Generelt notat vedr. Recovery-projektet

Oversigt over principper, metoder, problemer og resultater omkring konverteringen af Rigsarkivets bestand af edb-arkivalier[1] på magnetbånd til CD-R medie.

Udarbejdet af: Dan Tørning, BK-afdelingen
Dato: 19.7.2006

1. Indledning.

I årene 2001-2003 gennemførtes en konvertering af Rigsarkivets samlede bestand af magnetbånd til CD-R-medie, det såkaldte mediekonverteringsprojekt. En mindre del af bestanden af magnetbånd viste sig herved at volde problemer, idet det ikke umiddelbart lod sig gøre at foretage fejlfri konvertering.
Hensigten med det foreliggende notat er at dokumentere det projekt, som gennemførtes i årene 2004-2006 under navnet 'Recovery-projektet'[2], og som i korthed bestod i at forsøge at genskabe data fra disse magnetbånd. Notatet skal udgøre en generel beskrivelse af de faglige overvejelser og tekniske metoder, som blev bragt i spil. Foruden dette notat vil der for hvert arkivalie, som blev berørt af 'recovery', blive udarbejdet et notat til dokumentation af de specifikke problemer vedrørende pågældende arkivalie. På denne måde sikres det, at fornødne oplysninger om konverteringen er tilgængelige ved den fortsatte tekniske vedligeholdelse af arkivalierne.

2. Baggrund.

Siden begyndelsen af 1970-erne har Rigsarkivet modtaget afleveringer af edb-arkivalier.
De nærmere principper for aflevering blev defineret i 1973 efter oprettelsen af edb-sektionen. Det bestemtes her at edb-arkivalier skal afleveres på magnetbånd, nærmere betegnet 9-spors spolebånd med datatæthed 1600 bpi eller 6250 bpi (bpi= bytes pr. inch)[3]. Der blev afleveret to eksemplarer af båndene, således at det ene eksemplar kunne udgøre en sikkerhedskopi. Det ene eksemplar (RA-bånd) opbevaredes i Rigsarkivet, det andet på Landsarkivet for Sjælland (LAK-bånd). Alle bånd blev testlæst efter modtagelse. I årene fra 1973 til1986 foretoges testlæsningen ved de regionale edb-centre RECKU eller NEUCC, men fra 1987 på Rigsarkivets eget edb-anlæg. I 1995 gennemførtes en større undersøgelse af båndenes bevaringstilstand, hvor 20% af båndene blev testlæst. Der konstateredes ved denne undersøgelse ganske få båndfejl (størrelsesordnen 3 % af båndene), og data kunne i disse tilfælde uden vanskelighed tages fra kopieksemplaret.
Imidlertid var det klart allerede i 1995, at spolebånd nu var et forældet datamedie. Derfor indførtes i 1998 krav om at aflevering skulle foretages på CD-R, og efterfølgende besluttedes det at gennemføre den i indledningen omtalte konvertering af hele båndbestanden til CD-R.
At dette også var velbegrundet ud fra et holdbarhedskriterium blev kort tid efter konstateret i forbindelse med sag om udlevering af data, idet der her viste sig problemer med læsning af adskillige bånd.

---

[1] I dette notat anvendes betegnelsen edb-arkivalier og ikke den nu mere almindelige: IT-arkivalier, da notatet omhandler perioden 1970 – 1998, hvor 'edb-' endnu ikke var erstattet af 'IT-'

[2] Dette noget udanske navn anvendes om projektet. Der hvor der er brug for et verbum, bruges i notatet derimod ikke ordet recover men i stedet danske ord som retablere eller genskabe.

[3] Fra 1975 modtoges kun 6250-bpi-bånd og bestanden af 1600 bpi konverteredes til 6250 bpi.

