ON),STACK(, ,ANYWHERE)'
```

Note that this program assumes that CBLOPTS(ON), which is the default, is specified as an LE/VSE installation run-time option. This option states that LE/VSE overrides must/will come after, not before, a user program's PARM= values, as in the following example.

```
PARM='ASNT=Y,DUPS=N/ALL31(OFF),STACK(, ,ANYWHERE)'
```

- 5 Up to 50% increased performance can be achieved simply by 'de-editing' the macros used to assemble the DFHSNT or for any

other CICS assembly. Unlike the VSE macros, which are ‘de-edited’ and placed in PRD1.MACLIB, IBM does not yet ship ‘de-edited’ macros for CICS. In order to ‘de-edit’ them, you need to perform three steps:

- a The first is to ‘de-edit’ the CICS macros used. The JCL statements needed to ‘de-edit’ the DFHSNT macros are shown below. Note that you may modify the DLBLs/EXTENTs/ASSGNs for IJSYSPH/IJSYSIN SYSPCH/SYSIPT as you see fit, or replace them with IESINSRT. Also, if service is applied affecting any of the macros listed below (or any others you may have added), you must remember to ‘de-edit’ the macros again to pick up any fixes.

```
// DLBL IJSYSPH,'DP.SYSPCH.FILE',1971/001
// EXTENT SYSPCH,DOSRES,1,0,15000,1500
ASSGN SYSPCH,DISK,VOL=DOSRES,SHR
// LIBDEF SOURCE,SEARCH=(PRD1.BASE,PRD2.GEN1) (A)
// EXEC ESERV,SIZE=256K
    PUNCH DFHSNT
    PUNCH DFHCOVER
    PUNCH DFHVM
    PUNCH DFHSYS
    PUNCH DFHPRMCK
/*
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'DP.SYSPCH.FILE',1971/001
// EXTENT SYSIPT,DOSRES
ASSGN SYSIPT,DISK,VOL=DOSRES,SHR
// EXEC LIBR,SIZE=256K,PARM='ACC S=USR1.IBM' (B)
CLOSE SYSIPT,READER
```

Note that, in the above JCL, the LIBDEF highlighted as (A) points to the place where the edited (ie .E MEMBERTYPE) macros for CICS currently reside. The ‘ACC’, for access, (highlighted as (B)) indicates where the ‘de-edit’ macros are to be placed. You can use an existing LIB.SUBLIB or create a new LIB.SUBLIB.

- b The second is to replace your // EXEC ASMA90 EXECUTE statement with one that does not include the EXIT operand that invokes ‘EDECKXIT’. Change your current // EXEC ASMA90 statement, removing the EXIT operand that invokes

'EDECKXIT' to one similar to that shown below (note that the // EXEC is not shown because of space limitations).

```
ASMA90,SIZE=(ASMA90,128K),PARM='SIZE(MAX-200K,ABOVE)'
```

- c The third is to add/change a new/existing LIBDEF, pointing to where the 'de-edited' macros were placed, in your ASMA90 execution JCL. Add/change a // LIBDEF statement pointing to the LIB.SUBLIB where the 'de-edited' macros were catalogued, as in the following:

```
// LIBDEF *,SEARCH=USR1.IBM
```

Note that none of the above will work if you are not yet using ASMA90. Moreover, you must not use // EXEC assembly if you are running VSE/ESA 2.2 or higher, as this statement is changed automatically by IBM to ASMA90 with the EXIT operand mentioned above.

- 6 If you are running VSE/ESA 1.3.1 or above and are using four-character userids rather than the eight-character ones imposed by the IUI, you can apply the following zap to save the user having to press the tab key after entering their four-character userid to position the cursor at the password. The zap has two job steps: the first copies the original map to another LIB.SUBLIB of your choice, which must be LIBDEFed ahead of IJSYSRS.SYSLIB in your CICS start-up JCL, and the second performs the actual zap using MSHP's patch function. You should respond 'GO' to message M087D. Issue 'CEMT SET PROG(IESESOP) NEW ENA' in each CICS in which the zapped map is intended to be used after the job completes.

```
// EXEC LIBR,SIZE=256K,PARM='MSHP'  
CON S=IJSYSRS.SYSLIB:LLLL.SSSS <--- CHANGE TO YOUR LIB  
COP IESESOP.PHASE R=YES  
/*  
/* USE THE FOLLOWING ZAP STATEMENTS IF YOU ARE RUNNING  
/* VSE/ESA 1.3.?.  
/*  
// EXEC MSHP,SIZE=900K  
PATCH SUBLIB=LLLL.SSSS          <--- CHANGE TO YOUR LIB  
AFFECTS PHASE=IESESOP  
ALTER 0AC0 0008034005676D6D : 0004034005676D6D
```

```

ALTER ØAC8 6D6D6D6D6D6DØØØØ : 6D6DØØØØØØØØ1Ø1FØ
ALTER ØADØ ØØ29Ø2FØØ57Ø4Ø4Ø : Ø56CØØØØØØØØ25Ø2FØ
ALTER ØAD8 E388854Ø95819485 : Ø57Ø4Ø4ØE388854Ø
ALTER ØAEØ 4Ø82A84ØA6888983 : 958194854ØA38885
ALTER ØAE8 884ØA388854ØA2A8 : 4ØA2A8A2A385944Ø
ALTER ØAFØ A2A385944Ø929596 : 929596A6A24ØA896
ALTER ØAF8 A6A24ØA896A44B : A44B4Ø4Ø4Ø4Ø4Ø

/*
/* USE THE FOLLOWING ZAP STATEMENTS IF YOU ARE RUNNING
/* VSE/ESA 2.2.?.
/*
AFFECTS PHASE=IESESOP
ALTER ØABE ØØØ8Ø34ØØ5676D6D : ØØØ4Ø34ØØ5676D6D
ALTER ØAC6 6D6D6D6D6D6DØØØØ : 6D6DØØØØØØØØ1Ø1FØ
ALTER ØACE ØØ29Ø2FØØ57Ø4Ø4Ø : Ø56CØØØØØØØØ25Ø2FØ
ALTER ØAD6 E388854Ø95819485 : Ø57Ø4Ø4ØE388854Ø
ALTER ØADE 4Ø82A84ØA6888983 : 958194854ØA38885
ALTER ØAE6 884ØA388854ØA2A8 : 4ØA2A8A2A385944Ø
ALTER ØAEE A2A385944Ø929596 : 929596A6A24ØA896
ALTER ØAF6 A6A24ØA896A44B : A44B4Ø4Ø4Ø4Ø4Ø

/*

```

Note that applying the zap will prevent the use of any userids greater than four characters. This includes the IBM-supplied 'FORSEC' userid.

- 7 If viewing a user's password is a sensitive issue in your installation, it is highly recommended that you use the PWD= operand, in the \* \$\$ LST statement for the execution of this program and in the \* \$\$ JOB statement for the execution of the DFHSNT assembly, assuming that PASS=Y and ASNT=Y were specified, respectively. Note that the ICCF allows password-protected queue entries to be viewed using the /LP command, without a password having to be entered. Some third-party software may also allow this to be done.
- 8 This program uses SYS007 for CARDIN, SYS008 for CARDOT, and SYS009 for REPOUT (ie both reports). If your system does not have these SYSNNNs permanently assigned to a real or virtual reader, punch, and printer, it's recommended that you change them to conform to your installation. Alternatively, you can simply add // ASSGN statements in the execution JCL, pointing SYS007 to a card reader, SYS008 to a card punch, and SYS009 to a printer.

- 9 This program contains constants of 'DPCSNTJ' for the POWER and VSE job names and 'DPCSNT' for the program name. If you execute this program using a different POWER or VSE job name, or have changed the name of the program, you may want to change these statements.
- 10 This program specifies the COBOL/VSE compiler option TEST(ALL,SYM). This option triples the size of the .PHASE MEMBERTYPE that is created when this program is link edited. If the program cancels, this option may provide enough information to determine the exact reason why.
- 11 This program contains the following MOVE statement as one of the first statements in the procedure division:

```
MOVE 'IGZSORT' TO SORT-CONTROL.
```

As far as is known, the SORT-CONTROL special register is needed only if DFSORT is used. It is not needed if you are running CA-SORT 8.1. However, it is not known whether it is needed for CA-SORT 8.2 or SYNC-SORT (any VSE/ESA version). You should read up on the new sort special registers, so that you know what this statement does. In any case, if the statement is not removed or commented, it will cause the following error message, irrespective of whether or not it is used, unless you already have IGZSORT.C in a LIBDEFed library when the program is executed.

```
IGZ0027W THE SORT CONTROL FILE COULD NOT BE OPENED.
```

You can eliminate the error message by removing or uncommenting the 'MOVE' statement (if not using DFSORT), or leaving the statement as it is and executing the following job:

```
* $$ JOB JNM=IGZSORT,CLASS=0,DISP=D,PRI=8
* $$ LST DISP=H,PRI=8,CLASS=0
// JOB IGZSORT CATALOG SORT-CONTROL RECORD.
// EXEC DTRIATTN,PARM='L LST,IGZSORT'
// EXEC LIBR,SIZE=256K,PARM='ACC S=PRD2.SCEEBASE'
CATALOG IGZSORT.C REPL=YES
  OPTION PRINT=CRITICAL,ROUTE=LOG
/+
LIS IGZSORT.C
```



```
/*  
/&  
* $$ E0J
```

- 12 This program sets the VSE return code to eight on initial execution, and to zero immediately before any 'GOBACK' statements. This allows the \$SRC value to be checked by placing a '// IF \$SRC > 7 THEN' statement immediately after the // EXEC DPCSNT ... statement.
- 13 There are a number of display statements in this program that contain the character 'D' in column 7. These can be used to display information that may be helpful if you think the program is not functioning correctly. Invoking these display statements will cause a vast amount of output to be written to SYSLST; in addition to displaying the display statements, COBOL/VSE will also display the procedure division paragraph names invoked throughout the program's execution. To invoke these display statements, simply uncomment the 'SOURCE-COMPUTER. IBM-390 WITH DEBUGGING MODE.' statement, change the PROCESS statement from TEST(ALL,SYM) to TEST(NONE), and compile the program. Then add/modify the PARM operand on the EXEC statement, as in the following example:

```
// EXEC DPCSNT,SIZE=(DPCSNT,128K),PARM='.../DEBUG'
```

The above example assumes that the LE/VSE run-time option CBLOPTS=((ON),OVR) has been specified. Despite what the documentation states, this option cannot be overridden at execution time. If your CBLOPTS was specified as 'OFF', then your execute statement must appear as follows:

```
// EXEC DPCSNT,SIZE=(DPCSNT,128K),PARM='DEBUG/...'
```

- 14 If you are running multiple CICSs in the same machine, it is highly recommended that you have only one DFHCSD file. First, this simplifies maintenance. And second, and more importantly, because access to a single DFHCSD is possible from any up and running CICS, additions/changes can be made even when other CICSs aren't running. This would not be possible if each CICS had its own DFHCSD file. Then, if an error in a DFHCSD entry

prevented a CICS from starting, you wouldn't be able to fix the problem from another CICS.

The following subroutines are called, dynamically, by this program:

- DPBITX
- DPSORT
- DPSRCH
- DPHTOC
- DPPOWP (optional).

The first three of these were published in the September 1998 issue of *VSE Update*. The fourth will be published in a future issue, at the end of this program. The fifth was published in September 1997.

#### DPCSNT.CBL

