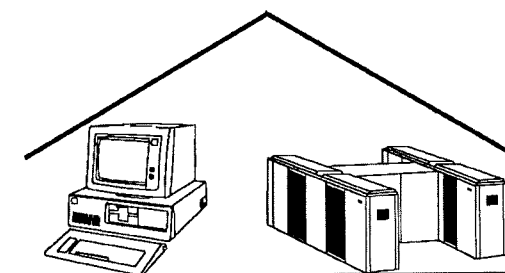


VOLVO

UTGIVARE: Volvo Data AB
2500 ADB-Teknik



TEKNIK - basen

TEKNIK - basen / Mars - 1992

Hej igen! Nytt år och nytt nummer av TEKNIKbasen. Vi är nu inne på årgång 6 (!) om vi tillåter oss att räkna med föregångaren DATABasen. Tiden går sannerligen fort.

I detta nummer finns lite av varje. Projektinformation av olika slag blandad med lite mer detaljerade upplysningar om den nya datalagringsstandarden i stordatormiljön.

För 1992 planerar vi följande ytterligare utgivningstillfällen:

W225, W239, W250

Manusstopp för material att publicera är veckan innan varje utgivning. Men för ett bra bidrag kan vi alltid ordna förlängt öppethållande några dagar, vi är inte omöjliga på det viset. Skicka ett memo till memoid **VD.TEKNIK** med Ditt bidrag så tar vi in det i nästa nummer.

TEKNIK - basen distribueras f.n. till de för oss kända teknikkontakterna. Är det nå-

gon vi missat, kontakta i första hand Din lokala PI-administratör som ordnar saken. Alternativt kan Du sända ett memo till **VD.TEKNIK** så hjälper vi till.

För formens skull påpekas till sist att självklart står varje artikelförfattare för respektive artikels innehåll, vi på 'redaktionen' utövar ingen censur eller kontroll, möjligen lite redigering av insänt material.

Med vänlig hälsning

Henrik Holst / 2502 Computer Technology Development

Nilsson, Thomas
AVD 30652
HC2N

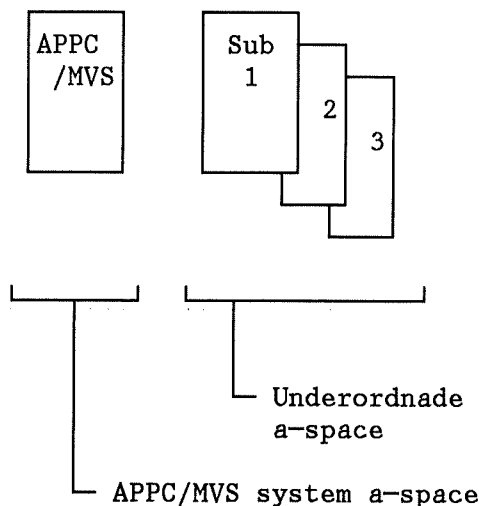
APPC/MVS, nu börjar det

Den minnesgode kanske kommer ihåg en blänkare i TEKNIKbasen från juni -91 som handlade om TSO/E och APPC/MVS. Vi fortsätter här med nästa avsnitt i ämnet. APPC/MVS är namnet på en ny funktion inom MVS, man skulle nästan kunna kalla det en ny miljö. Jag skall här beskriva lite närmare vad APPC/MVS är, och vilka aktiviteter som f.n. finns inom teknik/prod-divisionen att arbeta vidare med APPC/MVS.

Vad är APPC/MVS? En något förenklad beskrivning är att APPC/MVS är implementationen av program-till-program kommunikation i stordatormiljön. Eller implementationen av kommunikationsprotokollet LU6.2 i stordatormiljön. Båda är korrekta beskrivningar, men säger bara en del av sanningen.

I och med APPC/MVS introduceras ett nytt sätt att få program körda. Förutom de traditionella typerna av address-space Batch, TSO/E och STC¹ tillkommer ett nytt sätt att få program körda.

Den nya omgivningen kan beskrivas så här:



APPC/MVS håller sig med ett antal underordnade address-space (eng. subordinate) som är reserverade för APPC/MVS applikations program.