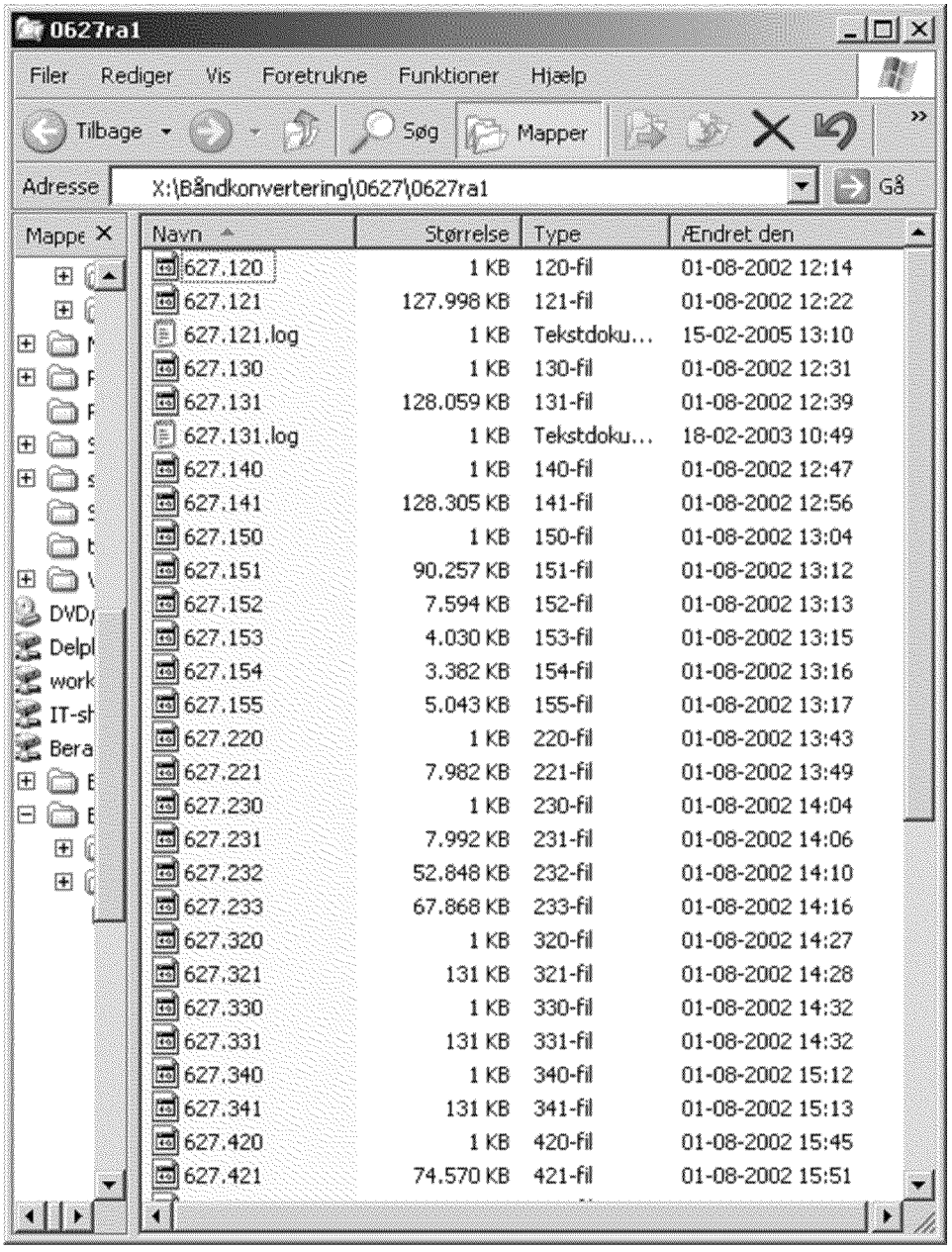
```
VarRecNy-test af filen: 462

******** Her kommer test på blokniveau ********

For stor eller for lille bloklængde:
Bloknr   6158 Bloklængde:  16448 startbyte:        92750749
Forrige blokstart=     92735736
ny blokstart fundet byte nr.  92742176  Bloklængde=     15124

For stor eller for lille bloklængde:
Bloknr   6486 Bloklængde:  16448 startbyte:        97684017
Forrige blokstart=     97668990
ny blokstart fundet byte nr.  97685366  Bloklængde=     15101

For stor eller for lille bloklængde:
Bloknr   6657 Bloklængde:  27619 startbyte:       100261742
Forrige blokstart=    100246598
ny blokstart fundet byte nr. 100257113  Bloklængde=     15068

filslut efter bloknr   7108
slutbytenr:       107067495

******** Her kommer test på record niveau ********

filslut efter bloknr   7108
slutbytenr:       107067495

Antal blokfejl:    3

 bloknr        startbyte         slutbyte      antal bytes

   6157         92735737         92742175             6439
   6485         97668991         97685365            16375
   6656        100246599        100257112            10514

Kopieres til rettet fil:

    fra byte         til byte      antal bytes      bytes i alt
           1         92735736         92735736         92735736
    92742176         97668990          4926815         97662551
    97685366        100246598          2561233        100223784
   100257113        107067495          6810383        107034167
```

Antal bytes læst: 107067495
Antal bytes skrevet: 107034167

Slut på VarRecNy

Bilag 3. Eksempel på udlæsning af fragmenter af fil

I nedenstående eksempel ses resultatet af forsøg på udlæsning af data fra en multi volume fil.
Navngivningen af fragmenterne forstås således:
Fragmentet 627.231 omhandler fil nr. 627, bånd nr. 2, læsningsforsøg nr. 3, fragment nr. 1.
Fragment nr. 0 angiver i altid filens header label, som jo er en separat fil og ikke et fragment af selve filen.
De to log-filer er output fra TestVar programmet, som viste antallet af blokke i fragmentet.