```
* $$ JOB JNM=DPCSNT,CLASS=0,DISP=D,PRI=8,USER=*BOTSIS*
* $$ LST DISP=H,CLASS=0,PRI=8
// JOB DPCSNT MS00 COBOL/VSE BATCH 24-BIT. TM
// EXEC DTRIATTN,PARM='L LST,DPCSNT'
* $$ SLI ICCF=(DPVLBL),LIB=(02)
// LIBDEF PHASE,SEARCH=(USR1.IBM,PRD2.SCEEBASE,PRD2.PROD)
// LIBDEF OBJ,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,USR1.TECH)
// LIBDEF PHASE,CATALOG=USR1.DFHSNT
// OPTION ERRS,SXREF,SYM,NODECK,CATAL
        PHASE DPCSNT,*
        INCLUDE CEEUOPT
// EXEC IGYCRCTL,SIZE=IGYCRCTL
PROCESS ADV,APOST,AWO,BUFSIZE(32760),NOCMPR2,NOCMPPILE(E)
PROCESS NOCURRENCY,DATA(24),NODBCS,NODECK,NODUMP,NODYNAM
PROCESS FLAG(I,E),NOFLAGMIG,NOFLAGSA,NOFLAGSTD,LANGUAGE(UE)
PROCESS LIB,LINCOUNT(60),NOLIST,MAP,ONAME,ONUMBER
PROCESS NUMPROC(MIG),OBJECT,OFFSET,NOOPTIMIZE,OUTDD(SYSOUT)
PROCESS RENT,NOSEQUENCE,SIZE(MAX),SOURCE,SPACE(1),SSRANGE
PROCESS NOTERM,TEST(ALL,SYM),TRUNC(STD),NOVBREF,NOWORD
PROCESS XREF(FULL),ZWB
PROCESS NOFASTSRT
*PROCESS OPTIMIZE TEST(NONE)
ID DIVISION.
PROGRAM-ID.        DPCSNT.
DATE-WRITTEN.     06/24/1991.
DATE-COMPILED.
AUTHOR.          R. BOTSIS.
```

ENVIRONMENT DIVISION.  
 CONFIGURATION SECTION.  
 \*SOURCE-COMPUTER. IBM-390 WITH DEBUGGING MODE.  
 OBJECT-COMPUTER. IBM-390.  
 INPUT-OUTPUT SECTION.  
 FILE-CONTROL.  
   SELECT CARDIN ASSIGN SYS007-S-CARDIN.  
   SELECT CARDOT ASSIGN SYS008-S-CARDOT  
     FILE STATUS CO-STATUS.  
   SELECT NAME-TABLE ASSIGN DPNKTAB  
     ORGANIZATION INDEXED  
     ACCESS RANDOM  
     RECORD KEY NTAB-KEY  
     FILE STATUS IS NTAB-STATUS NTAB-VSAM-STATUS.  
   SELECT IESCNTL ASSIGN IESCNTL  
     ORGANIZATION INDEXED  
     ACCESS SEQUENTIAL  
     RECORD KEY IE-KEY  
     FILE STATUS IE-STATUS IE-VSAM-STATUS.  
   SELECT DFHCSD ASSIGN DFHCSD  
     ORGANIZATION INDEXED  
     ACCESS SEQUENTIAL  
     RECORD KEY DF-KEY  
     FILE STATUS DF-STATUS DF-VSAM-STATUS.  
   SELECT OUTFILE ASSIGN OUTFILE  
     FILE STATUS OT-STATUS.  
   SELECT CPYFILE ASSIGN CPYFILE  
     FILE STATUS CP-STATUS.  
   SELECT SORT-FILE ASSIGN SORTWK1.  
   SELECT REPOUT ASSIGN SYS009-S-REPOUT.  
 DATA DIVISION.  
 FILE SECTION.  
 \* CARD FILE IN (PARAMETERS/JCL).  
   FD CARDIN  
     RECORDING MODE F.  
   01 CARD-REC                   PIC X(128).  
 \* CARD FILE OUT.  
   FD CARDOT  
     RECORDING MODE F.  
   01 CARD-OUT                   PIC X(80).  
 \* NAME-TABLE FILE.  
   FD NAME-TABLE.  
   01 NTAB-RECORD.  
 \*                                   01-08.  
   03 NTAB-KEY                   PIC X(8).  
 \*                                   09-10.  
   03 FILLER                     PIC XX.  
 \*                                   11-13.

```

    03 NTAB-OPID          PIC XXX.
*                               14-19.
    03 FILLER             PIC X(6).
    03 NTAB-USER-DATA.
*                               20-39.
    05 NTAB-USER-DEPT-LOC PIC X(20).
    05 NTAB-USER-NAME.
*                               40-59.
    07 NTAB-USER-FST-NAME PIC X(20).
*                               60-79.
    07 NTAB-USER-LST-NAME PIC X(20).
*                               80-80.
    03 NTAB-FLAG         PIC X.
* IESCNTL FILE.
  FD IESCNTL
    RECORD VARYING FROM 1 TO 1000 CHARACTERS
      DEPENDING ON IE-REC-LEN.
* IESCNTL FILE RECORD LAYOUT.
  01 IE-RECORD.
*                               0001-0012.
    03 IE-KEY.
    05 IE-TYPE           PIC XX.
    05 IE-USERID        PIC X(8).
    05 FILLER           PIC XX.
    03 FILLER REDEFINES IE-KEY.
    05 FILLER           PIC XX.
    05 IE-USERID4       PIC XXXX.
    05 IE-USERID-5-8   PIC XXXX.
    05 FILLER           PIC XX.
*                               0013-0014.
    03 IE-VER           PIC XX.
*                               0015-0016.
    03 IE-RSVD-1       PIC XX.
*                               0017-0018.
    03 IE-TAB-LENGTH   PIC S999 COMP.
*                               0019-0019.
    03 IE-IND-BYTE     PIC X.
*                               0020-0020.
    03 IE-SIGN-OFF-TIME PIC X.
*                               0021-0026.
    03 IE-PASS-1-6     PIC X(6).
*                               0027-0028.
    03 IE-PASS-7-8     PIC XX.
*                               0029-0032.
    03 IE-SEC-BLK-ADR  PIC S99999 COMP.
*                               0033-0040.
    03 IE-SCTY-64-01.
    05 IE-SCTY-64-57  PIC X.

```

```

    05 IE-SCTY-56-49 PIC X.
    05 IE-SCTY-48-41 PIC X.
    05 IE-SCTY-40-33 PIC X.
    05 IE-SCTY-32-25 PIC X.
    05 IE-SCTY-24-17 PIC X.
    05 IE-SCTY-16-09 PIC X.
    05 IE-SCTY-08-01 PIC X.
*
    0041-0043.
    03 IE-RSL-KEY.
    05 IE-RSL-KEYS-24-17 PIC X.
    05 IE-RSL-KEYS-16-09 PIC X.
    05 IE-RSL-KEYS-08-01 PIC X.
*
    0044-0044.
    03 IE-OPR-PRI PIC X.
*
    0045-0047.
    03 IE-OPR-CLASS.
    05 IE-OPR-CLASS-KEYS-24-17 PIC X.
    05 IE-OPR-CLASS-KEYS-16-09 PIC X.
    05 IE-OPR-CLASS-KEYS-08-01 PIC X.
*
    0048-0050.
    03 IE-OPERID PIC XXX.
    03 IE-RSVD-2 PIC X.
    03 IE-USER-ID-LENGTH PIC X.
*
    0053-0060.
    03 IE-USER-ID PIC X(8).
*
    0061-0062.
    03 IE-USE-COUNT PIC S999 COMP.
*
    0063-0063.
    03 IE-OPR-NAME-LENGTH PIC X.
*
    0064-0083.
    03 IE-OPR-NAME PIC X(20).
    03 FILLER PIC X.
*
    0085-0088.
    03 IE-RSVD-3 PIC XXXX.
*
    0089-0100.
    03 IE-ICCF-OPTIONS PIC X(12).
*
    0101-0102.
    03 IE-RSVD-4 PIC XX.
*
    0103-0111.
    03 IE-PANEL-STUFF.
    05 FILLER PIC X(9).
* X'00E1' = POST ADMINISTRATOR (POST). (IE. DECIMAL 225).
* X'00D9' = DIAL UP ADMINISTRATOR. (IE. DECIMAL 217).
* X'00D1' = ADMINISTRATOR. (IE. DECIMAL 209).
* X'0081' = PROGRAMMER WITH ICCF. (IE. DECIMAL 129).
* X'0071' = PROGRAMMER WITH ICCF (IE. DECIMAL 113).
* X'0061' = GENERAL USER WITH ICCF (IE. DECIMAL 97).
* X'0051' = PROGRAMMER WITH ICCF. (IE. DECIMAL 81).
* X'0021' = GENERAL USER WITHOUT ICCF. (IE. DECIMAL 33).

```

```

* X'0011' = GENERAL USER WITHOUT ICCF. (IE. DECIMAL 17).
*
      0112-0131.
      05 IE-PANEL-USER-TYPE PIC S999 COMP.
      05 FILLER          PIC XXX.
      05 IE-PANEL-NAME  PIC X(8).
      05 IE-PANEL-USRCAT PIC X(7).
*
      0132-0135.
      05 IE-IDUNNO      PIC X(4).
*
      0136-0138.
      03 IE-PASSWRD-CNTRL-INFO PIC S9(5) COMP-3.
*
      0139-0140.
      03 IE-PASSWRD-DAYS-TO-EXP PIC 999 COMP-3.
*
      0141-0148.
      03 IE-GREGORIAN-EXP  PIC X(8).
*
      0149-0156.
      03 IE-MODEL-USER-ID-SYNOM PIC X(8).
*
      0157-0202.
      03 IE-RSVD-5        PIC X(46).
*
      0203-0298.
      03 IE-OLD-PASSWORDS.
      05 IE-OLD-PASSWORDS-ENTRY PIC X(8) OCCURS 12.
* DFHCSD FILE.
FD DFHCSD
  RECORD VARYING FROM 1 TO 500 CHARACTERS.
* DFHCSD FILE RECORD LAYOUT.
01 DF-RECORD.
  03 DF-KEY.
    05 DF-NAME.
      07 FILLER          PIC XXX.
      07 DF-NAME-DOLLAR PIC X.
      07 FILLER          PIC XXXX.
    05 DF-LOW-VALUE     PIC X(4).
    05 DF-REC-TYPE      PIC S99 COMP.
    05 DF-REC-TYPE-NAME PIC X(8).
  03 DF-RECORD-LENG    PIC S999 COMP.
  03 DF-TRANID-INFO    PIC X(44).
  03 DF-TRANID-TYPE    PIC X.
  03 DF-TRANID-LENG    PIC X.
  03 DF-TRANID.
    05 DF-TRANID-DATA   PIC X OCCURS 430.
FD CPYFILE
  RECORD VARYING FROM 1 TO 298 CHARACTERS
  RECORDING MODE V
  LABEL RECORD STANDARD.
01 CPY-REC              PIC X(298).
* OUT-REC 'MUST BE' THE SAME SIZE AS SORT-RECORD.
FD OUTFILE
  RECORDING MODE F

```