Dessa a-space kan vara av två typer. I a-space av **standard**-typ körs program som initieras på nytt för varje exekvering. Dvs de resurser (tex filer) som behövs allokeras i samband med programexekveringen, antingen innan programstart eller dynamiskt under exekveringen. Efteråt friställs resurserna igen. På så sätt får varje nytt program en ren och fräsch omgivning. Det a-space som använts finns dock kvar, det avslutas inte. Denna variant skall nog användas med försiktighet då det finns krav på svarstider, eftersom den ofrånkomligen innebär extra system overhead (resursförbrukning). Jämför med traditionell batch.

Den andra typen av a-space kallas **multi-trans**. Här kan ett program ligga aktivt under längre eller kortare tid och behandla de transaktioner som APPC/MVS sänder över i den takt de anländer. Efter varje programexekvering återgår programmet till vänteläge i avvakten på nästa transaktion. Under tiden kan resurser finnas permanent allokerade. Denna variant skall närmast jämföras med traditionell online.

Den som tycker sig känna igen IMS sätt att arbeta i detta är inte helt fel ute, det finns vissa likheter mellan APPC/MVS och IMS kontrollregion och mellan de underordnade a-spacen och MSG-regionerna i IMS. Men mycket längre skall nog inte parallellerna dras.

Program i de underordnade a-spacen kan göra allt ett konventionellt program kan. Alla typer av data, både vanliga filer (VSAM, QSAM m.m.) och databaser (DB2 och DL/1) kan nås, må vara på lite olika sätt beroende på typ.

Man kommunicerar med APPC/MVS med hjälp av kommunikationsprotokollet LU6.2, dvs program-till-program kommunikation. Det spelar ingen roll om 'programmet i andra änden' befinner sig i samma maskin eller i en maskin på andra sidan jordklotet.

¹ STC ovan betyder Started Task. Dessa används av systemfunktioner i MVS och diverse subsystem. Tex körs IMS online delvis som STC. Ett annat exempel är MEMO.

Vad kommer nu att hända med APPC/MVS på VD? F.n. finns det många frågor kring APPC/MVS. Vad skall det användas till? Vilka funktioner är viktiga att känna till? Hur kommer APPC/MVS att kunna förbättra nuvarande applikationsmiljö? osv osv.

Som ett första avstamp för att försöka bena ut dessa och andra frågor har ett PU-projekt startats inom teknik/prod-divisionen. Målen för projektet är att:

- etablera nödvändiga standards för att effektivt kunna nyttja APPC/MVS. Avser främst standards för definitioner och specifikationer.
- kartlägga behovet av administration och underhåll vad gäller nätsidan av APPC/MVS
- gå igenom säkerhets- och behörighetsaspekterna av APPC/MVS. Hur är detta implementerat i APPC/MVS?

- positionera APPC/MVS med avseende på organisatoriska frågor. Produkten går lite 'på tvären' mot etablerade strukturer. Den innefattar såväl kommunikation som basteknik och berör även systemutveckling.

Den uppfattningen vi har, och den delas av de flesta bedömare, är att APPC/MVS kommer att spela en mycket viktig roll i processen att placera stordatorn (MVS) som en viktig server i den Client/Server-miljö som är på väg att växa fram.

Projektet kommer att avslutas med ett seminarium, där vi kommer att belysa mer ingående APPC/MVS och hur produkten passar in i Volvo miljön.

Deltagarna i projektet kommer från 2200, 2500 och 2800 på Volvo Data just med tanke på produktens bredd. För ytterligare information om projektet kontakta undertecknad.

/Henrik Holst (VD.HH)

Vad händer inom utdata?