Bilag 4: Oversigt over slutresultat af Recovery-projektet: Fejlramte filer

Filnr	Arkivskaber	Systemnavn	Filnavn	Fejlbånd	Kommentar
97	Undervisningsministeriet	Skole- og klassestatistik 1967/68	Skole og klasse 1967/68	1	86 records ud af 24885 mangler.
104	Undervisningsministeriet	Elevstatistik 1972/73	Elev 1972/73	1	ca. 8500 records ud af 320340 mangler
358	Kgl. Grønlandske Handel	Regnskabsopgave 1980	1980	1	128 records ud af 29948 mangler
424	Kgl. Grønlandske Handel	Regnskabsopgave	1984	1	Ingen data bevaret
433	Direktoratet for Arbejdsmar-kedsuddannelserne	Uddannelsesstatistikregister	1977	1	108 records ud af 71415 mangler .
494	Statens Regnskabsdirektorat	SCR	Internt regnskab 1985	1	ca. 700 records ud af 638105 mangler
579	Kgl. Grønlandske Handel	Forsendelsessystem	Modtagelse 1986	1	28 records ud af 127530 mangler
650	Arbejdsdirektoratet	Centrale register over arbejdsmarkedsstatistik	CRAM register 1987	1	10 records ud af 2.615.200 mangler
942	Danmarks Statistik	Pensionsstatistikregister	Status, januar 1986	1	30 records ud af 918611 mangler
1248	Danmarks Statistik	Løn- og personalestatistik offentlig sektor	Statistikreg. Stat 1987	1	122 records ud af 130028 mangler
1253	Danmarks Statistik	Løn- og personalestatistik offentlig sektor	Totalreg. Kommunal 1986	2	Næsten ingen data bevaret
1255	Danmarks Statistik	Løn- og personalestatistik offentlig sektor	Totalreg. Kommunal 1988	2	Næsten ingen data bevaret
1261	Danmarks Statistik	Løn- og personalestatistik offentlig sektor	Statistikreg. Komm. 1988	1	96 records ud af 615728 mangler
1262	Danmarks Statistik	Løn- og personalestatistik offentlig sektor	Statistikreg. Komm. 1989	1	288 records ud af 626917 mangler
1269	Københavns Universitet	Ansøgere til ungdomsudd.	Ansøgningsregister	1	13 records ud af 82350 mangler
Antal fejl-filer i alt: 15					
Antal fejlramte arkiveringsversioner: 14 (se note)					
Antal arkiveringsversioner omfattet af mediekonverteringen: 657					
Antal bånd omfattet af mediekonverteringen: ca. 4000					
Antal fejlramte bånd: 17					
Note: fil 1255 og 1261 hører til samme arkiveringsversion					

3. Konvertering fra bånd til CD-R.

Mediekonverteringen blev foretaget på PC med anvendelse af båndstationer af 'bord-model'-type[4], idet Rigsarkivets hidtidige edb-anlæg med dets båndstationer (af 'main frame' type, lodret monteret) var blevet afviklet i 1996. Der skal ikke her redegøres mediekonverteringsprojektet som sådan, men enkelte tekniske forhold vedrørende konvertering fra bånd til CD-R beskrives i det følgende, da de har betydning for Recovery-projektet.

Magnetbånd er et sekventielt medie. Det betyder, at data kun kan læses 'fra en ende af'. Man kan ikke starte læsning et vilkårligt sted i en datafil. Data ligger i såkaldte 'blokke' som kun kan skrives/læses som en helhed, adskilt af blokmellemrum, som anvendes af båndstationen til start/stop-positionering. Den enkelte datafil består så af et større eller mindre antal blokke.
I *bilag 1* er forklaret nærmere om strukturen af data på magnetbånd, herunder betydningen af hhv. fast og variabelt format samt multi-volume filer.

Ved konverteringen indlæses data først til en fil. Den fysiske opdeling i blokke, som kun er relevant for båndmediet, fjernes ved indlæsningen. I alle de tilfælde, hvor indlæsningen forløb uden problemer, kunne filens samlede dataindhold derefter under ét skrives til CD-R mediet. Kriteriet for at et bånd var læst korrekt var, at antallet af læste blokke var det samme som angivet i båndets label, samt at antallet af records i den læste fil passede med det i dokumentationen angivne antal[5]. Til denne kontrol anvendtes programmerne TestVar og Tapedump (kort omtalt i *bilag 2*). I de tilfælde hvor der var problemer med læsning af et bånd, blev filen forsøgt læst fra kopibåndet, som regel med succes.

Efter gennemførelsen af mediekonvertering af ca. 4000 bånd, stod man med en rest på 203 bånd (212 filer), som ikke kunne udlæses fejlfrit, eller hvor fejlen ikke kunne genoprettes fra kopibåndet, idet begge bånd var fejlramt.

4. Recovery af fejlramte bånd/filer

Det besluttedes at lade firmaet HAYES i Norge (Stavanger) forsøge at læse de fejlramte bånd. Resultatet fra HAYES forelå i efteråret 2003 og en nærmere analyse viste, at knapt 30 % af de fremsendte bånd/filer var læst med succes. I resten af tilfældene var der læst en større eller mindre del af båndet, altid som ét fragment pr. fil.
Fra RA's mediekonvertering forelå et stort antal fil-fragmenter fra de mislykkede læsninger af problemfilerne. I *bilag 3* er vist eksempler. Disse fragmenter er fremkommet ved, at båndstationen ved detektering af en læsefejl har forsøgt at finde næste blokmellemrum og fortsætte læsningen herfra. Hvis dette ikke lader sig gøre, stoppes læsningen. Da det havde vist sig, at man efter rensning af læsehoveder ved et nyt forsøg kunne være heldig at læse en større del af båndet med succes, blev denne fremgangsmåde ofte anvendt, dvs. at der foreligger flere serier af fragmenter af den samme fil. Om disse fragmenter er yderligere at sige, at der ofte forekommer tilfælde, hvor den samme blok er udlæst et antal gange i rækkefølge [6]. For at kunne fjerne disse blokke anvendtes programmet RemoveDuplicateRecords (se *bilag 2*). Det forudsættes her, at alle blokke i den originale fil er forskellige, hvilket man med stor sikkerhed kan gå ud fra[7].

