Vol. 2

heavy weight paper - perfect for felt tip pens!

50 SHEETS 11x8½ 4x4 QUADRILLE 33-688

NATIONAL

NATIONAL BLANK BOOK COMPANY, INC. • HOLYOKE, MASS. 01040

MADE IN U. S. A.

CSR
scroll

CHEMICAL SYMBOLS & VALENCE CHART

B. Page

Chemical Elements, Symbols, Atomic Weights and Valence Chart

Atomic weights, based on the exact number 12 as the assigned atomic mass of the principal isotope of carbon, carbon 12, are provided through the courtesy of the International Union of Pure and Applied Chemistry and Butterworth Scientific Publications.

For the radioactive elements with the exception of uranium, thorium and radium, the mass number of either the isotope of longest half-life (marked with a star) or the better known isotope (marked with two stars) is given.

Chemical Element	Symbol	At. No.	Atomic Weight	Valence	Discoverer
Actinium	Ac	89	227*	.....	Debierne
Aluminum	Al	13	26.9815	3	Oersted
Americium	Am	95	243*	3, 4, 5, 6	Seaborg, et al.
Antimony	Sb	51	121.75	3, 5	Valentine
Argon	Ar	18	39.948	0	Rayleigh, Ramsay
Arsenic	As	33	74.9216	3, 5	Magnus
Astatine	At	85	210*	1, 3, 5, 7	Corson, et al.
Barium	Ba	56	137.34	2	Davy
Berkelium	Bk	97	247*	3, 4	Thompson, Ghiorso, Seaborg
Beryllium	Be	4	9.0122	2	Vauquelin
Bismuth	Bi	83	208.980	3, 5	Valentine
Boron	B	5	10.811a	3	Davy
Bromine	Br	35	79.909b	1, 3, 5, 7	Balard
Cadmium	Cd	48	112.40	2	.....
Calcium	Ca	20	40.08	2	Stromeyer
Californium	Cf	98	249*	.....	Davy
Carbon	C	6	12.01115a	2, 4	Thompson, et al.
Cerium	Ce	58	140.12	3, 4	.....
Cesium	Cs	55	132.905	1	Klaproth
Chlorine	Cl	17	35.453b	1, 3, 5, 7	Bunsen, Kirchoff
Chromium	Cr	24	51.996b	2, 3, 6	Scheele
Cobalt	Co	27	58.933	2, 3	Vauquelin
Copper	Cu	29	63.54	1, 2	Brandt
Curium	Cm	96	247*	3	Seaborg, et al.
Dysprosium	Dy	66	162.50	3	Boisbaudran
Einsteinium	Es	99	254*	.....	Ghiorso, et al.
Erbium	Er	68	167.26	3	Mosander
Europium	Eu	63	151.96	2, 3	Demarcay
Fermium	Fm	100	253**	.....	Ghiorso, et al.
Fluorine	F	9	18.9984	1	Scheele
Francium	Fr	87	223*	1	Perey
Gadolinium	Gd	64	157.25	3	Marignac
Gallium	Ga	31	69.72	2, 3	Boisbaudran
Germanium	Ge	32	72.59	4	Winkler
Gold	Au	79	196.967	1, 3	.....
Hafnium	Hf	72	178.49	4	Coster, Hevesy
Helium	He	2	4.0026	0	Ramsay
Holmium	Ho	67	164.930	3	Cleve
Hydrogen	H	1	1.00797a	1	Cavendish
Indium	In	49	114.82	3	Reich, Richter
Iodine	I	53	126.9044	1, 3, 5, 7	Courtois
Iridium	Ir	77	192.2	3, 4	Tennant
Iron	Fe	26	55.847b	2, 3	.....
Krypton	Kr	36	83.80	0	Ramsay, Travers
Lanthanum	La	57	138.91	3	Mosander
Lawrencium	Lw	103	257*	2, 4	Ghiorso, T. Sikkeland, A. E. Larsh, and R. M. Latimer
Lead	Pb	82	.....	1	.....
Lithium	Li	3	6.939	1	.....
Lutetium	Lu	71	174.97	3	Arfvedson
Magnesium	Mg	12	24.312	2	Welsbach, Urbain
Manganese	Mn	25	54.9380	2, 3, 4, 6, 7	Liebig, Bussy
Mendelevium	Md	101	256*	.....	Gahn
Mercury	Hg	80	200.59	1, 2	Ghiorso, et al.
Molybdenum	Mo	42	95.94	3, 4, 6	Hjelm
Neodymium	Nd	60	144.24	3	Welsbach
Neon	Ne	10	20.183	0	Ramsay, Travers
Neptunium	Np	93	237*	4, 5, 6	McMillan and Abelson
Nickel	Ni	28	58.71	2, 3	Cronstedt
Niobuim (Form. Columbium)	Nb	41	92.906	3, 5	Hatchett
Nitrogen	N	7	14.0067	3, 5	Rutherford
Nobelium (c)	No	102	256*	.....	Ghiorso, et al.
Osmium	Os	76	190.2	2, 3, 4, 8	Tennant
Oxygen	O	8	15.9994a	2	Priestly, Scheele
Palladium	Pd	46	106.4	2, 4, 6	Wollaston
Phosphorus	P	15	30.9738	3, 5	Brandt
Platinum	Pt	78	195.09	2, 4	Ullos
Plutonium	Pu	94	242*	3, 4, 5, 6	Seaborg, et al.
Polonium	Po	84	210**	.....	P. and M. Curie
Potassium	K	19	39.102	1	Davy
Praseodymium	Pr	59	140.907	3	Welsbach
Promethium	Pm	61	147**	3	Glendenin and Marinsky
Protactinium	Pa	91	231*	.....	Hahn and Meitner
Radium	Ra	88	226*	2	P. & M. Curie, Bemont
Radon	Rn	86	222*	0	Dorn
Rhenium	Re	75	186.2	.....	Noddack and Tacke
Rhodium	Rh	45	102.905	3	Wollaston
Rubidium	Rb	37	85.47	1	Bunsen, Kirchoff
Ruthenium	Ru	44	101.07	3, 4, 6, 8	Claus
Samarium	Sm	62	150.35	2, 3	Boisbaudran
Scandium	Sc	21	44.956	3	Nilson
Selenium	Se	34	78.96	2, 4, 6	Berzelius
Silicon	Si	14	28.086a	4	Berzelius
Silver	Ag	47	107.870b	1	.....
Sodium	Na	11	22.9898	1	Davy
Strontium	Sr	38	87.62	2	Crawford
Sulfur	S	16	32.064a	2, 4, 6	Eckeberg
Tantalum	Ta	73	180.948	5	Perrier and Segre
Technetium	Tc	43	99**	6, 7	Von Reichenstein
Tellurium	Te	52	127.60	2, 4, 6	Masander
Terbium	Tb	65	158.924	3	Crokes
Thallium	Tl	81	204.37	1, 3	Berzelius
Thorium	Th	90	232.038	4	Cleve
Thulium	Tm	69	168.934	3	Gregor
Tin	Sn	50	118.69	2, 4	.....
Titanium	Ti	22	47.90	3, 4	d'Elhular
Tungsten (Alternate Wolfram)	W	74	183.85	6	Klaproth
Uranium	U	92	238.03	4, 6	Sefstrom
Vanadium	V	23	50.942	3, 5	Ramsay, Travers
Xenon	Xe	54	131.30	0	Marignac
Ytterbium	Yb	70	173.04	2,3	Gadolin
Yttrium	Y	39	88.905	3	.....
Zinc	Zn	30	65.37	2	Klaproth
Zirconium	Zr	40	91.22	4	

[a] Atomic weights so designated are known to be variable because of natural variations in isotopic composition. The observed ranges are: hydrogen ± 0.00001; boron ± 0.003; carbon ± 0.00005; oxygen ± 0.0001; silicon ± 0.001; sulfur ± 0.003.

[b] Atomic weights so designated are believed to have the following experimental uncertainties: chlorine ± 0.001; chromium ± 0.001; iron ± 0.003; bromine ± 0.002; silver ± 0.003.

[c] The name for element 102 (Nobelium) is still in doubt because its discovery is in dispute.

PERIODIC CHART

1a	2a	3b	4b	5b	6b	7b	8	1b	2b	3a	4a	5a	6a	7a	0	Orbit
1 +1 H −1 1.00797 1															2 0 He 4.0026 2	K
3 +1 Li 6.939 2-1	4 +2 Be 9.0122 2-2									5 +3 B 10.811 2-3	6 +2 +4 −4 C 12.01115 2-4	7 +1 +2 +3 +4 +5 −1 N 14.0067 −2 −3 2-5	8 −2 O 15.9994 2-6	9 −1 F 18.9984 2-7	10 0 Ne 20.183 2-8	K-L
11 +1 Na 22.9898 2-8-1	12 +2 Mg 24.312 2-8-2	Transition Elements					Group 8		Transition Elements	13 +3 Al 26.9815 2-8-3	14 +2 +4 −4 Si 28.086 2-8-4	15 +3 +5 −3 P 30.9738 2-8-5	16 +4 +6 −2 S 32.064 2-8-6	17 +1 +5 +7 −1 Cl 35.453 2-8-7	18 0 Ar 39.948 2-8-8	K-L-M
19 +1 K 39.102 -8-8-1	20 +2 Ca 40.08 -8-8-2	21 +3 Sc 44.956 -8-9-2	22 +2 +3 +4 Ti 47.90 -8-10-2	23 +2 +3 +4 +5 V 50.942 -8-11-2	24 +2 +3 +6 Cr 51.996 -8-13-1	25 +2 +3 +4 +6 +7 Mn 54.9380 -8-13-2	26 +2 +3 Fe 55.847 -8-14-2	27 +2 +3 Co 58.9332 -8-15-2	28 +2 +3 Ni 58.71 -8-16-2	29 +1 +2 Cu 63.54 -8-18-1	30 +2 Zn 65.37 -8-18-2	31 +3 Ga 69.72 -8-18-3	32 +2 +4 Ge 72.59 -8-18-4	33 +3 +5 −3 As 74.9216 -8-18-5	34 +4 +6 −2 Se 78.96 -8-18-6	35 +1 +5 −1 Br 79.909 -8-18-7	36 0 Kr 83.80 -8-18-8	-L-M-N
37 +1 Rb 85.47 -18-8-1	38 +2 Sr 87.62 -18-8-2	39 +3 Y 88.905 -18-9-2	40 +4 Zr 91.22 -18-10-2	41 +3 +5 Nb 92.906 -18-12-1	42 +6 Mo 95.94 -18-13-1	43 +4 +6 +7 Tc (99) -18-13-2	44 +3 Ru 101.07 -18-15-1	45 +3 Rh 102.905 -18-16-1	46 +2 +4 Pd 106.4 -18-18-0	47 +1 Ag 107.870 -18-18-1	48 +2 Cd 112.40 -18-18-2	49 +3 In 114.82 -18-18-3	50 +2 +4 Sn 118.69 -18-18-4	51 +3 +5 −3 Sb 121.75 -18-18-5	52 +4 +6 −2 Te 127.60 -18-18-6	53 +1 +5 +7 −1 I 126.9044 -18-18-7	54 0 Xe 131.30 -18-18-8	-M-N-O
55 +1 Cs 132.905 -18-8-1	56 +2 Ba 137.34 -18-8-2	57*+3 La 138.91 -18-9-2	72 +4 Hf 178.49 -32-10-2	73 +5 Ta 180.948 -32-11-2	74 +6 W 183.85 -32-12-2	75 +4 +6 +7 Re 186.2 -32-13-2	76 +3 +4 Os 190.2 -32-14-2	77 +3 +4 Ir 192.2 -32-15-2	78 +2 +4 Pt 195.09 -32-16-2	79 +1 +3 Au 196.967 -32-18-1	80 +1 +2 Hg 200.59 -32-18-1	81 +1 +3 Tl 204.37 -32-18-3	82 +2 +4 Pb 207.19 -32-18-4	83 +3 +5 Bi 208.980 -32-18-5	84 +2 +4 Po (210) -32-18-6	85 At (210) -32-18-7	86 0 Rn (222) -32-18-8	-N-O-P
87 +1 Fr (223) -18-8-1	88 +2 Ra (226) -18-8-2	89** Ac +3 (227) -18-9-2														-O-P-Q

*Lanthanides	58 +3 Ce +4 140.12 -19-9-2	59 +3 PR 140.907 -20-9-2	60 +3 Nd 144.24 -22-8-2	61 +3 Pm (145) -23-8-2	62 +2 +3 Sm 150.35 -24-8-2	63 +2 +3 Eu 151.96 -25-8-2	64 +3 Gd 157.25 -25-9-2	65 +3 Tb 158.924 -26-9-2	66 +3 Dy 162.50 -28-8-2	67 +3 Ho 164.930 -29-8-2	68 +3 Er 167.26 -30-8-2	69 +3 Tm 168.934 -31-8-2	70 +2 +3 Yb 173.04 -32-8-2	71 +3 Lu 174.97 -32-9-2		-N-O-P
**Actinides	90 +4 Th 232.038 -19-9-2	91 +5 Pa +4 (231) -20-9-2	92 +3 +4 +5 +6 U 238.03 -21-9-2	93 +3 +4 +5 +6 Np (237) -22-9-2	94 +3 +4 +5 +6 Pu (242) -23-9-2	95 +3 +4 +5 +6 Am (243) -24-9-2	96 +3 Cm (247) -25-9-2	97 +3 +4 Bk (249) -26-9-2	98 +3 Cf (251) -28-8-2	99 Es (254) -29-8-2	100 Fm (252) -30-8-2	101 Md (256) -31-8-2	102 (254) -32-8-2	103 Lw		-O-P-Q

Atomic Number → 50 +2 ← Oxidation States
Symbol → Sn +4 KEY TO CHART
Atomic Weight → 118.69 ← Electron Configuration
-18-18-4

Numbers in parentheses are mass numbers of most stable isotope of that element.

7-22-83
Ken Okuba

Program Fundamentals

Program manager

Fetches load modules
Deletes " "
Transfers control between load modules
Tests the status of load modules
Searches libraries for load modules

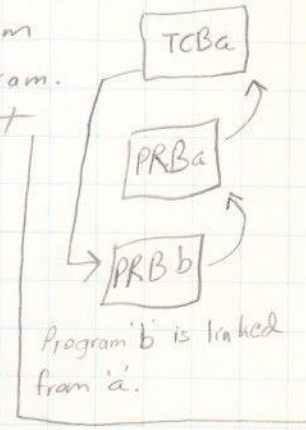
8 Macros

Load
Delete
Link, XCTL, Attach

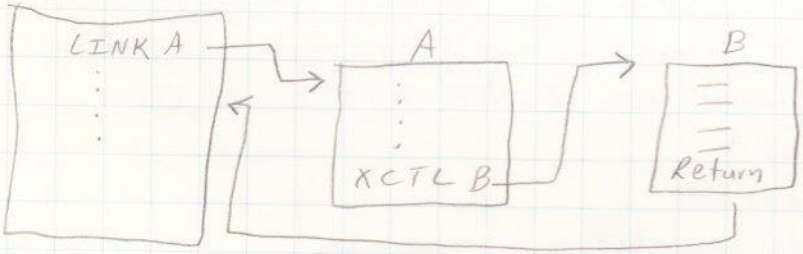
Load - when macro is expanded, the last instruction is an SVC 8 which causes the target program to be accessible to the calling program. Load may be used to quickly load tables of constants.

The SRB is a TCB for a system service. It has the capability to issue a cross-memory post. Once a program is loaded, the SRB returns with the EPA of the called program in R0. The application program can then do a BALR. Data or tables may also be loaded. Load does not create a PRB. All other PM macros create PRBs.

LINK - (SVC 6) Program manager executes a load and then gives control to the loaded program. When the called program terminates, Program Manager returns control to the original program. (IEHPROG and IEHLIST Link to the same program to interpret control cards - an example.) Runs under one TCB.



XCTL (SVC 7) - causes loading & transfer of control.



Program A disappears upon issuing the XCTL. This is used to completely transfer control.

Attach vs Link (Attach is SVC 42)

With Attach the second program has higher priority, but both the first and second programs compete for CPU time. - 2 TCBs

With Link, only one TCB exists & execution is serialized. The calling program manditorily waits on the execution of the called program.

Detach - opposite of Attach. This kills the TCB and the program which was attached.

Identify - returns with addres of program.

Synch works like link, except permits called program to have different state (prob/super) or different storage key. Must be Authorized

Abends

S0C× - program interrupt

All others may be translated as an SVC number:

S322 — SVC # hex 22

SB37 — SVC # hex 37 - EOV

S806 - SVC 06 Link

This SVC is used by attach, link, load, XCTL, + Synch.

RBs have PSW and Interrupt code + ILC 2,3,4 SVC

Know PRBs + SVRB.

(Link, XCTL, ATT + Synch)

SVRBs do not hold regs.

Program Management Control Blocks

(CDE) Contents Directory Entry

Holds EPA, # of users, + Module Name.

CDE is pointed to by the RB (SRB, SVRB, PRB)

CDEs themselves are chained.

CDE holds attributes of module so Program Management knows what to do with it. For instance, if a flag says a module is in PLPA, Prog. Man would use it there + not reload it.

Attributes EPA

Not reusable

Serially reusable

Reentrant - must used ENQ if it modifies itself.

Refreshable - absolutely reentrant. (All RPA modules)

A new copy of the program can overlay the original copy during program execution.

Name

The attributes are picked up from the directory of the PDS. Use count is kept so that it may be decremented by 1 with each delete issued. When use count hits 0 the module may be really deleted.

XL Extent List

Describes storage characteristics: location in storage and length. The location in storage is not the EPA.

LLE Load List Element

Controls loading and deleting.

Holds Task and System counts to show who has called the module. A system call adds 1 to both counters. A prob prog. call adds 1 to Task counter only. An LLE is created by a load. Points to CDE. Pointed to by another LLE or the TCB (which points only the latest).

7-25-83
Ken Okuba

Review

Attach - with an ECB operand, the mother task may be notified of the completion status of a daughter task. A daughter's Post is entered is the Event Control Block. The ETXR operand specifies an entry point in the mother task to receive control upon completion of the daughter task. (see handouts, PGMØIØØ)

Program Management Module Search

To locate a load module, Program Management will search:

Jobpack area (JPAQ)
Tasklib, * Joblib, * steplib
Active LPA
Link Library

Job Pack Area - TCB +2C is the pointer to the JPAQ. Program manager begins at 2C and goes to the first CDE to check the name + thus down the CDE chain. When it gets to the correct CDE it finds if the module is reentrant.

Active LPA - (FLPA, MLPA, PLPA) search order.

CVT+BC is pointer. May only contain reentrant, refreshable modules.

Link Library - SYS1.PARMLIB (LNKLSTxx) - contains a list of libraries (SYS1.LINKLIB) to be searched.

Load List Queue pointer begins at TCB+24. This is a list of Load List Elements.

Recovery / Termination Manager

The user may Abend with the macro: Abend XXXX (user code)

This will not produce a dump unless you include the dump option and a sysudump DD card. (The CMP field of TCB Summary shows completion codes).

The SPM command may be used to turn on the program mask bits in the PSW which will then intercept over and under-flow conditions.

SPIE will intercept ØC1s (Operation Exceptions). To issue a SPIE the program must be Lockholder, disabled, or in SRB mode. Catches only program checks.

STAE - gets control after abend. May be used by Tasks. STAE is an early version of ESTAE. ESTAE allows multiple analysis. Will catch program and other type abends. STAE ran on MVT. No percolation with STAE.

The System Diagnostic Work Area - holds GPRs, FPs, completion code, PSW

If a retry is specified in the ESTAI routine, and the retry fails, the system will not retry again. Instead, the system will chain to the next ESTAI routine (percolate). The ESTAI routine can do anything it wants.

Look these up.

ESTAE gets control before ESTAI.

A task may issue an ESTAE for itself. ESTAI is an operand of the attach macro and only effects the attached module.

System routines can issue SETFRRs which intercept program interrupts.

SYND exits are only applicable to I/O errors.

(See SPL: Supervisor, p. 67)

7-26-83

TCB (from Friday)

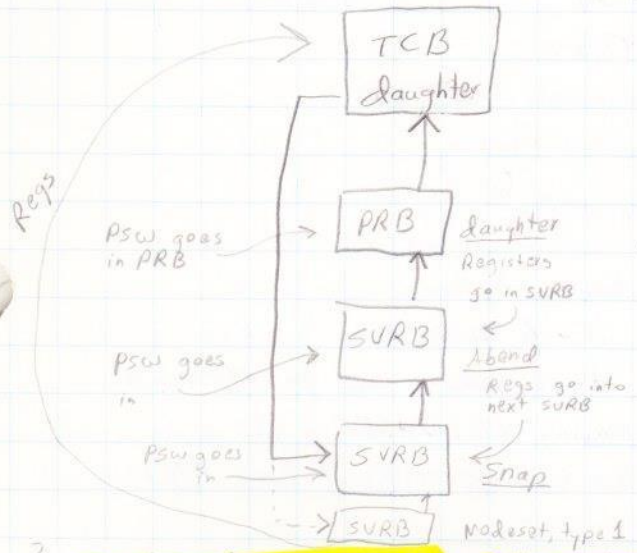
The RB field points to the most current RB and this may be either an SRB or an RB. The LLS is created with the LOAD macro and points only to the latest LLE. The ECB field within the TCB points to the mother's ECB.

The PRB holds the Interruption Code, PSW, & CDE pointer.

The CDE holds the program name, EPA

SVRBs are created by FLIH for type 2, 3, and 4 interrupts.

PRBs are created by ATTACH, LINK, XCTL, & SYNCH.



When daughter invoked Abend macro, the pointer from daughter TCB switched to the new RB. The Abend SVRB issued a recursive SVC which caused creation of an SVRB to handle SNAP dump. The pointer from daughter TCB shifted to the SNAP SVRB.

A PRB is created along with the TCB. The PRB holds PSW for the task. Type 1 & 6 SVCs cause Regs to be placed in TCB. All others save Regs in SVRB. The RB has no place for Registers. Daughter's registers are stored in the first SVRB.

When each SVC finishes, it branches to Reg 14, Exit Prologue. Exit Prologue checks to see if calling program is dispatchable. If so, it returns control to it. If not, it gives control to the dispatcher.