Pågående aktiviteter:

- Vi tittar på en arkitektur som Volvo Data kan rekommendera inom området och som syftar till att förenkla utprintning mellan de olika datorplattformar vi idag driver. Vår målsättning är att ha ut förslaget på remiss innan mars månad slut.
- Det pågår ett PU-projekt för att ge förslag till PDL-standard (Page Description Language) som bör stödjas av Volvo Data. De som ligger bäst till är IBM-IPDS, PostScript och HP-PCL. Inom dessa områden letar vi nu efter bryggor som skall klara av att ta oss mellan de olika PDL-erna. Vi har en installerad mellan PostScript och IBM-IPDS.
- Det finns en färdig font-header för mappning av PostScript font till ISO8859-1 (ECMA-94 Latin 1) som är koncernstandard.
- Utdata (avd 2130) har sedan en tid tillbaka möjligheter att interaktivt framställa printerresurser typ fonter, bilder och blanketer. De format som klaras av är IBM-AFP, Xerox, HP och PostScript när det gäller blanketter och fonter, när det gäller bilder klarar de IBM-pageseg, Xerox-img, PCX och TIFF.
- IBM-AFP utskrift på en AS/400 håller på att tas i drift för PV.
- De som har laser skrivare genade i VPS som HPI (HP-PCL skrivare med en IDA-PCU) eller HPE (HP-PCL skrivare med en J-PCU) kan nu utnyttja forms 2000 för liggande A4 och forms 3000 som stående A4 där kommandosträngarna för detta fås via VPS.

Jag hoppas att på detta sätt kunna hålla Er uppdaterade på vad Volvo Data håller på med inom området utdata. Om ni har

Konvertering av dataset (databaser) på TP-poolen

Bakgrund

Under de senaste åren har det kontinuerligt pågått en aktivitet att konvertera dataset till SMS (System Managed Storage). SMS ger möjligheter att administrera data på ett mera kostnadseffektivt sätt och ger dessutom större möjligheter att ge rätt service till varje enskild grupp av dataset. Det är också IBM's strategiska plattform inom MVS för områdena data-lagring och data-access.

Dokumentation

Allokeringar i en SMS-miljö beskrivs i "Storage Management Document" som kan läsas och skrivas ut via VILMA-funktionen SDSMHB. Språket i dokumentet beror på språkkoden i användar-profilen. Svenska och engelska finns.

Storage Class (SC) beskriver huvudtypen av dataset. Det ligger till grund för svarstids-service, separering, etc.

Management Class (MC) beskriver hur datasetet skall behandlas efter användning. Tillsammans med vissa andra parametrar, typ dsorg, ligger detta till grund för backup, migrering, idle space release etc.

Dags för databaser (DL1, etc. Ej DB2)

Nu har turen kommit till de dataset (databaser) som allokerats via UNIT=TPTEST samt VOL=SER=TP0... och motsvarande termer via IDCAMS DEFINE CLUSTER etc..

Före sommaren 1992 kommer dessa att konverteras till SMS via "convert in place", dvs. konvertering av resp. volym utan att flytta dataseten. Därefter gäller nedanstående mönster (**Figur 1**) för nyallokeringar:

Unit	Volser	översätts till	Storage Class	Management Class
-----	-----	====>	-----	-----
TPTEST			TP0T	TP0T
	TP0...		TP0	TP0
Efter konvertering kan också följande SMS-klasser användas:				
			TP0A	TP0
			TP0B	TP0

Figur 1. : Mönster för nyallokeringar

Databaser med SC TP0T, dvs. testbaser lagras tillsammans med övrigt "icke specialbehandlat data" (SC=STANDARD).

Databaser med Storage Class TP0, dvs. prod-baser lagras på för det speciella ändamålet lämpliga volymsgrupper.

Vi vet att många databas-administratörer har investerat en hel del tid i att separera en eller flera databaser till från varandra skilda volymer i avsikt att eliminera arm-konflikter. Via "convert in place" kvarstår separeringen och investerade manresurser bevaras initialt.

Dataseten kommer vid konverteringen att erhålla SC TP0 och MC TP0. SMS rutinerna är initialt inställda så att volym-sallokeringar hanteras såsom före konverteringen, dvs. allokeringen utförs på den volym som begärs.

Efter konverteringen, vilken kommer att meddelas i förtid via VICS-NEWS, kan man allokera nya databaser med SC TP0. SMS kommer då att välja lämplig volym av samtliga i TP-poolen ingående volymer. MC TP0 erhålles default för SC TP0. Dock råder full kompatibilitet, enligt ovan, med gammal allokering via ex.vis VOL=SER=TP.....