---

[4] Hhv. Hewlet Packard og Overland og Qualstar
[5] Recordantallet gælder hele filen, for multi volume filer altså mere end ét bånd.
[6] Dette må betragtes som en fejl ved båndstationen (og dens software), som antagelig opstod i forbindelse med visse temporære læsefejl (dvs. backspace og ny læsning).
[7] Derimod har der vist sig mange eksempler på, at man ikke kan regne med, at alle records i en fil er forskellige

Det skal også bemærkes, at alt tyder på, at faktisk udlæste blokke i RA- og LAK-fragmenterne er udlæst korrekt[8].

Da man formodede, at det i en hel del tilfælde ville være muligt at sammenstykke data fra RA-fragmenterne enten med andre RA- eller LAK-fragmenter eller med de fragmenter, som var resultat af ligeledes mislykkede HAYES-læsninger, blev det besluttet at udvikle et program til sammenfletning(merging) af filfragmenter, MD5Flet (se *bilag 2*). Filosofien er, at to fragmenter, som hver især indeholder et antal unikke blokke, og som overlapper hinanden, kan sammenflettes til ét fragment. Hvis man er så heldig, at manglende blokke i et fragment konsekvent findes på det andet fragment, vil resultatet af sammenfletningen blive et fragment indeholdende alle blokke fra den oprindelige fil fra første blok på det ene fragment til sidste blok i det ene eller det andet fragment, afhængigt af overlappets karakter.

Denne fremgangsmåde blev så anvendt på de resterende ca. 70% af de fejlramte bånd. Men fremgangsmåden forudsætter, at der arbejdes med fragmenter, som indeholder fejlfri blokke, hvilket som nævnt er tilfældet for RA- og LAK-fragmenterne.
Nu viste det sig imidlertid, at mange af Hayes-fragmenterne indeholdt fejlbehæftede blokke[9], og fragmenterne kunne derfor ikke uden videre anvendes til sammenfletning med andre fragmenter. Det var nødvendigt skære 'syge blokke' bort fra fragmentet. Hertil udarbejdedes et program RemoveBytes (se *bilag 2*), som kopierer specifikke områder i en fil, angivet ved bytenumrene, til en ny fil. Ved hjælp af programmerne TestVar og Tapedump var det ret enkelt at identificere positionen af starten på den første syge blok. Vanskeligere var det at finde positionen af starten af den første blok efter den 'syge blok', idet dette måtte gøres ved en tidskrævende visuel analyse af hexadecimale dumps af et passende stort område. Når det var gjort og den syge blok derefter fjernet ved hjælp af RemoveBytes, kunne en ny test med TestVar/Tapedump vise, om fragmentet nu var fejlfrit, eller om proceduren måtte gentages med fjernelse af den næste syge blok. For filer i variabelt format var det imidlertid muligt at udnytte formatets generelle karakteristika som basis for udarbejdelse af et program, VarRecNy (se *bilag 2*) til automatisk fjernelse af fejl-blokke i ét gennemløb. Hayes-fragmenter blev herved gjort tilgængelige for sammenfletning med RA- og LAK-fragmenter, hvorved det i en række tilfælde lykkedes at retablere den fejlramte fil.

5. Endeligt resultat

Med de ovenstående metoder og redskaber retableredes et anseeligt antal filer. Der resterede herefter ca. 66 bånd (56 filer), som ikke kunne retableres ved hjælp af de eksisterende udlæste fragmenter. Det besluttedes i oktober 2005 at sende 35 bånd til Data Recovery International i Dallas, USA, som i en tidligere sag havde vist sig at råde over exceptionelt effektive hardware og software redskaber for retablering af data fra magnetbånd. Resultatet heraf var, at 29 bånd blev læst fejlfrit, 5 bånd blev læst delvist[10], mens et bånd var ulæseligt.
De resterende fejlbehæftede filer kunne enten betragtes som redundante i forhold til andre bevarede filer, eller fejlene var så små, at de kunne anses for uvæsentlige.

Oversigt over det endelige resultat af Recovery-projektet er vist i *bilag 4*.

---

[8] Dette er sikret ved sædvanligt paritetscheck

[9] Man har åbenbart ikke anvendt paritetscheck, måske for at få så mange data med som muligt uanset fejlene.

[10] I tre tilfælde kunne respektive filer imidlertid retableres fuldt ud v.h.a. de foreliggende fragmenter..