```

BLOCK CONTAINS 20 RECORDS
LABEL RECORD STANDARD.
01 OUT-REC          PIC X(1193).
* SORT-RECORD 'MUST BE' THE SAME SIZE AS OUT-REC.
SD SORT-FILE.
01 SORT-RECORD.
*
          0001-0002.
03 SORT-TRANSEC-VALUE PIC XX.
03 SORT-TRANSEC-VALUE-N REDEFINES SORT-TRANSEC-VALUE PIC 99.
*
          0003-0003.
03 SORT-ID          PIC X.
   88 MAP-REC      VALUE 'M'.
   88 PROG-REC     VALUE 'P'.
   88 USER-REC    VALUE 'U'.
   88 TRAN-REC    VALUE 'V'.
*
          0004-0011.
03 SORT-PROGID     PIC X(8).
03 SORT-USERID REDEFINES SORT-PROGID PIC X(8).
03 SORT-TRANID REDEFINES SORT-PROGID PIC X(8).
*
          0012-0014.
03 SORT-OPERID     PIC XXX.
*
          0015-0022.
03 SORT-GRPLIST    PIC X(8).
*
          0023-0030.
03 SORT-PROG-NAME  PIC X(8).
*
          0031-0034.
03 SORT-TRAN-PFKEY PIC X(4).
*
          0035-0038.
03 SORT-REMOTE-SYSID PIC X(4).
*
          0039-0046.
03 SORT-DF-NAME    PIC X(8).
*
          0047-0054.
03 SORT-VSE-NAME   PIC X(8).
*
          0055-0056.
03 FILLER          PIC XX.
*
          0057-0057.
03 SORT-RSL-BYTE   PIC X.
*
          0058-0065.
03 SORT-PASSWORD   PIC X(8).
*
          0066-0068.
03 SORT-OP-PRTY    PIC XXX.
*
          0069-0116.
03 SORT-OP-CLASS-KEYS.
   05 SORT-OP-CLASS-KEY PIC XX OCCURS 24.
*
          0117-0164.
03 SORT-RSL-KEYS.
   05 SORT-RSL-KEY     PIC XX OCCURS 24.
*
          0165-0166.

```

```

      03 FILLER PIC XX.
*      03 SORT-TABLE-DESC PIC X(25).
*      03 FILLER PIC XX.
*      03 SORT-IE-RECORD PIC X(1000).
FD REPOUT
  RECORDING MODE F
  RECORD CONTAINS 132 CHARACTERS.
01 REP-OUT.
  03 REP-OUT-1-132 PIC X(132).
WORKING-STORAGE SECTION.
77 PIC X(14) VALUE 'STORAGE HERE. '.
77 TRACE-ON PIC 9 VALUE 0.
77 SORT-RTRN-CODE PIC 9(4) VALUE ZERO.
77 IE-REC-LEN PIC 9999 VALUE 0.
77 SAVE-TALLY PIC 99999 VALUE ZERO COMP.
77 LINE-COUNT PIC 9999 VALUE ZERO.
77 LINE-COUNT-MAX PIC 9999 VALUE 58.
77 PARM-SW PIC X VALUE '0'.
77 PARM-EOF-SW PIC X VALUE '0'.
77 PARM-DONE-SW PIC X VALUE '0'.
77 NTAB-OPEN-SW PIC X VALUE '0'.
77 COUT-OPEN-SW PIC X VALUE '0'.
77 DF-SW PIC X VALUE '0'.
77 IN-SW PIC X VALUE '0'.
77 EX-SW PIC X VALUE '0'.
77 CHAR-MAX PIC 999 VALUE 117.
77 OPPRI PIC 999.
77 LIB-ALTS-80 PIC X(80) VALUE SPACE.
77 LIB-ALTS-LENG PIC S999 VALUE +5 COMP.
77 LIB-ALTS-1-8 PIC ZZZZ.
77 HALF-WORD-46 PIC S999 VALUE +46 COMP.
77 SEC-TABLE-DESC-LENG PIC S999 VALUE +27 COMP.
77 DFRD-COUNT PIC 9(5) VALUE 0.
77 DFTR-COUNT PIC 9(5) VALUE 0.
77 DFPG-COUNT PIC 9(5) VALUE 0.
77 DFMP-COUNT PIC 9(5) VALUE 0.
77 DFRM-COUNT PIC 9(5) VALUE 0.
77 DFPR-COUNT PIC 9(5) VALUE 0.
77 DFDE-COUNT PIC 9(5) VALUE 0.
77 MAP-RECORD PIC S9(5) COMP VALUE +1000.
77 PROG-RECORD PIC S9(5) COMP VALUE +4000.
77 TRAN-RECORD PIC S9(5) COMP VALUE +5000.
77 SAVE-BYTE-1 PIC X.
77 WK-LENG1 PIC S99 COMP.
77 SUB-1 PIC S99 COMP.
77 HEADING-SW PIC X VALUE '1'.

```



77	THRE-SW	PIC X VALUE '0'.
77	FIVE-SW	PIC X VALUE '0'.
77	NINE-SW	PIC X VALUE '0'.
77	FIRST-SW	PIC X VALUE '0'.
77	ONCE-SW	PIC X VALUE '0'.
77	TOTL-SW	PIC X VALUE '0'.
77	PWR-NAME	PIC X(8) VALUE 'DPCSNTJ'.
77		PIC X VALUE SPACE.
77	VSE-NAME	PIC X(8) VALUE 'DPCSNTJ'.
77	SUB-NAME	PIC X(8).
77	IE-EOF	PIC X VALUE '0'.
77	DF-EOF	PIC X VALUE '0'.
77	NTAB-STATUS	PIC XX VALUE ' '.
77	IE-STATUS	PIC XX VALUE ' '.
77	DF-STATUS	PIC XX VALUE ' '.
77	OT-STATUS	PIC XX VALUE ' '.
77	CO-STATUS	PIC XX VALUE ' '.
77	CP-STATUS	PIC XX VALUE ' '.
77	INDX-COUNT	PIC 999 VALUE 0.
77	INDX-COUNT1	PIC 99 VALUE 0 COMP.
77	SAVE-INDX-COUNT	PIC 999 VALUE 0.
77	SUBM-COUNT	PIC 99 VALUE 0 COMP.
77	INDXG-COUNT	PIC 99 VALUE 1.
77	INDXG1-COUNT	PIC 99 VALUE 1.
77	INDXG2-COUNT	PIC 99 VALUE 1.
77	INDXG3-COUNT	PIC 99 VALUE 1.
77	INDXG4-COUNT	PIC 99 VALUE 1.
77	INDXY-COUNT	PIC 999 VALUE 1.
77	INDXY1-COUNT	PIC 999 VALUE 1.
77	INDXY2-COUNT	PIC 999 VALUE 1.
77	INDXY3-COUNT	PIC 999 VALUE 1.
77	INDXY4-COUNT	PIC 999 VALUE 1.
77	TABL-COUNT	PIC 99 VALUE 0.
77	SECT-COUNT	PIC 99 VALUE 0.
77	GRPL-COUNT	PIC 99 VALUE 0.
77	GRUP-COUNT	PIC 999 VALUE 0.
77	PARM-COUNT	PIC 999 VALUE 0.
77	PAGE-COUNT	PIC 9999 VALUE 0.
77	CNTL-COUNT	PIC 9(5) VALUE 0.
77	SAVE-SORT-ID	PIC X.
77	SAVE-SORT-TRANSEC-VALUE	PIC 99.
77	SAVE-SORT-USERID	PIC X(8).
77	SAVE-WK-BYTE	PIC 9999.
77	SAVE-ASNT-Y-OR-N	PIC X VALUE 'Y'.
77	SAVE-DUPS-Y-OR-N	PIC X VALUE 'N'.
77	SAVE-FULL-Y-OR-N	PIC X VALUE 'N'.
77	SAVE-NTAB-Y-OR-N	PIC X VALUE 'N'.
77	SAVE-OPNM-Y-OR-N	PIC X VALUE 'N'.
77	SAVE-PASS-Y-OR-N	PIC X VALUE 'N'.

```

77 SAVE-POWP-Y-OR-N          PIC X VALUE 'N'.
77 SAVE-REMT-Y-OR-N          PIC X VALUE 'N'.
77 SAVE-RPTS-1-OR-2-OR-B-OR-N PIC X VALUE 'B'.
77 SAVE-USID-F-OR-4-OR-8     PIC X VALUE '4'.
77 SAVE-TRAN-4-OR-8         PIC X VALUE '4'.
77 HEX-LENG                  PIC 9(4) COMP VALUE ZERO.
Ø1                             PIC X(5) VALUE 'HERE.'.
Ø1 PWR-INFO.
  Ø3 PWR-INFO-NAME           PIC X(8).
  Ø3                          PIC X(28) VALUE SPACE.
Ø1 SUB-2                      PIC S99 COMP.
Ø1 FILLER REDEFINES SUB-2.
  Ø3 FILLER                  PIC X.
  Ø3 SUB-2-R                 PIC X.
Ø1 CURR-DATE.
  Ø3 CURR-DATE-MM           PIC XX.
  Ø3 CURR-DATE-S1          PIC X.
  Ø3 CURR-DATE-DD          PIC XX.
  Ø3 CURR-DATE-S2          PIC X.
  Ø3 CURR-DATE-CC          PIC XX.
  Ø3 CURR-DATE-YY          PIC XX.
Ø1 NTAB-VSAM-STATUS.
  Ø3 NTAB-VSAM-STATUS-R15   PIC 99 COMP.
  Ø3 NTAB-VSAM-STATUS-FUN   PIC 9 COMP.
  Ø3 NTAB-VSAM-STATUS-FBK   PIC 999 COMP.
Ø1 IE-VSAM-STATUS.
  Ø3 IE-VSAM-STATUS-R15     PIC 99 COMP.
  Ø3 IE-VSAM-STATUS-FUN     PIC 9 COMP.
  Ø3 IE-VSAM-STATUS-FBK     PIC 999 COMP.
Ø1 DF-VSAM-STATUS.
  Ø3 DF-VSAM-STATUS-R15     PIC 99 COMP.
  Ø3 DF-VSAM-STATUS-FUN     PIC 9 COMP.
  Ø3 DF-VSAM-STATUS-FBK     PIC 999 COMP.
Ø1 VSAM-STATUS.
  Ø3 VSAM-R15               PIC 99.
  Ø3 FILLER                  PIC X VALUE '-'.
  Ø3 VSAM-FUN               PIC 9.
  Ø3 FILLER                  PIC X VALUE '-'.
  Ø3 VSAM-FBK               PIC 999.
Ø1 SAVE-DF-KEY.
  Ø3 SAVE-DF-NAME.
    Ø5 FILLER                PIC XXX.
    Ø5 SAVE-DF-NAME-DOLLAR   PIC X.
    Ø5 FILLER                PIC XXXX.
  Ø3 SAVE-DF-LOW-VALUE       PIC X(4).
  Ø3 SAVE-DF-REC-TYPE        PIC S99 COMP.
  Ø3 SAVE-DF-REC-TYPE-NAME   PIC X(8).
Ø1 HEX-CHRT.
  Ø3 HEX-CHRE                PIC X OCCURS 4Ø DEPENDING HEX-LENG.

```