DASD

Objectives:
Addressing
CCHHR
TTR
Read a VTOC

Look up #Trks/cyl, #cyls, #bytes for 3380

	Trks/Cyl	Cyls	Bytes/Trk
3330-1	19	404	13030
3330-2	19	808	13030
3350	30	555	19069
3380	15	885	47,476

Direct Access is not instantaneous:
1.) Head Access Time. (30ms) } average for 3330.
2.) Rotational Delay. (8.4 ms) }

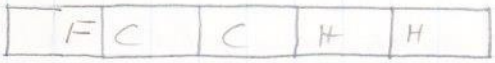
Tracks - the concentric circles of data. Each pack usually has several alternates. IEHDASDR will initialize packs + assign alternate tracks if necessary. (ICKDSF is replacing IEHDASDR)
Cylinder - the set of tracks accessible without movement of the R/W heads.
The R/W heads are numbered Top to Bottom in hex beginning with Ø.
One side of a platter acts as the Servo + the head on that track cannot write. That platter side serves to continuously calibrate head alignment. The servo platter is usually located in the center of the stack.

CC HH
0010 0012
→ picks which R/W head.
→ controls positioning of the arm.

Track layout:

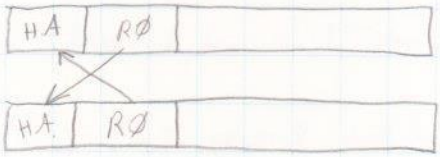
1. Each track has an index marker which can alert the CU as to where the beginning of track is.
2. Next, home address of track.
3. Track Descriptor Record -
4. Data Record
   Preceded with count and key.

Home address



0-4 unused
F = flag
bit 5 = 1 = CE, Ø = user
bits 6, 7
  00 = normal
  01 = alternate
  10 = defective
CC = cyl
HH = track #
ECC = used by hardware

Bits 6 & 7 are repeated in the count field as are CCHH. On a bad track, bits 6 & 7 will say 1Ø and the CCHH in Record Ø will point to the alternate track.



Key

CCW command "Search Key Equal or High" - used to locate (on key) a certain record in a file that is in key sequence. The key on the track holds the value of the highest key on that track. Thus, by checking only the key field, the system can quickly find which track has the record it needs.

Record Ø of every track cannot be used for data. It may only be used to point to alternate or bad track.

Record 3 - 8Ø bytes Volser and VTOC address.

An EOF marker is a Data Length of Ø.

TTR Addressing

Track/Record. Used like Base + displacement addressing.

TT = displacement

R = record on track.

For a dataset the TT is displacement from the beginning address of the dataset. For non-datasets, the TT is displaced from Ø.

In CVT, CVTPCNVT points to a Relative to Actual conversion routine. Actual-to-Relative is CVTPRLVT. The contents of these fields may be branched to. (see SPC: Data Management)

When subtracting disk addresses to get the extent, add 1 after the subtraction to account for the zero track.

```
  000E   ending address
- 0002   beginning address
------
  000C
+    1
------
  000D   length of DS in trks.
```

In CCHH format, if the ending HH is smaller than the beginning HH, borrow from the CC figure. Be sure to borrow the correct number of tracks for the device.

```
  0005   0002
  0004   0004
---------------
  0004   0002
  0004   0004
---------------
            28
          + 1
---------------
            29   tracks
```

3350

Volume Table of Contents

DADSM manages space using VTOCs. IEHDASDR and ICKDSF create the VTOC + can be placed anywhere on the pack.

The VTOC contains DSCBs.

- DSCB4 identifies VTOC
- DSCB5 identifies freespace
- DSCB0 are free DSCBs.
- DSCB1 are for Datasets.

The size of the VTOC may also be varied according to creation parameters.

A track is the smallest unit of DASD allocation.

An ISAM dataset has two DSCBs, one holds ISAM information (DSCB2) and the other points to the dataset (DSCB1).

A DSCB1 can only describe up to 3 extents. Therefore, DADSM creates a DSCB3 to hold up to 13 additional extents.

When a dataset is scratched, only the VTOC entries are changed.

MVS will support existing format 6 DSCBs, but it cannot create one.

Format 1 DSCB (see Debugging Handbook, Vol. 2)

Offset 2C of each DSCB is the DSCB identifier.

DS1SCALO - secondary allocation; 4 bytes

first byte

- 1000 0000 trks
- 0100 0000 blocks
- 1100 0000 cyl

next three bytes indicate quantity of above units.

LSTAR - last record address in TTR. Shows when space is being wasted.

NOEPV - number of extents.

Format 3 DSCB - holds 13 more extent descriptors.

Format 4 DSCB

Format Identifier is at 0. DSCB4 actually has 44 bytes of '04' prefixing the control block. This is always the first DSCB in the VTOC. The name of the VTOC data set is the 44 bytes of X'04.'

DEVSZ shows number of tracks/cylinder. We can figure device type from this.

Format 5 DSCB

First 4 bytes are X'05'.
Describes free extents with 5 bytes:
bytes 1+2 - relative track address (displaced from Ø).
3+4 - number of cylinders in Extent.
5 - number of additional unused tracks.

A single Format 5 DSCB can describe up to 26 free extents.

In examining a VTOC dump, use the count field and not the addresses on the left side of the page.

Data Management

7-27-83

Know:
DEB } function + creation
IOB }
DCB fields

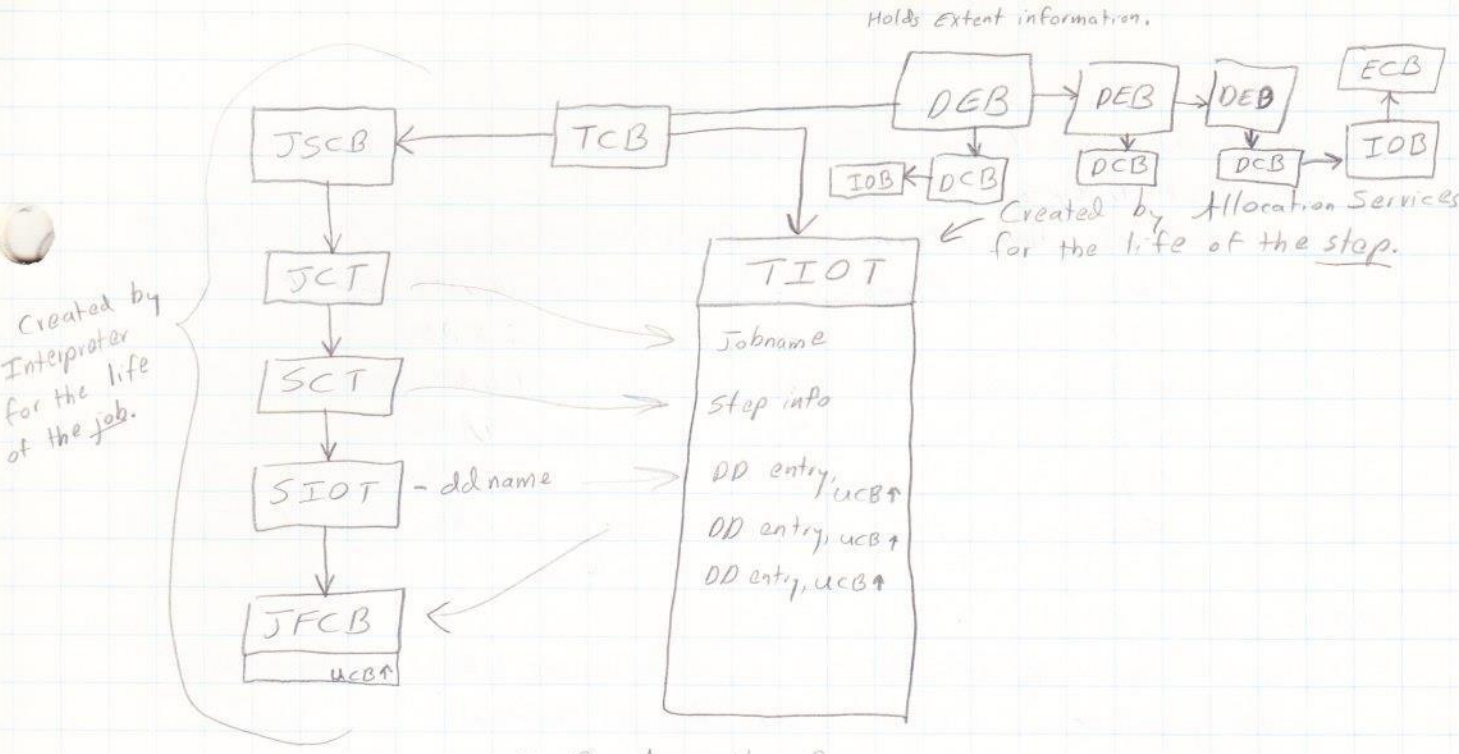
Contact with data management is established with I/O macros such as OPEN, GET, PUT, READ, WRITE, INPUT, and CLOSE. In macro expansion the last instruction branches to an Access Method which builds the Channel Program (using DCB) and issues EXCP to call SVC Ø. SVC interrupt handler does page fixing and issues SIO macro which expands and branches to IOS.

IOS has three sections:
Front End - prefix CCW, setup CAW, issue SIO.
Interrupt Handler. -
Back End -

After issuing SIO, IOS notifies Access Method which marks the user task in a wait (TCB RB).

Data Management is active during the execution phase of a job.

The Initiator's Allocation Services sets up the environment for the job's communication with data management. Allocation services uses the JFCB to know which UCBs to access.



Q+D Assembler Program

```
        OPEN    (INDD,,OUTDD,OUTPUT)
        TM      INDD+48,X'10'
        BZ      Abend
        TM      Outdd+48,X'10'
        BZ      Abend
GET     GET     INDD
        LR      R3,R1
        PUT     OUTDD,(R3)
        B       GET
EXIT    CLOSE   (INDD,,OUTDD)
        .
        .
        .
```

test status of opens.

GET locate mode → is faster than Move mode.

GET move mode take data from buffer + moves it to DS location. This is slow. Locate mode gets data + return address of buffer in R1.

```
INDD    DCB   DDNAME=INDD,DSORG=PS,MACRF=GL,EODAD=Exit
OUTDD   DCB   DDNAME=OUTDD,DSORG=PS,MACRF=PM
```

JCL

```
//INDD    DD   DSN=ABC,DISP=SHR
//OUTDP   DD   Sysout=A,DCB=INDD
```

The DCB has six formats. To determine which to use:

1.) Find DSORG

if PS	then Format	1 DCB	(uses QSAM, BSAM)
PO	Format	1	BPAM, BSAM
IS		2	ISAM
DA		3	BDAM
CX		4	BTAM
TX		5	TCAM
GS		6	GAM (Graphic Support)

OPEN will

1. Cause volser to be read
2. Verify volser
3. Verify DSN
4. Perform DCB Merge.
5. Get Access Method addresses
6. Allocate buffers
7. Flag DCB open.

DEB - holds extent information which is obtained from the VTOC. At DEB $+10_{16}$ is number of extents + number of sections in the DEB (one for each extent, holding starting + ending CCHH). DEB is created at open. The DEB is protected from users by a key of 5. (in subpool 230 in private area). DEB points back at the related DCB.

The TCB will point at the first opened DEB. Other DEBs are chained from the first. Thus, in a dump, the DEBs will appear in their open order.

Review UCBs.

TIOT is created at step initiation and is good only for the life of the step. It is created by the SWA Control Blocks. It holds DD name and pointers. When the DCB is altered during OPEN, a routine puts the address of the DDName in the TIOT in the DCB. Later, at CLOSE the name is grabed as an offset from the start of the TIOT and replaced in the DCB.

(See handouts, p. DM00140)

Access Method naming convention: IGG019aa (two alphabetic)

The EPAs of access methods are obtained by issuing LOADs and then placed in the DCB at OPEN.

DCB information can be found in the program, jcl, and data set label. Order of priority:

1. Program
2. JFCB (DD statement)
3. Data set Label

These data are combined in a forward merge + a reverse merge:

Input: Forward merge:

DSCB merges into JFCB empty fields.
JFCB merges into DCB empty fields.

Reverse merge:

DCB merges back into JFCB empty fields. (This merge simply updates the JFCB so all fields are filled. The JFCB must be complete in case the DCB is used in a refer-back).

Thus, only code Blksize and Lrecl on Output JCL statements. Do not code it in the program. On input, the Blksize and LRECL can be taken from the label.

Output

Do forward merge and then do a reverse merge with over-riding of everything in the JFCB, and then overlaying the DSCB. Why? On creation, all descriptors should match. Be careful on opening existing datasets for output, if your DCB is incorrect, the data set label will be overlaid.

The size of the buffer is usually the blocksize.
QSAM will perform automatic blocking/deblocking.
BSAM + BDAM perform an I/O for every READ/WRITE operation. Automatic blocking/deblocking is not performed. This is slow. No anticipatory buffering or synchronized I/O (thus you must issue a WAIT after a READ or WRITE). This has less overhead and more control.

To synchronize, issue a WAIT or CHECK. CHECK will honor an EODAD pointer. With WAIT the user must check for EOV.

To get to the access method, load its address in Reg. 15 and branch. Since the access method is branched to, it also runs in protect key of 8. The access method is not doing anything that the problem program cannot do.

The Access Method gets storage in the private area to build the IOB and channel program. The IOB is the communication area between Access Method and IOS.

The IOB points to the channel program. It gives this address to the SVC Ø routine (EXCP) which page fixes and translates virtual addresses in CCW. Control is then passed to IOS for scheduling. IOS loads the CAW and issues the SIO. Upon return, the IOS interrupt handler will post the CSW in the IOB.

IOB has a prefixed area which varies from -4 to -8 with the access method used. The IOB points to the ECB

Close processing

1. Write partially-filled buffers
2. Update DSCBs
3. Restore DCBs.
4. Remove DEB from DEB chain.
5. Flag DCB not Open
6. Perform data set disposition if FREE=CLOSE was specified.
7. Create trailer labels for tapes.

EXCP

7-28-83

The problem program issues a GET/READ which invokes the Access Method. Access Method creates IOB and Channel Program, then issues EXCP macro. Control goes to SVC Ø routine which page fixes, translates the channel program into another copy with real addresses, and issues STARTIO macro. Control goes to IOS which adds three CCWs to the front of the channel program: SEEK, SFM (Set file mask to prevent another seek), and TIC to the next CCW. IOS gets the SEEK destination from the IOB.

CCW must be aligned on doubleword boundary, and has four operands:

- Command Code
- Data Address
- Flags
- Byte Count

Read Data:

```
CP   CCW X'Ø6',WKAREA,Ø,12000      (12000 = size of block)
WKAREA DS  CL12000                 ← buffer
```

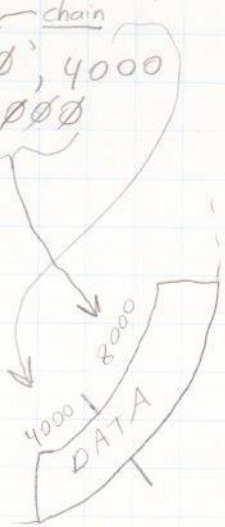
Command Chain

```
CP     CCW  X'12',KNT,X'40',8      Read Count CC
       CCW  X'06',WKAREA,Ø,12000   Read Data
WKAREA DS   CL12000                Buffer for Data
KNT    DS   CL8                    Contains Count
```

Data Chain - implies command chaining

```
CP     CCW  X'Ø6',DATA1,X'8Ø',4000   Begin Read   (X'8Ø' = chain)
       CCW  X'Ø6',DATA2,Ø,8ØØØ       Continue
Data2  DS   CL8000
Data1  DS   CL4000
```

ignored



The byte counts must sum to the block size.

The second CCW command is the same as the first.

Data Chaining

Read Key and Data

```
CP      CCW   X'0E', wkkey, X'80', 14      in this example the length
        CCW   X'0E', wkarea, 0, 12000       of the key is 14 bytes.
wkarea  DS    CL12000
wkkey   DS    CL14
```

Skip and Data Chaining

```
CP      CCW   X'06', DATA1, X'80', 2000     Read Data. DC
        CCW   X'06', 0, X'90', 6000
                      ↑ buffer 0, thus skip these 6000 bytes
        CCW   X'06', DATA2, 0, 4000         Continue reading
```

SLI

```
CP      CCW   X'06', DATA1, X'20', 3000
```

Suppress Length Indicator (SLI)

Read 3000 bytes

Be careful with this one. No error indication will be issued for an incorrect length. This is used for reading only part of a block. The DASD will read the whole block, but the channel keys the first 3000 and dumps the rest in the bit bucket. The device cannot transmit less than a block.

If a physical block is smaller than the requested amount the channel will insert into the residual count field of the CSW the number of bytes not read.

Selecting Records

CCHHR

This location minus 8

```
CP      CCW   X'31', SID, X'40', 5        Search ID=, CC
        CCW   X'08', (*-8), 0, 0          TIC back until found
        CCW   X'06', wkarea, 0, 4000      Read Data
```

```
SID     DC    X'000000000E'               ← record 14₁₀ (from IOB).
wkarea  DS    CL4000
```

When the control unit hears an equal comparison it skips the next CCW (which is the TIC). This is how the correct record is read after a seek has positioned the head.

If the correct record number is not found after two passes of the Index Point, the channel will signal a unit check.
(see handouts EXØØ23Ø)

32K is the maximum possible blocksize. (3 digits in CCW)

Indirect Address List (IDAL)

```
CCW     X'Ø6' IDALIST, X'Ø4', 7168
                              → use IDAL
IDALIST DC  X'00010800'   → Real storage addresses
        DC  X'0001A000'
        .
        .
        .
```

When system sees 'Ø4' it will go to the address list and store 2K of data beginning at each address contained in the list. (Handout, EXØØ25Ø)

DEB

At OPEN the DEB is created with information from VTOC. Each DEB has a Basic Section and then a DASD Section for every extent. Check the #Extents in the Basic section for the number of DASD Sections. The DASD section will have the beginning CCHH of every extent. This information is given to the IOB.

IOB

We must supply:
- Address of Channel Program
- " " DCB
- " " ECB
- Disk seek address
- Flags describing type of channel program. (We will use command chaining, X'42')

IOS will supply:
- Completion code → the right 7 bytes of the CSW. } General Info
  - Unit Exception
  - Unit check
- Sense Information (see Debugging Handbook, vol 1., p. 4-32)
  - Device dependent codes.

ECB

A one byte flag and an address

Flags:

7F	Successful I/O
41	Channel Program Exceptional Condition
42	Out-of-Extent track address
44	Previous request had device error, reissue
4F	

To code an ECB:

```
DC   F'0'
```

To code a DCB:

```
INDCB    DCB   DSORG=PS,DDNAME=,MACRF=E
```

Only required parameters.

The IOB could be coded with a series of DCs and DSs. However, a macro will build a DSECT. The macro is IEZIOB. It allocates storage and assigns the field names. (see EX0036 in handouts)

The DEB can be coded with the macro IEZDEB.

Since EXCP is slow, a faster version exists which eliminates the Access Method — EXCPVR. JES2 uses this.

```
IC    Reg, Stor    1 byte
STC   Reg, Stor    1 byte

LA    R1, 1(R1)    - adds 1 to R1

STCM  R5, B'1001', Storage   3 operands
```

STCM
ICM

R5 | / | / | / | / | ← the mask dictates the quantity placed in register.

Storage | X | X | / | / |

VM

7-31-83
Don Stern

When VM is IPLed, it first looks in real storage for a flag to see if he has already been up. The field can contain "COLD," "CPCP," "WARM," or zeros.

VM tries to establish communication with the primary system console. If unavailable, it will go through a list of alternate consoles.

The first line of display after IPL will show the system name and release, and give the date the nucleus was generated.

Types of VM starts:

COLD - formats spooling and paging area. All spool files will be lost. Must be done between major releases of VM.

WARM - assumes a normal shutdown had been done. This is the default option.

CKPT - No status saved at previous shutdown. System looks at page, spool, and EREP to re-establish pointers. Takes more time than warm.

FORCE - System will try to re-establish pointers if possible but will sacrifice data if necessary.

DRAIN is an option with any startup option. It leaves all terminals except console disabled, as well as the printers and the punch.

SHUTDOWN may also be issued as a sub-option. This causes a normal, soft, shutdown after completion of IPL.

After OPERATOR is logged on, the system will display storage, nucleus, and paging statistics.

VM Messages and Codes:

CCC MMM NNN T

- T → type code → Warning, Information, Error, Severe
- NNN → number
- MMM → module code
- CCC → component code

Abend Code (in VM Messages and Codes)

MMMNNN
- NNN → code number
- MMM → module code

SET FAVOR - the operand is a percentage of time slicing.

VM/PE

Supports MVS and VM to run independently under one CPU.
Note: MVS is not running under VM.
MVS runs ½ to ¾ times slower under VM than standalone.
With PE, MVS almost runs with standalone efficiency.
VM comes up first, then MVS, and then PE is turned on.
MVS is brought up as a virtual machine in a V=R area.

MVS

MVS can support a maximum of 9,997 address spaces.

K, A, NONE will cause the entire MVS console to be used for system messages.
K, D, PFK will display PF key assignments.
K, S, DEL=RD, RTME=1
- RD → Roll Delete
- RTME → Roll Time

K, E, D - clears the lower message area.
K, S, CON=Y, SEG=20 ← clears 20 lines at a K command
D PFK - displays PFkeys

D A, L - List all active tasks. (VTAM is usually called NET)
D A, TS - Displays TSO users.
D R, L - Displays outstanding requests.
D C - Displays console address assignments and authority.
D M=CPU - Display Message
- DEV
- STOR
- High

D, U, DASD - Display Unit (DASD)
TAPE (Tape)
,,, XXX, 1 device

SEND 'message', BRDCST
, OPERATOR=routecode
, CN=console
, USER=userid, LOGON
, NOW

IEA101 Responses: Specify System Parameters
U - nothing (or hit Enter)
SYSP - identifies secondary parameters
CLPA - clears LPA
CVIO - Clear VIO datasets.

To set time:
R 00, 'DATE=83.349, CLOCK=21.02.00'

Cancel, jobname , DUMP - immediately kills task
U=userid , DUMP
unitaddr.
devicetype
identifier
S DEALLOC - starts a dummy job to force an offline to complete.
STOP, P } jobname - soft close of task
procname
identifier
HALT, Z } EOD - EOD= End of Day, shuts MVS down and closes SMF. This quiesces MVS.
Vary XXX, ONLINE
OFFLINE
The S DEALLOC is necessary in order to cause JES to look at the queues to see the offline request.
V Path XXX, ONLINE
OFFLINE
M XXX, VOL=(SL, volser), USE=PRIVATE - mounts DASD
PUBLIC
STORAGE
UNLOAD unitaddr - de-mounts a device
QUIESCE - loads disabled wait. Reverseable only with RPSW.
Vary STOR (ddddK, ddddK), OFFLINE
CH
CPU

JES

Several JES systems can share a common spool (workload). JES will talk directly to VTAM which can connect it to RJE locations.