TP0-volymererna delas in i två grupper. Detta för att man efter konverteringen skall kunna separera konfliktande databaser via SMS attributet SC. Via SC TP0A allokeras databasen på den ena halvan (A-halvan) av TP-poolen och via SC TP0B allokeras databasen på den andra halvan

(B-halvan). Dessa två typer av dataset kommer alltid att hållas åtskilda från varandra. MC TP0 erhålles default för båda.

En viss tid efter konverteringen kommer SMS-reglerna att ändras så att ingen hänsyn tas till gammal volser allokering så till vida att en allokering via VOL=SER=TP0... kommer att hamna på vilken som helst (vald av SMS) av i TP-poolen ingående volymer. Innan dess skall resp allokeringsinformation i nödvändiga fall ha ändrats av resp. databasansvarig för att utnyttja de nya SMS-attributen SC TP0A/TP0B.

Om ytterligare separering önskas, då två typer inte räcker, skall separering via DD-definitioner användas. I en SMS miljö utförs en nyallokering, om möjligt, aldrig på en volym som redan är allokerad till steget i fråga, oavsett om de andra allokeringarna är OLD/SHR/MOD/NEW. Exempel följer:

```
//STEP1      EXEC PGM=...
//DD1        DD DSN=DATABAS1,DISP=(NEW,CATLG,...),
//            STORCLAS=TP0A,...
//DD2        DD DSN=DATABAS2,DISP=(NEW,CATLG,...),
//            STORCLAS=TP0B,...
```

Figur 2. Exempel 1: Separera DATABAS1 från DATABAS2 via JCL.

```
//STEP1      EXEC PGM=IDCAMS
//SYSIN      DD *
  DEFINE CLUSTER (NAME      (DATABAS1)          -
                  STORCLAS (TP0A)                -
                  ...                             )
  DEFINE CLUSTER (NAME      (DATABAS2)          -
                  STORCLAS (TP0B)                -
                  ...                             )
  DEFINE CLUSTER (NAME      (DATABAS3)          -
                  STORCLAS (TP0B)                -
                  ...                             )
//...
```

Figur 3. Exempel 2: Separera DATABAS1 från DATABAS2 (viktigast) samt separera dessa från DATABAS3 via IDCAMS.

```

//STEP1      EXEC PGM=...
//DD1        DD DSN=DATAS1,DISP=(NEW,CATLG,...),
//           STORCLAS=TP0A,...
//*
//STEP2      EXEC ....
//DD1        DD DSN=DATAS2,DISP=(NEW,CATLG,...),
//           STORCLAS=TP0B,...
//*
//STEP3      EXEC PGM=IDCAMS
//DD1        DD DSN=DATAS2,DISP=SHR
//SYSIN      DD *
              DEFINE CLUSTER (NAME      (DATAS3)          -
                             STORCLAS (TP0B)            -
                             ...                          )
//...

```

Figur 4. Exempel 3: Samma separering som exempel 2, men i tre steg.

Nya SMS-klasser

Följande nya klasser kommer att läggas in i "Storage Management Document":

Storage Class

- TP0** Traditionell DL1 databas och liknande. Central dag för dag performance-bevakning på data-sets- nivå och åtgärd vid behov efter kontakt med databasansvarig.
- TP0A** Samma som ovan, fast separerad från TP0B
- TP0B** Samma som ovan, fast separerad från TP0A
- TP0T** Samma som ovan, fast testdata. Lagras bland dataset med SC STANDARD.

Management Class

- TP0** Default för SC TP0/TP0A/TP0B. Migrering tidigast 400 dagar efter senaste open. Ingen idle space release, ingen backup.
- TP0T** Default för SC TP0T. Migrering tidigast 30 dagar efter senaste open. Ingen idle space release, ingen backup.

DB2

Motsvarande konvertering för DB2's tablespace är under utvärdering.