```

Ø1 HEX-Ø1 PIC S999 COMP VALUE +1.
Ø1 FILLER REDEFINES HEX-Ø1.
    Ø3 FILLER PIC X.
    Ø3 HEX-Ø1-R PIC X.
Ø1 HEX-Ø2 PIC S999 COMP VALUE +2.
Ø1 FILLER REDEFINES HEX-Ø2.
    Ø3 FILLER PIC X.
    Ø3 HEX-Ø2-R PIC X.
Ø1 HEX-Ø5 PIC S999 COMP VALUE +5.
Ø1 FILLER REDEFINES HEX-Ø5.
    Ø3 FILLER PIC X.
    Ø3 HEX-Ø5-R PIC X.
Ø1 HEX-Ø6 PIC S999 COMP VALUE +6.
Ø1 FILLER REDEFINES HEX-Ø6.
    Ø3 FILLER PIC X.
    Ø3 HEX-Ø6-R PIC X.
Ø1 HEX-Ø7 PIC S999 COMP VALUE +7.
Ø1 FILLER REDEFINES HEX-Ø7.
    Ø3 FILLER PIC X.
    Ø3 HEX-Ø7-R PIC X.
Ø1 HEX-Ø9 PIC S999 COMP VALUE +9.
Ø1 FILLER REDEFINES HEX-Ø9.
    Ø3 FILLER PIC X.
    Ø3 HEX-Ø9-R PIC X.
Ø1 HEX-11 PIC S999 COMP VALUE +17.
Ø1 FILLER REDEFINES HEX-11.
    Ø3 FILLER PIC X.
    Ø3 HEX-11-R PIC X.
Ø1 HEX-7F PIC S999 COMP VALUE +127.
Ø1 FILLER REDEFINES HEX-7F.
    Ø3 FILLER PIC X.
    Ø3 HEX-7F-R PIC X.
Ø1 HEX-FE PIC S999 COMP VALUE +254.
Ø1 FILLER REDEFINES HEX-FE.
    Ø3 FILLER PIC X.
    Ø3 HEX-FE-R PIC X.
Ø1 SNT-TIMEOUT.
    Ø3 FILLER PIC X(8) VALUE 'TIMEOUT='.
    Ø3 SNT-TIMEOUT-VALUE PIC XX.
    Ø3 FILLER PIC X VALUE ', '.
Ø1 WK-BYTES.
    Ø3 WK-BYTE-L PIC X.
    Ø3 WK-BYTE-R PIC X.
Ø1 WK-BYTE REDEFINES WK-BYTES PIC S99 COMP.
Ø1 WK-BYTES-2.
    Ø3 WK-BYTE-2-L PIC X.
    Ø3 WK-BYTE-2-R PIC X.
Ø1 WK-BYTE-2 REDEFINES WK-BYTES-2 PIC S99 COMP.
Ø1 WK-BYTES-3.
    Ø3 WK-BYTE-3-L PIC X.

```

```

    Ø3 WK-BYTE-3-R          PIC XXX.
Ø1 WK-BYTE-3 REDEFINES WK-BYTES-3 PIC S99999 COMP.
Ø1 WK-PASS1                PIC XX.
Ø1 FILLER REDEFINES WK-PASS1.
    Ø3 WK-PASS11          PIC X.
    Ø3 WK-PASS12          PIC X.
Ø1 FILLER REDEFINES WK-PASS1.
    Ø3 WK-BIN1           PIC S9999 COMP.
Ø1 WK-PASS2                PIC XX.
Ø1 FILLER REDEFINES WK-PASS2.
    Ø3 WK-PASS21          PIC X.
    Ø3 WK-PASS22          PIC X.
Ø1 FILLER REDEFINES WK-PASS2.
    Ø3 WK-BIN2           PIC S9999 COMP.
Ø1 WK-IN-PASS              PIC X(6).
Ø1 FILLER REDEFINES WK-IN-PASS.
    Ø3 WK-IN-PASS-CHAR    PIC X OCCURS 6.
Ø1 RSL-TABLE.
    Ø3 RSL-TABLE-ENTRY    PIC XX OCCURS 64.
Ø1 OP-CLASS-TABLE.
    Ø3 OP-CLASS-TABLE-ENTRY PIC XX OCCURS 64.
Ø1 GRPLIST-NAME.
    Ø3 GRPLIST-NAME-ENTRY PIC X OCCURS 8.
Ø1 FILLER                  PIC X(80) VALUE HIGH-VALUE.
Ø1 GRPLIST-TABLE.
    Ø3 GRPLIST-TABLE-ENTRY PIC X(8) OCCURS 7.
Ø1 FILLER                  PIC X(8) VALUE HIGH-VALUE.
Ø1 GROUP-NAME.
    Ø3 GROUP-NAME-ENTRY   PIC X OCCURS 8.
Ø1 FILLER                  PIC X(80) VALUE HIGH-VALUE.
Ø1 GROUP-NAME-TABLE.
    Ø3 GROUP-NAME-TABLE-ENTRY PIC X(8) OCCURS 99.
Ø1 FILLER                  PIC X(8) VALUE HIGH-VALUE.
Ø1 SECURITY-TABLE.
    Ø3 SECURITY-TABLE-DESC OCCURS 64.
        Ø5 SEC-TABLE-DESC PIC X(25).
        Ø5 SEC-TABLE      PIC 99.
Ø1 SAVE-SECURITY-TABLE.
    Ø3 SAVE-SECURITY-TABLE-DESC OCCURS 64.
        Ø5 SAVE-SEC-TABLE-DESC PIC X(25).
        Ø5 SAVE-SEC-TABLE      PIC 99.
Ø1 FILLER                  PIC X(27) VALUE HIGH-VALUE.
Ø1 USER-SECURITY-TABLE.
    Ø3 USER-SEC-TABLE      PIC XX OCCURS 64.
Ø1 TOTL-SECURITY-TABLE.
    Ø3 TOTL-SEC-TABLE      PIC S9(5) COMP OCCURS 64.
Ø1 PARM-IN.
    Ø3 PARM-IN-1-1         PIC X.
    Ø3 PARM-REC.
        Ø5 FILLER          PIC X(70).

```

```

        Ø5 PARM-REC-CONT    PIC X.
        Ø5 PARM-REC-73-8Ø  PIC X(8).
Ø3 FILLER                    PIC X(48).
Ø1 FILLER REDEFINES PARM-IN.
Ø3 PARM-IN-1-5              PIC X(5).
Ø3 PARM-REC-SEC-1.
    Ø5 PARM-REC-SEC-NUM-1 PIC 99.
    Ø5 PARM-REC-SEC-NUM-1-A REDEFINES
        PARM-REC-SEC-NUM-1 PIC XX.
    Ø5 PARM-REC-SEC-DESC-1 PIC X(25).
Ø3 PARM-REC-SEC-2.
    Ø5 PARM-REC-SEC-NUM-2 PIC 99.
    Ø5 PARM-REC-SEC-NUM-2-A REDEFINES
        PARM-REC-SEC-NUM-2 PIC XX.
    Ø5 PARM-REC-SEC-DESC-2 PIC X(25).
Ø3 FILLER                    PIC X(21).
Ø1 FILLER REDEFINES PARM-IN.
Ø3 FILLER                    PIC X(5).
Ø3 PARM-REC-GRP              PIC X OCCURS 62.
Ø3 FILLER                    PIC X(13).
Ø1 SRCH-PARM1.
Ø3 SRCH-PARM1-DIR          PIC X VALUE 'F'.
Ø3 SRCH-PARM1-REL          PIC X VALUE 'E'.
Ø1 SRCH-PARM2.
Ø3 SRCH-PARM2-SRC-LQUOTE PIC X VALUE QUOTE.
Ø3 SRCH-PARM2-SRC          PIC X(8) VALUE SPACE.
Ø3 SRCH-PARM2-SRC-RQUOTE PIC X VALUE QUOTE.
Ø3 FILLER                    PIC X(6Ø) VALUE SPACE.
Ø1 SRCH-PARM3.
Ø3 SRCH-PARM3-TAR          PIC X(8Ø) VALUE SPACE.
Ø1 SRCH-PARM4.
Ø3 SRCH-PARM4-RTC          PIC X VALUE SPACE.
Ø1 SRCH-PARM5.
Ø3 SRCH-PARM5-STR          PIC S99 COMP VALUE +1.
Ø3 SRCH-PARM5-END          PIC S99 COMP VALUE +8Ø.
Ø3 SRCH-PARM5-NUM          PIC 99.
Ø1 BITS-1-8.
Ø3 BITS-8                  PIC X.
Ø3 BITS-7                  PIC X.
Ø3 BITS-6                  PIC X.
Ø3 BITS-5                  PIC X.
Ø3 BITS-4                  PIC X.
Ø3 BITS-3                  PIC X.
Ø3 BITS-2                  PIC X.
Ø3 BITS-1                  PIC X.

```

*Editor's note: This article will be continued in the next issue.*

---

*Robert Botsis*  
*Senior Systems Programmer (USA)*

© Xephon 1998

---

## VSE/ESA 2.4 announcement review

VSE/ESA Version 2 Release 4, announced on 29 September, offers a number of enhancements, including CICS Transaction Server for VSE/ESA, new security options, and IXFP/SnapShot for VSE. It is Year 2000-ready, supports the Euro currency symbol, and offers VisualAge 2000 tools such as IBM Opti-Audit and VisualAge COBOL MLE.

### CICS TRANSACTION SERVER FOR VSE/ESA

The CICS Transaction Server for VSE/ESA (CICS TS for VSE/ESA) provides increased reliability and availability, with:

- Storage protection
- Virtual Storage Constraint Relief
- Shared data tables
- External call interface
- Front-end programming interface
- Resource definition on-line for files
- Network computing and e-business capabilities.

It comprises the following components, delivered as a single product:

- CICS Transaction Server Base – the new version of CICS that is based on CICS for MVS/ESA Version 4.1.
- CICS/VSE Version 2.3 – the coexistence release to aid in migration.
- Report Controller – previously a priced feature of VSE/ESA, which will work with both CICS/VSE Version 2.3 and with CICS TS for VSE/ESA.
- CICS Universal Clients Version 3 – an enhanced replacement for CICS clients, which runs on all popular desktops and provides client access to CICS servers.
- CICS Transaction Gateway Version 3 – which provides Web

access to CICS servers and replaces both the CICS Internet gateway and the CICS gateway for Java.

## SECURITY OPTIONS

The new security options are as follows:

- There is a new Basic Security Manager for CICS sign-on and transaction ID protection.
- CA-TopSecret for VSE/ESA is an optional product shipped with V2.4.

## E-BUSINESS

If you want to use your VSE system as a server for browser-based Internet/intranet solutions:

- TCP/IP for VSE/ESA enables open connectivity and supports interoperability via the 'socket' programming interfaces. It offers common TCP/IP applications such as file transfer (FTP client and server), terminal emulation (Telnet client and TN3270 server), print distribution and serving (LPR/LPD), basic Web serving, and VSE file serving (NFS).
- CICS TS VSE/ESA will contain the CICS Web interface and 3270 bridge, allowing Web access to CICS applications, services, and data.

## OTHER ENHANCEMENTS

Version 2.4 includes new releases of high-level Assembler, the HLASM toolkit, and DITTO/ESA. New VSE optional products include MQSeries Version 2 and DL/I VSE Version 1.11. Note that BTAM/ES is now an optional product instead of being in the base system.

Version 2.4 supports the whole range of System/390 servers, as well as RVA SnapShot for high-speed copies. (IXFP/SnapShot for VSE/ESA will also be available for Version 2.3.1 in December.)

---

© Xephon 1998

## Restart prevention revisited

The June 1992 issue of *VSE Update* contained an article entitled 'Preventing automatic restart of in-flight jobs' by Frank A Braman. Although the method presented in the article gives you full control, you have to insert a statement like

```
// EXEC PROC=$$MODE
```

into each job.

In the March 1994 issue, Kevin E Crandall described an alternative method, which involved inserting the NORUN=YES parameter in the VSE/POWER start-up procedure. With this parameter, all jobs in-flight during a system crash are altered by VSE/POWER to a disposition of X when the system comes up again. This solution is attractive because it is very easy to implement. However, it does have a drawback in that jobs stacked behind a failing in-flight job will start automatically.

We therefore tried to find a third solution, which, like the first one, would give us full control, but, unlike it, would not require each job to be changed. At the end of the IPL procedure \$1JCL (F1 is the VSE/POWER partition), we inserted a statement for each batch partition F4, F5, ..., BG that sets the corresponding environment variables XSTATpid (pid replaced by the partition ID) of the SETPARM procedure CPUVARn to INACTIVE.

