Mimer 3.2

Mimer 3.2 Användarhandbok

Uppdaterad för Mimer/DB Version 3.2.8

Diab Data AB

Box 2029 S-183 02 TÄBY
SWEDEN

Revisionsinformation

Revisionsinformation

Revision	Datum	Kommentar
A	870609	Första versionen av dokumentationen

Innehåll

1. Introduktion

1.2 Målgrupp för dokumentationen

Denna dokumentation vänder sig huvudsakligen till tillämpningsprogrammerare och databasansvariga (databasadministratörer, DBA).

1.3.1 Mimer/DB

Mimer/DB är ett relationsdatabassystem för fleranvändarmiljö utvecklat vid Uppsala Universitets Datacenter. Mimer/DB används av övriga moduler i Mimer-familjen men kan även användas av tillämpningsprogram skrivna i olika programspråk.

Fysiskt består en Mimer databas av en eller flera databanker (fysiska filer). De olika användarnas behörighet till databasen bestäms av databasadministratören (DBA).

1.3.3 Mimer/PG

Mimer/PG används för att skapa tillämpningsprogram genom att använda Mimer databashanteringssystem och skärmhanteraren Mimer/SH. PG är både ett kraftfullt interaktivt frågespråk och programgenerator. Programmen som skapas är antingen Fortran- eller Cobol källfiler.

1.3.5 Mimer/SH

Mimer/SH är en terminaloberoende skärmhanterare åtkomlig från tillämpningsprogram. Det är även ett program för interaktiv generering av bildskärmslayouter.

1.4 Filsystemet som används av Mimer

Alla filer (databanker och definitionsfiler) vilka används av Mimer skall finnas under biblioteket /usr/mimer. (se fig. 1.1.)

Privilegierna för åtkomst av /usr/mimer skall vara drwxr-xr-x och ägaren skall vara 'root'.

Biblioteken under /usr/mimer är:

sysdb - Innehåller systemdatabanker.
Privilegier för åtkomst drw-------.
Ägare 'root'.

Privilegierna för åtkomst ovan försäkrar att ingen annan än systemanvändaren ,'root', har tillgång till databankerna i detta bibliotek.

Databanker:

SYSDB	- Systemdatabank
LOGDB	- Logdatabank
TRANSDB	- Transaktionsdatabank
WORKDB	- Temporär databank

usrdb - Innehåller alla användardatabanker.
Privilegier för åtkomst drw-------.
Ägare 'root'.

sh - Innehåller alla definitionsfiler för Mimer/SH.

ql - Innehåller frågespråket QL's systemdatabank, SYSQL.

pg - Innehåller PG-program och PG's systemdatabank, SYSPG.

2. Att använda Mimer/DB

2.2 Multi Databashanteraren

Alla Mimer lågnivå databankförfrågningar skickas från tillämpningsprogrammen till Mimer databashanteraren, dblm.

Denna hanterare är ett program som kräver två startparametrar. Den första startparametern är hanterarbibliotek och den andra parametern är det fysiska filnamnet på systemdatabanken.

Exempel:

```
# dblm /mimerdbl /usr/mimer/sysdb/sysdb &
```

Hanterarbiblioteket måste finnas och ha namnet /mimerdbl. Systemdatabanken specificeras eftersom det kan finnas flera systemdatabanker i systemet (endast en kan användas åt gången). Ett och-tecken (&) behövs eftersom hanteraren arbetar som ett bakgrundsjobb.

En shellrutin, kallad /usr/bin/startdb, startar Mimer hanteraren. Rutinen kontrollerar att hanteraren inte redan körs och att hanterarbiblioteket existerar.

Shellrutinen kontrollerar också att processen inte är specificerad med 'respawn' i /etc/inittab. Om filen /usr/mimer/backup/ERR finns, ges en varning.