$SPRT1 - starts printer 1
$DU - displays unit record devices + shows device assignments.
$PPRT1 - Drains printer 1
$HPRT1 - immediate stop
$FPRT1 - immediate flush
$I - interrupt (flush with hold) print
$E - restart printer
PUN1 - punch device
RDR1 - reader device
LNE1 - line
RLN1 - remote line
$T - set
$PJES2 - drain JES2.
$SI2 - start initiator 2
$PI1 - drain initiator 1
$DA - Display active JES jobs
$DI - displays initiators
$DN,Q=XEQB - displays all jobs in queue class 'B'
$DQ - displays queues
$PA - drains everything
$SA - start "
$DJxxx - display job by JES number
$H - hold
$AJxxx - release job xxx

8-1-83

VM Overview

Components:

1. CP - highest level of operating system; Interrogates the hardware.
2. CMS - runs on a virtual architecture.
3. IPCS
4. RSCS

VM simulates a CPU environment. It creates a Virtual architecture. Each Virtual Machine "has" a CPU, storage, and channels. These are simulated by CP. CP translates all virtual activities into real actions. CP intercepts any privileged instructions via a program check and reissues the request for a real service.

VM is not the best environment for a guest batch system because CP must step in and translate every privileged request. Interactive users run much better because they make fewer privileged requests.

If MVS runs virtual under VM and can handle it's own paging, three storage levels result.

1. Real Storage - handled by VM
2. Virtual Storage which MVS sees as Real Storage.
3. Virtual Virtual Storage controlled by MVS.

Unless care is taken, VM can page out a page which MVS wants to page out - causing VM to do a page in so that MVS can page it out.

In VM, everything except tape drives can be virtual. This includes DASD, TP lines, printers, punches, and consoles.

A Virtual Machine is described to VM in a Directory. The Directory entry contains the userid, password, max storage, normal storage, command class privileges, account number, Distribution code, virtual UR devices (RDR, PUN, PRT), links to other virtual machines, and mini-disks.

Any IBM device can be simulated by CP; and this device does not need to be backed by a real device.

CMS lives on MAINT's 190 mini-disk. Languages and utilities live on Maint's 19E.

A real VM pack has a dummy VTOC which tells an OS system that the whole pack is allocated. VM does not use the VTOC. The corresponding information is kept in the Directory. The Directory lives on disk but is imbeded into the nucleus at IPL and at other times.

A CMS minidisk may be formatted into an OS pack. It has a VTOC and a relative cylinder Ø. It looks just like an OS pack but, if shared, has fewer cylinders. An OS virtual machine may also own real DASD. VM allocates only in cylinder increments.

A particular virtual machine may issue CCWs, but CP will intercept these and translate them into valid cylinder-track addresses.

CP Commands

#CP - prefix will permit communication with CP. The # bypasses the CMS command translation search order.

At logon, CP will read the directory entry and allocate a virtual machine. The virtual machine is not completed until the password is entered.

Entering "#CP" causes a virtual machine to stop. So does reconnecting to a disconnected machine. Thus, after reconnect, enter Begin to resume running.

Messages:

DMK - CP
DMS - CMS
DMM - IPCS
DMT - RSCS
} component

DMKCSPØ22E
- DMK, CSP: component
- Ø22: message code
- E: type code
- module (usually ignore this).

If you explicitely IPL 190, you will get a whole new copy of CMS. Instead, issue IPL CMS and access the shared segment CMS nucleus. The shared segment is locked in storage and is faster.

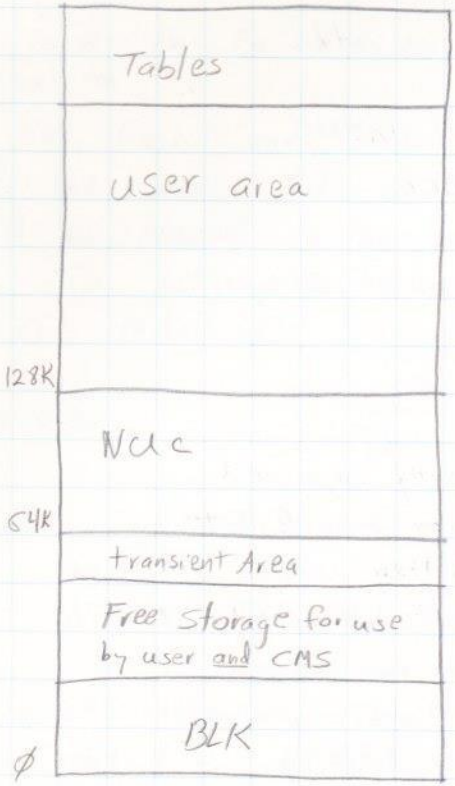
A # may be entered in a PF key if it is preceeded by a double quote "".

SP PRT FOR CMSØØ2
- FOR: uses the header of CMSØØ2 for the printout.

When Status is:

CP READ - the virtual machine is stopped.
VM READ - the virtual machine has issued a Read to the console.
Not Accepted - VM can buffer three commands.

CMS Storage Layout



CMS does not issue privileged instructions. Instead, it issues Diagnose commands.

CMS runs in BC mode with no multitasking

CMS commands are really pointers to the routine in the nucleus to carry out the function.

Usually considered as the Nucleus.
This area is usually shared between all CMS users.

Command Search Order:

1. #CP
2. EXEC
3. Module - all CMS commands are modules } looks at every disk.
4. CP

Format creates a: Master File Directory
Volser

Disk Access defaults:

191	A
192	D
190	S
19E	Y

Mini disks are searched in descending FM order.

Reserved Filetypes

Listing	Script
Text	Exec
Module	Macro
Txtlib	Copy
Maclib	Cobol
Map	Fortran
	Assemble

COPY fn ft fm fn ft fm fn ft fm fn ft fm (Append Oldate
↑ uses date of last file.

concatenates these files into this

COPY MY FILE A = = = (LRECL 40 TRUNC
Copies a file into itself changing the LRECL.

The native CMS sort is a descending hex sort which stinks.

8-2-83

CMS Editors

CMS came out with a poor line editor. This was greatly improved with Edgar, a full screen editor. However, Edgar could not be used with TTY devices. SPF/TSO was written under VM and even was released for VM. XEDIT is the most powerful, versatile editor in existence. It is fullscreen and can run on a TTY device as a line-editor. XEDIT can look like any of the other editors. XEDIT can perform four file split screen editing. It can handle recursive editing.

STATUS will display the edit parameters for a particular file. The parameters may be different from file-to-file depending on the File Type defaults.

XEDIT is run with Macros. Therefore, the user may write his own macros to customize editing or to make XEDIT look like another editor. Commands may also be customized with a synonym table.

LOCATE - may be used in reverse: -/
-/ = find a line that does not contain the string
/|/ = ORs two strings per line.
/&/ = ANDs two strings per line.
Ands and ORs may be combined.
FIND only searches strings beginning in column one.

Enter a CH command but do not press Enter. Press PF5 & XEDIT will do a locate. Press PF6 and XEDIT will process the change.

PF Defaults:

1 Help	6 see above	11 Join Line
2 Add a line	7 FO	12 Go home
3 Quit	8 BA	
4 Tab	9 =	
5 see above	10 Split Line	

Program Development

All compilers under VM are the same as the OS compilers.

A Fn Fn ← assembles two source files

The Object and Listing files will be placed on the home disk of the source file if access is write; if not, the output will be placed on the highest disk with write access.

To set up the Mac and TXT libs, use a Global:

GL MACLIB CMSLIB OSMACRO
FN FN ← assumed Ft of "maclib"
this is also the search order

GL TXTLIB Fn Fn
↳ assumes Ft of TXTLIB.
↳ Defines object libraries necessary for link-edit

LOAD - performs link edit and produces an in core load module. Output in LOAD MAP which lists all CSECTs and DSECTs in the module. Loads from 128K in user area. This load point may be explicitely stated.

START transfers control to program residing at 128K. If you want to start at another origin, an address may be specified. Program PARMS may also be passed at START.

Filedefs are the DD statements describing the I/O files.

FILEDEF INPUT TERM
OUTPUT "
} use terminal to display + read.

RDR
PRT
PUN
} this is sufficient

DISK Fn Ft Fm (DCB . . .
optional

LKED will directly produce a relocatable load module and write it to disk. If a program has been LKED'ed the OS lked will be used and to execute the program issue the OSRUN command.

A CMS user may create his own macro libraries. To do that, issue:

MACLIB GEN Fn Fn Fn Fn ...

The Ft is assumed to be MACLIB. The library will be placed on the first write disk. Each FN will be a PDS member of a new MACLIB.

MACLIB ADD Fn Fn ← will add PDS members to the Maclib.

The first Fn is the name of Macro library, the following FNs are the macros.

MACLIB DEL Fn ← deletes a macro.

" REP Fn ← replaces an existing macro.

" MAP Fn ← creates a list of all members.

Object libraries may be created in the same way. (REP excepted) The assumed Ft is TEXT.

The GENMOD Fn command will take a load module in storage and create a relocatable load module with a Ft of MODULE. Thereafter, to execute this program, simply enter the Fn.

Synonyms are created using a file with a Ft of SYNONYM. The entries list the CMS/CP command and the synonym to use, and the number of significant characters.

EXEC2

8-3-83

EXEC arrays may be built with:

&&2 Since translation occurs from Right to Left, the first &2 will be substituted, and then the second

&2 = 1
&&2 = &(value of &2) = &1

(UMDOC - RPASS)

EXEC2 was supposed to be a macro processor for XEDIT and was not really intended to replace EXEC.

L (E ← creates a list of files with the name CMS EXEC. Each entry will appear as:

Ln1 Ln2 Fn Ft Fm

An EXEC2 file with an Ft of XEDIT functions as a macro command for XEDIT. This file may contain synonyms, for example.

EXEC2 pulls the entire EXEC2 program into storage for execution. The old EXEC used to reference the disk for each line.

VM Update Commands

VM does not use SMP. Instead, one command is used to apply all maintenance: UPDATE.

EXECs control the use of the UPDATE command.

UPDATE Fn Ft Fm (source to be updated) Fn Ft Fm (contains update) (CTL ← specifies multiple updates. → CNTRL

This produces three files:

Fn UPDLOG - A list of updates (all changes) - insertions, deletions, replacements
List of all cards affected by the changes.

Fn UPDATES - A list of all files used to perform the updates.

$Fn Ft - Ft is usually Assemble.
This is the output; the updated file. The actual source is never over-written. All PTFs are applied to the static source, not to a previously updated file.

The CNTRL files gets a file of control cards. Each card references the sequence number of a particular source card. The sequence numbers appear in columns 73-80 of source.

Update control commands:

./ I - means insert

./ I 100 # 110 10

- I → Insert
- 100 → after line 100
- # → perform renumbering
- 110 → start resequencing at 110
- 10 → sequence by 10

./ D 300 ← delete line 300

./ D 400 500 ← delete lines 400 through 500, inclusively.

If an error occurs, UPDATE will not stop. It will issue messages to screen and UPDLOG and then continue.

./ R 600 ← Replace one card
./ R 400 500 # ← replace a range and resequence.

./ R 600 # 615 5 ← replace 600 with the next five cards and renumber beginning at 615. It is good to renumber even if only one card is being replaced so that others will know the line has been changed.

./ * → Comment Line. Only appears in the UPDLOG.

The user is responsible for insuring that a usermod will not interfer with application of the PTF.

The controls cards and updates are contained in files

ModnameComponent PTF number
↳ Fn ↳ Ft

When update is issued, the source is read into core. Then it opens the CNTRL file and reads from the bottom. System will then chain to the update files. The update log will be opened to record the changes.

Every PTF tape includes all previous PTFs which IBM is interested in having applied. Also, APARs are obsoleted by the new PTF.

Program Products

VM rel 1 through 6 was free. Then: BSEPP - Basic System Extension Program Product and SEPP which cost + had to be added to the free VM.

BSEPP enhanced CMS.
SEPP enhanced an OS under VM.
VM/SP replaced the above + cost.
Then, VM/SP rel. 2 had HPO option.
See the Amdahl Software Availability Schedule to correlate all PP levels.
VM/XA will be coming soon.

RSCS comes with the base VM. However, RSCS has several related software packages. For the super version of RSCS the user must use the Program Product RSCS.

Same for IPCS. The IPCS which comes with VM is barebones. The IPCS Program Product has all the bells and whistles which everyone really needs.

The RSCS PP + IPCS PP have release versions independent of VM itself.

PVM interfaces another CPU to a VM system. The terminals on the other CPU "Pass-through" PVM and PVM makes these devices look like local terminals.

DMS - Display Management System: builds a menu system. (like SPF on CICS)

QBE - Query By Example: a RDBMS. One of the first good relational systems.

DIRMAINT - a PP for directory maintenance.
VMAP - Performance Analysis and Capacity Planning.
SMART - Realtime Performance Analysis.
IIS - Interactive Instructional System.

8-4-83

MVS/SE - Microcode Dependent.
MVS/SEA - Permits MVS/SE to run on a 470. Translates new privileged instructions and speeds up other instructions. Will run on an IBM.
VM/SE - helps MVS run under VM
VM/SP - " " " " "
VMA - hardware unit for expansion of certain instructions. 20-30% improvement.
VM/SA - Amdahl's improved VMA, but runs entirely in software. apx $500/mo.
VM/PE - Permits both VM and MVS to run concurrently in supervisor state. (Amdahl) Now at rel. 4.

SESSION (VM/SES) - future Amdahl product:
One tube dialed to session can then act like up to twelve tubes. PA2 takes you to the menu. Thus, one system programmer can have multiple tubes for system testing. SESSION itself runs in a virtual machine. The user DIALs to SESSION.

VM/PE (covered in a 3 day class): Modifies a major piece of MVS as well as VM. PE owns page Ø.
VM comes up first. A V=R area is allocated for MVS. Within the VM area, PE lives with a VM page Ø, and a PE page Ø. From VM, enter SET VM/PE ON. VM then waits for MVS. On MVS, enter: S VMPEINIT. Thereafter, MVS cannot load a LPSW with a wait bit, a SIO, or certain error instructions. VMPEINIT identifies each of these very privileged instructions in MVS and passes the locations to VM. VM then goes through MVS code and replaces each VPI with an SVC. Thereafter, interrupts branch to PE which decides who the interrupt belongs to. PE will

then load the appropriate Page Ø into real Page Ø and allow the system to run.

Unfortunately, PE is a user mod and nearly every PTF affects storage areas stolen by PE. The site SE usually must resolve PTF conflicts.

MVS deals with real devices.

MVS performance is only about 3% slower than standalone. VM performance does not degrade appreciably.

This system avoids VM having to simulate and translate every privileged instruction issued by MVS.

If either machine crashes the other can continue. In addition, the down system can come back up without having to bring down the up system (however, MVS must be put in stopped).

PE sometimes swaps page zeros up to 700 times per second.

Machine checks are handled by whoever has the machine at the time (except for PE which gives it to VM).

PE adds 5 modules to VM & changes 29 others. MVS gets 1 new SVC and 3 programs.

Amdahl has over 60 users of PE.

5 79
1 SVC
3 prog's

VM and OS datasets

A CMS user may link to an OS disk, then CISTDS the VTOC, & CISTPDS any PDS files. The OS dataset may then be accessed with FILEDEFs and MOVEFILE.

(label) (keyword for OS dataset)

FILEDEF IN * DSN Sys1 Proclib (member

CMS will then search the OS packs for the data set and member. FILEDEF OUT DISK (Call Macro A

(new name)

MOVEFILE is the module which will move the dataset or member from the OS disk to the CMS disk.

DIAG Interface

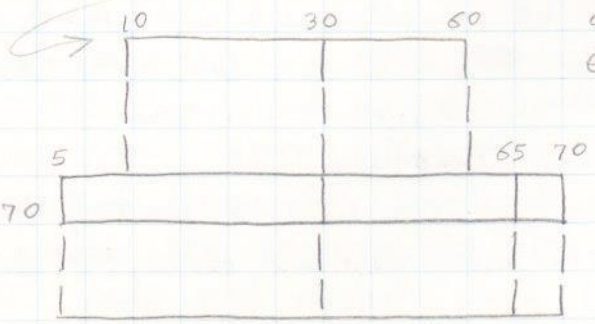
CMS does not issue SIOs. DIAG (X'83') is an instruction used to talk to virtual devices and issue CP and CMS instructions.

SCRIPT

Some script commands come with the base VM. These commands are used for the HELP facility. These are:

.cm - comment in script file + will not print (or .*).
.fo - Format command
"on" - perform reformatting
"off" - print exactly as in source.
.il n - indent line (has an implied break)
.in n - left margin indentation. (May use "+" "-" to be relative).
.of n - offsets text n spaces after the next line. This is relative to current margin.
.sp n - Blank line. If "n" is omitted, one line is given.
.ll - line length (set effective right margin). This is relative to the edge of the page.
.pa - page eject.
.br - break; will end a formatting unit and begin another.
.us text - underscore the text on the same line.
.us on - turns on underscore
.us off - turns off underscore
.us n - underscore next n line
.ce on/off | data | number - centering on/off, text, following lines.
.tt L/C/R - Top title
L = left position on top of page
C = center
R = right
Use slashes as delimiters for contents in positions.
.bt L/C/R - bottom title
.im filename - imbed filename
.bx n n n - Box; n is the vertical bar columns.

.bx 10 30 60



The columns are absolute. They are not effected by indentation.

.bx 5 30 65 70
.bx

.bx off
.bx 10

8-3-83

VM Update

DMKIOS ASSEMBLE ← the source file

DMKIOS MODUF1
Fixes MODX1
6.

DMKIOS MODX1
usermod
5.

usermod

DMKIOS UPDTFØØ1
Apar
4.

Apar from IBM

DMKIOS 5123980K
PTF
3.

DMKIOS 5110130K
PTF
2.

DMKIOS 5101030K
PTF
1.
* / cards
* updates

From PTF tapes →

DMKIOS AUXMOD
hold Fts of usermods.
MODUF1 * comments
MODX1 * comments
listed bottom up.

DMKIOS AUXPTF
Hold Fts of PTFs
5123980K
5110130K
5101030K
listed bottom up

A type of global file.

DMKSP CNTRL
last card on top.
TEXT MACS CMSLIB
MOD AUXMOD
fix UPDTFØØ1
assumes it points directly to the update.
PTF AUXPTF
listed bottom up.

specified in the UPDATE command

UPDATE DMKIOS ASSEMBLE * DMKSP CNTRL *
CTL NOSEQ8 STK
usermods don't use all seq. numbers.

VM System Generation (See handouts for examples)

IPO is a set of programs which assist in the installation of VM. It prompts for specification of gen parameters. VM changes are contained in SLUs. After SLUs are applied the user must then apply a PUT from Amdahl which adds RMS and other fixes.

AMS - Amdahl Maintenance System is our answer to IPO. It is easier to use and is inclusive of Amdahl options.

An upgrade tape is the Amdahl PTF for Amdahl RMS to be applied after an IBM SLU.

Know the PUT and SLU levels when calling the support center.

A VM disk differs from OS disks.

Cyl Ø - blank 8 bytes in RØ
R1 - IPL record
R2 - DMKCKP - looks at core to see if re-initialization is necessary.
R3 - volser
R4 - Allocation byte map
R5 - DSCB4 } OS type DSCBs telling OS that the pack is full.
R6 - DSCB5 }

IBCDASDI - OS program to format new disks.

FMT - used to allocate VM-only disks. Runs under VM with prompted answers:

Dev addr
Dev type
volser
type: Temp, Perm, TDSK, DRCT, PAGE
(→ mini disks, directory)
END

Different types of areas may reside on one physical disk:

TDSK ØØ1 Ø1Ø
TEMP Ø11 Ø2Ø
(beginning + ending cylinders)

The allocation bit map contain this allocation information.

DDR is then used to "restore" the system from tape onto the formatted disk. DDR will run either standalone or under CMS. DDR may also copy Disk to Disk. DDR is also used to back up a VM system to tape. DDR will automatically skip spool and page areas when dumping a volume to tape with the ALL option.

Necessary Files

1. The Directory describes all Virtual Machines. (see VM Sysgen Guide) The first line in the Directory lists the cuu, devtypp, and volser of where the Directory is kept.
2. DMKRIO - describes the Real I/O Configuration.
3. DMKSYS - describes the system - AP, UP, MP; V=R; Remote TP, etc.
4. DMKSNT - System Name table
5. DMKFCB - forms control block
6. DMKBOX - system logo

The above files are full of macros which, when assembled, expand into control blocks.

DMKRIO - describes Clusters
Terminals
RDevice - tape + dasd, ...
RCTLUNIT - control units (last digit Ø or 8)
RChannel
RIOGEN - consoles

Some non-existant devices may be gened.

The RIO gen is usually done in address sequance. A typical DMKRIO is 2000 to 4000 lines. The EXEC VMFASM issues the appropriate MACLIBs.

DMKSYS - macros:

SYSOWN - lists system-owned packs
SYSRES - where + layout
SYSOPR - identifes OPERATOR
SYSCOR - Real storage size
SYSTIME - time zone offset to GMT
SYSMON - describes monitor data to be collected.
SYSJRL - Journaling for security
SYSACNT - specifies accounting into spooling
SYSFORM - output forms + classes

SYSPCLAS - permits output of classified data.
SYSID - identifies a particular system.
SYSORD - Order of page/spooling device priority.
SYSLOCS - catch-all of pointers, etc.

DMKSNT - contains names of operating systems which may be IPLed from a Virtual Machine (i.e. CMS). This also holds form control for 3800 laser printers. Specifies shared segments for a given operating system.

DMKFCB - describes all types of forms available as FORM options. This does the form layout — an electronic carriage tape.

DMKBOX - logo for VM screen. This is basically a series of DCs, no macros.

These files are assembled with the VMFASM command. This issues the necessary UPDATE command and then issues the Assembly command. The UPDATE log, Update file, and Update Listing are merged into one listing.

VMFASM filename ctlfile (options

Five assemblies are necessary to build a VM system. This takes about 10 minutes once all the files are ready.

CP Nucleus Files

The command NUCBUILD brings in a list of all VM object modules, performs a massive link edit, and outputs a nucleus to the nucleus area (which was defined in DMKSYS by the SYSRES macro). It also writes the bootstrap program and the IPL text.