```
CATALOG $1JCL.PROC DATA=YES REPLACE=YES
// JOB POWSTART
...
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTAT&XPARTPW=INACTIVE'
/*

// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTATF4=INACTIVE'
/*
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTATF5=INACTIVE'
/*
...
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;SET XSTATBG=INACTIVE'
/*
/&
/+
```



With this solution, when the system shuts down and VSE/POWER is terminated normally, the environment variables XSTATpid of the batch partitions are set to INACTIVE. If this value is not found at IPL time, the batch partition executes a PAUSE statement, and jobs can be manually altered or deleted from the reader queue. The environment variable is then set to ACTIVE.

Because of this checking and setting, the IPL procedures \$\$JCL of the batch partitions now look like those shown below (partition F4) :

```
CATALOG $4JCL.PROC DATA=YES REPLACE=YES
// JOB F4START
- LIBDEF and ASSGN statements -
// SETPARM XNCPU=,XSTATF4=
// EXEC PROC=$COMVAR,XNCPU
// EXEC PROC=CPUVAR&XNCPU,XSTATF4
// IF XSTATF4 NE 'INACTIVE' THEN
// PAUSE RECOVERY MODE - CHECK JOBS READY TO RUN
// EXEC DTRSETP,PARM='CPUVAR&XNCPU;;;SET XSTAT&PID=ACTIVE'
/*
/+
```

Note that jobs starting subsystems like POWER, CICS, and VTAM set the environment variables of their partitions in a similar way.

---

*Walter Richters*  
(Germany)

© Xephon 1998

---

## Intranet download EXEC

When we were building our Intranet, we decided to include print listings originating from our VSE mainframe system. We devised the following OS/2 REXX EXEC, named INLOAD, to automatically sense when listings were waiting in a specific VSE/POWER queue, download them, add HTML codes, and rewrite them to a disk where the Intranet could pick them up.

INLOAD runs on an OS/2 Warp machine, and wakes up every 45 seconds to send a VSE/POWER display command to the mainframe. It then RECEIVES and queries the results to see whether listings are waiting in the proper queue. If nothing is found, the EXEC goes back

to sleep for another 45 seconds. If listings are found, they are RECEIVED to the PC and then rewritten along with HTML header and trailer codes and a report title which appears on the title bar as the report is viewed in the Intranet. Finally, the VSE/POWER queue entry is deleted, and INLOAD goes back to sleep for another 45 seconds.

INLOAD has worked well for some time, and provides a fully automated download process which requires no human intervention.

## INLOAD

```

*****
***  INLOAD.CMD OS/2 REXX EXEC                               ***
*****

/*****/
/****                                     ****/
/**** Program Name   - INLOAD                               ****/
/**** Installation   - Arkansas Farm Bureau                 ****/
/**** Author         - Steve Bernard, Technical Support     ****/
/**** Date Written   - June 6, 1997                         ****/
/****                                     ****/
/**** Purpose       - Provide an automated method of       ****/
/****                downloading listings from VSE/POWER   ****/
/****                to the Intranet.                       ****/
/****                                     ****/
/****                This EXEC periodically queries VSE/POWER ****/
/****                for eligible listings.  When found, they ****/
/****                are downloaded, HTML codes are added and ****/
/****                they are re-written to a disk where the ****/
/****                intranet can access them.              ****/
/****                                     ****/
/*****/

    call rxfuncadd 'sysloadfuncs','rexxutil','sysloadfuncs'
    call sysloadfuncs

/*****/
/**** Titles for Intranet-destined reports.  These titles   ****/
/**** appear on the title bar of the viewing window.       ****/
/*****/

ttlC1571M = 'AGTSEC   Agent and Secretary Listing'
ttlA3175D = 'CALLOG   On-Call Response Log'

worktime  = '5:30am' /* Time when exec should scan queues */

```

```

DEL PWRDISP.TXT

do forever

    if time(c) = worktime then do

        say 'INLOAD01 Checking for eligible VSE/POWER listings...'

        'RECEIVE PWRDISP.TXT a: (file=pcmd format) d lst,q '

        if lines(PWRDISP.TXT) > 0 then
            call pwrrcv
        else
            say 'INLOAD02 No listings found. Resuming interval
                checks...'

    end

    say 'INLOAD03 The time is now 'time(C)'. Pausing until
        'worktime

    call sysssleep 45                /* wake up every 45 seconds */

end ; exit

/*****
/**** Bump through the VSE/POWER queue display, receive files, ****/
/**** add HTML codes and title, then rewrite the file to the ****/
/**** Intranet disk... ****/
/*****/
PWRRCV:

say 'INLOAD04 Listings found; beginning download sequence...'

do until lines(pwrdisp.txt) = 0

    cls                                /* clear the screen */
    iline = linein(pwrdisp.txt)        /* read line of pwr que disp */
    qname = substr(iline,2,8)         /* extract power lst name */
    qname = strip(qname)              /* remove blanks from name */
    qnbr = substr(iline,10,5)         /* extract power lst number */

    'RECEIVE' qname'.PWR a:' qname qnbr '(file=lst delete)'

    tname = ' ' ; ttl = ' '          /* clear table name & title */
    tname = 'ttl'qname                /* build table name */
    interpret 'ttl='tname             /* and activate it */
    ttl = substr(ttl,10)              /* bypass output disk name */
    dname = substr(ttl,1,8)           /* save output disk name */
    call lineout tname,ttl            /* write title file for copy */

```

```

        fileclose=lineout(tname)          /* close title file          */

        'copy htmlh1+'tname'+htmlh2+'qname'.pwr+htmlt1
        C:\WWW\AFB\REPORTS\'dname'.html'

/*****
/****
/**** The above copy command appends a series of files to
/**** create a new html file as shown below;
/****
/**** copy
/****   htmlh1      - contains 1st half of html header codes
/****   tname       - contains the report title
/****   htmlh2     - contains remaining html header codes
/****   qname.pwr  - contains file downloaded from VSE/POWER
/****   htmlt1     - contains html trailer codes
/****
/**** qname.html   - merged file containing all of the above
/****
/****
/*****/

        'DEL' qname'.PWR'                /* cleanup work files          */
        'DEL' tname

end

        fileclose=lineout(pwrdisp.txt)   /* close power file            */

return

*****/
*** HTMLH1 header (part 1)...
*****/

<|----->
<html>
<head>
<title>

*****/
*** HTMLH2 header (part 2)...
*****/

</title>
</head>
<body bgcolor=#FFFFFF text=#000000 link=#2B00AA>
<font size=3>
<pre>
<|----->

```