Nedan är ett exempel på rutinen **startdb**:

```
:
: Startup file for mimer v 3.2.8 handler
:
: Check the inittab file for 'respawn'

RESP=`grep db3: /etc/inittab ö grep -c respawn`
if test ØRESP != 0
then
      echo WARNING: dblm is specified with Ö'respawnÖ' Ö
                                                in /etc/inittab
      echo This is not recommended, it should be Ö'onceÖ'
      exit 1
fi

: Check if ERR file from mdbchk

if test -f /usr/mimer/backup/ERR
then
    echo Startdb: Error file /usr/mimer/backup/ERR exists
    echo this indicates databank problems, please check!
    echo dblm NOT started.
    exit 1
fi
```

eller

```
# ps lx          ( process-statuslista )
# kill -9 XX     ( där XX är processnummret på
                   dblm-hanteraren )
# startdb        ( Startdb är en kommandofil för att
                   starta hanteraren )
```

En annan metod är att använda hjälpprogrammet 'dbadm' och göra val 12 - Restart.

Om du vill att Mimer Databashanteraren skall startas automatiskt efter återstart av systemet, ändra då en rad i startfilen /etc/inittab.

Exempel:

```
db3:2:once:dblm /mimerdbl /usr/mimer/sysdb/sysdb
```

2.3 Program för databasadministratören

Följande program används av databasadministratören, DBA, för underhåll av Mimer databassystem:

dbadm - Program för att skapa/ändra/visa olika databanker, användare och privilegier. Programmet finns också som en underfunktion i QL och PG.

dbchk - Databankkontroll.
Kontrollerar databankfilerna så att den fysiska informationen inte är förstörd.

dbbru - Hjälpprogram för Backup.
Programmet gör det möjligt att kopiera innehållet i en databank till en backupdatabank eller för att återskapa en databank, vilken har förstörts eller förvanskats, från en backupdatabank och logdatabanken.

dbtab - För att skapa tabeller i specificerade databanker.

expimp - Export/Import hjälpprogram för att flytta Mimer databanker mellan olika datortyper som exempelvis DS90 och VAX.

sdbgen - Program för att generera en ny systemdatabank, SYSDB.

mimer - Generellt statusprogram för övervakning av Mimer.

dbcopy - Program för kopiering av tabeller från en databank till en annan.

Det fysiska filnamnet som efterfrågas i alla dessa program är D-NIX path-namnet på filen som innehåller databanken.

Anmärkning: path-namnet måste börja med / och får vara maximalt 64 tecken långt.

Anmärkning: Observera att biblioteken i path-namnet måste existera när nya databanker skapas.

Exempel:

```
/usr/mimer/usrdb/testdb
```

2.5 EXITA1-rutinen

Allokerings/exit-rutinen EXITA1 används för att öka storleken på systemkontrollutrymmet. Om detta utrymme överskrids returneras en felkod, xx31, till användarprogrammet.

Den normala EXITA1-rutinen i /usr/lib/libMdbm.a är:

```
      SUBROUTINE EXITA1
C-------------------------------------
C   Mimer/DB Allocation routine
C-------------------------------------
      INTEGER  W
      COMMON /STORE2/ W(4096)
C
      W(1) = 4096
      RETURN
      END
```

Om felkod xx31 uppträder måste detta utrymme (4096 ord) utökas.

Exempel:

Ett tillämpningsprogram kallat 'app1' körs och fel xx31 skrivs ut av EXITC1 (Se avsnitt 2.6). Utrymmet 4096 i EXITA1 ökas då till 8192.

Rutinen kompileras med:

```
f77 -c exital.f
```

och programmet 'app1' länkas på nytt med:

```
cc -x appl.o exital.o exitcl.o -lMdbm -lm77
```

Om man vill skriva exita1 i C kan det se ut så här:

```
/* Mimer DB Allocation routine. The default      */
/* size in /usr/lib/libMdbm.a is 4096 for MSIZE  */

#define MSIZE 8192

int store2_ÄMSIZEÅ;

exital_()
ä
    store2_Ä0Å=MSIZE;
å
```

2.6 EXITC1-rutinen

När ett fel detekteras av Mimer/DB, anropas felrutinen EXITC1 med felkoden som parameter.

Den normala EXITC1-rutinen, placerad i Mimer/DB biblioteket /usr/lib/-libMdbm.a, skriver ut felet på standard output (vanligtvis din terminal) och avslutas.

När ett tillämpningsprogram testas är det bra om ett meddelande skrivs ut då fel uppstår. När tillämpningen är klar kan exitc1-rutinen ersättas av en tom rutin.