Maint is the userid responsible for building and maintaining the system. System disks are Maint minidisks.

PUT tapes are cumulative from the base to the date of the PUT. VMSERV is a giant slick EXEC which helps apply PUTs.

MVS Installation Workshop

8-29-83
Ken Okuba

We will init a pack from VM to initialize a sysres with an IPL text. The GEN00 userid will be running an MVS to do the init. We must submit our init job to GEN00.

Our own MVS systems will also come up under VM. All sysgen jobs will run on system 0 in ccc.

use sysout=Y + use tube. Don't print. Save all sysout on spool pack.

If the sysres pack is not online, issue a message to the operator to "Mount Disk volser as cut," and wait for message.
see p. CBSP08P

To do the pack init create the init job + then from the GEN00 MVS system, start a reader to get the specific job:

S SETRDR,SYSALLDA,SET10X,DSN=Gen0x.mydata(member)

An Operating System is a collection of data sets.
A Sysres has an IPL text

SYS1.Nucleus
RIMs
IEANUC01
SYS1.SVCLIB (contains appendages to certain routines)
SYS1.LOGREC - hardware errors

} Must be on Sysres since no catalog is available at first.

The nucleus has a SYSCATLG member which specifies the Master Catalog.

Linklib - utilities
Parmlib - system parameters
Proclib - procs
Maclib - macro library
STGINDEX - records VIO dataset info
SAMPCIB - ipl text source; sample jobs for debugging.

DCMLIB - Display Console Module - holds PF key settings for each display console.

DUMP00 - dump data sets. (1 to 10) will hold dump for system tasks (run disabled or holds a lock, or on SRB routine)

IMAGELIB - holds UCS information to synchronize print trains; also holds FCB information.

INDMAC - a maclib for obtruse applications. (Optional)

TELCMLIB - telecommunications related.

PAGE

PAGE.LPA - link pack area
.CSA -
.LOCAL1 (to 64)
.SWAP (to 25)
} must have these 3

DCLIB - input to stage 1 + stage 2.

all begin with "A"

APARMLIB - copied into SYS1.PARMLIB in stage 2.
APROCLIB - "
ALINKLIB - "
ACPALIB - "

AGENLIB - sysgen macros
AMODGEN - DSECTs for DEB, IOB, + other CBs + some macros.
HASPSRC - JES2 source module

SMP - system Modification Program
TSO - Time Sharing Option

Brodcast - messages
CMDLIB - TSO commands
UADS - User attribute datasets - id, password, etc.
HELP - HELP in script format.

VTAM

VTAMLIB - load modules
VTAMLST - parameter list
VTAMOBJ - an interpreted copy of the parameter list.

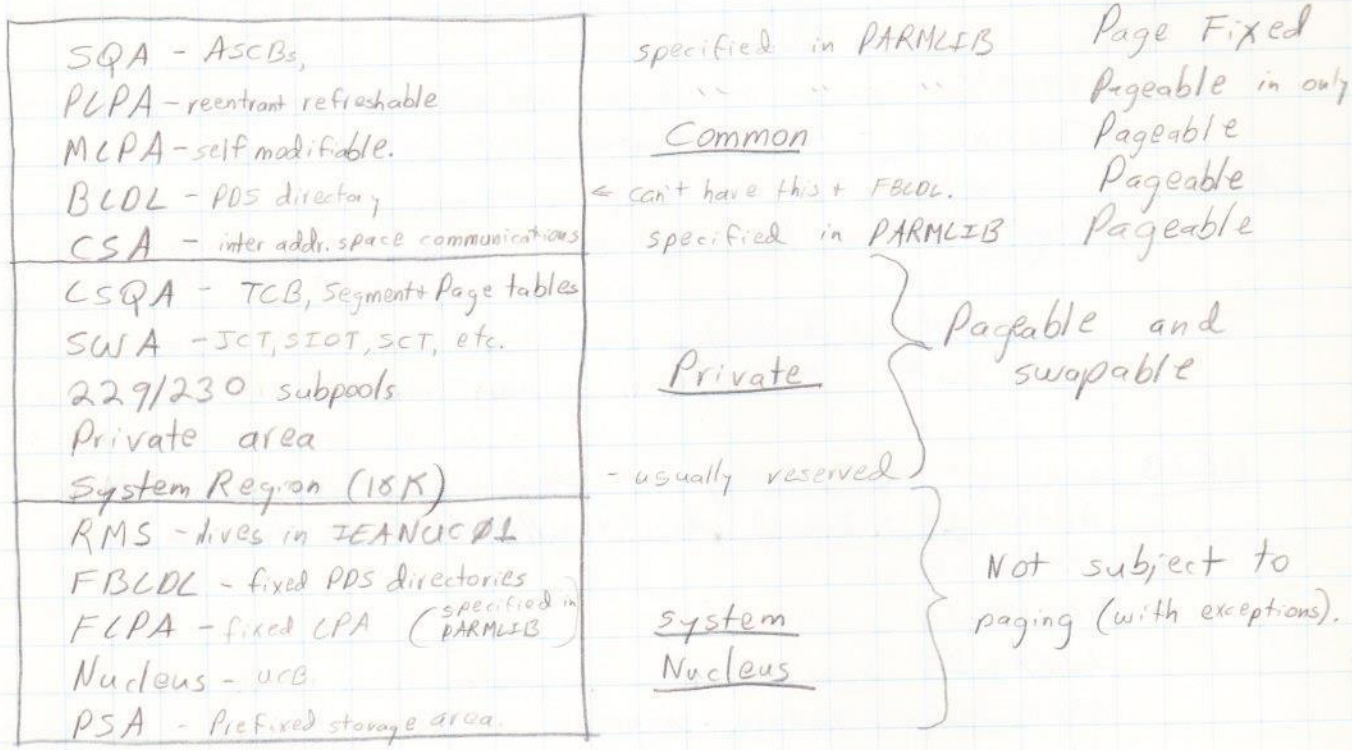
SMF

MAN% (0-9, A-Z)
MAN1
:

JES

HASPACE - spool
HASPCKPT - holds checkpoint info.

Virtual Storage



PARMLIB specifies the existence, location, and contents for many of these system components; for instance, FLPA, BLDL. Contents may come from Linklib or libraries concatenated to linklib.

The MLPA pages to PAGE.CSA, as does BLDL, & CSA. If no SWAP dataset is specified, swapping will occur on to the PAGE.LOCAL datasets.

At IPL, the nucleus is written out to the front of the local page dataset so that it could be refreshed if necessary.

How to create an MVS system:

1. Copy all the datasets from an existing system,
2. IPO (Installation Productivity Option) - a full sysgen of a simple system is performed by the vendor and dumped on tape.
3. Full Sysgen: stage1/stage 2.
4. I/O Gen: code I/O macros only.
5. EDT : update eligible device table.

Catalog

Holds: DSN
volser
devicetype
data set sequence (tape only)

Three types of catalogs:
VSAM
OS CVOL
ICF

VSAM

A VSAM catalog entry takes 512 bytes to hold the 70-80 bytes of needed information. It also holds a time stamp which must match the time stamp in the DSCB4. This makes backup and restore of the catalog very difficult. VSAM is also not suitable for GDG processing.

When MVS first came out, it did not support OS CVOL catalogs, only VSAM. The Integrated Catalog Facility is designed to circumvent the difficulties.

ICF - uses 6280 bytes for each entry; restore is possible; GDG is also possible. DF/EF is the Program Product. ICF uses the least I/O of all catalog methods.

OS CVOL -

Catalogs are indexed by dsn qualifiers. Each block points to its continuation and each entry in each block points to the next level of qualifiers.

several catalogs may exist in a system. However, a single Master catalog must contain pointers to the seconday catalogs. It may be either VSAM or ICF.

SYS1.Nucleus entry:

SYSCATLG
- 1-6 volser
- 7 catalog type — 00, 40 } VSAM; any other ICF
- 8 device type
- 11-54 Master catalog Name - must be specified for ICF. With VSAM it is not necessary because VSAM cat names are distinct and reserved and thus may be found just using the volser info.

An ICF catalog is defined using IDCAMS.
We will use a VSAM catalog for the lab. It is defined as a user catalog and is a master catalog only because it is refevenced in the SYSCATLG member of SYS1.Nucleus.

Catalog Search Order

1. stepcat or jobcat
2. First DSN qualifier
   check for alias to secondary catalog.
3. Master catalog.
4. JCL error.

Catalog Space Requirements

VSAM

$aN+b$

N = low key range record estimate.

a	b	Device
.1364	5	2314
.09	3	3330
.125	5	3340
.0667	3	3350

ICF uses 1/4 space of VSAM.

OSCVOL - allocate 1 cylinder.

SYSGEN

ICKDSF initializes packs.

DSF control commands:

specify this for an offline device.

INIT DDNAME(ddname) | UNIT(cua)
VERIFY(volser) | NOVERIFY
CHECK(n) | NOCHECK ← used to test pack
INDEX(cyl, trk, extent)
IPLDD(ddname) ← source of IPL text
VALIDATE | NOVALIDATE ← check for home addrs. on ea. trk.
VOLID(ser) ← changes volser to this name.
VTOC(cyl, trk, extent) ← placement of VTOC

lab: init pack.
allocate master catalog.

8-30-83

During the early phases of sysgen, the SYS1 prefix is not used. The high level qualifiers are changed to sys1 before IPL.

Sysgen macros are defined in SYS1.AGENCLIB and AMODGEN.

Machine Configuration Macros:

CHANNEL - code one per channel
IODEVICE - specifies I/O device, creates UCBs.
CONSOLE - specifies alternate consoles
UNITNAME - assigns esoteric names (SYSDA, TAPE, etc)

Non-existent devices may be gened.

The IODEVICE macro has the most operands.

Be sure to define SYSSQ as an esoteric name for disk. The IVP job which tests a new MVS system will use SYSSQ.

For VIO, the esoteric name entry will specify an address or range of addresses. These address must appear in the I/O gen section (i.e. the object of a IODEVICE macro). This is so the system knows what type of device to simulate.

Control Program Macros:

These establish the environment for the OS.

AFFINITY
CKPTREST
CTRLPROG
EDIT
JES
SCHEDULER
TSO

Data Management Macros

ACSMETH
GRAPHICS
IND
TABLE
UCSDFLT

use the SVCTABLE macro to code user SVCs.

SVC-nnn-Tn-Ln-FCOn-NP
- Tn: type
- Ln: lock
- FCOn: function code
- NP: Non-preemptable

The layout must be in reverse order, 255 to 1.

Generation Macros

Dataset - permits allocation of all system datasets.
Generate -
EDTGEN -

Alway preallocate system datasets - for performance reasons and to avoid the expiration date/operator reply problem. Do not use a vol. or space parameter on any Dataset macro.

use lots of comments

On an IBM 3081 or 3084, an IOCP gen must be done to gen the channels.

8-31-83
Ron Wolf

Linkage Editor

The Assembler outputs an External Symbol Dictionary on externally referenced entry points. This proceeds the TXT cards which hold the code. RLDs contain entries of references which still need to be relocated prior to execution.

Relocation Dictionary: RLDs contain A-cons and V-cons which specify the displacement of each program CPA within the load module.

The linkage editor combines several object (or load) modules, resolves external references among the modules, and creates a relocatable load module.

The load module is preceeded by a CESD image which lists the CSECTs which follow in the module. The txt records are usually the maximum trk size available. The RLDs are now 256 bytes long and still contain the displacement information needed at load time.

Each program in a load module will have its own TXT and RLD records. Also, the programs appear in the load module in the order in which they are referenced in the linkage editor.

{one RLD per TXT record}

Link Edit Options

Module Attribute:
- SCTR - scatter (used to lked the nucleus); permits scatter loading into storage. This is as opposed to block loading.
- RENT - reentrant
- REUS - reuseable (serially)
- REFR - refreshable
- AC=n - Authorization Code (AC=1 creates a CP module) (the library of the load module must be listed in the IEAAPFxx member of PARMLIB to be truely authorized).

Output Options:
- LIST - prints lked messages to sysprint
- MAP - prints detailed map
- XREF - creates cross reference (implies MAP + LIST).

Special Processing Options

XCAL - exclusive "call" external name; used for overlays. Causes auto overlays.
LET - execute load module even with a link edit which incurred unusual problems.
NCAL - No automatic resolutions of external resolutions.

DD Cards

SYSLIN - input to linkage editor.
SYSLIB - PDS of object or load modules refered to by the program undergoing link edit. specifying NCAL will prohibit the lked from looking at the syslib data set.
SYSLMOD - target library for output load module (will have undefined Record length)
SYSPRINT - output of messages and listing.
SYSUT1 - work file.

Some link edit control cards can follow the object code in SYSLIN. An alias may be assigned in this manner, for example. The syslin may also hold INCLUDE control cards which point to another DD statement which will be concatenated to syslin. The data set specified by INCLUDE may contain object decks or link edit control cards.

```
//INPUT    DD   DSN=object.deck
//SYSLIN   DD   *
     INCLUDE INPUT
     NAME  pgmname
     ALIAS pgmname1
/*
```

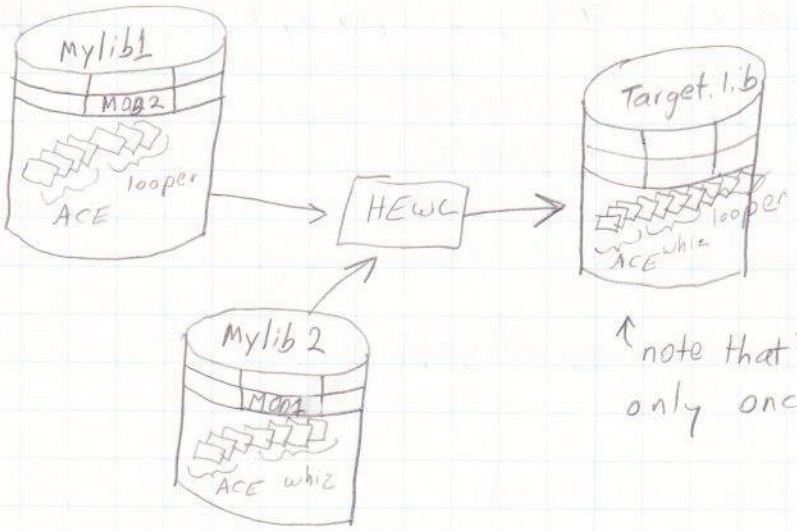
} This permits using control cards with an object deck.

The INCLUDE may also specify a member name along with the reference to a DD statement:

INCUDE INPUT(member)

Interestingly, the SYSLMOD may be referenced for <u>input</u> with include statements. This makes it possible to incorporate a previously linkedited module in a new module, and depositing the new module on the same PDS as the previous load module.

The CSECT names within programs are important. The true CSECT name need not be the same as the member name in a PDS. Also, during link edit if the same CSECT name appears twice in two load modules which are being link edited together, only the first occurence of the CSECT name will be included in the output load module. This happens even if the two identically named CSECTs are really different programs! To force inclusion of a subsequent duplicately named CSECT, use the CHANGE control card to change the CSECT name of the appropriate modules.



INCLUDE Mylib2
INCLUDE Mylib1

↑ note that "ACE" appears only once.

REPLACE may be used to specifically remove a CSECT from a load module. The REPLACE takes effect when the next INCLUDE is read.

Stage 2

Job Ø Defines the Master Catalog.
Job 1 builds CPA lib - contains copies & link edits } very long
Job 2 " link lib
Job 3 " SVC lib - (SVC lib holds I/O appendages) - very small job.
Job 4 " Nucleus and DCMLIB (Display Console Module - holds PFkeys for display consoles).
Job 5 Create miscellaneous datasets. Run this one first.
Job 6 Re-link edits some of the earlier load modules. Must be run last.

Job1

Will allocate the LPALIB with the index specified in the GENERATE statement.

Creates a device name table for use with the IEH utilities (which specify device type in the control cards).

Creates IEFSD060 which specifies initiator characteristics.

Creates TSO and SMF exits (which are dummy at this time).

Also creates IEFBR14.

Job2

May allocate Linklib.

Creates MSTRJCL (to provide the JCL to start address spaces) member.

Creates JES name table (IEFJESNM) - lists primary and secondary subsystems (up to 15).

Creates IFOX00 GENER
HEWL UPDTE
DASDR IEHLIST

Job3

Creates SVCLIB of I/O appendages.

Job4

May allocate Nucleus & DCMLIB.

holds svctable, UCBs, CVT, & NIP modules.

Job5 - Run first.

Allocates Logrec.

Specifies TSO edit parameters.

Creates (TSO) CMDLIB and VTAMLIB (to hold VTAM code).

Creates a SYS1.Parmlib.

Defines SYS1.STGINDEX for checkpointing VIO page datasets.

Job6

Link edits into LPA & Link libs.

Has lots of Listvtocs, listpds, & listcats - usually good to leave these in.

9-1-83
Ken Okuba

SYS1.PARMLIB

The system obtains parameters from SYS1.PARMLIB and the operator's console. PARMLIB can have only one extent and must have lrecl of 80 bytes. If NIP cannot read Parmlib it issues a 64 wait state code.

Members in Parmlib are created by specifying certain macros is stage one, such as JES, +APFLIB. Additional members may be added with SPF.

Member IEASYS00 is the director to other members. It contains the nn suffix to various members, such as APF, etc. Two formats:

	Example:
keyword=suffixcode	APF=00
keyword=parameter	CSA=1024

See the Initialization and Tuning Guide (figure 3-1, p. 3-7) to find which keywords require suffixcodes and which require parameter values.

The internal coding requirements of Parmlib members are different. Refer to examples.

IEASYS00 specifies the page datasets in the order:

LPA
CSA
Local
:

The actual names do not influence their function; it is the order of specification which determines the function.

Be sure to IPL with a CLPA to load the LPA page dataset. The system will force a CLPA if you forget. CLPA use is a "cold start."

It is possible to specify DUPLEX which then causes all pages written out to LPA and CSA page areas to also be written to another volume. This is inefficient. A ,L" following the IEASYS00 entry for PAGE will cause a Listing of the page dataset names to the operator console.

VIO may be prohibited from occupying select page datasets with the NONVIO parameter. CVIO will Clear all VIO datasets in existence at system initialization. This basically circumvents use of SYS1.STGINDEX. Use of CVIO is known as a "Quick Start" of MVS.

The SQA keyword specifies the number of 64K segments which will be added in addition to the four segments already provided.

CSA space is set on a 64K segment boundary.
Allocate 100K REAL for V=R users. (V=R users get a protect key greater than 8). VRREGN specifies the default region size given to V=R users. This should be Ø to force the user to code a REGION parameter on the jobcard.
The WTOBFRS value should be 10x the maxusers. This allocates buffers for Write To Operator messages.
APG value is the default DPRTY for all jobs.

If an alternate IEASYSnn is specified during IPL (via SYSP=nn command), the system will still fetch IEASYSØØ first and then overlay a copy of it with the parameters in the alternately specified IEASYSnn. Thus, several IEASYSnn members may be merged to produce a final table - PARMTAB. Additionally, specific fields in IEASYSnn may be specified from the console. The last entered parameter takes precedence.

LNKLSTnn

The suffix is specified in IEASYSØØ.
LNKLST specifies the default system libraries for program libraries. Any dataset listed in LNKLST is considered authorized These datasets must be catalogued in the master catalog. They will be opened at NIP. A maximum of 15 additional datasets may be concatenated to LINKLIB. The LINKLIB entry is not included in LNKLST. It is implied.
Beware of duplicating entries. This will cause the system to doubly search the PDS directories.

IEAPAKØØ specifies LPA members which should by placed together in the same page frame if possible. This procedure, however, leaves gaps in page frames which causes PLPA size to increase (sometimes severely). This facility may be going away soon. Scratch this member if present.

IEAFIXnn

Specifies members of fixed LPA. We must specify the module name and its library name: i.e. SVCLIB, LPALIB, & LINKLIB.

IEALPAxy

Specifies members of MLPA. Must specify library name and module name.

IEABLDxx

A BLDL table is a in-storage copy of various PDS directories. Having the directories in storage eliminates the I/O associated with reading the PDS directory to find the member. Thus, the only I/O will be that of retreiving the member. BLDL tables are built with macros that are available to any user. The system may have either a fixed or a pageable BLDL.

For Pageable, specify BLDL in IEASYS00.
For Fixed, specify BLDLF in IEASYS00.

The IEABLDxx member will contain the data set names of PDS's which will have their directories placed in a BLDL table.

IEAAPFxx

Lists authorized libraries. Cinklib and SVClib have implied authorization and will not appear on the list. The member must contain the dsn and the volser.

COMMNDxx

Commands in this member will be executed at Master Scheduler Initialization. (Cannot contain JES commands). The console display commands may not be contained here (K, S etc.)

Example:

```
TOD= Prompt
COM='S JES2'
```

MPFLSTxx

Permits suppression of selected informational messages. These message will not appear on the console, but will appear on the console log (marked with "@").

To invoke this facility, enter

SET MPF=NO|YES

This command may be placed in COMMNDxx.

VATLSTxx

Contains the list of volumes to be made available to MVS after IPL is complete.

field	bytes
Volser	1-6
Mount	8
use	10
Dev. Name	12-19
Suppress MSG	21 (N|Y)

Example

SETRES, Ø, 2, 3330-1 , N

↑ must use commas to separate

Mount may be specified as Removeable (default), Reserved, or Permanently Resident. Avoid removeable. Use Reserved for user packs, and Resident for system packs.

USE attributes are Private, Public, + Storage.

Private = volume must be specifically referenced.

Public = A temporary, non-specific dataset may be place on the volume.

Storage = can take any type dataset.

Additional volumes may be mounted from the console:

M cua, VOL=(SL, volser), USE= STORAGE / PUBLIC / PRIVATE

An "N" in suppress message means not to issue a message to the operator if a volume is not available.

VAT = Volume Attribute Table.

IECIOSxx

Selects the IOS use algorithm for Channel/Optional Channel decision:

default → Sequential - try primary first

Reverse Sequential - try alternate first

popular → * Rotate (alternate back + forth)

LCU (last channel use)

→ * Balance - balance I/O activity.

IEADMPØØ
Controls what information is provided in SYSUDUMP.

IEAABDØØ
Controls what information is provided in SYSABEND.

IEADMRØØ
Controls what information is provided in SYSMDUMP.