Nuläge

Den uppmärksamme iakttagaren kan knappast ha undgått att se att nyallokeringar av dataset redan nu konverteras direkt vid nyskapandet enligt följande mönster:

Unit	Vol- prefix	översätts till	Storage Class	Management Class
PP		=====>	STANDARD	NOBACKUP
	PP....		STANDARD	NOBACKUP
PLP			STANDARD	BACKUP7
	PLP...		STANDARD	BACKUP7
PLB			STANDARD	BACKUP1
	PLB...		STANDARD	BACKUP1
MD			STANDARD	DEL2
	MD....		STANDARD	DEL2
TS			STANDARD	DEL2
	TS....		STANDARD	DEL2

Figur 5. : Konvertering av gamla beteckningar

Även gammalt online-data har konverterats till motsvarande SMS-attribut. Gammalt migrerat data konverteras automatiskt när det återanvänds (recall).

Volvo Datas strategi vid konvertering till SMS är att kunden/användaren inte skall behöva ändra existerande JCL eller annan styrinformation i samband med konverteringen. Däremot skall SMS-attributen SC och MC användas för att erhålla specific service som inte erbjudits tidigare för non-SMS-dataset via de gamla termerna UNIT och VOL.

Ny JCL och andra definitioner om nyallokeringar av dataset kan skrivas med de nya attributen SC och MC så fort dessa är tillgängliga i resp dator-miljö.

Vidare upplysningar

Vis frågor angående ovanstående förändringar kontakta i första hand resp no-

deansvarig, avdelning 2120 på Volvo Data.

VD/VT Lars Hammar,
tel 67052,
memo VD.HAMMAR

V1/V5/V6 Sven-Olof Knapasjö,
tel 67481,
memo VD.SOK

V2 Börje Joakimsson,
tel 62208,
memo VD.BÖRJE

V4/VK Birgitta Elmäng,
tel 67205,
memo VD.ELMÄNG

BM Hasse Wicktorin,
tel 69639,
memo VD.HW

eller undertecknad
Lars Danielsson, avd 2510, tel 67479, me-
moid VD.LD

Slutsummering av projekt IMS31

Detta är en kortfattad summering av projekt IMS31, dvs byte av IMS-version från 2.2 till 3.1, med avsikt att belysa problem som uppstod, de resultat projektet uppnått och fortsatta aktiviteter som följd av detta och den nya IMS-versionen.

Rapporten är sålunda av mindre intresse för de som inte sysslar med IMS-orienterat arbete. En mer fullständig rapport har distribuerats till IMS-kontaktpersoner på de olika kundbolagen.

Uppgift

Projektet avsåg migrering från IMS 2.2 till version 3.1 i samtliga IMS-system som underhålls av Volvo Data/Gbg. Samtidigt infördes ny namnstandard (3-delade namn) på samtliga datamängder i de migrerade IMS-systemen.

Genomförande

Projektet genomfördes under hösten-91 tom februari-92, dvs med ca 2 månaders fördröjning mot ursprunglig plan. Installation återstår idag för CICS-system, VME och VEDA.

Projektet har förbrukat ca 1.880 timmar (47 manveckor), vilket är ca 70% över projektets ursprungliga bedömning. Detta beror mycket på det samtidiga namnbytet, men vi kan åter konstatera att ett IMS versionsbyte fortfarande är en 'manintensiv' uppgift på Volvo Data.

Själva projektet bemannades med enbart Volvo Data-personal, men kontaktpersoner på respektive kundbolag har också utfört mycket viktiga uppgifter i projektet.

Problem

De problem av större betydelse som inträffade under installation, uttestning och införande kan summeras under följande gruppering:

- ej full IMS 3.1-support i vissa subprodukter (ex. DATAPACKER/IMS, DL/1 BRIDGE, HSSR)
- vissa subprogram ej anpassade till 31-bitarsadressering (bl.a vissa viktiga randommoduler)
- felstatuskoder vid vissa DL/1-anrop i program (AR vid 4 parametrar i

CHNG-anrop, A4 vid ISRT mot alternativ-PCB)

Resultat av ny version

Efter övergång har IMS 3.1 visat stor pålitlighet och bidragit till bibehållen hög tillgänglighet i IMS-systemen.

De 'trögningar' som ibland förekommer i online-verksamheten beror inte på IMS internt utan har 'yttre' orsaker som:

- tungt lastad maskin (CPU, I/O)
- 'känsliga' BMP-jobb som påverkar online
- låsningar i DB2

Dessa situationer är ofta svåra att kartlägga, vi jobbar på att hitta en enhetlig 'metodik' för akut svarstidsanalys.