Bilag 1. Datastruktur på magnetbånd

Data på magnetbåndsmedie skrives og læses sekventielt. I den enkelte skrive- eller læseproces overføres en såkaldt blok af data. De enkelte blokke adskilles af et blokmellemrum på 0.76 cm.

En fil på magnetbånd består således af et antal blokke, og efter sidste blok skrives et filemark (End of file, EOF). Næsten alle de bånd, som RA har modtaget anvender herudover såkaldt tape-label, hvilket betyder at hver fil er omgivet af header- og trailerlabels, som hver især er en fil. Disse labels blev brugt af det operativsystem, som producerede pågældende magnetbåndsdata, og de indeholder informationer, som også er af betydning for recovery processen.
Filens header label indeholder angivelse af, om filen er i fast format (Fixed blocked, FB) eller variabelt format (Variable blocked, VB). Desuden angives recordlængde (RL) og bloklængde (BL).
FB-format betyder, at alle blokke indeholder samme antal records, sidste blok dog muligvis færre. Både recordlængden og bloklængden er fast. Eksempel RL/BL= 78/7800, dvs. 100 records pr. blok.
VB-format anvendes, hvor filen indeholder records af varierende længde, hvorved bloklængden også vil variere. I dette format angiver RL og BL ikke de faktiske længder af records og blokke men derimod de maksimalt tilladte længder. Hver blok indeholder en blokheader på 4 bytes. De første to indeholder blokkens faktiske længde (inkl. header), de sidste to er tomme.
På samme måde har alle records en 4-bytes header, hvor de første to bytes indeholder recordens faktiske længde (inkl. header) og de sidste to er tomme. Dette format anvendtes ofte af ældre systemer, hvor hver fil kunne bestå af forskellige recordtyper[11].
Eksempel: En fil indeholder 3 recordtyper med længderne 78, 102 og 279. Filen er skrevet i VB-format med RL=300 og BL= 10000[12]. Når filen skrives, vil det for hver enkelt record blive checket, at bloklængden ikke kommer over 10000. Hvis eksempelvis de hidtidige records i den aktuelle blok fylder 9812 bytes og den næste record, som skal skrives er af længden 279 bytes, så er der ikke plads inden for den maksimale bloklængde. Blokken må derfor lukkes og får længden 9812, og en ny blok påbegyndes med recorden af længde 279.

Trailer label indeholder information om, hvor mange blokke pågældende bånd indeholder. Desuden fortæller trailerlabel, om filen er afsluttet eller om den fortsætter på et andet bånd, jf. nedenfor.

I mange tilfælde indeholder det enkelte magnetbånd adskillige filer. Hver enkelt fil er så omgivet af header og trailer labels og filemarks som forklaret ovenfor. Den sidste fil på båndet er (bør være) afsluttet af 2 filemarks, for at angive at der ikke er flere filer på båndet.

---

[11] Det skal tilføjes, at der intet er til hinder for, at en fil med kun én recordtype kan skrives i VB-format. Selv om alle records således er lige lange, har de en foranstillet header, som angiver længden. Det tilsvarende gælder alle blokkene .

[12] I Rigsarkivets regler for aflevering var det fastsat, at bloklænden ikke måtte være større end 10000. Valget af bloklængde har betydning for den effektive udnyttelse af båndets samlede kapacitet. Jo større bloklængde des bedre udnyttelse af båndet, idet hver blok jo efterfølges af et blokgap på 0.76 inch. På den anden side udgør store bloklængder en risiko for, at flere data kan gå tabt i tilfælde af læseproblemer, idet det normalt vil være sådan, at man enten kan læse en blok eller ikke kan, dvs. hardware og software tillader ikke, at en fejlbehæftet blok videregives til læseprogrammet. I HAYES læsningerne blev fejllæste accepteret og indlæst, jf. hovedtekst.

Omvendt forekommer det også ofte, at en fil er for stor til at kunne rummes på et enkelt bånd. Sådanne filer betegnes multi-volume files. Når slutningen af båndet nås[13], skrives en trailer label, som angiver, at filen ikke er afsluttet men fortsætter på et nyt bånd. I forbindelse med Recovery-projektet er det en vigtig pointe, at bånd af 'samme størrelse'[14] faktisk ikke er præcis lige lange. Det betyder, at tilsvarende bånd i de to eksemplarer af en multi-volume file sjældent er egentlige kopier.
Et konkret eksempel: En fil består af 30621 blokke, som fordeler sig således:

	RA-bånd	Lak-bånd
Bånd1	15302	15231
Bånd2	15166	15203
Bånd3	153	187
I alt	30621	30621

Hvis der nu eksempelvis er problemer med læsning af datablokke i slutningen af RA-bånd1, findes disse datablokke ikke på LAK-bånd1 men skal i givet fald hentes fra LAK-bånd2.

Ved modtagelsen af afleveringer af magnetbånd blev disse testlæst, og informationen fra tape-labels om filformat, record/bloklængder og blokantal blev registreret i Rigsarkivets afleveringsdokumentation. Desuden registreredes oplysning om recordantal, hvis den forelå.