GRSCNFxx
Global Resource Serialization is designed to circumvent problems with the reserve of peripheral devices via hard lock features. With GRS, each CPU asks the other for a specific resource before commiting itself to an ENQ.
This members specifies the GRS environment. IEASYSØØ has a GRS parameter. For a UP, specify NONE.

CONFIGxx
Specifies the "standard" system configuration - of devices, channel online or off, certain volumes mounted, etc. Then, after a night of special runs, the operator can issue

DISPLAY M=config(xx)

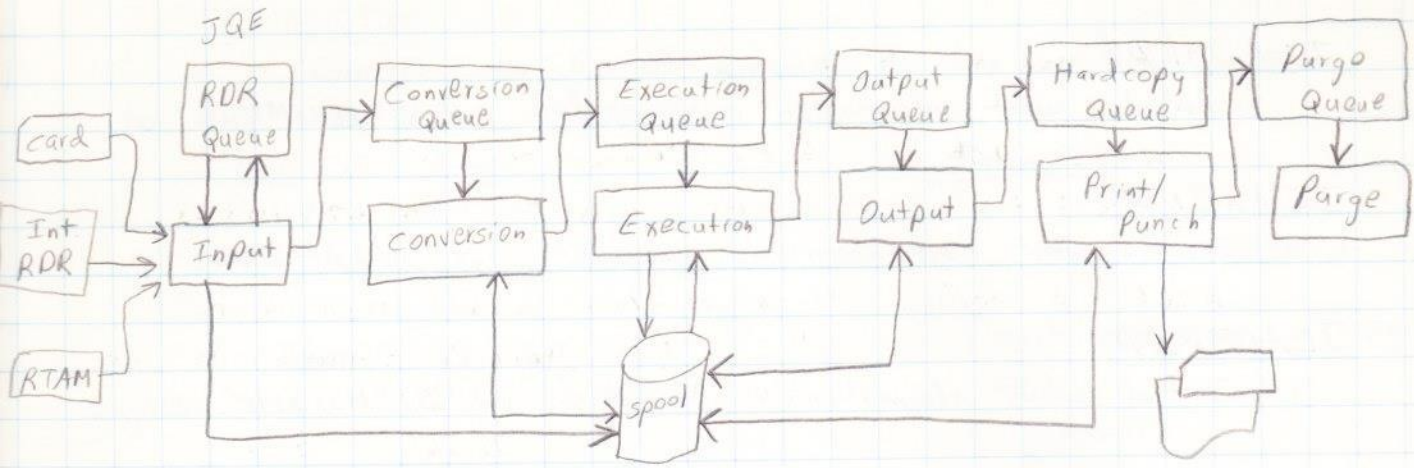
which will cause the system to compare its current state with the parms in CONFIGxx and issue appropriate messages. (written by Ron Wolfe originally for McDonald Douglass)

9-2-83
Ken Okuba

JES 2

JES2 has the capability to use a "shared spool" on a loosely coupled system. Jobs can enter JES through card reader, RJE, Internal Reader, START command (STC), MOUNT command, and TSO LOGON command. All jobs require JCL.

(see JESØØ7Ø of handouts for JES Job flow diagram)



The JQE (Job Queue Element) is the controlling control block. A single byte in the JQE controls which queue the job is in.

JES has two main modules:
HASJES2Ø - runs in its own address space (LINKLIB)
HASPSSSM - lives in PLPA (LPALIB)

DCIBs:
SYS1.AOSH1 - RTAM
SYS1.HASPRC - JES2 Source
SYS1.AOSH3 - JES2 Load Modules

SYS1.Proclib has three members to help install JES2.
JES2JOB - JCL to assemble.
JES2BLD - JCL for Lked.
JES2LNK - link edit control statements.

JES2 datasets:
SYS1.HASPACE - JCL internal text, sysin + sysout, + JES control blocks; Entire volume.
SYS1.HASPCKPT - warm start info.
- copy of JES2 Queue.
- must be only 1 extent.

The volumes which contain these datasets are considered spool volumes and should be Permanently Resident, Private.

Starting JES

The command S JES2 starts JES2. This command may be issued from COMMNDxx. Normally, JES2 is automatically started by the Master Scheduler. To avoid this requires use of superzap.

JES2 initialization options:

FORMAT | NOFMT ← write / Nowrite binary zeros. (implies COLD)
COLD | WARM ← erase / Noerase spool files.
NOREQ | REQ ← Norequest / Request the $S entry from operator.
NOLIST | LIST ←
NOLOG | LOG ← Don't Print / Print Hasp messages on console log.
HASPPARM = ddname | HASPPARM=HASPPARM ← where are JES2 parms.

The JES startup JCL may contain ddnames of alternate parm datasets.

Once JES2 is up, it will issue a wait macro and not actually begin processing until the operator replies $S
To accept the default initialization options, enter a null, u, NONE, or N.

The operand of the Start command is a procname. The JES2 PROC has a dd statement PROC00 which contains the concatenated list of all valid proclibs.

JES2PARM (1)

72
X

Format: Parameter, subparm, subparm, subparm, . . .
subparm, subparm

* in column one is comment line.
Other comments may follow subparms after a blank.

&SPOOL = volser of spool volume. JES2 does not go through the catalog.
&CHKPT = volser - this is the checkpoint volume. The default is the &SPOOL volume but this is very bad for performance reasons.
&CHKPTIME = nnn ← checkpoint time in seconds. Default = 60.
&DSNPRFX = name. ← high level qualifier of dsn. Default = SYS1.

Define sysout class:
$$x x = sysout class

Example

Dummy | SYSOUT
HOLD | NOHOLD
PUNCH | PRINT
TRKCEL | NOTRKCEL
used for higher speed printers: anything faster than 1403.

equivalent { $$A Sysout, Nohold, Trkcel
$$A TRKCEL

If a sysout class is not defined, it gets all the defaults.

STCMCLAS=n ← sysout class for Started Task msgclass.
TSUMCLAS=n ← sysout class for TSO Logons.
Default is class A.
To prevent these from printing, define one sysout class as a dummy (usually Z).

Job Classes:

certain classes may be set up with special characteristics, for instance class S for SCAN only.

Options:

COPY | NOCOPY ← simply print JCL, dont run.
HOLD | NOHOLD ← hold execution
use Nojourn → NOJOURN | JOURN ← provides for auto restart. Does not pay.
NOLOG | LOG ← joblog
NOOUTPUT | OUTPUT ← save output
SCAN | NOSCAN ← test JCL
PROCLIB=nn ← refers to nn PROCnn ddstatement in JES2 startup deck.
PERFORM=nnn ← assigns specific performance group.

Certain job classes can be restricted to certain Proclibs.

Conversion Parameters

During job conversion, JES must supply defaults for parameters that were not specified in the job.

&RDROPSL=string for logon
&RDROPST=string for STC
&RDROPSU=string for batch

String:

bppmmmmssccrlaaaaef

b = accounting + programmer name field
pp = not used
mmmmss = step time
ccc = region size
r = commands for MVS permited
l = BLP; Permits Bypass Label Processing

aaaa = command group
e = MSGLEVEL=(x, y)
f = MSGLEVEL=(x, y)
only valid for OPSC and OPSU
x = 0, 1, 2 = JCL listing
y = 0, 1 = Allocation listing

JES2PARM must also contain entries for each available Unit Record device. Among the operands for printers are

CKPTLNS = check point # of lines printed. Default = Ø
CKPTPGS = check point # of pages printed. Default = 1

These parameters permit JES to correctly restart a print job after a system crash. Otherwise, JES would begin printing the interrupted job at the beginning. This library also contains parms which specify the default class and status (Drained, etc) of each printer.

JES2PARM contains entries to set the default estimated lines exceeded, the lines interval to display the message, and the action determination for a job exceeding the limit.

&ESTLNCT = nnnn
&ESTPUN = nnnnnnn
&OUTXS = nnnnnnn ← message interval
&OUTPOPT = Ø|1|2 ← fate
Ø = allows job to continue.
1 = cancel w/o dump.
2 = cancel w/ dump.

Teleprocessing: VTAM and TSO

9-6-83
Ron Wolf

The access method interfaces an application program to terminals. Therefore, to install an access method we must define the characteristics of the application (via Message Control Program, MCP) and describe the terminal network (SYS1.VTAMLST).

VTAM

VTAM was pretty bad at first, but is now a much improved performer.

Components

Three Libraries:

SYS1.VTAMLIB - holds user exits.
non-reentrant (reentrant are in LPALIB).
SYS1.VTAMLST - a parmlib to describe network.
SYS1.VTAMOBJ - machine readable form of VTAMLST.
Members of LST and OBJ have timestamps which are compared. If a change is made a new copy is assembled and placed in OBJ.

Three Tables:

Logon Mode - describes logon characteristics

latest and greatest > USS - interpretation of logon; system logo (Unformatted System Services)

Logon Interpret Table - command interpretor for pre-logon communication (ie: LOGON).

USS + Logon Interpret Table are mutually exclusive.

"Net" - Proc to start VTAM

"TSO" - Not starting TSO, but a Proc for Terminal I/O coordinator.

SYS1.VTAMLST

Terminal Definitions: type of terminals and address.

Application Programs: name of program.

The member names for both these lists are arbitrary.

Member ATCCONnn lists the members containing terminal and program definitions.

ATCSTR00 is the start-up list for VTAM itself. It points to ATCCONnn.

A "major node" is the name of the Terminal Definition table.

The Application Program Definition Table contains one statement for the application program (TSO) and one entry for every possible logon. These entries create control blocks which are used in the terminal session.

VTAMLSI(ATCSTR00)

SSCPID = 01, COLD

↑ WARM ← leave nodes as they were when VTAM went down

processor ID (used for 4705)

CONFIG = 00 - The suffix of ATCCONnn

CSALIMIT = 0 = no storage limit.

Any other value limits amount of storage that VTAM can use. A limit is good.

MAXSUBA - interpreted, specifies the number of terminals via major + minor nodes.

MAXAPPL - maximum applications running on VTAM.

SUPP = NOSUP - suppression of error messages.

VTAMEAS - max network address units.

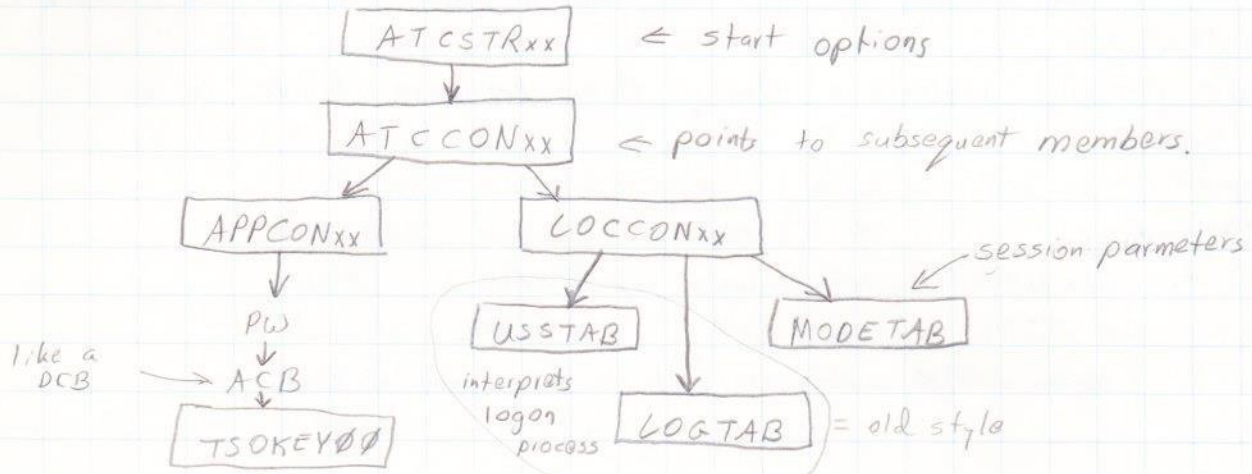
CIST = 00 - suffix of the ATCSTRnn member (self referential).

ATCCON$\emptyset\emptyset$ — specified in ATCSTRnn

Holds arbitrary names of application description members and links them to the requisite terminal lists. Only one member may be specified per 80 byte entry.

Logon Mode Table

Contains default session parameters. The IBM version is ISTINCLM A user can code, assemble, and link his own if required.



USS Table macro statements generate executable code when assembled. This table specifies pre-logon commands, replacement strings

VTAM Proc

The NET Proc lives in SYS1.Proclib. Use the standard name of "NET" to avoid complications. VTAM should execute at a higher DPRTY than any job using VTAM. The proc requires three DD cards:

1. VTAMLIB
2. VTAM OBJ
3. VTAM LST

VTAM Teminal I/O Coordinator (VTIOC)

called TSO in the start command. This interfaces data flow between VTAM and the individual TSO address spaces. Parms are specified in the SYS1.PARMLIB member TSOKEYØØ.

TCAS - Terminal Control Address Space - part of MVS/TSO to handle creation of the the address space by MVS.

TSOKEYØØ in SYS1.Parmlib holds
maxusers of TSO users: (the maxuser in IEASYSØØ limits the number of address spaces).
Reconlim - reconnect limit time; max time for an inactive disconnected user.
Bufrsize = 132, ← required by VTAM.
ACBPW = pwtso ← must match password in APPCONØØ.

VTIOC ("TSO") requires a startup Proc because it lives in an address space.

Sequence:
S NET ← brings up VTAM
S TSO ← " " VTIOC
Logon BRAØ ← user (IKJACCNT member of Proclib).

TSO

User Exits
- Submit, Status, Cancel - IKJEFF53
- Pre-logon - IKJEFLD
- APF - IKJEFTE2, IKJEFTE8

"Logon IBMUser Nonotices Nomail" → the logon to use for the very first logon of TSO.
Account - permits building of userids. (ADD subcommand)
SYNC - formats broadcast ds (subcommand of ACCOUNT)
The Account commands update SYS1.UADS

IKJACCNT holds the logon JCL necessary to create a TSO address space. The TSO administrator should have a SYSUADS dd statement.

Exits

IKJEFF53 is the exit to scan and approve/deny the submit, status, and cancel commands. To approve, zero R15 and return; Otherwise, return a non-zero in R15. Place module is SYS1.LINKLIB.

IKJEFLD is the pre-logon exit. To decide whether to permit logon (+ how) it can look at UADS, JCL proc specified, region size, and password.
This module must be INCLUDED in IKJEFLA.

```
//SYSCIN   DD *
         INCLUDE  SYSIN(member)   ← object deck of IKJEFLD. Make sure this is first!
         INCLUDE  SYSLMOD(IKJEFLA) ← existing IKJEFLA
         ENTRY    IKJEFLA         ← defines entry point for execution
         ALIAS    . . .   } Make sure all aliases are update so that
         ALIAS    . . .   } none remain pointing to the old module.
         NAME     IKJEFLA
```

Ordinarily TSO is not APF authorized. IEBCOPY won't run unless authorized. Thus, to permit IEBCOPY to run under MVS, it must be authorized. Two tables are used
APFCTABL — to authorize commands
APFPTABL — to authorize programs (invoked via CALL command)
If a command appears in APFCTABL than the system will permit key change and then go execute the appropriate program. If the command does not appear in the table, the key will not change and the program will abend for being in the wrong key.

Account

ADD - userid
Change - pw, logonid, proc
Delete - deletes userid
END - exit
HELP -
LIST - display particular IP
LISTIDS - list all ids
SYNC - format broadcast ds

Example:

ACCOUNT

ADD(userid pw * IKJACCNT) JCL ACCT OPER

(* = account number or *; IKJACCNT = Procname; JCL = permits submit; ACCT = use ACCOUNT command; OPER = use oper command)

Different passwords under the same userid will lead, if wanted, to different attributes.

Use of the ACCOUNT command creates or alters entries in the SYS1.UADS dataset.

OPER - Operator mode

Cancel - cancel TSO session
Display
END
HELP
Monitor - performance
SEND - update SYS1.Broadcast
SLIP - Source Level Indication Processing - a Trap
STOPMN - stops Monitor.

Dial session
press PFK
Logon VM
Def Grat 090
IPC 100
:
S NET
S TSO
PA2
press PFK
Dial GEN05

LAB

Make sure SYS1.CMDLIB is concatenated to Linklib with the Linklstxx member of SYS1.Parmlib.

✓ Net, Act, id = Loccon00 to bring up 091

9-7-83
Ron Wolf

System Management Facility

SMF collects configuration, performance, and workload accounting data. SMF gets a record at step end and job end. It also can record sysout lines and data set accesses (EXCPs).

SMF also provides for user exits which can enforce system standards or collect additional performance information.

SMF is widely used for accounting and billing. SMF can even examine the pgm name in execution and thus bill accurately for the use of a program product (charge for actual use). SMF is also used to track job throughput and turnaround time. This information is used for job scheduling.

SMF uses lots of system resources if it is collecting many different kinds of records.

SMF uses buffers in CSA. SMFPRMxx of PARMLIB controls SMF parms including the number of buffers. By examining the SMF reports, one can see how well SMF is using its buffers. When a buffer fills it is written to the active SMF dataset. SMF requires at least two datasets (+ a maximum of 36). An SMF buffer is 4K in size.

All SMF records have a standard header (14 bytes):

- Time -
- Record type -
- Date -
- SID - System Identifier.
- Record Length - (variable spanned)

} commonly used by the sort routines.

SMF does not perform its own data collection. This is done by MVS and then deposited in SMF control blocks. RACF, RMF, and other programs also write to SMF buffers. An SVC 83 causes a write to SMF. The SMF datasets are ESDS VSAM datasets. They are named SYS1.MANn.

Parmlib member SMFPRMxx contains the names of the datasets which SMF should use.

To define:

```
Define Cluster (NAME(SYS1.MANX) VOL(volser) -
  CYL(2) Nonindexed (Reuse) Spanned -
  Speed Controlintervalsize (4096) -
  Recordsize(4096,32767) Shareoptions(2))
```

4096 ↑ average size; 32767 ↑ max

(Reuse) → Each time the dataset is opened the use byte is reset so the dataset looks empty.

To format an SMF dataset, execute IFASMFDP which will either dump the data or reformat.

Control card:

INDD(ddname, OPTIONS(CLEAR))

IFASMFDP can also do preliminary sorting which will route different record types to separate DDcards and still write all records to another ds. It can also be used to merge SMF files.

SMFPRMxx Processing Options

The suffix of this member name may be specified explicitely or by default (ØØ). The parm file may switched during operation by issuing "IEASYSØØ

SET SMF=XX

The command

SETSMF

will permit the changing of individual parameters.

DSNAME(SYS1.MAN1, SYS1.MAN2, ...)

[NO]LISTDSN - list dsnames to console

SID(XXXX[, serial#]) - system id + serial number.

REC ALL - data set information (all datasets)
PERM - (just permanent datasets)

BUFNUM(MIN, MAX) - number of buffers, default = (4, 9)
The system can dynamically acquire buffers up to the maximum. These are getmained out of CSA. It always holds the min. but will freemain others.

[NO]MAXDORM(mmss) - amount of time to leave data in buffers before writing a record.

[NO]STATUS(HHMMSS) - how often to write statistics about SMF buffer usage. The number of buffers is expressed as an average of this interval.

JWT(HHMM) - Job Wait Time; maximum time for 522.

[NO]PROMPT (IPLR, LIST, ALL)

IPLR - IPL reason prompt when system comes up.

LIST - LIST SMF parameter options and give the operator a chance to save them.

ALL - do both.

(YY:ZZ) → specifies range

SYS ([NO]TYPE (XX, YY, ZZ, ...) ← record these record types.
[NO]INTERVAL (HHMMSS) ← interval for type 30^{*32} records.
EXITS(NAME1, NAME2, ...) ← which exits to take, by name.
[NO]DETAIL) ← controls detail in type 30+32 records.

SUBPARM (name, parameter) ← 60 bytes of parameter info may be passed to a specified subsystem (JES2) upon request.

SUBSYS (Name, Options) ← specifies different SMF options for a specific subsystem, like TSO or STC.

SMF Commands

SET SMF=XX ← XX = a suffix of SMFPRMxx parameter list. The dump dataset in use will not change, and the SID cannot be changed.

SETSMF JWT(10) ← example; Changes a single parameter for the life of an SMF session. This cannot change SID, Exits, or Prompt options.

DISPLAY SMF, O - options in SMFPRMxx.
DISPLAY SMF, S - Status of dump datasets

SMF Exits

These are actually exits from MVS code. Sample Exits may be found in SYS1.Samplib. User exits must be link edited into LPALIB (check stage 2 for link edit control options).

IEFUJV - Job Validation scans input job stream (can change JCL).
IEFUJI - Job Initiation (or TSO or STC)
IEFUSI - Step Initiation
IEFUTL - Time Limit to examine job that exceeds JWT or CPU time.
IEFUSO - Sysout Limit for exceeding OUTCLM JES2 parameter.
IEFU83 - SMF Record entered when SVC 83 is issued.
IEFACTRT - Termination Exit; Accounting Routine, (Step End, Job End).
IEFU84 - SMF Record for when SVC83 is branched into.

IEFUJP - Job Purge Exit for JES2.
IEFU29 - SMF Dump Exit; May execute when an SMF dataset fills up (i.e. Submit a job).

Each exit is passed a common parameter list.
Register 1 points to the pointer which points to the list:

- Job Name
- SID
- User Data (8 bytes) ← a work area. Not used normally.
- Step Number
- Job Class
- User Communication (4 bytes) ← preserved for life of job.

SMF also has a set of macros which may be used in SMF exits. (see handouts, SMF0230 for a list)

SVC

9-8-83
Ken Okuba

Types of SVCs:

	Type 1	Type 2	Type 3/4	Type 6
Refreshable	No	No	yes	No
Locks Req.	Local	sometimes	sometimes	No
Recursive SVCs	No	Yes	yes	No
Disabled	No	No	No	Yes
Location	Nuc	Nuc	LPA	NUC

All SVCs must be reentrant. Locks are required to serialize resources in tightly coupled (shared memory) CPU environments. Locks are obtained by the SVC FLIH. A lockholding SVC cannot recursively call another SVC (it can, however, branch into the needed routine). Type 6 SVCs are the most restricted. This is because it runs disabled for interrupts.

User SVCs, if type 3/4, may live in Linklib, SVClib, or LPALIB. They may be read into FLPA, MLPA, or PLPA. The user must carefully select the type for a home grown SVC.