Inga direkta klara förbättringar i fråga om resursförbrukning och svarstider har kunnat ses, men bra möjligheter finns nu, genom t.ex. utökning av arbetsbuffrar och pooler.

Flera förändringar har också gett klara positiva effekter, t.ex. mindre minnesutnyttjande (CSA) och reducerad loggning.

Ur applikationssynpunkt finns en del nya funktioner, mest intressant är kanske IMS Spool API, direktutskrift mot JES2. Dessa testas nu som 'följdaktiviteter' till projektet, och dokumenteras sedan i IMS metodhandbok.

Erfarenheter i projektet

Inom flera områden kommer ansträngningar att göras för att uppnå förbättringar inför nya versionsbyten:

- bättre kartläggning av support i subprodukter
- utökade och förbättrade funktions-tester, kompletterade med kundapplikationstester, i test-maskinen VT01
- en mer strömlinjeformad process av tillverkning (i VT01), distribution och installation ut i respektive miljö
- utökad 'kundtest' efter övergång

Eftersom all produktion och SU i framtiden körs i samma MVS, måste vi etablera en helt ny testmetodik med all uttesning i test-maskinen VT01 och samtidig förändring i prod och SU.

Följdaktiviteter

Som följd av projektet har alltså ett antal följdaktiviteter definierats, t.ex.

- installation i återstående miljöer (CICS, VME, VEDA)
- införande av ny testmetodik
- förbättrad IMS performance (online och batch)
- utvärdering och utnyttjande av nya applikationsfunktioner

Dessa och andra IMS-aktiviteter på 2540 utan direkt anknäytning till IMS 3.1 kommer förhoppningsvis att ge ett allt bättre utnyttjande av produkten.

Slutsatser

Projektets genomförande måste betraktas som lyckat, trots vissa problem efter övergångsförsök som i två fall också framtvängde fallback.

För att till fullo utnyttja möjligheterna i ny IMS-version finns dock mycket att göra såväl centralt (driftmässigt) som i kundapplikationer. Låt oss jobba vidare ! ...

/Perry Lundqvist

V1-sammanslagningen, två maskiner blir en

Nedan följer en sammanfattning av den projektrapport som producerats efter projektet V1-sammanslagningen. För ytterligare detaljer och information kontakta undertecknad.

Bakgrund och förutsättningar: Sedan ett antal år har systemutveckling (SU) och produktion (PROD) varit separerade i skilda maskinmiljöer inom MVS-världen. Undantag är V4-miljön (Volvo Lastvagnar) som är en integrerad miljö sedan ett drygt år tillbaka.

Emellertid har beräkningar visat att kostnaden för separata maskinmiljöer i dagens läge är större än de intäkter (i form av fördelar) detta ger. Som en följd av detta utarbetades direktiv att slå samman SU och PROD i både V1 och V2 miljöerna.

Detta projekt har haft som mål att integrera SU och PROD inom V1-miljön. Den omfattar följande kundbolag:

- Volvo Truck Parts AB

- Volvo Penta AB
- Volvo Svenska Bil AB
- Volvohandelns Affärssystem AB
- VOLVIA

Den omedelbara vinsten med att slå samman SU och PROD har varit möjligheten att friställa en logisk maskin (LPAR)², vilket innebär möjlighet att effektivare utnyttja installerad hårdvara. Det går åt mycket extra resurser för varje logisk maskin, främst i form av diskutrymme.

I förutsättningarna ingick att kvalitét avseende tillgänglighet och svarstider inte fick försämrats. Ej heller fick integriteten på databaser och filer äventyras genom frånvaron av det extra skydd som skilda maskiner ger.

Sammanfattning: Projektet startades i aug.-91 och avslutades i feb.-92, då sammanslagningen genomfördes helgen W208. Totalt lades ner 23 mv inom Volvo Data.

² Möjligheten att dela en fysisk maskin i flera s.k. logiska har funnits ett antal år inom IBM stordatorer. Funktionen som nyttjas kallas PR/SM. Varje logisk maskin är att betrakta som en helt separat maskin ur användningssynpunkt.