Exempel:

Detta är en tom EXITC1-rutin, skriven i C, som bara returnerar utan att utföra någonting.

```
exitcl_(errcode)

int *errcode;  /* You get a pointer from a Fortran
                  program */

  ä
     /* nothing is done */
  å
```

2.8 Programexempel

Det följande avsnittet innehåller två exempel på hur du använder Mimer/DB från ett C-program. För definition av Mimer/DB anropen se 'the Mimer/DB reference manual' kapitel 2.

Alla "lables" i Fortran genereras med 'underline' (_) som avslutning. Det är därför som du måste lägga till ett 'underline'-tecken efter namnet på Fortran-rutiner i dina C-program.

Exempel 1:

```
/*
 *=========================================================
 *
 *    EXAMPLE of how to use Mimer/DB.
 *
 *    A databank MIMTST contains two tables:
 *
 *        INFO     DATE    C6   *
 *                 PNR     C10
 *                 VAL1    I4
 *                 VAL2    F8
 *                 COM     C30
 *
 *        PERS     PNR     C10  *
 *                 NAME    C30
 *                 ADRESS  C30
 *
 *    A date is entered from the terminal and is then
 *    searched for in the table INFO. If the date a found,
 *    a list of all entries from this date is written on
 *    the screen.
 *
 *    Example:
 *                                              Date: 850528
 *
 *    Pnr         Name        Adress     Vall  Comment
 *    ------------------------------------------------------
 *    1111111111 Mr Fictive Not known    15  No real person
 *    2222222222 Mr Smith   Not known   177  No comment
 *
 *=========================================================
 */
```

```
    begin2_(&icond,user,passw);
    if (icond)
      ä
         if (retry-- < 0)
            ä
              /* No more retries */
              exit(1);
            å
          else
            ä
         printf("Wrong username or password. Retry.Ön");
            å
       å
   å
  while ( icond != 0 );

/*  Open the databank ...  */

opend2_(did, MIMTST, S);
if (didÄ0Å)
  ä
   /* Can't open databank. Fatal! */

   printf("Can't open the databank MIMTST. Fatal.ÖnÖn");
   exit(2);
  å

/* ... and the two tables.
       Exit if error in open     */

opent2_(tidl, did, INFO, S);
if ( tidlÄ0Å )
  ä
  printf("Can't open INFO-table. Fatal!.ÖnÖn"); exit(3);
  å
opent2_(tid2, did, PERS, S);
if ( tid2Ä0Å )
  ä
  printf("Can't open PERS-table. Fatal!.ÖnÖn"); exit(4);
  å

/* Make all projections into the output buffer */
/* VAL1 is projected as a char string and will be con-
   verted from integer to char automatically.   */
```

```
                if (tid2Ä0Å)
                  ä
          printf("Name entry for person not found.ÖnÖn");
                  å
                else
                  ä
                    /* print out entry */

                    bufÄ79Å='Ö0';
                    printf("%sÖn",buf);
                  å
             for(icond=0;icond<79;icond++) bufÄicondÅ=' ';
                get2_(tid1,BASE);
              å /* end while */
         å /* end else */
    å /* end main loop */
å
```

Filen i exempel 1 är kallad 'ex1.c' och den kompileras/länkas med:

```
¤ cc ex1.c exit.o -lMdbm -lm77 -o ex1
```

'Exit.o' innehåller rutinen EXITC1 beskriven i avsnitt 2.6. -lm77 biblioteket behövs eftersom Fortran-kompilatorn genererar anrop till rutiner för enkel precision vilka inte existerar i standard biblioteket libc. -lm77 biblioteket rättar till dessa skillnader.

3. Export/Import

3.2 Restriktioner i Export/Import

Nedanstående gäller för både EXPORT och IMPORT.

1. Maximalt tre felaktiga försök till login tillåts innan hjälpprogrammet avslutas. Därefter återgår kontrollen till operativsystemet. Detsamma gäller om inte både användarnamn och lösenord anges.

2. Om du bara svarar med en tom inträng när programmet väntar på en inträng kommer du att gå till den högre menynivån. Ett undantag från denna regel är den högsta menyn där du måste ge en inträng för att någonting skall hända.