SVC Naming Conventions

Types 1, 2, & 6:

IGCnnn

nnn = SVC number in decimal

Types 3, 4:

IGCyynnz

yy = ØØ

nn = SVC number

z = zone unpacked decimal sign

Example:

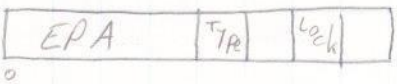
IGCØØ25A = SVC 251

A = EBCDIC C1 ← value, ↖ sign

Following the naming convention is very important. A module with an incorrect name will not be executed.

SVC Table

Each entry is 8 bytes. The table has 256 entries, numbered zero through 255.



↳ Ø = type 1

(see handout for bit map)

The APF bit specifies that the calling program must be APF authorized. All SVC are themselves authorized. IBM reserves the first 200 SVCs.

ESR bit says the EPA is really a pointer to another SVC table. This increases the number of possible SVCs. When ESR bit is on, R 15 must contain the entry displacement into the second SVC table. The second SVC table is formatted exactly like the first SVC table. Only IBM SVCs will use this second level lookup.

The only SVCs which have ESR bit on are:

109	– for type 3/4	Each type has its own second level table.
116	– 1	
122	– 2	
137	– 6	

(See p. 136 of Sysgen Reference for CSECT of SVCTable.)

To code a user SVC we must code several macros:

To install an SVC at Sysgen time:

SVCTABLE SVC-nnn-Tn[-(Ln,...)] [-FC{01/00}]

↑ SVC number ↑ type ↑ Locks ↑ Function Code (01 = APF on, 00 = APF off)

Example:

SVC-255-T2-(L1,L3)-FC01

At NIP time, the RIM will check the SVCtable for all type 3 & 4 SVCs. These entries contain zeros in the EPA. The RIM will scan the PCPA for the address of the SVC and will supply the correct EPA for the SVCTable. It does this for every 3/4 SVC entry.

After Sysgen:

Type 1, 2, 6
update SVC table
relink edit NUC01

Type 3/4
Zap SVCtable

System Maintenance Program (SMP 4)

9-9-83
Ron Wolf

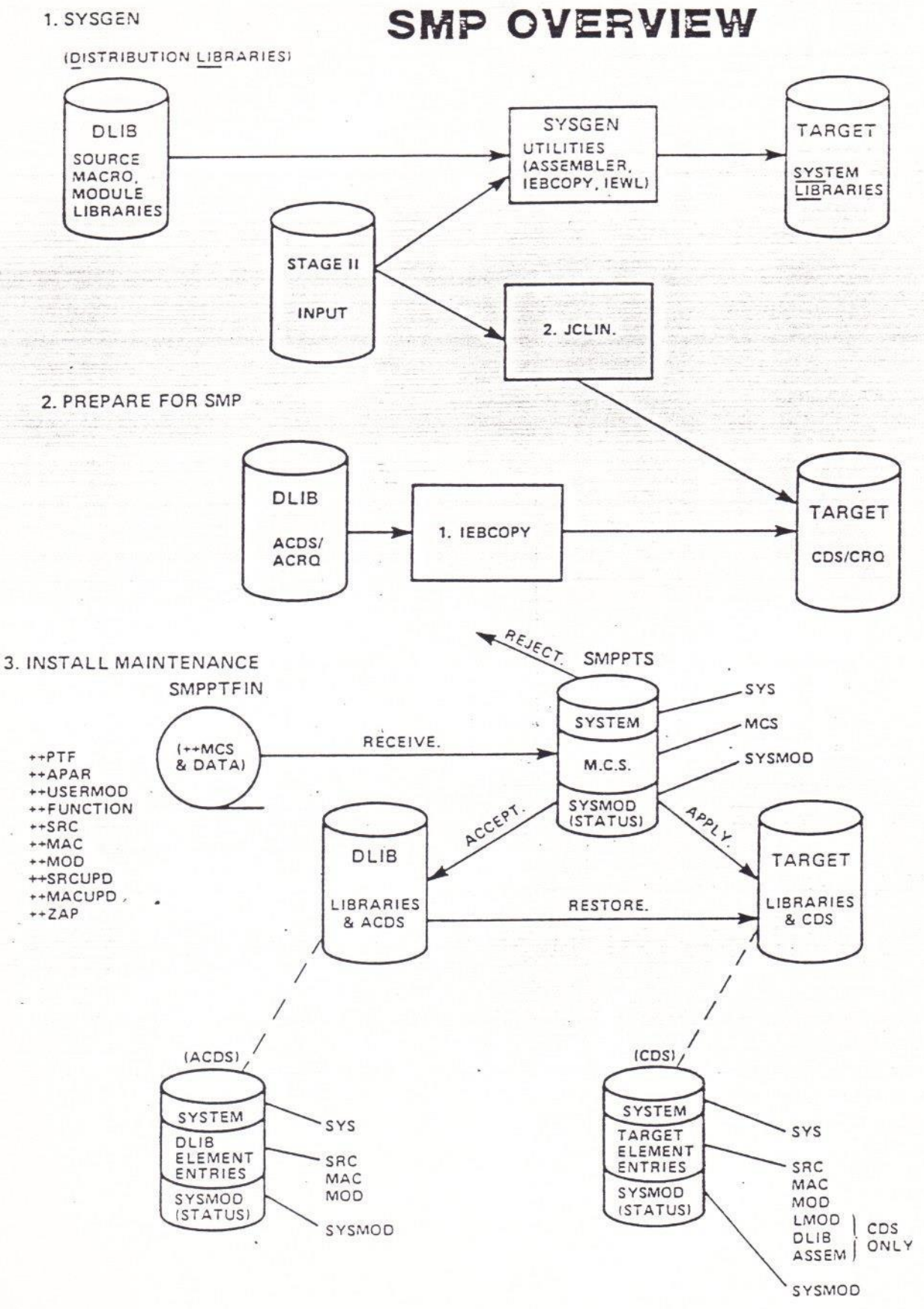
Output of sysgen is used by SMP to build tables. SMP is used to apply system maintenance (PUT) and Functions (major additions of code). SMP can only apply a single PTF to one Function at a time.

Four types of sysmods:

at least an entire object module:
- Function — major system component (Distribution Code)
- PTF — Program Temporary Fix (Considered Permanent) object deck
- APAR — Authorized Program Analysis Request: the fix for an APAR is an element update and is applied via IMASPZAP to the actual load modules. (IEBUPDTE is used to "zap" source or tables). An APAR is a Corrective measure.

Usually fixes single problem.

SMP OVERVIEW



QSTNS0100

REVIEW

SMP JCL

```
//EXAMP1 EXEC PGM=HMASMPED

//SYSUT1 DD UNIT=SYSVIO,SPACE=(TRK,(4,4))
//SYSUT2 DD UNIT=SYSVIO,SPACE=(TRK,(4,4))
//SYSUT3 DD UNIT=SYSVIO,SPACE=(TRK,(4,4))

//SMPWRK1 DD UNIT=SYSVIO,SPACE=(TRK,(4,4,4))
//SMPWRK2 DD UNIT=SYSVIO,SPACE=(TRK,(4,4,4))
//SMPWRK3 DD UNIT=SYSVIO,SPACE=(TRK,(4,4,4))
//SMPWRK4 DD UNIT=SYSVIO,SPACE=(TRK,(4,4,4))
//SMPWRK5 DD UNIT=SYSVIO,SPACE=(TRK,(4,4,4))

//SMPOUT DD SYSOUT=A

//SYSPRINT DD SYSOUT=A

//SMPPUNCH DD SYSOUT=A

//SMPLOG DD DSN=SMP00.LOG,DISP=SHR  — sequential

//SMPPTS DD DSN=SMP00.PTS,DISP=SHR

//SMPSCDS DD DSN=SMP00.SCDS,DISP=SHR

//SMPMTS DD DSN=SMP00.MTS,DISP=SHR

//SMPSTS DD DSN=SMP00.STS,DISP=SHR

//SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR

//SMPCRQ DD DSN=SMP00.CRQ,DISP=SHR

//SMPACRQ DD DSN=SMP00.ACRQ,DISP=SHR

//SMPACDS DD DSN=SMP00.ACDS,DISP=SHR

//SMPCDS DO DSN=SMP00.CDS,DISP=SHR

//DLIBN DD DSN=SMP00.DLIBN,DISP=SHR

//TARGET DD DSN=SMP00,TARGET,DISP=SHR

//TEXTIN DD DSN=SMP00.TEXTIN,DISP=SHR

//SMPCNTL DD *
RECEIVE .

//SMPPTFIN DD DATA,DLM=AA

++FUNCTION(FUN0001).

++VER (Z038).

++MOD (OBJMOD) DISTLIB (DLIBN) LEPARM (STD) TXLIB (TEXTIN) LMOD (TARGLOAD)
```

input; may be PUT.

must be 7 characters. 8th digit is used in Directory entries.

verify that Z038 (MVS 3.8) matches the PTFs received.

CDS ENTRIES

```
SMPCDS LOAD MODULE ENTRIES

TARGLOAD LASTUPD          = FUN0001     TYPE=ADD
         SYSTEM LIBRARY   = TARGET
         LKED ATTRIBUTES  = COPY   STD

SMPCDS MODULE ENTRIES

OBJMOD   LASTUPD          = FUN0001     TYPE=ADD
         LIBRARIES        = DISTLIB=DLIBN
         FMID             = FUN0001
         RMID             = FUN0001
         UMID             = FIX0001
         LMOD             = TARGLOAD

SMPCDS   SYSMOD ENTRIES

FUN0001  TYPE             = FUNCTION
         STATUS           = REC APP
         FMID             = FUN0001
         JCLIN            = YES
         DATE/TIME REC    = 80.024  09:51:19
                   APP    = 80.204  09:51:33
         MOD              = OBJMOD
```

Usermod - entirely user defined. May be an entire element, function, or zap. This is not supported by IBM.

SMP requires lots of data sets. It uses about 12 of its own just to run, plus most all system datasets. SMP heavily uses the user space within PDS directories. Thus, SMP datasets should have huge directories. (The new SMP/E uses VSAM).

The DLIB contains single element load modules. (Only one CSECT per element). A copy statement in stage ~~II~~ retrieves only one CSECT. Three primary functions occur in stage 2: Assemblies, Link Edits, and copies.

SMP directs modification to both the target (Sysres) and the DLIB.

The Service Level is the PTF/APAR level of a given element.

PTF UZnnnnn
- UZ → PTF for MVS (VS1 is UX)
- nnnnn → sequential PTF number

APAR AZnnnnn
- AZ → APAR for MVS
- nnnnn → sequential APAR number

Usermods must not infringe upon IBM reserved names and they must follow the naming format if SMP is going to be used.

PUT YYMM for example: 8302 = Feb. 1983
- YY → year
- MM → month

IBM puts out about 10 PUTs per year.

SMP will also keep track of sysmod relationships: the pre- and co-requisites for a given modification. Any PTFs which have common elements must therefore belong to the same function.

SMP's Control Data Set (CDS) describes the relationship of the modules on the target. The Alternate CDS (ACDS) lists the relationship of elements on the DLIB (comes on DLIB).

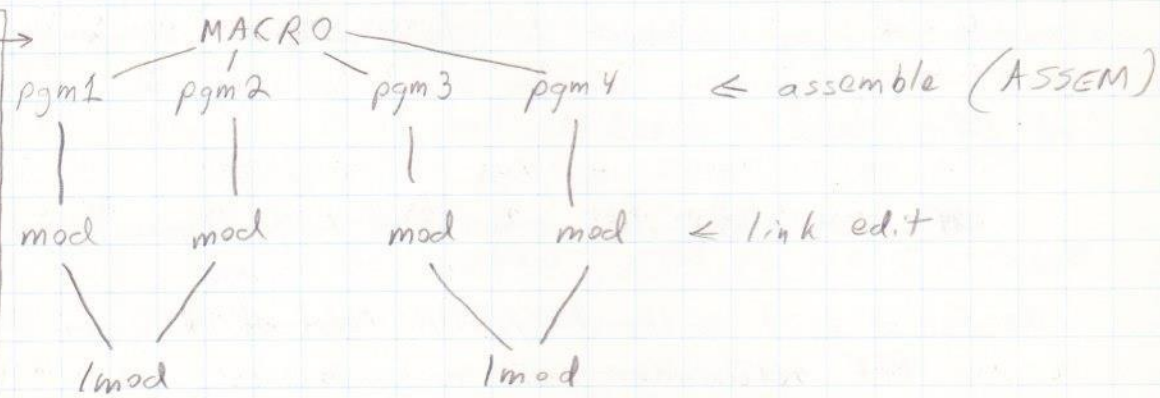
SMP permits complete backout of sysmods after they have been applied (but not accepted) to the target.

The user must decide when to accept elements on the DLIBs.

SMP must keep track of how elements are related as well as how sysmods are related. For instance, if a certain Macro is changed, SMP knows which modules contain this macro and which, therefore, must be reassembled and link edited. In this case, the Macro is an element.

SMP uses the standard utilities to actually implement any changes: assembly, linkedit, copy, update, etc.

If this macro is changed, SMP knows to re-assemble the programs, and then perform the appropriate link edits to actually implement the change in the correct load modules



The DLIB and ACDS must be updated synchroniously.

The JCLIN function of SMP takes the stage 2 sequential dataset and updates the CDS. SMP looks for assembles to create "assem" entries in the CDS. Also, SMP looks for Macros to update all its "Mac" entries. (Macros must be six characters or longer). The stage 2 is scanned for link edits and builds "MOD" elements (one per include card) in CDS. Each name card causes an "LMOD" entry to be created. "Copy" statements generate appropriate mod and "lmod" statements.

This is how the element and sysmod relationships are generated and maintained.

The first step (in full sysgen) is to copy the ACDS to the CDS. Then run the new stage 2 using JCLIN to update the CDS.

FMID: Functions have a naming convention: (IBM does not always comply).

+ppvrff

+ = feature
- E - SCP + free
- F - free feature
- H - licensed base
- J - licensed feature

pp = base program initials
v = version
r = release
ff = system

SMP control statements are supplied through //SMPCNTL dd card. Some statements:

- Receive – move PTF (APAR, or Function) from tape to temporary library (PTStore).
- Apply – install in target library (updates CDS).
- Accept – install in DLIB (updates ACDS) from PTS (<u>not</u> target).
- REJECT – removes modification from PTS.
- Restore – over-writes modification in target with module in DLIB. (Backout change)
- JCLIN – instruct SMP to scan stage 2 job stream.
- UCCLIN – User Control Language Input – updates SMP libraries.
- LIST – List contents from SMP libraries, CDS, ACDS, etc.

The PTS is a PDS. It has a system member and a MCS (Modification Control Statement) member which hold the actual PTF member. Another member is sysmod status which records the current disposition of the sysmod (REC, Apply, etc). The user may decide to automatically delete PTS members when they are "accepted." (Not good idea) The system member also holds the Function IDs for all modules. Thus, SMP need not clutter PTS with PTFs for functions not available on the system.

<u>PTS Members</u>

- System : FMID + Release information
- MCS : PTFs, APAR, Function
- sysmod : Status of each sysmod

<u>CDS</u> has 3 major entries:

- System : FMID + SREL (system release), CDSID - name of this CDS.
- Target Element Entry : contains Element relationships.
- sysmod : status

To list any entry:

LIST PTS System. (PTS ← dataset; System ← member; . ← terminator character)

LIST PTS MCS (member).

<u>ACDS</u>

- System :
- DLIB Element Entry :
- Sysmod :

Most folks leave all mods in apply status until the next PUT comes in. At that time they do the Accept and apply the new sysmods.

During processing, the element is marked with a status of ERR. Then, if the system goes down, the status is correct. If processing is successful, the ERR flag is removed.

Many installations have one PTS and then separate CDS datasets for their major systems. Then, PTFs need only be received once. Thus, SREC on the PTS volume must be complete for all systems in the shop. On the individual CDS volumes, the SREL will only identify system and release numbers for the appropriate system.

Other SMP datasets:

CRQ/ACRQ - a record of necessary relationships for features and functions which are not installed but which might be.

MTS (Macro Temporary Store) - holds macros not on target until Accept time.

STS (Source Temporary Store) - holds source code that has been applied but not yet accepted to DLIB.

SCDS (Saved CDS) - audit log type of CDS record. Used during RESTORE.

TLIB - Temporary Libs - used during unload of PUT.

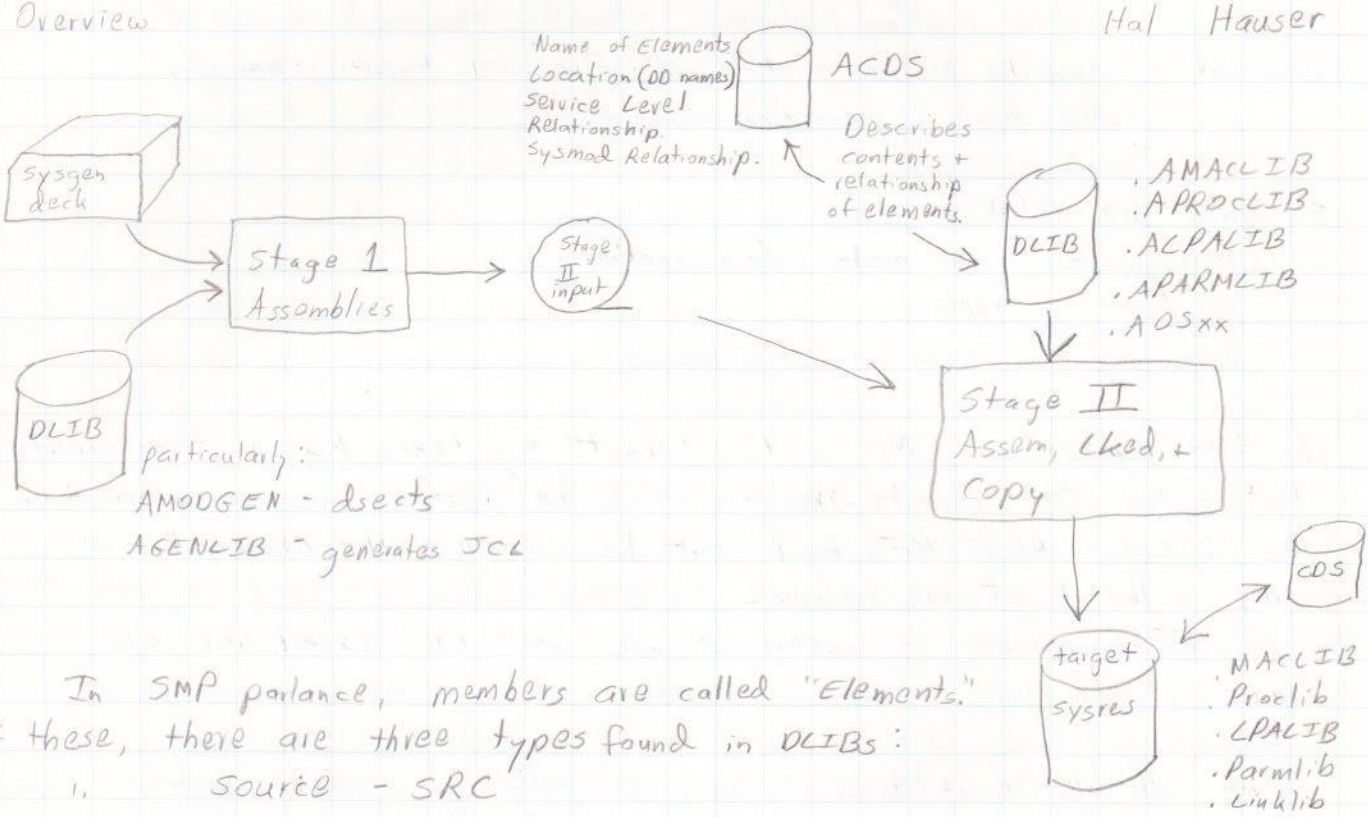
9-15-83
Cole Baker, Wally

"68% of customers who change suppliers do so because they perceive an attitude of indifference on the part of the original supplier."
Not price, performance, or service.

SMP 4

9-19-83
Hal Hauser

Overview



In SMP parlance, members are called "Elements." Of these, there are three types found in DLIBs:

1. Source - SRC
2. Macro - MAC
3. Load Module - MOD (not considered executable)

The target system may also contain three types of elements:

1. Macro - MAC
2. Source - source code placed by the user
3. Load Module - LMOD (considered executable).

The ACDS describes the contents and relationship of the DLIB. The ACDS contains the name of each element in the DLIB, the location of datasets correlated by DD name, service level, relationship of each element to other elements, and relationship of sysmods to each other.

The CDS is created for the first time by copying the ACDS.

Four types of changes:
Function - a major functional change. This is an operating unit.
PTF - "Program Temporary Fix" - a permanent fix.
APAR - A temporary "quick" fix. (Limited test). Used only to solve existing problems. Usually replaced by a PTF.
Usermod - can be either a PTF or Function type change. The trend is away from usermods.

Sysmods are related as
Prerequiste - one mode before another.
Supplant - replace
Requiste - coreguiste (together).

The test system in SMP is the target system. Rather than create a test system, SMP permits the target to be restored using information in the DLIB. RESTORE backs out the change & the changed elements in the target are replaced by corresponding elements on the DLIB. The RESTORE process is useful because SMP creates all the job streams itself.

SMP calls all the utilities necessary to implement changes, examine the return code from the utilities, and acts accordingly.

IEBUPDTE
IFOX00 — SMP supplies the appropriate DD statements
IEWL — for each of these utilities.
IEBCOPY
IMASPZAP

The CDS is created from the ACDS and then subsequently modified by SMP as it scans the stage ~~II~~ sysgen job stream. SMP looks for the appropriate EXEC statements, and utility control cards. The JCLIN process updates the CDS. It picks up name, location, and element relationships from the stage ~~II~~ input.

Sysmodid - 7 characters which identify a sysmod.
FMid - function modification id.
(Functions introduce new elements & SMP knows which function introduced each element).
The FMID is used frequently to look up the source code for a certain module.

FMID = tppvrff

t = E - system control program.
F - free feature (features depend on base).
H - licensed base.
J - licensed feature.
pp - base program initials
V - version
r - release
ff - system
ØØ - VS1 or MVS
Ø1 - VS1
Ø2 - MVS
} usually

SMP 4 is really SMP 1.4. SMP/E is really SMP 2.
SMP/E is an extension of SMP 4. Most information is still applicable.
SMP is executed usually by calling the module HMASMP.
Two types of functions:
Base -
Feature - depends on base.
SMP will examine the sysmod to determine whether a change is a feature or base.