```
*****
*** HTMLT1 trailer ***
*****
```

```
<|----->
</pre>
</font>
</body>
</html>
```

---

*Steve Bernard*  
(USA)

© Xephon 1998

---

## VSE/ESA – what’s in a name?

VSE/ESA Version 1 Release 1 was first announced on 18 December 1990, and Version 2 Release 3 became available on 5 December 1997. Any acronym that has been used without explanation for this length of time can pose a problem, and anyone foolish enough to utter it with a non-mainframe person present will probably be asked what it is, what it stands for, and how it got such a weird name. The problem, of course, is knowing and/or remembering the answers to these questions.

The first answer is easy: “VSE/ESA is the name of an operating system for large IBM computers that used to be called mainframes.” But the other two answers are a bit more complicated ...

### ESA

VSE/ESA is an acronym for Virtual Storage Extended/Enterprise Systems Architecture. Although this is not very helpful, since it does not even hint at it being an operating system, it is certainly not alone in this regard: VM stands for Virtual Machine, and MVS is Multiple Virtual Storage, although it was recently renamed OS/390 (though IBM claims that the ‘OS’ stands for Open Server, not Operating System). Most other platforms are just as bad: the desktop has NT for New Technology, and who remembers what VAX/VMS stands for?

So, how did we get here – to VSE/ESA? The ESA part is the newest, added in September 1990 when IBM decided that the word ‘mainframe’

had too many negative connotations. Mainframes became Enterprise Systems. The latest mainframe technology was named Enterprise Systems Architecture (ESA) and branded System/390.

Like its predecessors, ESA increased the amount of addressable memory available to a program. Rather than being limited to 2GB, access registers allowed many 2GB chunks of memory to be used in a program, with an access register specifying the starting address of the relevant chunk. There could now be a technical debate as to whether large databases should be kept resident in virtual memory.

### 31-BIT ADDRESSING

The predecessor technology to ESA was XA, eXtended Architecture, which allows the simultaneous addressing of 2GB of virtual memory. Addresses are specified in the lower 31 bits of a 32-bit General Register. Since each byte of memory is addressed separately in System/3x0 architecture, 2<sup>31</sup> bytes of memory can be addressed with a 31-bit address.

But why only use 31 bits, when there are 32 available? After all, there's no need to reserve the high order bit for a sign (plus or minus) – addresses cannot be negative. In fact, the high order bit is used to differentiate between addresses specified in XA (and now ESA) and those in the predecessor technology to XA, which used only 24 bits. This differentiation is required not so much for upward compatibility, but for coexistence. The switch from 24- to 31-bit addressing requires more than just a management decree. In fact, it is not even enough to recompile every program; vendor products also need to be considered, since source code is generally not supplied for them.

But by far the scariest prospect is that of Assembler programs in which every effort has been made to save every possible bit. Many an Assembler program was written in which the top bits of an address were used to store other information – typically bit flags, but sometimes even the number of a memory pool by memory management routines.

Admittedly, VSE avoided XA technology, skipping directly from 24-bit addressing to ESA. There were several releases of VM/XA, but it was never very popular – it was MVS where XA technology was the standard for most of the 1980s.

## 24-BIT ADDRESSING

Only in retrospect was there anything resembling a standard name for 24-bit addressing technology. After newer technologies were released, systems software products and operating systems using the old technology were either renamed in order to differentiate them, or their previous name took on new meaning. SP (System Package or System Product, depending on the product) and 370 were the most common terms used. MVS/370 only got the 370 in its name after MVS/XA was released, and the already existing SP in VSE/SP suddenly meant 24-bit addressing.

There was one final predecessor technology. No, unlike PCs, addresses never got shorter than 24 bits, which allows the addressing of 16MB. This seemed like an infinitely large amount of memory to the original developers of the System/360, where today's IBM mainframe architecture began. Part of the reason that it sounded so large was that the System/360, with one exception, had no concept of virtual memory. All addresses referred to real memory. There was only one memory address, 19576 – it began with the 176,184th magnetic core of memory. Each byte required nine pieces of core memory to represent the eight data bits, with a parity bit for detecting data loss.

The one exception was the System/360 Model 67. As its name implies, it was released in 1967, one of the last System/360 models. It included hardware that allowed the storage and updating of page tables, for quick translation from virtual to real addresses. Each program – actually an address space or partition – had its own version of memory, where the same address location could exist in multiple partitions but point to different real memory locations with no fear of conflict. The key to making it all work was paging, where blocks of memory – a page – could be resident on disk to avoid running out of real memory. Plus, there was no need to reload the page later to the same real memory location.

Even some of the early System/370 computer models lacked virtual memory. Systems software and operating systems that supported virtual memory were indicated with VS for Virtual Storage, as, for example, in DOS/VS.

## VS TO VSE

And that takes us full circle. By tracing back the history of the ESA part of the VSE/ESA name, we stumbled across the VS, too. Later improvements ('Extensions') saw the addition of an 'E', for Extended.

Before adding support for ESA's multiple chunks of 2GB blocks of memory, VSE was known as VSE/SP. Rather than defining a technology for addressing memory and other machine architecture issues, the 'SP' (System Package) originally referred to the packaging of IBM operating system and systems software products together into one item from an ordering, installation, and support perspective. Ironically, while this was going on in the VSE and VM worlds, MVS was being carved up into more products, not fewer. But that is another story, although it is worth pointing out that OS/390 has since reversed the trend and done the same kind of packaging for MVS and its many systems software products.

## DOS/VSE AND SYSGENs

Before VSE/SP, there was a divergence into two quite different products – DOS/VSE and SSX/VSE – both of which could claim parentage to VSE/SP. Both ran the same application software with the same JCL. DOS/VSE (Disk Operating System/Virtual Storage Extended) gave you the full capabilities of the operating system as it existed at the time. But it required a SYSGEN.

SYSGEN is short for System Generation, and was actually an assembly (Assembler language equivalent of compilation) of Assembler macros, where the systems programmer would change the values of any default systems parameters, such as the number of partitions. It was a scary feeling to know that an entire organization would be using an operating system for which you were now modifying the source code. And it was easy to make mistakes, since the syntax rules for Assembler macros are similar, but slightly worse than those for JCL: no spaces around commas, equals signs, or parentheses; column 72 is for continuation marks and nothing else; no blank lines; multiple parameter values must be surrounded with parentheses; and odd spellings to live with eight-character naming limitations.



Of course, this meant that one bad estimate for a value, or a change in the environment, meant another SYSGEN. Changing one value often required the change of another. For this and several other reasons, it usually took quite a few SYSGENs for an initial install or major change. Because so much source code was involved, a SYSGEN took a long time to run.

And it was not just the operating system. Other IBM and non-IBM systems software products required the same SYSGEN process. Some, like CICS, also had separately assembled tables. Tables reduced the number of SYSGENs required because of changes in the environment and the number of things that had to be set, and could go wrong, in the initial SYSGEN.

DOS/VSE delivered the maximum performance and functionality, but it took a lot of time and effort to get there.

#### SSX/VSE

SSX/VSE (Small Systems eXecutive/Virtual Storage Extended) was a prepackaged system that did not require a SYSGEN. In fact, it was impossible to perform a SYSGEN in SSX. It was aimed at the mini-computer marketplace, namely:

- Smaller organizations (100-1,000 employees) getting their first computer.
- Companies considering IBM as a vendor to replace their existing non-IBM minicomputer, either because they were outgrowing it or because their vendor was getting out of the business.
- IBM minicomputer customers outgrowing their present platform.

SSX/VSE came with a fixed set of systems software on a single installation tape at a very attractive price. It was installed as an image of a couple of prenamed FBA (Fixed Block Architecture) DASD volumes. The few things that could be changed were separated into multiple small files that could be modified and reassembled in isolation, rather than the huge monster that SYSGEN represented, where you had to get everything right or nothing worked.

A few vendors signed on, too. They packaged their products – both

systems and application software – in a specific tape installation format and required no SYSGEN. If they got it right, installation was complete in less than an hour.

Other products were also available to install on SSX/VSE. PL/I, for example, was not prepackaged for SSX, but it could still be installed in the old-fashioned way. And therein lies a story: SAS was one of the pioneering vendors who signed on for SSX. But the base SAS product required the PL/I Transient Library, and IBM didn't include this with SSX/VSE, nor was it packaged in SSX format. So much for eliminating the need for the experienced DOS systems programmer.

The level of commitment of some of these vendors also had to be questioned. Despite its standardization – there was only one way to install an SSX product – one major package failed to install, clearly indicating that the install had never been tested in its final form, as delivered to the customer.

SSX/VSE only ever had a few sites world-wide, and IBM's goal of providing a mainframe alternative to minicomputers arguably died with it, but its approach formed the basis of VSE/SP – only extended to provide the flexibility of DOS/VSE. My personal opinion is that the DOS got dropped from the name in the transition from DOS/VSE to VSE/SP out of respect for SSX/VSE. Of course, there are those who argue that DOS was removed from the name to avoid confusion with a new operating system by a certain Mr Gates ....

## DOS/360

DOS, of course, stands for Disk Operating System. Before DOS/VSE, there was DOS/VS. The original System/360 operating system was named DOS/360, and, like its elder brother OS/360, only addressed real memory. Like many elder brothers, OS/360 was originally planned as an only child. But it soon became clear that one was not enough, and DOS/360 was born.

A radical concept at the time, the System/360 was originally promised to be a single family of compatible computer systems with one operating system that ran on all models of hardware. As the development of this operating system progressed, it became clear that the

functionality and hardware flexibility originally envisioned necessitated a complexity that could not be coded small enough to run on the smaller models of the System/360 – specifically, those with only 16KB of memory. Thus was born DOS/360.

This realization actually gave birth to three new operating systems, not just DOS/360. There were also TOS (Tape Operating System) and BOS (Basic Operating System), the latter for System/360 models sold with only 8KB memory. There was also a variant of DOS, best remembered for its abbreviated version of JCL, developed a few years later for the System/360 Model 20. None of these variants survived through System/370.

Like OS/360, the JCL processor for DOS/360 was adapted from the macro processor of the System/360 Assembler. This decision was made by IBM to deal with the huge cost overruns of the OS/360 development project and the unplanned costs of developing these new operating systems.

## OLD CODE

Speaking of Assembler and macros, system macros and data areas often provide an interesting stroll down memory lane. For example, offset 62 (X'3E') in the Format 1 DSCB of a VTOC is the DS1SYSCD System Code field. The current value of that field on an OS/390 system is:

```
IBMOSVS2
```

which indicates it has not been updated in about 20 years.

Until High Level Assembler, there was no standard way to determine what version and release of the operating system an Assembler program was being assembled under, nor even what operating system. Systems software developers discovered their own techniques, but they were often unique to a vendor. They could be as simple as passing the name of the operating system as a parameter at Assembly time, and referenced through &SYSPARM:

```
AIF ('&SYSPARM' EQ 'DOS').DOS
```

to subtle differences between releases of the operating system that could be checked with one or two Assembler instructions at run-time.

The High Level Assembler provides the `&SYSTEM_ID` variable, which contains a relatively lengthy string identifying operating system, version, release, and modification level. For example:

```
VSE/AF 5.1.2
```

## VSE'S BIRTH AND DEATH

VSE has come a long way in its more than 30-year history. The first decade of both DOS and OS (MVS) is well documented in *IBM's 360 and Early 370 Systems* by Pugh, Johnson, and Palmer (1991, MIT Press). In it, they state that "Despite its inauspicious beginnings, DOS was destined to become the most widely used operating system in the world." I think they may have got it confused with another operating system called DOS, but there is no arguing its popularity.

There have been many attempts to kill VSE over the years, and the two most recent occurred in the past decade. When it became clear that development of VSE had ground to a halt in the early 1990s, a major campaign by several large users at industry conferences and in the computer trade press renewed interest in VSE. More recently, plans by IBM to merge VSE and VM were scrapped after discussions with customers elicited a decidedly negative reaction.

Tom Watson Junior's autobiography, *Father Son & Co*, also gives significant detail about the development of the System/360. For a glimpse of the computer world just before System/360, Ned Chapin's 1963 Second Edition of *An Introduction to Automatic Computers* makes fascinating reading. Although both are out of print, they can often be found in public libraries, and occasionally in used bookstores. And, because they were best sellers in their day, bookstores with out-of-print search services can often obtain them; amazon.com lists both in its catalogue, and will quote price and availability without requiring a deposit.

---

*Jon E Pearkins*  
(Canada)

© Xephon 1998

---

## IBM's VSE/ESA Web site

*There's such a flood of information available that it's easy to miss key resources that can actually solve problems, offer improvements, help interact with colleagues, and so on. This article is the first in a series of reviews which will identify, describe, and navigate Web sites of interest to the VSE community. In it, we visit IBM's VSE/ESA Web site, which can be accessed at <<http://www.s390.ibm.com/vse/>>*

VSE has evolved through the decades into today's powerful and well-regarded mainframe operating system, one of IBM's three mainstream System/390 environments (see 'VSE/ESA – what's in a name?', pp 45-52). But in spite of the diverse and widespread worldwide VSE community, it's easy for an individual or an installation to feel isolated, especially when told by a vendor that they're the only one reporting a problem or specifying a requirement. Just as user groups have done for decades, the Internet immunizes against the feeling of being alone with a problem, question, or opportunity.

IBM's VSE Web site opens with the cheery welcome and declaration: *"Welcome to the IBM VSE/ESA home page! VSE/ESA is a member of the S/390 operating system family. Together with VM/ESA, VSE/ESA forms the basis of S/390's on-going support for its Medium-sized Enterprise sector."*

It's clear that people working on the Web site take a personal interest in conveying the positive message and delivering content of value to VSE sites. And it's refreshing to have VSE positioned both squarely within the System/390 spectrum, and also as a partner in supporting medium-size enterprises. In fact, IBM recently placed VSE and VM in a separate organization, the VSE/VM Independent Business Unit, to allow both operating systems to develop their own marketing plans to serve their customers.

Besides the greeting, the VSE Web page is simple in construction: a System/390 toolbar at top, several VSE links below the greeting, and a number of topic categories at the left margin. The current first link is very timely, and doubtless very much on peoples' minds: *VSE and Year 2000*. After loading a mysterious 38Kbyte graphic, the link

yields three sub-categories: *Info from IBM*, *Info from VSE*, and *Vendor Solutions*. There's also a prominent note: "This page is always "under construction". If you know of sources related to VSE and Year 2000 that are not shown here, please let us know." This illustrates that IBM has conquered the 'not invented here' syndrome of years back and now recognizes the need to actively team with customers and vendors to ensure VSE's survival and success.

The *Info from IBM* page in turn offers several links: *IBM and the Year 2000*, *IBM Year 2000 Support Center*, *Year 2000 question? (Contact IBM Year 2000 Support Team)*, *FAQs for Year 2000 (includes FAQ for VSE)*, *VSAM and the Year 2000 White paper*, and *VM and the Year 2000*.

The *IBM and the Year 2000* page is a good starting point for technical and industry information – it offers quick access to diverse information, organized by computing platform (eg System/390, RS/6000, OS/2, and DOS), industry size or niche, and tasks to be accomplished (eg planning, testing, etc). It also links to non-IBM information sources such as Ed Yardeni and ITAA, which are well worth reading. *IBM's Year 2000 Support Center* offers a comprehensive list of IBM resources and related information. IBM's Year 2000 Support team notes: "The *IBM Year 2000 Technical Support Center in Perth, Australia*, offers VSE customers a free question & answer service. If you have a question related to VSE and Year 2000, complete the form below and select Send Mail. An answer within 24 hours is guaranteed."

The *FAQs* page includes documents on programming languages, operating systems, databases, hardware, methodology, software products, and testing, but only a few questions are included in two VSE FAQs. The next link provides a paper on VSAM and Year 2000, followed by a link to VM's elaborate Year 2000 Web page.

The *Info from VSE* page is a catalogue of technical information, resources, tips and techniques, and presentations. The first item is *IBM Deutschland Receives ITAA\*2000 Certification*, which begins: "The Information Technology Association of America (ITAA) today announced that IBM Deutschland Entwicklung GmbH, Boblingen Programming Lab, Boblingen, Germany has received ITAA\*2000 certification. ITAA\*2000 is the industry's century date change

*certification program. The program examines processes and methods used by companies to perform their Year 2000 software conversions.”*

The *Info From VSE* page also allows you to download the *VSE/ESA Newsletter Year 2000* special edition, or selected articles from it, including ‘Full four-digit date year support in VSE/POWER’, ‘It’s the right time for COBOL!’, and ‘Year 2000 support for VSE access methods’. *Vendor Y2000 Solutions for VSE* includes alphabetical listings/links from Allegiant Legacy Solutions and Barnard Software to Xpress Software and Y2Kplus.

Returning to the main VSE page, the next link, *Related user, vendor, and IBM home pages* lists more than five screens of resources. Many of these Web sites are entire worlds, offering content, people, information, software, and more, justifying browsing and searching.

VSE professionals can benefit from user group affiliations, and the *World Alliance of VM and VSE (WAVV)* and *VSE User Group* pages can meet that need, highlighting respectively the yearly WAVV conference (1999 in Cincinnati) and the joys and rewards of local user grouping: “*VSE is not a spectator sport... Get involved! Over the years, VSE has been fortunate to have a strong community of users, systems programmers and other professionals intent on supporting VSE’s continued success in an ever-changing environment. Meeting together on a periodic basis is an essential element in sustaining this environment of community and cooperation.*”

It’s hard to describe to a non-participant, but professional activities such as WAVV and local user groups take on an added aura of socialization and friendship that makes them all the more valuable. One anticipates meetings as gatherings with friends, not just for talking shop. The vendor list can aid gathering information on products, tracking down useful facilities, or comparing alternative solutions in the comfort of your own office.

The next link on the main VSE page, *Other VSE home pages, related products, etc*, is a catch-all, containing more VSE home pages and general topics/interest areas. The first category includes: *VSE Turbo Dispatcher home page*, *LANRES/VSE home page*, *VSE Workdesk home page*, and *TCP/IP configuration support home page*. The latter

category indexes manuals, teleconferences, publications, software, and an inventory of VSE-related products from IBM.

The next main page link, *Documents, demos, and coding examples available via FTP*, points to a wealth of material that would otherwise be quite hard to assemble. For example, it offers the full text of *Hints and Tips for VSE/ESA*, Third Edition (December 1997) in both PDF and PostScript formats. Further down the main page is *HTML files with technical information from VSE Development*, which opens “This page is always “under construction.” Newest items are at the top of the list”. The newest additions are *Online Message Explanations for LE/VSE* by Gerhard Schneidt, VSE Product Planning; *VSE/ESA Version 2 Console Support* by Manfred Baudisch, VSE Development; *What’s New for REXX/VSE in VSE/ESA 2.2* by Ursula Braun-Krahl, REXX/VSE Development; and *P/390 and R/390 Performance Attracts VM and VSE Customers* by Gary Eheman, System/390 VM/VSE Technical Marketing. The last main page link is *Web Sites on VSE* which lists a few VSE Web hosts (IBM would like to hear of any others).

Returning to the main VSE page, the first topic category on the left is *Information auf Deutsch*, reflecting VSE’s home base in IBM Germany. The next topic, *VSE/ESA Newsletter*, provides the current issue, available as a merged and several individual article PDF files. Under *Main Technical Articles* it includes ‘Effective Enterprise Data Access to VSE’, ‘LE/VSE Socket Programming’, and ‘How to Do Physical Attachments to TCP/IP for VSE/ESA’. The *More Articles and News* category covers user group meetings, Redbooks and other technical publications, and customer success stories. The next main page topic is the *System/390 Bulletin*, whose most recent several issues are available for download in PDF. The first article category is *Customer Success Stories*, which highlight business-related applications of System/390 computing platforms, demonstrating mission-critical results and return on investment. Customer stories include ‘Sallie Mae Uses Tivoli Software and Services’, ‘Modernizing the Belarussian Interbank Settlement System’, and ‘Program Development Under S/390 — The Amadeus Case’. The *System/390 Bulletin* also includes System/390 product news such as ‘Coexistence Offering for VM and VSE’, and information on business partners and updates.



The next three topics relate to VSE conferences – and it’s gratifying that IBM gives equal weight to its own conferences and the WAVV user group meeting. You can click to send suggestions about the IBM VM/VSE Technical Conference for 1999, which will be held on 24-27 May in Orlando, Florida. The next two items provide access to two 1998 conferences: IBM’s VM/VSE Technical Conference in Reno, NV, in May, and WAVV ’98 in Albany, NY. Both conferences featured dozens of presentations by IBMers, customers, vendors, and consultants. These two download libraries offer great value in acquiring current VSE information, and can both reduce disappointment for missing the conferences and provide incentive and justification for planning to attend next year. The *IBM Tech Conference* page offers topics such as *Storage Systems Update for VSE and VM*, *Developments in Languages for VSE*, *Roads from the Web to VSE/ESA: The CICS Gateway Solutions*, and *Real World Turbo Dispatcher Use*. The WAVV page opens with a photo of three VSE stalwarts – Bennie Pugh, Pete Clark, and Charles Rice – and the caption “*Thanks, gentlemen and ladies, from all of us!*”. The page continues: “*The World Alliance of VSE and VM (WAVV) held its 1998 Conference in Albany, New York. Thanks to the conference’s organizers and presenters, we now have a number of the presentations available here. We’ve done this to support VM/VSE users who were not able to attend the conference.*”

Presentations available include ‘VSE/ESA Hints for Performance Activities’ by Wolfgang Kraemer, ‘CICS/VSE Problem Determination – Demystifying the Transaction Dump’ by John Lawson, and ‘Web Enabling CICS Applications – Methodologies and Techniques’ by David Kennedy.

The first topic in the second grouping is *Education Information*, which opens by saying: “*This site provides ‘one-stop shopping’ for all your S/390 product, technical and sales training information needs. Whether you are an IBM customer, an IBM S/390 Business Partner, or an IBM sales, marketing or technical support person, keeping your skills at peak fitness is critical. Here you’ll find guidance to training offerings that explain the latest trends in large systems technology, help you to maximize your in-house expertise, and increase*

*your own 'market value' in the industry.*” In spite of the introduction being somewhat ‘marketing speak’, this page is worth searching for topics of interest, to stay abreast of evolving IBM hardware, software, networks, and related technologies.

The next topic offers information on *Upcoming Events*. For those with a wanderlust, or those wanting to know when a VSE event will be near them, this page lists many months – right now, covering September 1998 (WAVV in Albany, NY) through May 1999 (VM and VSE Technical Conference in Orlando, FL), passing through many cities worldwide (Munich, Milan, Capetown, Madrid, Des Moines, Hamburg, etc) with events such as ‘IBM VM & VSE Seminar’, ‘TCP/IP from A to Z’, and ‘Pathway to the Future – S/390 Roadshow’. Almost all event listings are linked to additional information.