[13] Dette er markeret af et EOT (end of tape) mærke på båndet i form af aluminiumsstrimmel.
[14] Der er i næsten alle tilfælde anvendt 2400 fods bånd.

Bilag 2. Oversigt over programmer anvendt til recovery.

I dette bilag omtales de programmer, som blev anvendt i recovery projektet. Programmerne beskrives kun ganske summarisk og kun i forhold til deres funktion i recovery-projektet.

2. 1 Tapedump

Dette program er et tidligere udviklet generelt program til brug for visning af indholdet af en fil. I Recovery-projektet blev det anvendt på forskellig måde, alt efter hvilken type fil der var tale om, samt hvilket problem der var med filen.

2.1.1 Anvendelse for FB-filer

Tapedump anvendes enklest for FB-filer. Her kan man ved angivelse af recordlængden umiddelbart få konstateret, om filen indeholder et helt antal records. Hvis det er tilfældet og antallet svarer til det forventede i henhold til dokumentationen, så er filen anset for at være læst korrekt. Hvis recordantallet er for lille, er der åbenbart tale om et fragment, som evt. kan sammensættes med andre fragmenter.
Hvis Tapedump viser, at der er en 'recordrest', betyder det , at recordlængden enten er angivet forkert, eller at der forekommer fejl i filen, som medfører, at filens totale længde ikke svarer til et helt antal records[15].
I tilfælde af fejl i filen, kan man ved at bladre frem record for record eller ved at springe til et givet recordnummer[16] vurdere, om der er noget der 'ser forkert ud'. Hvis der f.eks.– som ikke sjældent forekommer – forventes et CPR-nr. i en bestemt position af recorden, kan man med Tapedump, ved angivelse af en recorddefinition i overensstemmelse med dette, forholdsvis let finde det sted i filen, hvor der første gang forekommer en fejl, hvorved strukturen bliver forskubbet. Hermed har man altså fundet starten på et 'sygt' område. At finde starten på det næste 'raske' område i en FB-fil er mere vanskeligt, idet det kræver visuel inspektion af data med henblik på at identificere 'karakteristiske' data som f.eks. et CPR-nr. Man kan ikke umiddelbart med Tapedump angive en ny record-start på et udvalgt sted. Man vil være nødt til med programmet RemoveBytes at fjerne det syge område og derefter anvende Tapedump på det rettede fragment. I tilfælde af mere end én eller ganske få fejl af denne art, vil det være uoverkommeligt at nå frem til et fejlfrit fragment på denne måde. Hertil kommer, at man ved denne metode uundgåeligt mister lokaliseringen af blokstrukturen, hvorved muligheden for fletning af fragmentet med andre fragmenter bliver mere usikker (se pkt. 2.3).

2.1.2 Anvendelse for VB-filer

Tapedump anvender for VB-filer informationen i blokheaderne til trinvis at identificere starten på en given blok, som så vises. På denne måde kan man bladre eller 'hoppe' frem i filen – men ikke baglæns, da positionen af den foregående blok ikke huskes af Tapedump. Hvis der som i Hayes-læsningerne er fejl i data, vil dette typisk vise sig ved, at den variable blokstruktur 'bryder sammen', fordi manglende eller forskubbede bytes medfører, at en forventet blokheader indeholder data, som ikke kan fortolkes som en korrekt blokheader. Man må så ved gentagne forsøg identificere den første forkerte blok. Når det er gjort, må man forsøge at finde starten på den næste blok. Dette kan kun ske ved at specificere hexadecimal læsning, men til gengæld er der som nævnt i bilag 1 karakteristiske træk ved blok- og record-headere i VB-format, som gør det principielt muligt at identificere en blokstart med ret stor sikkerhed. Herefter kan man med

---

[15] Dette forekommer som ikke for RA-læsninger, der som nævnt altid indeholder korrekt læste blokke, men derimod ofte for Hayes-læsninger

[16] Positionen af første byte i en bestemt FB-record, f.ex. nr. N, er (N-1)*RL+1

RemoveBytes fjerne det 'syge' område, og gentage analysen med Tapedump. Ligesom for FB-filer er det dog overordentlig tidkrævende, og da det for VB-filer var muligt at automatisere processen med et program, blev dette gjort, jf. pkt. 2.6.

2.2 TestVar

Dette program udvikledes til brug i mediekonverteringsprojektet til kontrol af antal blokke og records i VB-filer. I Recovery-projektet anvendtes programmet desuden til at finde strukturfejl i VB-filer eller - fragmenter .
Et kørselseksempel:

```
TEST AF VARIABEL FIL MED BLOKHEADER[17]

Min bloklængde   = 15004
Max bloklængde   = 15199
Blokke ialt      =   192

_____________________________________________________
Min rekordlængde = 23
Max rekordlængde = 61680
Rekords ialt     = 23188

_____________________________________________________
Indfil længde    = 151693389 bytes (ca. 144 MB)
Udfil længde     = 0 bytes (ca. 0 MB)

_____________________________________________________
STATUS           = AFBRUDT -Rekordlængde > Bloklængde 09:43:31:
```