//SMPCNTL is the dd which supplies the control cards to SMP. For instance, when SMP sees a RECEIVE it builds a PTS (PTF Temporary Store) entry.
Receive determines which sysmods to select for processing. The selection criteria is contained in the SREL (system + release) subfields of the system member of the PTS. The PTS may have multiple SRELs - one for each system: MVS, CICS, IMS, etc.

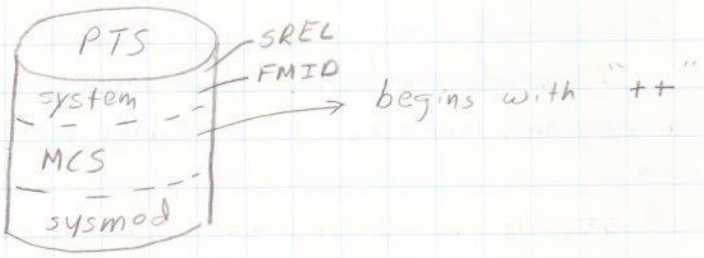
The system member also contains an FMID for each function in the system.

The FMID and SREL are compared to the label of the sysmod + are then copied to the PTS or excluded.

The PTS also has an MCS (Modification Control Statements) area which contains the actual code of the sysmod. Each PTF is a separate member.

Also, a Sysmod entry in the PTS is created for each MCS entry.

//SMPPTS



Elements may be inline with the MCS entry, or pointed to in another library (the MCS entry will point to this library). Elements may also live in SMP-allocated libraries. (Relative File Package) SMP allocates and manages Temporary Libraries (TLIBS).

Reject permits selected PTFs to be removed from the PTS. A Reject can remove an FMID but will not remove an SREL. It will remove appropriate MCS and sysmod entries. If any TLIBs were allocated in association with the PTF, they also will be rejected.

Apply causes sysmods to be taken from the PTS, processed according to the MCS information in the PTS and installed in the target system. A CDS has only one SREL. Thus, each system must have its own CDS.

The Target Element Entries of the CDS are of six types:

MAC - name of macro, location of source, and which programs use this macro.
SRC - name of Source Element Entry
MOD - module element entry is referenced during link edits.
LMOL - Load module element entry.
DLIB - a completely copied library from OLIB.
ASSEM - contains source code for stage II assemblies. (from Stage II input job streams)

Apply will create or update appropriate Target Element Entries. Also, back on the PTS, it will update the sysmod entry to show that the sysmod has been applied. This also holds a APPID field which identifies the system to which the sysmod has been applied. The APPID was picked up from the CDSID on the CDS volume.

Accept causes the sysmods to be moved to the DLIB and the ACDS updated. The system area of the ACDS holds a single SREL and CDSID. The ACDS has a DLIB Element Entry section which has three types of entries:

MAC
SRC
MOD

Accept also updates the sysmod entry of the PTS to reflect the acceptance. An option of Accept permits the sysmod on the PTS to be purged at acceptance. This is not a good practice. Turn off the purge flag so that the change may be accepted on multiple systems. The purge flag lives in the system area of the PTS.

SMP cannot back out changes from the DLIB.

Restore backs sysmods off the target system data sets. If an applied element is an update of an existing element, the back out copy will be taken from the DLIB. If the element is wholly new, it will simply be deleted. The DEE information will also be copied from the ACDS to the TEE to reflect the step back in maintenance level.

The PTS also has a Reject flag (default = ON) which will automatically reject any sysmod that is RESTORED. This is not desirable. Like the Purge flag, the reject flag lives in the system area of the PTS.

Other data sets:

CRQ, ACRQ - Conditional Requisite Queue & Alternate CRQ - relationships of PTFs for optional features which are not yet on the system.

MTS - Macro Temporary Storage - holds applied macros destined for DLIB maclib.

STS - Source Temporary Storage - holds source which has been applied but not yet accepted (SYS1.HASPSRC).

SCDS - Saved CDS: built at apply time so that the Restore will function. Acts like an audit log of CDS entries that are about to be changed by an apply.

TLIB - repository of "Relative File Package" elements during Receive (used for functions).

Data Set Entries

PTS

System entry: One per PTS + is required.
Must contain at least one SREL. Holds the Purge and Reject flags. To change the flags:

```
UCLIN PTS.
  DEL sys Purge Reject. <- turns off
ENDUCL.
```

These statements are read from //SMPCNTL.

MCS built as a result of RECEIVE. Each member has a seven character name and may be viewed with regular SPF browse. SMP heavily uses the user area of the PDS directory. SPF also uses it. Therefore, do not change anything with SPFedit because the Save will screw the directory.

Sysmod built as a result of RECEIVE. Holds status and summary information.

The system entry of the PTS contains a subentry DSSPACE which contains the space parameter, in tracks, for any TLIBS. SMP dynamically allocates TLIBs and will release any space not used.

The system area also contains subentries which permit the DD names of utilities to be changed. That way, each utility can print out of separate dd names. Standard parms may also be passed to selected utilities.

CDS

has SREL and CDSID mandatory entries.
One sysmod entry per APPLY'd sysmod.
It also has a NUCID which determines what name the current nucleus will be renamed to when a new NUCØ1 is built.
It also holds two flags: savemts + savests. If these are on (=YES) then the MTS and STS members will not be deleted at accept time. These members should be deleted.
The RETRYDDN permits avoidance of certain abends, such as SB37s, by compressing the dataset. CINKLIB cannot be compressed because this is where IEBCOPY lives (which is more than one module). "All" may be specifed, but cinklib had better not have an SD37 or SB37.

Target Element Entries:

Stores the Names of elements, the dd names of the libraries where the elements live, the service level of each element.
The MAC entry names the source elements which use the macro.
The SRC element specifies the mod. The MOD describes the modules which are link edited together to form LMOD.
The SRC entry is only created with a ++SRC statement or UCLIN. Each source entry names the distribution and system libraries which hold the element. The FMID entry describes which function created the element. The RMID specifies which element replaced the element. The UMID is the update id of elements which updated the existing element.
If FMID=RMID, the element is new.

9-20-83

Hal Hauser, Elliot Hamilton

The FMID is the "owner" of an element. The last update field (LASTUPD) shows how the particular element entry was last changed- UCLIN, FMID, etc. The DISTLIB field is the ddname and the low-level dsn qualifier for the owning library; For example: DISTLIB=HASPSRC.

UMIDs:

UZ = PTF
AZ = APAR
#R = RMS Amdahl user mod
#Z = Amdahl APAR

Changes to the F, R, & UMIDs will not cause the LASTUPD fields to be altered. (The LASTUPD fields seem to be fairly useless). SRC element entries are never built by JCLIN.

An ASSEM entry is usually created by JCLIN processing. The source contained in the stage II deck is placed in the ASSEM entry.

MAC entries are usually built by JCLIN processing. Any OP code of six characters or longer is considered a macro. The macro code itself will live in an ASSEM entry and the MAC entry will point to the appropriate ASSEM entry as well as to all the SRC element entries which use the element. This is done with the GENASM option.

MOD entries and LMOD entries are built by the presence of IEWL or IEBCOPY.

The name of the MOD element entry is taken from the INCLUDE statement. The DISTLIB library ddname is also taken from the include statement. The LMOD subentry names the destination load module. This comes from the NAME statement. The FMID is also a mandatory subentry.

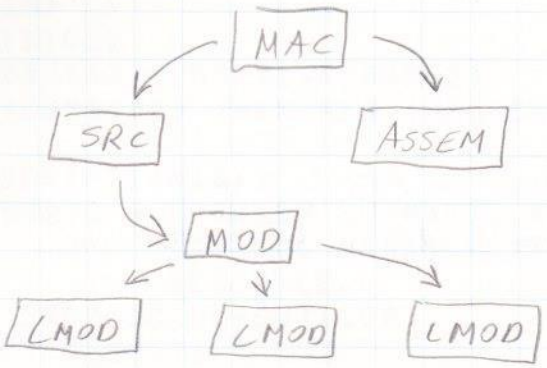
The MOD entry exists to trigger a re-link edit in case any MOD changes. A single Mod element entry may point to multiple LMODs.

The LMOD entry represents a load module in the target system. The entry name comes from the IEWL NAME statement. Also, the entry contains the ddname of the SYSLIB where the load module lives. It holds the linkedit parms, too. Any statements in the IEWL deck other than INCLUDE and NAME statements are incorporated under ++LMODIN. (Order, etc). The SYSLIB comes from the low level qualifier of the SYSLMOD.

LMOD entries are also built from IEBCOPY jobs. SMP picks up the SELECT statements for the LMOD name and will refer to the OUTDD value for the SYSLIB.

DLIB element entries are created with JCLIN and UCLIN. These are generated by an IEBCOPY with no SELECT statement (thus entire libraries are copied). The DLIB links a DLIB and a syslib. The DLIB Element entries are named as the INDD file.

If a macro is changed, and the MAC entry has a GENASM subentry, then SMP searches for an SRC entry that matches a MOD entry. If found, SMP will perform an assembly of the effected module. Then SMP checks the MOD entry for an LMOD subentry. If this exists, SMP will then go to the LMOD element entry to get information to do the link edit.



Sysmod Status Entries

RGN - "Regen"; may appear in the ACDS at accept time. This is left over from SMP 3.
BYP - Bypass; says user requested to bypass in case of problem and a problem did occur with the sysmod thus flaged.
ERR - ERROR status. Sysmods are marked with ERR during processing
RES - Restore
REC - Receive
APP - Applied
ACC - Accept

Element that have been superseded are retained in the sysmod file so that SMP knows not to apply this sysmod if it ever comes up again.

The ACDS is very similar to the CDS. There are no LMOD, ASSEM, or DLIB entries.

Instream JCLIN: JCLIN processing may be invoked with the ++JCLIN MCS card which comes in via SMPPTFIN ddname. The JCLIN commands go on the PTS and are not executed until the particular PTF is Applied.

When JCLIN comes in instream with a PTF, the CDS entry to be changed is copied to the SCDS. Instream JCLIN is restorable. The source JCL follows the ++JCLIN card.

UCL/UCLIN

UCLIN cannot change MCS entries in PTS.
UCLIN can only delete SCDS, STS, & MTS elements.

Modification Control Statements (MCS) (see pages 251-291)

MCS comes in through //SMPPTFIN. Statements begin with ++ in columns one and two.

Each sysmod has one header statement containing Function, PTF, APAR, or usermod designation. Then it has the VERify or relational environmental specifications. The ++JCLIN preceeds inline JCL.

A sysmod id can be Ø-9, A-Z, or #, $, @; and can be 7 characters in length. IBM reserves A-K and U-Z as beginning characters. SMP doesn't care.

The header statement also specifies whether the sysmod is instream, Relative File Packaging, or in a user library.

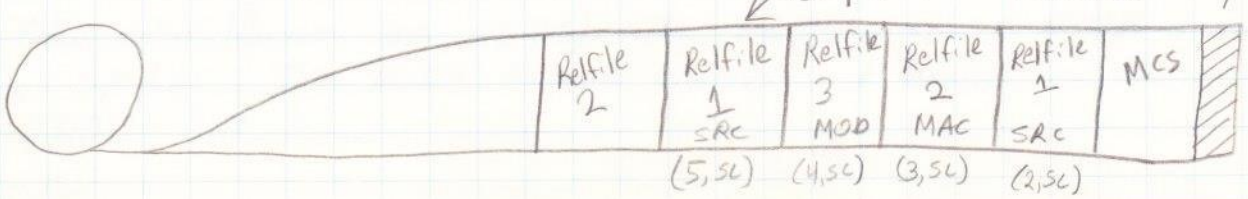
Example:

++FUNCTION (FUNØØØ1) FILES(3)

↖ SMP will allocate 3 files for TLIB.

Relative File Packaging

The MCS are on the first file of the tape (Label=(1,SL)). The first relative file in File 2 (2,SL). Even if a tape contains more than one function, the MCS for all functions are on the first file. The sysmods themselves are still numbered relative to the first within a function.



 TAPE:M

```
PRINT DATA ENTRY  P030075

H P030075 D=MAY82 E=MAY83 S=PTF C=GX5 L=25
T PTF UZ54911 5752 SC1UC TAPE:  8203

++ PTF (UZ54911)         /*    5752-SC1UC-EUT1102-SCP
//UZ54911  JOB 5752-54911-0,SC1UC,MSGLEVEL=(1,1),CLASS=A     */  .
++ VER (Z038)
   FMID(EUT1102)
   SUP  (UZ30162,AZ59271,AZ38892)                              .
++ IF FMID(EDM11C2) THEN REQ (UZ30065,UZ30062)
/*
PROBLEM DESCRIPTION(S):
        OZ59271 - PROBLEM SUMMARY: ABEND WAS DUE TO AN ERROR IN
                  CATALOG MANAGEMENT THAT ERRONEOUSLY DEQUEUED THE
                  RESOURCE.
COMPONENT: 5752-SC1UC-EUT1102
APARS FIXED: OZ59271
SPECIAL CONDITIONS:
 NONE:
COMMENTS:          NONE
                  20 JANUARY 1982
 */.
++ MOD (IEHMVESC) DISTLIB(AOSU0)
          LEPARM (DC,LET,LIST,NCAL,REUS) .

$EOM
```

↑ This became a PTF in Error (PE)
+ two APARs were issued; finally,
a new PTF was issued.

 TAPE:MA

PRINT DATA ENTRY E124571

H E124571 D=SEP82 S=MVSEWS82 C=GX4 L=00047
T F999-57924-0Z62396--5752-SC-1UC--PE-PE54911
T FIX FOR ABEND138 ON SYSCTLG ENQ FOR UZ54911 IN ERROR.

FESN5501706-UT1
REPORTED RELEASE R102
ERROR DESCRIPTION:
APPLICABLE PTFS PE54911-T8203
FIX FOR ABEND138 ON ENQ OF SYSCTLG AND ABEND130 ON DEQ OF
SYSCTLG FOR APARS 0Z61161 AND 0Z59271 ON PTF UZ54911
IS IN ERROR. THE FIX CHANGED THE RNAME LENGTH FROM 12 TO 8, THUS
ENQUEUEING ON SYSCTLG ONLY AND NOT SYSCTLG AND THE UCB ADDRESS.
PROBLEM SUMMARY:
ABEND138 ON ENQ AND ABEND130 ON DEQ OF SYSCTLG WHILE COPYING
CATALOG USING IEHMOVE.
PROBLEM CONCLUSION:
NEITHER THE ENQ NOR THE DEQ SPECIFIED A CONDITIONAL REQUEST FOR
THE RESOURCE 'SYSCTLG'. PROGRAM WILL BE CHANGED TO SPECIFY RET=
HAVE ON BOTH THE ENQ AND THE DEQ FOR SYSCTLG.
TEMPORARY FIX:
IF UZ54911 IS ON, THEN USE THIS ZAP:
++ APAR(AZ62396).
++ VER(Z038) FMID(EUT1102) PRE(UZ54911).
++ ZAP(IEHMVESC) DISTLIB(AOSU0).
NAME IEHMVESC
VER 00A4 C008,C000
VER 0254 C008,4000
REP 00A4 C00C,C100
REP 0254 C00C,4100
/*
IF UZ54911 IS NOT ON, THEN USE THIS ZAP:
++ APAR(AZ62396).
++ VER(Z038) FMID(EUT1102).
++ ZAP(IEHMVESC) DISTLIB(AOSU0).
NAME IEHMVESC
VER 00A4 C00C,C000
VER 0254 C00C,4000
REP 00A4 C00C,C100
REP 0254 C00C,4100
/*
ZAPS ADDED 82/06/11
MODULES/MACROS: IEHMVESC
SRLS: GC26390200
APPLICABLE COMPONENT LEVEL/SU:
R102 PSY UZ57924 UP82/08/06 P DLL5 8208
CIRCUMVENTION:
REPINNED RP82/09/24 ANSWR TO CORRECT FORMAT
82/09/24,CHICAGO FS

$EOM

TAPE:H

```
PRINT DATA ENTRY  P080055

H P080055 D=SEP82 E=SEP83 S=PTF C=GX5 L=26
T PTF UZ57924 5752 SC1UC TAPE:  8208

++ PTF (UZ57924)         /*    5752-SC1UC-EUT1102-SCP
//UZ57924  JOB 5752-57924-0,SC1UC,MSGLEVEL=(1,1),CLASS=A      */  .
++ VER (Z038)
   FMID(EUT1102)
   SUP  (UZ54911,UZ30162,AZ62396,AZ59271,AZ38892)              .
++ IF FMID(EDM1102) THEN REQ (UZ30065,UZ30062)                 .
++ IF FMID(HDP1102) THEN REQ (UZ57965)
/*
PROBLEM DESCRIPTION(S):
        OZ62396 - PROBLEM SUMMARY: ABEND138 ON ENQ AND ABEND130 ON
                  DEQ OF SYSCTLG WHILE COPYING CATALOG USING
                  IEHMOVE.
COMPONENT: 5752-SC1UC-EUT1102
APARS FIXED: OZ62396
SPECIAL CONDITIONS:
 NONE:
COMMENTS:
                  14 JUNE 1982
 */.
++ MOD (IEHMVESC) DISTLIB(AOSU0)
          LEPARM (DC,LET,LIST,NCAL,REUS) .

$EOM
```

Each function on a tape has a different DSN consisting of SYSMODID.Fn
← where n = relative file number.

When SMP does dynamic TLIB allocation it uses these dsns prefixed by the prefix specified in the PTS System entry:

DSN = Prefix.Sysmodid.Fn

These may be preallocated and cataloged by the user if desirable. SMP uses an SVC 32 for dynamic allocation of datasets. SMP does not automatically catalog. Thus, beware DBMS products which delete uncataloged datasets.

9-21-83

++VER - Verifies that the operand of ++VER matches the SREL in the PTS. The FMID may optionally coded (required on PTF, APAR, + usermod).

++VER(Z038) FMID(FUN0001).

The SUPersedes and REQuisites may also be coded with the VER. The VER always directly follows the header MCS. The FMID is checked against the Sysmod entry of CDS/ACDS at Apply/Accept time.

The VER statement may be used to delete a function. The preceeding statement must be introducing a new function.

++VER(Z038) DELETE(FUN0001)

This deletes the function and all service (features and base). Thus, a single delete may remove multiple functions. A delete cannot be restored or reapplied. The original function may be reapplied if its CDS Sysmod entry is first deleted.

The NPRE option of VER specifies a negative prerequisite. It specifies mutually incompatable sysmods.

++VER(Z038) FMID(EVT0108) NPRE(EVT1102, JVT1112)

NPRE is ravely used.

VERSION option of VER: this names another function which is henceforth considered to be inferior to the new function. If the new function and the inferior function share any elements, the new function now owns these common elements. This is a way of changing the ownership of elements.

VER may also have a PRE option which specifies any Prerequisite functions.

++VER(2Ø38) FMID(FUNØØØ1) PRE(uzØØØ1Ø, uzØØØ11)

The functions named by PRE must be in place (applied) by the time FUNØØØ1 is applied. This statement is often used with the ++ZAP statement to make sure the module is there before trying to perform the ZAP.

If SMP is told to Apply a sysmod with a prereq, and that PREreq is not already applied but is on the PTS, SMP will try to Apply the PREreqs out of the PTS. PREreqs may be avoided by specifying BYPASS.

The REQ option of VER specifies sysmods which are required but does not specify an order of application of the sysmods. The new sysmod, or the one named in the REQ must both go on, but it does not matter which one goes on first.

The SUPersede option of VER specifies that the new sysmods obsoletes the one specified by the SUP. The ones specified by the SUP do not have to be in the system.

Up to 255 VER statements may be coded.

Conditional Processing

The ++IF is always paired with a VER statement. The VER/IF pairs follow the header. At Apply, an IF effects the CRQ. At Accept, it acts on the ACRQ.

++IF FMID(FUNOOO1) THEN REQ(UZØØØØ1).

↖ If this function is on, then apply ↑ this sysmod.

The IF comes in on a PTF that hits another function than that specified by the PTF. Therefore, if you use the PTF as a whole, then, via IF, you must apply another sysmod to another function.

If the IF condition is not true, then a record of it goes into the CRQ (for Apply) or ACRQ (for accept) in case you ever install the object of the IF.

Example:

DS/EF is a feature that depends on DS/DF. Then a PTF arrives to fix DS/DF. If you use the PTF and you have DS/EF then apply a sysmod to DS/EF too.

If you use the PTF to DS/DF and do not have DS/EF, then SMP will record the sysmod needed by DS/EF in the CRQ/ACRQ in case you ever install DS/EF. When DS/EF is installed, it will supersede any sysmods which are no longer applicable.

Construction Statement

The ++JCLIN statement invokes JCLIN processing. The JCL may follow the ++JCLIN statement (direct) or the ++JCLIN statement may point to a library of JCL (indirect).

Indirect:

```
++JCLIN  TXTLIB(TEXTDD)
```

Direct:

```
++JCLIN
//  EXEC
//
  :
```

The JCLIN statement must be after the header and relational statements. (Not necessarily immediately after) Only one JCLIN statement per sysmod is allowed (maximum). The PDS member name must be the sysmod name. (sysmodid).

Element MCS

++SRC
++MAC
++MOD

} These are used for module replacement, addition, and deletions. If the FMID is equal to the RMID, then the element is new, an addition.

Example:

```
++SRC(SRC01) DISTLIB(DLIBSRC) SYSLIB(SRC).
SRC01  CSECT
  :
  :
END
++SRC(SRC02) DELETE.
```

} instream code (or use TXTLIB(DDname) for indirect).

In case of a RESTORE, SRC02 will be successfully restored. Its source lives on the DLIB & its SRC element entry was moved to SCDS.

A ++SRC entry may force the creation of a MOD entry if DISTMOD option is coded:

++SRC(SRC01) DISTMOD(ALIB) ...

In addition, this may cause an LMOD entry to be created.

Minor changes may be implemented with:

++SRCUPD } use IEBUPDTE
++MACUPD }
++ZAP - uses IMASPZAP

When these are employed, the UMID field is filled in. The user must supply the IEBUPDTE control information for SRCUPD or MACUPD. IEBUPDTE keys on line sequence numbers (column 73 for 8 bytes). Coding is identical for SRC or MAC updates.

For using ZAP, the user must code the IMASPZAP control cards. SMP will perform all the super zap verifies before doing any changes. With ZAP control cards, one may specify a particular CSECT within a loadmodule.

Regression

If a user attempts to replace a sysmod that has already been replaced, an ID check will occur. Thus, to preceed, SUP the sysmod which replaced the sysmod you wish to replace.

SMP will allow an update of a replaced module but "results are unpredictable."