Organisation: Projektet var uppdelat i följande delaktiviteter.

Volvo Data aktiviteter

- DB (IMS och DB2)
- VICS/IC
- MVS/JES2/sys
- SU
- Nät

Från kundbolagen

- PARTS
- Penta
- Volvo Sv. Bil
- VOLVIA

Genomförande: Själva överläggningen genomfördes utan några allvarliga störningar. Några mindre saker noterades, se nedan under erfarenheter. Allmänt kan sägas att själva överläggningen var ganska enkel såtillvida att några egentliga dataflyttar aldrig var aktuella. All data fanns på plats redan innan sammanslagningen. De saker som behövde göras i samband med överläggningen inskränkte sig till:

- ompekning av resurser (tex NJERST), från V102-maskinen (gamla SU-maskinen) till V101 (den nya gemensamma miljön)
- justering av parametrar i VICS-miljön. Gjordes för alla VICS-användare i V1.
- diverse anpassningar av systemdata

Erfarenheter: Nedan följer några av de viktigaste erfarenheterna och iakttagelserna samt hur problemen löstes eller undveks.

Problem: Att undvika att oavsiktlig uppdatering av produktionsdata, framför allt från VICS.

Lösning: Först och främst ett grundläggande ACF2-skydd av produktionsdata. Allt produktionsdata skall vara oåtkomligt såvida inte tillräcklig auktoritet tilldelats. Ett kompletterande sätt att skydda produktionsdata utvecklades, Bolagsspecifik EDIT-macro vid SUBMIT. För närmare

beskrivning av denna, kontakta VICS produktansvarig på avdelning 2520.

Problem: Hur skall kvalitén på systemmjukvaran kunna garanteras även efter en sammanslagning av SU och PROD? Nya releaser av operativsystem och subsystem installeras med jämna mellanrum, med de risker för störningar det innebär.

Lösning: I Nuvarande teknikmiljö, VT01, kommer förutom de normala tekniktестerna utökade tester av systemmjukvaran att genomföras. Dessa tester bygger på att utvalda (kompleta) applikationssystem tas över i VT01 från resp. produktionsmiljö. Kritiska och känsliga applikationer skall väljas i första hand.

Efter det att nya versioner/releaser av operativsystem eller subsystem installerats i teknikmiljön VT01 skall de utvalda applikationerna alltså först flyttas över till VT01 från V101. Sedan skall dessa köras igenom och testas med godkänt resultat. Först därefter kan den förändrade systemmjukvaran installeras i den kombinerade SU- och PROD-miljön. Orsaken till detta förfaringssätt utvecklas lite ytterligare i avsnittet förändringar nedan.

Förändringar

Systemförändringar: Den viktigaste förändringen som introducerades i och med detta projekt är det sätt på vilket subsystemens laddmodulbibliotek är organiserade. I den nya miljön finns endast en version av programkoden för subsystemen. Detta innebär att subsystemen (tex. DB2) i både SU och PROD bygger på samma programmoduler. En programändring i tex DB2 slår alltså igenom direkt både i SU och PROD.

En annan förändring är det sätt på vilket en användare loggar på VICS i den nya miljön. Eftersom det bara finns ett TSO-system i maskinen används skilda LOGON-procedurer för att isolera SU respektive PROD från varandra.

Organisation och ansvar: Sammanslagningen av PROD och SU har inte resulterat i några organisatoriska eller ansvarsmässiga förändringar.

/Henrik Holst

Innehåll

APPC/MVS, nu börjar det	2
Vad händer inom utdata?	3
Konvertering av dataset (databaser) på TP-poolen	4
Bakgrund	4
Dokumentation	4
Dags för databaser (DL1, etc. Ej DB2)	4
Nya SMS-klasser	6
DB2	6
Nuläge	6
Vidare upplysningar	7
Slutsummering av projekt IMS31	8
Resultat av ny version	8
Erfarenheter i projektet	8
Följdaktiviteter	9
Slutsatser	9
V1-sammanslagningen, två maskiner blir en	9