I dette eksempel er læsning af filen stoppet ved blok nr. 192, fordi programmet er stødt på data, som ikke kan være rigtige. Recordlængden for den aktuelle record har et header indhold, som angiver en recordlængde på 61680. Den trinvise identifikation af blok-headere - og inden for hver blok recordheadere - kan derfor ikke fortsætte meningsfuldt. Den 'syge' blok må fjernes før man kan gøre et nyt forsøg med TestVar.
Andre fejltyper kan forekomme (fx. bloklængde =0), men fælles for dem er, at programmet går i stå ved pågældende fejl.
Hvis der ikke er datafejl i filen, anvendes programmet til at fjerne blokheaderne, der jo egentlig er et 'suppement' til selve data (records), som kun har relevans for magnetbåndsmedier.
Eksempel på en sådan konvertering:

```
KONVERTERING AF VARIABEL FIL MED BLOKHEADER

Min bloklængde   = 5904
Max bloklængde   = 5956
Blokke ialt      =   29635

_____________________________________________________
Min rekordlængde = 52
Max rekordlængde = 236
Rekords ialt     = 740877

_____________________________________________________
Indfil længde    = 174965144 bytes (ca. 166 MB)
Udfil længde     = 174846604 bytes (ca. 166 MB)

_____________________________________________________
STATUS           = UDFØRT 17:25:43
```

I eksemplet er det kontrolleret, at det læste antal blokke/records er korrekt.

[17] Man kan også angive, at filen er uden blokheadere. Hvis filen faktisk har blokheadere betyder det blot, at filen læses som om blokkene var records (dvs. nederste niveau i VB-strukturen).

2.3 MD5Flet

Fletteprogrammet, som blev udviklet specielt til brug for Recovery-projektet, anvendtes til at sammenflette to fil-fragmenter af en fil. Programmet anvender de enkelte blokkes MD5-værdi[18] som identifikation.
Eksempel:

```
4300 rekords i fragment 1     (filnavn: 868.131NoDupes)
9342 rekords i fragment 2     (filnavn  868.112NoDupes)

9946 unikke rekords

1  0  True  4300
2  3696  True  5646
********************************************************************************
Merger created at 11-07-2005 18:02:29
Sequence 1 has 4300 unique entries.
Sequence 2 has 9342 unique entries.
There are 9946 unique entries in total.
Merge operation started at 11-07-2005 18:02:29
StartSource = 1.  ForceSafeStart = True.
Found an 1-event at (p1, p2) = (0, 0).  Reading from sequence 1.
Found an 1-event at (p1, p2) = (4300, 3696).  Reading from sequence 2.
Reading the rest of sequence 2 at (p1, p2) = (4300, 3696).
Found 0 unsafe subsequence pair(s).
Merge operation ended at 11-07-2005 18:02:30
```

Der skal ikke her redegøres for de nærmere datalogiske detaljer i programmets funktionsmåde, men i eksemplet har fragmentet på 9342 blokke[19] kunnet suppleres med 604 blokke fra fragmentet på 4300 blokke til et samlet fragment på 9946 blokke.
Programmet anvendtes næsten udelukkende til VB-filer. Anvendelse på FB-filfragmenter var mindre relevant, da blokstrukturen her er ofte er invalid og records ikke er unikke.

2.4 RemoveDuplicateRecords

Dette program er anvendt til at fjerne dubletter af blokke, som er fremkommet ved indlæsning af af filer fra bånd(jf hovedafsnittet). Der angives navn på input-fil og outputfil. Programmet kan anvendes både på VB-filer og FB-filer. I sidste tilfælde skal bloklængden angives.
Eksempel på uddata fra dette program:

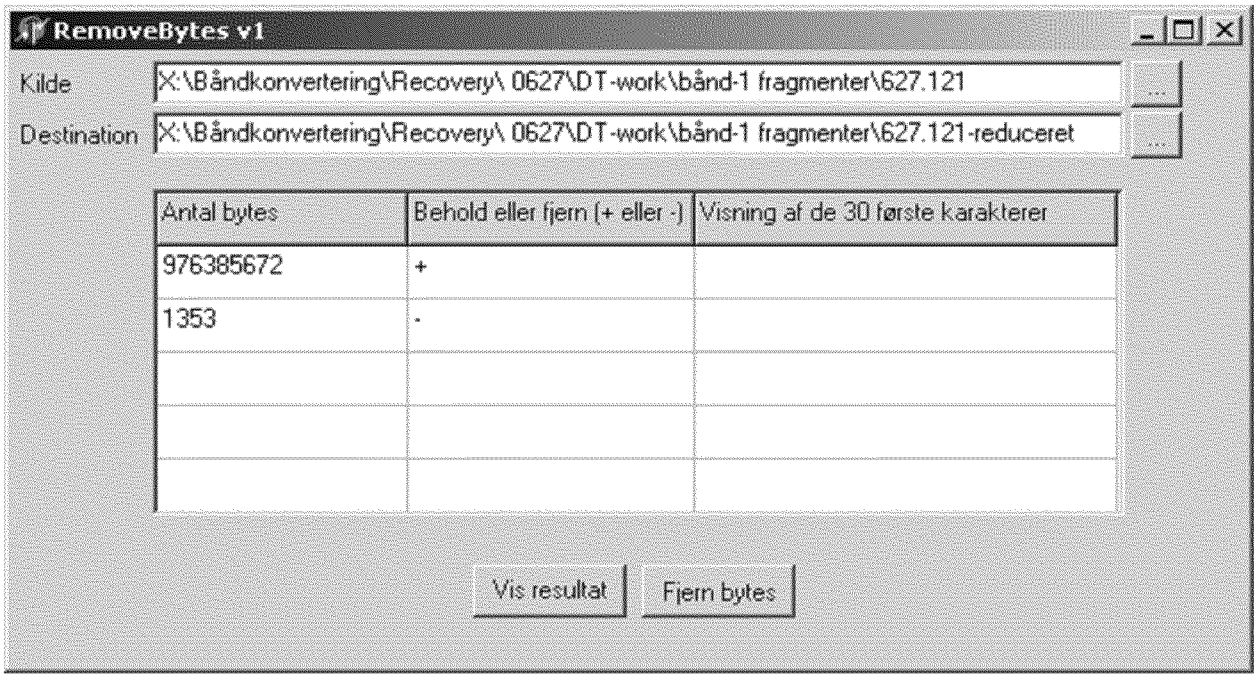
```
Duplikat fundet - blok/rekord nr: 541   MD5 værdi: F31EB457B99118AB0DAA51747F15480E
Duplikat fundet - blok/rekord nr: 862   MD5 værdi: DDA4D8AFE765AEF3BEE16C2D2B8DE8B7
Duplikat fundet - blok/rekord nr: 1184   MD5 værdi: BC8FD41CEF17CC0D2BEC8607A9B645D6
Duplikat fundet - blok/rekord nr: 1514   MD5 værdi: AE0CF305C60EDBDCA137AFA73E4557E5

---------- 59 linier med duplikater ----------

Duplikat fundet - blok/rekord nr: 12601   MD5 værdi: E66162905EE80D097D70B324BBCEE623
Duplikat fundet - blok/rekord nr: 12688   MD5 værdi: BE705A4BA3EB73CBDA9990BF1E8616B6
Duplikat fundet - blok/rekord nr: 12689   MD5 værdi: BE705A4BA3EB73CBDA9990BF1E8616B6
Duplikat fundet - blok/rekord nr: 12690   MD5 værdi: BE705A4BA3EB73CBDA9990BF1E8616B6

Samlet antal blokke/rekords: 13608 .
Heraf unikke blokke/rekords: 13541.
67 dublikate blokke/rekords fjernet.
```

[18] MD-5 værdien er en slags 'tværsum' baseret på en type algoritme, som også anvendes ved kryptering, og som entydigt identificerer pågældende blok.

[19] I output fra programmet bemærkes at der tales om records. Programmet kan ikke vide om de læste 'records' egentlig er blokke, idet blokheaderen ikke er fjernet.

2.5 RemoveBytes

Programmet anvendes til at fjerne et givet antal bytes fra en given byte-position i en fil. Desuden angives input og outputfil. For nemheds skyld vises her et skærmdump.



I eksemplet kopieres de første 976385672 bytes, hvorefter de næste 1353 bytes skippes. Derefter kopieres resten af filen til destinationsfilen.

2.5 VarRecNy

Dette specielt udviklede program anvendtes til at fjerne alle syge blokke fra en VB-fil eller -fragment. Input til programmet er filnavn, max-bloklængde, max recordlængde.
Programmet læser blokheadere trinvis og tester, om de kan være korrekte. Som kriterium anvendes, at bloklængden skal være mindre end max-bloklængden men også større end max-bloklængden minus max-recordlængden. Desuden skal blokheader og 1. recordheader indeholde tomme 3.- 4. bytes. Hvis man støder på en blok, som ikke kan være rigtig ifølge disse kriterier, søges videre i data, indtil der findes 8 bytes, som til sammen kan udgøre en korrekt blok- og recordheader. Som en ekstra kontrol testes de 3 første record-headere også. Herefter markeres (huskes) start og slutbyte på den 'syge' blok, og programmet fortsætter læsningen af data. Programmet kører i 2 gennemløb, idet der først testes på blok-niveau, dernæst på record-niveau. Grunden til det sidste er, at der i HAYES-læsningerne viste sig eksempler på, at en fil, som var korrekt i struktur på blok-niveau men ikke på record-niveau, hvilket viste sig ved at TestVar fejlede. I sådanne tilfælde markeres hele den pågældende blok som fejl-bytes, som skal fjernes[20]. Efter de to gennemløb er alle fejlblokke markeret og tilsvarende bytes fjernes.

Eksempel:
Nedenfor er vist et output fra en kørsel med VarRecNy. I eksemplet er der kun fundet fejl på blokniveau, ikke på record-niveau (2. gennemløb).

[20] Grunden til dette er, at efterfølgende fletning af fragmenter som nævnt bør foregå på blokniveau.