SMP Control Statement Details

9-22-83
Hal Hauser

LIST - used to print SMP data sets. May also be used to produce a cross-reference (LIST XREF) of sysmod relationships. Output goes to SMPLIST.

UNLOAD - used to convert from SMP3 to SMP4. This produces a ACLIN jobstream out SMPPUNCH. Unload may also to unload a a specific FMID as a backup copy. This may be used to backup a function prior to deleting it via a VER. This is not frequently used.

sub entries: ("list")

Select (-,-,-) -

Group (-,-,-) ← Group changes meaning depending on control statement.

Mass Mode - process everyone that qualifies.

Exclude - may be used with mass. ← Don't compare FMIDs at RECEIVE.

BYPASS - PRE, REQ, IF REQ, FMID, + ID. ← Lets regression happen. This is common.
→ conditional maintenance

RC - Return Code; Waive return codes between SMP functions.
RC(function = nn) < if RC > nn do not process.

COMPRESS [ALL] [ddname list] - SMP will do compresses of libraries when processing is complete.

DIS (Read | write) - place directory in memory.

CHECK - dry run of operation. Mainly used to check for the appropriate dd statements.

RETRY [NO | YES] - turns off/on automatic retry for the operation.

NOJCLIN - ignore any ++JCLIN statements at Apply time.

NUCID - supplies alternate NUCID for Apply. The current nuc is renamed to this value.

USERMODS - required when accepting usermods.

APARS - required when accepting APARS.

NOAPPLY - used to bypass apply on an Accept.

PURGE - goes with REJECT and will force delete of all sysmods that have been accepted. Used to clean PTS.

REUSE - Preserves SMPWRK3 (which holds object mods).

Control statements

RECEIVE - input is from SMPPTFIN. Each sysmod on the PTF is compared to the SREL and FMID in the system entry of the PTS (the Function sysmods do not check FMID). RECEIVE may dynamically allocate TLIBs (using OSSPACE values in PTS). The FILES subparm in the sysmod header will specify the number of TLIBs. BYPASS(FMID) may be specified at RECEIVE to avoid the FMID check.

Sysmods cannot be "Re-received." They may be Rejected and then Received again, but not received on top of previously existing entries.

(see CSD Ø4ØØ for logic diagram)

REJECT - deletes entries from MCS and Sysmod areas of PTS. If the REJECT is deleting a function that has not been applied or accepted, then the system FMID entry is also deleted.

Mass REJECTs will delete sysmods that have only the REC flag on.

With the PURGE option, SMP will delete from the PTS any sysmods that have been accepted successfully. Often found in the form:

REJECT PURGE COMPRESS(SMPPTS).

APPLY - input is from the PTS, + TLIB, or Userlib (TXTLIB or LKLIB). Output is to CDS. During processing the SREL and CDSID are checked in the system area of CDS. SMP will build the sysmod CDS entry. SMP will finally build the TEE and perform the necessary Assemblies and linkedits and place the module in the system libraries. SMP may also place the sysmod in the MTS. If the sysmod contains an IF, a CRQ entry is built. If a function is being installed, the CRQ is refuenced. If a sysmod deletes or updates an element, the CDS entry is moved to the SCDS prior to implementing the change. (SCDS will also be referenced for ++JCLIN processing).

A Mass Apply will take all sysmods in the PTS which have not already been successfully applied and apply them. A selective Apply will re-apply a sysmod that has already been applied or accepted.

If Group is specified, SMP will check the PRE + REQs of the named sysmod, and will then attempt to apply any requisite sysmods that are not already applied.

APPLY GROUP(UZ0000002)

The sysmod(s) named in the group list will be re-applied regardless of whether it is (they are) already applied. A sysmod that has a ++VER DELETE will not automatically be added to a group list by inference. Nor will a base function be automatically applied by naming a feature in the GROUP statement.

Apply will not zap the same element twice in one pass.

Apply check will not invoke assemblers and link editors. It will check for PREs, REQs, and the correct dd cards. Usermods may easily be regressed; thus, check is useful for spotting conflicts.

(See Apply logic diagrams beginning at p. CSD11ØØ)

RESTORE - Causes the Element Entry RMID and UMID to be copied from the ACDS to the CDS. The element modules are moved from the DLIB back to the target.

Restore may also use the PTS and the TLIB (particularly if the Reject Flag is on), SCDS, & S/MTS.

RESTORE does not have a mass mode. Sysmods may be specifically selected via SELECT.

Restore will also use the GROUP option. This causes SMP to examine the REQ, PRE, FMID, and SUP chains. SMP will restore in both directions on the chain so long as all sysmods involved are not yet accepted. The prerequisite of a sysmod named in a group field will be restored along with the specifically named sysmod. Objective: name the oldest sysmod that is PRErequisite to the sysmod you wish to RESTORE. The oldest sysmod may be spotted in the UMID field as the left-most entry.

Moral: Accept a function before applying maintenance to it. Otherwise, if a PTF is bad, it will want to take out the function.

Restore will not work on a sysmod that contained a ++VER DELETE card.

(See Logic Diagram, p. CSD26ØØ)

Perform a "Restore Select() Check" naming a bad sysmod to find out the names of all sysmods which must come off with the bad sysmod.

ACCEPT - Input is from PTS and TLIB or userlib (TXTLIB or LKLIB). SMP also looks at the Purge flag to know if it should reject the sysmods from the PTS after a successful Accept.

SMP then checks the SREL and CDSID of the ACDS, updates the DEEs, and moves the elements to the DLIB. SMP may also use the ACRQ (to record conditional relationships), S/MTS to delete elements being accepted into DLIB, and SCDS to delete the backup copy of the CDS sysmod entries. It touches the CDS only to update the sysmod entry to show acceptance. (If NOapply is specified, SMP will not access CDS, SCDS, or S/MTS).

Accept may be used in mass form. Candidate entries must be in both REC and APP status.
The Group apply functions like Apply.
Optrons APARS and USERMODs must be coded with Accept to accept these types of sysmods.

Syslib:
During Apply, the syslib DD should be:

```
//Syslib DD SYS1.MACLIB
       DD SYS1.SMPMTS
       DD SYS1.AMODGEN
       DD SYS1.AMACLIB      <- by tradition
```

During Accept, the syslib DD should be:

```
//Syslib   DD  SYS1.AMACLIB
           DD  SYS1.AMODGEN
```

Some shops are running SMP using only the Apply configuration of Syslib. If they do a partial accept, they may get into trouble because of a macro update that does not appear in the target system.

Program Update Tape — Elliot Hamilton

Five types of PTF sysmods:
- Normal installation
- UCLIN statements required (Delete CMOD, for example).
- Sysgen is required
- Engineering change required - hardware + software change.
- Documentation change only (i.e. Avoid fixing problem)

Occasionally a PTF is discovered to be bad after it is packaged as a PTF. These kind are known as PEs (PTFs in Error). A PE chain (PECHN) is a list of PTF which depend on a PTF which is a PE. Thus a PE-APAR is issued to fix the PE. Eventually, a new PTF is issued which supersedes the formerly bad PTF.

A file on the PUT becomes the PUTPDS. This contains a member (CNTLMSTR) which lists all PEs known to be bad at release. A user can phone Level 1 for the PUT Bucket which lists the PTF which have gone bad since release. The user can manually enter the PUT Bucket entries into member INPUTCTL. These two members are used to form the SMP Exclude List. Another member DLLEXCL (Data Link Library Excludes) can also be used to supply excluded PTFs. This member is created via a tp link between an IBM mainframe and IBM.

PUTCNTL causes SMP to list PTFs which have been received to the PTS. The output of PUTCNTL is a list of sysmods which are on the particular system but which will be excluded.

The Program Update Tape holds PTFs, JCL, SMP control information, and documentation. IBM releases about 10 PUTs per year. They are not cumulative.

Name: YYMM ← month
~ year

[Cumulative maintenance is supplied on a Service Update Tape (CUM tape) and is usually delivered with a new feature.]

PUTs are NL format.
Files:

1. PTFs
2. PTF List
3. Dummy
4. SMP/E
5. JCLIN (Not Used)
6. UCLIN (Not Used)
7. PUTLOADER Program - creates PUTPDS.
8. PUTPDS (one per SREL)
9. DOCPRINT program.
10. PUT Document
11. PTFXREF (one per SREL)

File 7, Putloader, can read file 8 and build the PUTPDS. Since 8305 the cross reference refers only to the fixes on the tape. Prior to this, the cross reference was huge: inclusive for all time.

Procedure

1. Read Shipping Memo
2. Run Docprint
3. Read Put document

A PUT Document has four sections:

GenR - General information
SING - Single PUT technique
MULT - Multiple PUT technique
PGEN - Pre-Gen a system (Accept Noapply, followed by a sysgen).

4. Load PUTPDS
5. Chose SING or MULT
6. Do Research
   A. Perform Accept of last PUT
   B. Backup system
   C. Get PUT Bucket
7. Do setup
   A. Build INPUTCTL. < Run SMP Receive
   B. Run PUTCNTL Program.
   C. Run SMP Apply Check
   D. Resolve SMP check errors.
      Most errors are solved by adding the offending sysmod to the Exclude list.
   E. Re-run PUTCNTL
8. Install (Apply)
9. Resolve Regressions (i.e. Amdahl RMS)
10. Post
    A. Accept sysmods (optional)
    B. "Reject Purge" unused sysmods from PTS

This PUT process requires about a day.

Sysmod Construction

9-23-83

Why sysmod?

1. Provide a service not meet with the OS.
2. Install an un-named APAR.
3. Add an SVC.
4. Alter MSTRJCL (Master Scheduler startup JCL).

A change to IBM code must be recorded in SMP or else it could be regressed by a PTF (it can still be regressed, but SMP will issue a warning).

SMP provides standard documentation of all system modules, including sysmods. It also governs the order in which these modules are put together. The price of these features is properly coding the necessary PRE and REQ statements for particular usermods.

Sysmodids:

A - K	— IBM (A-Apar; E,F,H,J-Functions)
U - Z	— IBM
¢ - T	— Available for user
#, @, $	- Available for user
Ø - 9	- Available for user

Try to make the sysmodid meaningful for the installation. The sysmod should have its own FMID (i.e. should be a function). If no PREs or REQs are specified, SMP will apply usermods last, after all Functions, PTFs, and APARs.

Sysmods come in three types of packaging: Relative File, In-line, + Indirect.

For usermods, do not accept the code into the IBM DLIBs. Usermods are only Received and Applied.

Midwest office: (312) 692-7520
Hal - HLHØØ
Elliot - ESHØØ

A usermod can be introduced to SMP by coding a ++Function, ++VER, and ++SRC statements followed by an IEBCOPY stream under ++JCLIN. The copy of distribution module libraries to target libraries generates the MOD and CMOD entries.

Review

Elliot

Prepare for SMP:

1. Sysgen
2. Allocate PTS, SCDS, CDS, CRQ, MTS, STS
3. UCLIN system entries
4. Copy ACDS to CDS
   Copy ACRQ to CRQ
5. JCLIN processing of Stage II input.
6. Install maintenance.

//SMPCNTL Maintenance:

RECEIVE → open SMPPTFIN

REJECT

APPLY

RESTORE

ACCEPT

LIST

UCLIN dsn.

ADD, DEL, REP, ENDUCL

JCLIN. → open //SMPJCLIN

* UNLOAD → out //SMPPUNCH

LOG ' '. → enters message in SMPLOG.

* RESETRC → Reset return code. (Does not effect 'Highest CC' field.)

* DEBUG → leave trace info in SMPOUT.

* - Probably will never be used.

Sysmods: Input from //SMPPTFIN

++ Function
++ PTF
++ APAR
++ Usermod

++ VER
++ IF
++ JCLIN
++ SRC, MOD, MAC, SRCUPD, ZAP, MACUPD

//Exec Pgm = HMASMP

PARM = DATE=(U, IPL, Repl, yydd, ← current-date, ← query operator

NORECOVERY ← will not try to recover from abends.

FMID=nnnnnnn ← SMP4 will process SMP3 sysmods.

"Restore" of accepted sysmod:

1.) Backup DCIB and ACDS
2.) Receive, Apply, Accept
3.) Pack restore of DLIB {The ACDS no longer thinks sysmod is accepted).
4. Perform SMP Restore (may have to change flags with UCLIN).

Another way would be to rework the original sysmod as a usermod and let it supesede the one which is bad. Be sure to get the PRE + REQ correct.

SMP/E

Hal

FMID = HMP1100. This supports MVS 3.8 and VS1. MVS/XA does not require SMP/E. SMP/4 is still supported.

Documentation may be better.

The major difference is the use of VSAM rather than PDSs. The data base organization has been reworked. CDS, ACDS, CRQ, + ACRQ are gone. A CSI (consolidated software Inventory) is new. PTS Sysmod Status is now in CSI.

SMP/E can use ISPF Dialogue. This causes much of the overall system work to be done interactively. ISPF also has a tutorial.

SMP/E is installed with SMP4. SMP4 datasets may be CONVERTed to SMP/E format.

Amdahl Research Tools

9-26-83
Carol Edmunds

Naming Conventions:

MVS	5752	Z	Microfiche goes by these numbers.
SVS	5742	Y	
VS1	5741	X	
MVT	5380	S	
VM	5749	V	
More than one	5744	R	← things like EREP + SMP

SMP requires a seven character FMID, PTF, or APAR.

IBM	AMDAHL
UZnnnnn ↖ PTF	#Z21234
AZnnnnn APAR w/fix ↑ APAR (or "I"). ↓ OZnnnnn	#Z51234 ↖ 2 or 5 mean Amdahl PTF
	#Z91234 ↖ APAR

When generating an APAR, you get a six digit number. This does not include the "9" in #Z9nnnn.

Infosys / Infomvs - an IBM product. Can be accessed sequentially or by keyword. Has information about PTFs. Made up of files such as

B. Programming announcements
E Early Warning System
G Support Center Document
U User Generated.
H Help

Enter "DEF A" to correctly define files & access anything.

The KWG (keyword glossary) is useful for finding the naming format used in Infosys.

After a KWS, hit PF12 for an abstract of each hit. Pressing PF12 again will cause it to switch back to the list. When looking at detail, press PF11 to see the next detail.

"F word" may be used within text to find a word.

Help is PF1 & is useful.

H102 provides a summary of all files.

RECALL will redisplay last command.

KWS Abend Loop Wait MVS
these are Anded for OR:

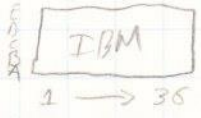
KWS Abend | Loop | Wait | MVS

KWS A xxx xxx
searches more libraries.

EWS Microfiche - source of most current information. This is 6 weeks to 6 months ahead of Infosys. Distributed monthly.

In lower left corner of first card is an index.

IBM is read from Left to Right



card # frame #
↑ coordinates

Look at APAR list, for instance & pick up .t card #, frame number. If an APAR references a PTF, then the APAR has been superseded.

New PIN Abstracts are organized by operating system.

Ten valid symptoms:

Abend/xxx	INCORROUT ← misc.
Loop	PERFM
Wait	MSG xxx-x
DOC	Wait xxx
Info	

PIN Abstracts: third group

Amdahl Software

RMS - Machine check
Channel check
EREP

SEA, SES, SPA - take advantage of extended facilities.

ECS - adds 16 channels for use as alternate paths.

Distributed in COMET and Release tapes.

Release:
- Microfiche
- Product Memo
- Installation Job Stream

Comet - follows PUT by 30 days. Corrects regressions caused by IBM PTFs. Comet provides one tape per user covering all software licensed by the user. Each COMET is cumulative between releases.

FMID: third character
- 2 = RMS
- 5 = Amdahl PP
- 9 = APAR

QWIK

Menu for

- * ITS - incident tracking
- HITS - hardware tracking
- * GS4 - search Services (Symptom searches) - High Level (very useful)
- * ERIN - Maintenance Research & Info Services (Search PUT+COMET) (for PTF searches & PE)
- AWS - Amdahl Warning System (add-an APAR)

HELP toolname → will describe how to use particular tool.

Tools are documented in Field Operations Information Systems.
To delete an Incident, send a MB to QWIKDEL.

GS4 (select EWS & AWS libraries)

Glossary Search - Searches for APARs. The User defines the libraries to be searched.

search operands are ANDed.

Pressing PF5 from GS4 goes to ERIN LIST where you enter a PTF#. This gives the PTF cover letter. PF3 then goes back to GS4.

PF6 from GS4 goes to ERIN search. PF3 returns to GS4.

GS4 will also accept a date range to narrow the search
DATEYYMM-DATEYYMM

ERIN

Not as friendly. This is based on SMP/E keywords:

MOD/
PUT/ [PE] ← finds errors on PUT.
ELM/
C/ ← comet
MAC/

Entries may be Selected or Listed.
ERIN is useful for finding what is wrong with a PTF that has been applied.

The EA flag is an exclude only for the AMS level.

Amdahl Warning System (AWS) - fixes for anybody's code. AWS generates microfiche every two weeks. Distributed to Amdahl field people, not customers.
(Also is a HWS for hardware).

Not Qwik

PTF Command → Engineering Software Group
PSSTOOLS
PTFINFO
Mailbox

PTF command → a primative ERIN.
Format:

PTF MVS
PPMVS
SVS
MVT
DOC ← Documentation. IBM/Amdahl conflicts
L *Index ← help
L FIXLIB ← a repository for Q+D fixes.

PSSTOOLS - A member describes all tools. "SERVICE" describes what is current (COMET, PUT) in the field. (TSO Proc may have to be redefined).
PTFINFO - originally used when Amdahl put out PTFs. This is used to enter info into ERIN.

Get on the mailing list for the "Software Availability Schedule." 9-27-83

VM Maintenance Philosophy

VM maintenance does not use SMP. VM in Amdahl is updated every 60 days regardless of IBM scheduling. VM maintenance is tracked in AWS. (VM maintenace is running about every 140 days).

FOIS - mailbox id for tools comments/complaints.

DATA CONTROL BLOCK--SAM

Offset	Bytes and Addresses	Field Name	Field Description, Contents, Meaning
			FOUNDATION SEGMENT--BEFORE OPEN
		DCBMACR	Code
		Byte 1	QSAM - Input
50 (32)		0...	Always zero for QSAM.
		.1.. .... G	GET
		..0.	Always zero for QSAM.
		...1 .... M	Move mode.
		.... 1... L	Locate mode.
		.... .1.. T	Substitute mode.
		.... ..1. C	CNTRL
		....1 D	Data mode.
		Byte 2	QSAM - Output
51 (33)		0...	Always zero for QSAM.
		.1.. .... P	PUT
		..0.	Always zero for QSAM.
		...1 .... M	Move mode.
		.... 1... L	Locate mode.
		.... .1.. T	Substitute mode.
		.... ..1. C	CNTRL
		....1 D	Data mode.
		Byte 1	BPAM - Input
50 (32)		00..	Always zero for BPAM.
		..1. .... R	READ
		.... .1.. P	POINT (which implies NOTE).
		...x x.xx	(Reserved bits)
		Byte 2	BPAM - Output
51 (33)		00..	Always zero for BPAM.
		..1. .... W	WRITE
		.... .1.. P	POINT (which implies NOTE).
		...x x.xx	(Reserved bits)
			FOUNDATION SEGMENT--AFTER OPEN
40 (28)	2	DCBTIOT	Offset from the TIOT origin to the TIOELNGH field in the TIOT entry for the DD statement associated with this DCB.
42 (2A)	. . 2	DCBMACRF	Contents and meaning are the same as those of the DCBMACR field in the foundation segment before OPEN.

Offset	Bytes and Addresses	Field Name	Field Description, Contents, Meaning
			FOUNDATION SEGMENT--AFTER OPEN (Cont'd)
44 (2C)	1	DCBIFLGS	Contents and meaning are the same as those of the DCBIFLG field in foundation segment before OPEN.
45 (2D)	. 3	DCBDEBAD	Address of the associated DEB.
			NOTE: The above fields are overlaid on the DCBDDNAM field during OPEN and are restored to their original form at CLOSE.
48 (30)	1	DCBOFLGS	Contents and meaning are the same as those of the DCBOFLGS field in the foundation segment before OPEN.
			FOUNDATION SEGMENT--BEFORE OPEN (Cont'd)
49 (31)	. 1	DCBIFLG	Used by I/O supervisor in communicating error conditions and in determining corrective procedures.
		00..	Not in error procedure.
		01..	Error correction in process.
		11..	Permanent error condition.
		..10	Channel 9 printer carriage tape punch sensed.
		..01	Channel 12 printer carriage tape punch sensed.
		.... 00..	Always use I/O supervisor error routine.
		.... 11..	Never use I/O supervisor error routine.
		.... 01..	Never use I/O supervisor error routine.
		.... 10..	Never use I/O supervisor error routine.
		xx	(Reserved bits)
50 (32)	. . 2	DCBMACR	Macro instruction reference. Major macro instructions and various options associated with them. Used by the open routine to determine access method. Used by the access method executors in conjunction with other parameters to determine which load modules are required.

Offset	Bytes and Addresses	Field Name	Field Description, Contents, Meaning
			EXCP ACCESS METHOD
			Code
		Byte 1	
50 (32)		1...	Execute channel program (EXCP).
		.1..	Foundation extension is present with EXCP.
		..1.	Appendages are required with EXCP.
		...1	Common interface is present with EXCP.
		.... .1..	User's program maintains accurate block count.
		.... x.xx	(Reserved bits)
		Byte 2	
51 (33)		xxxx	(Reserved bits)
		.... 1...	5-word device interface is present with EXCP.
		.... .1..	4-word device interface is present with EXCP.
		.... ..1.	3-word device interface is present with EXCP.
		1	1-word device interface is present with EXCP.
		Byte 1	BSAM - Input
50 (32)		00..	Always zero for BSAM.
		..1. .... R	READ
		...x x..x	(Reserved bits)
		.... .1.. P	POINT (which implies NOTE).
		.... ..1. C	CNTRL
		Byte 2	BSAM - Output
51 (33)		00..	Always zero for BSAM.
		..1. .... W	WRITE
		.... 1... L	Load mode BSAM (create BDAM data set).
		.... .1.. P	POINT (which implies NOTE).
		.... ..1. C	CNTRL
		1	BSAM create BDAM processing of unblocked spanned records, with BFTEK=R specified: The user's program has provided a segment work area pool and stored the address of the segment work area control block in DCBEOBW (offset 76).
		...x	Reserved.

bkup

AM3296

AM9917 Gen05. Backup4. Set105

247-0561