The next topic, *Information Sources*, offers PDF files for the second edition of ‘VSE/ESA – Keeping You Informed’ (G511-4224-00, December 1997), and instructions for downloading or ordering it.

The next topic, *VSE and Coexistence*, addresses a topic of great interest and concern to the VSE community. Written by IBM’s Jerry Johnston, and entitled “*What Is the Future of VSE and What Does OS/390 Coexistence Mean to You?*”, it addresses history, noting that: “*Since the introduction of VSE/ESA in 1990, VSE has pursued an ‘OS/390 Affinity’ technical strategy. That means whenever we address customer requirements for VSE, we try to make the solutions consistent with OS/390 solutions.*”

The present and future are described: “*In October 1997, we previewed a major new step in the evolution of VSE/ESA. IBM previewed VSE/ESA V2.4. The main innovation in V2.4 will be a new CICS Transaction Server for VSE based on CICS/ESA for OS/390. ... IBM will continue to support and enhance VSE/ESA.*”

IBM notes reality: “*However, there are limits to what we can do even with OS/390 affinity. The most significant limitation is that VSE has no equivalent of OS/390 Unix services. Moreover, the cost of adding Unix services to VSE/ESA is prohibitive.*”

And concludes: “*It’s important VSE customers understand OS/390 coexistence is simply one more option for them to consider. We’re*

*committed to improving the option and believe it offers some unique advantages. Still, it's only one option. Many VSE customers will decide VSE continues to meet all of their basic I/T needs. Some VSE customers will decide that, although they are basically satisfied with VSE, they need new applications that are only available for OS/390. They should consider OS/390 coexistence. Others will decide they want to migrate their entire VSE workload to OS/390. Any of the options is valid. The choice of which is best is up to each customer. All are 'fail safe' options. That is, having once selected an option, the customer can later decide to pick one of the other options. The customer's investment in S/390 is secure and nothing is lost."*

The next topic, *VM and VSE*, links to a page entitled *The Value of VM for VSE Enterprises*, which can be summarized with one of its section headings, 'It Has Never Been a Better Time to be a VM and VSE Customer'.

That's followed by a link to a useful product availability/support matrix, a VSE Web site index/map, and a feedback page. The last items at bottom left are an evolving list of new links and content on the Web page.

The final aspects of the VSE page to examine are in the toolbar at the top, linking to *Solutions* (solving business needs), *Services* (enjoy total System/390 support), *Products* (hardware and software products work together), *Sitemap*, *Success Stories* (see how other companies accelerate their operations), *News*, *Reference* (everything about System/390 – from tech to guidebooks), and *Contact System/390*.

Even just a few years ago, it would have been impossible to imagine IBM publishing such a complete collection of information about an IBM operating system, featuring a great deal of content not under its control. This Web site is an admirable demonstration of IBM's commitment to participating in the VSE community, collaborating with VSE customers, software and hardware vendors, and many other players. IBM clearly welcomes interaction with and feedback from the VSE world, and is both benefiting from and helping its diverse collaborators.

---

*Gabe Goldberg*  
*Computers and Publishing (USA)*

© Xephon 1998

---

## Program-driven output segmentation – correction

In the article entitled ‘Program driven output segmentation’ published in the September 1998 issue of *VSE Update*, the character representing the paragraph was misprinted in the string consisting of a star (\*), a blank, two paragraph characters, and a further blank.

This error can be remedied by replacing the paragraph character with any character except ‘S’ – eg ‘%’ or ‘&’.

The code needs the following corrections and additions:

- 1 Page 38, line 3 : The string reads ‘\* SS ’. Replace ‘S’ with your chosen paragraph character, to give, say: ‘\* && ’
- 2 On page 42, towards the bottom of the page, in the line with the comment ‘KEEP OLD OUTPUT ATTRIBUTES’, the two paragraph characters have been omitted and need to be reinserted.
- 3 On page 43, in line 16 (Comment ‘RESTORE OLD CONTENT’), the two paragraph characters have been omitted and need to be reinserted.

We apologize for any inconvenience caused by this error.

---

© Xephon 1998

---

### **Leaving?**

### **You don't have to give up *VSE Update***

You don't have to lose your subscription when you move to another location – let us know your new address, and the name of your successor at your current address, and we will send *VSE Update* to both of you, for the duration of your subscription. There is no charge for the additional copies.

## March 1991 – December 1998 index

Items below are references to articles that have appeared in *VSE Update* since March 1991. References show the issue number followed by the page number(s). Individual copies of all issues from that date are available.

\$COMVAR	12.60-61	console scanner	22.55-63
\$SYSOPEN	2.7-11	control block	14.3-55
accounting	7.19-55, 17.29-62, 20.3-17	Control File	10.8-39
AFP processing	25.48	copy book	13.3, 14.56
AIX	24.47-61	CPU-id	1.9-12, 6.31-33, 8.26-29
alias phase names	22.33-34	cross-job parameters	1.3-8
APPC	3.3	CSP	1.51-52
ARIDBS	30.21-30	CTCA	4.55, 11.18
ASIPROC	1.11	CTLSPOOL	8.23-25
automated compile	28.21-53	cut and paste	11.40-61
automated operations	4.7-54	DASD	17.3-25
automation	6.13-27, 13.63	data exceptions	7.8-12
back-up	20.18-57, 25.52-63, 26.14-32	data integration	25.42-44
banner pages	20.57-58	dataspaces	9.5-7
batch	4.62-63, 21.47-59, 28.3-15	date	2.7-11, 29.11-24, 30.3-21
batch commands	2.43-52, 3.29-30, 6.3-13, 6.33-34, 9.13-34, 9.40-42, 10.49, 11.24-37, 12.59	DETACH	28.57-61
BLL	10.39-49	device display	1.62-69
block size	4.64-65	DFHSNT	32.3-37
BMS	11.3-16	DFSORT	32.24
calling program execution	28.54-56	directory	12.33-59, 14.59
CDLOAD	31.22-31	DL/I	3.51-61, 21.3-25, 22.3-7
CHPID	7.14-17	DOS/VSE	32.48-51
CICS	4.62-63, 6.33-34, 13.3-28, 13.29-61, 13.62, 15.39-40, 18.29-48, 19.3-51, 23.15-20, 24.47-61, 32.5-26	DPR	7.13
CICS TS	32.38-39	DPS	7.13
client/server	18.61-63	DSECT	13.3
CMS	29.51-63, 30.32-51	DTSIESDC	12.32
CNTLUNIT	7.16-18	dynamic partitions	11.38-40
COBOL	2.12-14, 4.58-61, 5.62-63, 7.3-7, 7.60-63, 10.39-49, 13.62-66, 23.21-22, 26.46-63, 32.18-26	dynamic partition tables	8.47-56, 12.60-61, 18.62-63
compare files	15.3-38	EPIC/VSE	12.61-66, 28.57
compiles	14.57-58, 22.34-55, 28.21-53	ESCON	7.17, 9.3-9
compression	15.53-63	EXEC PARM	7.60-63
Computer Associates	19.63	expanding bytes	31.35-37
COMREG	2.4, 4.17	fonts	10.3-4
conditional JCL	3.18-22	fullscreen	21.25-46
console	2.36-42, 3.47-50, 6.13-27, 11.23-24, 12.32, 13.3-28	function packages	31.50-62
console automation	29.30-50	GDDM	11.61-63
		GDG	6.34-71
		GETJA	9.38-39
		GETVIS	6.14, 9.3-6, 31.22-31
		hardcopy	9.34-37
		HDAM	3.51-61
		history	17.29-62, 32.45-52

HSA	7.13	PL/I	3.18-22
ICCF	3.22, 4.56-57, 12.32, 23.23-27	POWER	1.18-50, 2.12-14, 3.29-30, 5.24-61, 5.64-67, 6.28-30, 6.3-13, 7.19-55, 8.23-25, 8.30-47, 11.16-20, 12.59, 17.29-62, 18.49-61, 22.11-22, 25.3-31, 26.8-13, 26.33-44, 27.3-44, 28.21, 29.28-29, 31.44-46, 32.40-45
IESCNTL	32.3-37	POWERNET	27.58-63
IESINSRT	14.57-58, 26.8-13, 28.21	PR/SM	4.54-55
Internet	25.44-45	printers	10.3-4
Internet access	24.39-40	print files	4.58-61, 11.16-20
interrupt	5.21-23	printlog	3.31-46
intranet	32.41-45	PROCs	11.38-40
I/O	20.58-61	PSF/VSE	25.48-51, 27.56-57
IOCP	1.11, 7.13-19	PSQ	15.41
IODEVICE	7.16-18	PTF	14.55-56
IPL	2.7-11, 13.63	PUB	1.62-69, 3.22, 14.3
IPSF	15.41	Punch output	26.8-13
IPWSEGM	31.37-43	QMF	16.31-41
ISQL	16.42-50	queue	1.18-50, 8.30-47
IUI	19.3-51, 32.3	RAMAC	29.3
IXFP/SnapShot	32.38-39	reader exit	18.49-61
JES2/NJE	27.58-63	real storage	15.41-53
job summary	8.57-63	reply-ids	16.50-54, 21.60-63
job synchronization	10.50-67	restart prevention	32.40-41
legacy applications	25.31-47	restore	20.18-57
LE/VSE	27.45-47, 28.56	return code	5.62-63, 7.3-7
Librarian	1.3, 8.14-23, 9.9-10, 9.43-45	reverse engineering	25.38-39
libraries	12.3-32, 22.23-32, 29.51-63	REXX	23.3-14, 23.27-63, 24.3-39, 27.45-47, 28.56, 29.24-29, 29.30-50, 31.44, 31.50-62
Library Access Services	12.3, 12.32	RSCS	5.24-61
LIBRDCB	12.3, 12.32	saving data	25.52-63, 26.14-32
LIBRM	12.3, 12.32	scheduler	4.3-6
LISTCAT	18.9-28	scheduling	5.64-67, 6.28-30
load lists	16.54.61	scrolling	11.23-24
lock file	8.3-14	searching strings	31.3-21
logical unit	28.57-61, 31.44-50	security	1.60-62
LPAR	14.60-63	SETPARM	1.3, 28.15-20
macros	30.32-51	sharing files	27.48-55
maintainability	25.41	SIR command	19.62-63
MSHP	14.55	sort	2.56-58, 31.32-34
MVS	27.58-63	SP3 libraries	2.15-35
MVS conversion	9.46-67, 27.58-63, 30.56-63	spooling	27.3-44
objects	22.23-32	SQL	3.3-14, 16.3-30, 18.62, 19.52-61, 30.21-30
on-line messages	1.12-18	SSX/VSE	32.48-50
ORE	21.47, 21.60	start-up	15.39-40
OS/2	32.41-45	storage dump	3.62-67
output segmentation	31.37-43, 32.60	SVA	5.3-21, 9.3, 10.5, 16.54.61, 18.3-9
packed-unsigned decimal	3.15-17		
passwords	11.20-23		
PCB	14.3		
PC download	1.52-60		
PDUMP	8.14-15		
performance	4.64-65		
PF Keys	3.47-50, 13.62		

SVC30	16.50	VM	21,25-46, 25.52-63, 26.14-32
synchronizing batch tasks	28.3-15, 30.51-56	VM/VSE interface	24.41-47
system unit records	7.56-60	volume characteristics	4.66-67
tables	26.46-63	VSAM	18.62, 20.18-57, 23.21-22, 25.52-53
tape	3.22-28, 5.63-64, 10.5-8, 14.60-63	VSECMD	2.36, 29.30
tracking system	23.27-63, 24.3-39	VSE/ESA	1.70-71, 2.3-7, 3.46, 9.3-12, 26.3-8
transferring files	27.48-55	VSEMSG	2.36
UCA	4.17	VSEREP	2.36
upgrading	26.3-8	VTAM	27.58-63
user experiences	26.3-8	VTOC	17.25-29
version 2.1	22.7-10	XPCC	3.3-6
version 2.4	32.38-39	Web	30.31, 32.53-59
view	13.29-61	Year 2000	25.45-46, 26.3-8, 29.3-11
virtual CTCA	2.52-55		
VLA	9.3		

## Contributing to *VSE Update*

In addition to *VSE Update*, the Xephon family of Update publications now includes *CICS Update*, *VM Update*, *MVS Update*, *SNA Update*, *VSAM Update*, *DB2 Update*, *RACF Update*, *AIX Update*, *Domino Update*, *NT Update*, *Oracle Update*, and *SQL Server Update*. Although the articles published are of a very high standard, the vast majority are not written by professional writers, and we rely heavily on our readers themselves taking the time and trouble to share their experiences with others.

If you have ever experienced any difficulties with VSE, or made an interesting discovery, you could receive a cash payment, a free subscription to any of our *Updates*, or a credit against any of Xephon's wide range of products and services, simply by telling us all about it. For a copy of our *Notes for Contributors*, which explains the terms and conditions under which we publish articles, please write to the editor, Fiona Hewitt, at any of the addresses shown on page 2, or e-mail her on [100336.1412@compuserve.com](mailto:100336.1412@compuserve.com)

## VSE news

---

IBM has announced Release 3 of High Level Assembler for VSE, which replaces all previous System/370 and System/390 Assemblers.

This includes tools for developing and maintaining all Assembler applications, converting symbolic source statements to machine language object code with extensive checks for common coding errors. There is support the ESA/390 IEEE binary floating-point instructions, data, and additional floating point registers, and there are new diagnostic and cross-reference features and a range of language enhancements.

For further information, contact your local IBM representative.

\* \* \*

IBM has announced its VM:Webgateway, which provides Web browser interfaces for all VM, VSE, OS/390, and TPF applications. VM:Webgateway is said to serve all types of Web data, protect data by combining mainframe-class authentication and access control with Secure Sockets Layer (SSL) technology, simplify most administration tasks with a Web browser interface, and scale to meet the needs of thousands of users. With it comes the VM:Webgateway OfficeVision interface, which provides a graphical browser interface for OfficeVision's e-mail and calendar functions.

For further information, contact your local IBM representative.

\* \* \*

At the time of going to print, IBM's new System/390 Integrated Server, which supports current levels of VSE/ESA, was due to ship on 12 November. The machines are aimed at users who need something less than the System/390 Multiprise, but still want to replace older System/370 and System/390 processors. The machine is positioned as a packaged application server for System/390 software, and for development using old and new tools.

For further information, contact your local IBM representative.

\* \* \*

IBM has announced a limited-time offer in which you can get a free System/390 Channel-to-LAN Connection Card if you have a qualifying service to help use the mainframe as a LAN data and/or print server. The deal is only available in the US and Puerto Rico, and runs until 29 January. The service in question is the IBM SmoothStart Service for OS/390 LANRES, which includes help in implementing a working System/390 LAN and print server. You'll need a PC Server with 128MB or more memory, BusMaster/LANStreamer LAN Cards, a LAN of course, and any System/390 ESA with an available ESCON channel.

For further information, contact your local IBM representative.

\* \* \*

IBM has announced VSE/ESA Version 2.4—see pages 38-39 of this issue for an announcement review.



**xephon**