3.3.1 Hjälpprogrammet Export

Det första som händer när du använder EXPORT är att du får frågan om vilken databank du vill skriva ut. Om denna databank inte är definierad i SYSDB kommer du att få felmeddelandet <*** ERROR databank not found in SYSDB ***>. Om databanken finns i SYSDB kommer du att frågas om det fysiska filnamnet på den fil som databanken skall skrivas till. Längden på filnamnet får inte överskrida 45 tecken. Om ett fel uppstår då den fysiska filen skall öppnas kommer felmeddelandet <*** ERROR while opening physical file ***> att visas på din terminal.

Därefter kommer du att få frågan om du vill skriva ut hela databanken (tabelldefinitioner och tabellrader) eller om du bara vill skriva ut tabelldefinitioner för varje tabell. Under tiden som databankerna skrivs ut kommer du att få meddelanden om vilka tabeller och hur många rader som skrivits ut. Efter det att utskriften är klar kommer du att återgå till huvudmenyn.

Exempel:

```
Dump databank : DRUGDB
Dump on file  : drugfile.dat

1. Dump table definitions and table rows.
2. Dump only definitions for each table.
Enter number  : 1

*** All table definitions dumped ***

Dumping table : TABLE1

    100 row(s) dumped.
    200 row(s) dumped.

    248 row(s) dumped.

Dumping table : TABLE3

        . . .
        . . .
```

Efter alla dessa databankdefinitioner börjar IMPORT att definiera alla tabellkolumner i den nya databanken. Följande varningar/felmeddelanden kan uppträda.

1. <*** WARNING. Error may occur while loading 'table name' and column 'column name' with floating(s) ***>

Motsvarande felmeddelande finns för heltalsvärden. Den här typen av varningar uppträder endast då du har skrivit ut en databank till en datafil och skall ladda ned den till en ny dator med kortare intern ordlängd. Till exempel, din databank har skrivits ut på en IBM dator (5 byte intern ordlängd) och skall nu laddas ned till en DIAB dator (4 bytes intern ordlängd).

När IMPORT laddar den nya databanken med flyttal och heltal, kontrolleras även om underflow/overflow förekommer.

2. <*** ERROR unexpected end of file ***>

Den här felkoden indikerar att IMPORT inte kunde hitta ett korrekt filslut i datafilen. Ett korrekt filslut definieras som en rad bestående av 78 '*', ett inledande blanktecken och textsträngen <END OF FILE> i mitten.

Om denna typ av fel uppträder kommer alla databanker och filer att stängas omedelbart och huvudmenyn kommer upp på din terminal.

Efter alla definitioner och kontroller kommer IMPORT att ladda varje tabell med tabellrader från datafilen. Du kommer att informeras om vilka tabeller och hur många rader som har laddats. När databanken har laddats återvänder du till huvudmenyn.

Möjliga fel när databanken laddas är något av följande:

1. <*** ERROR in row 'row.no' while converting form character to floating value ***>.

Motsvarande felmeddelande finns för heltalsvärden. Orsaken till detta felmeddelande är någon av följande:

I. Teckensträngen innehöll inte numeriska tecken eller deras attribut, därför är strängen odefinierad som numeriskt värde.

II. Teckensträngen är korrekt definierad men overflow eller underflow har uppträtt under konverteringen av teckensträngen till numeriskt (heltals eller flyttals) värde.

3.3.3 Vanliga felkoder

Om det uppstår något fel vid användandet av någon av de sekventiella åtkomstrutinerna, kommer dessa rutiner att stänga samtliga databanker och filer. Kontrollen lämnas sedan till operativsystemet.

Tänkbara fel som kan orsaka detta är något av följande.

```
<*** ERROR while reading from file ***>

<*** ERROR while writing on file ***>

<*** ERROR while closing file ***>
```

4. Hjälpprogram

4.2 SDBGEN - Systeminitialisering

I Mimer/DB version 3 finns kontrollen av användarbeskrivningar, befogenheter, databankbeskrivningar och åtkomstprivilegier i en systemdatabank. Innan någon Mimer/DB tillämpning (i enanvändar- eller fleranvändarmod) kan köras måste en systemdatabank definieras.

Hjälpprogrammet 'sdbgen' kan användas för att skapa systemdatabanken.

4.2.2 Tillträde

Först av allt får du svara med användarnamn (username) och lösenord (password) för databasadministratören, DBA. Både användarnamn och lösenord är teckensträngar på upp till 8 tecken. För att få tillträde fler gånger måste du komma ihåg dessa tecken. Alla små bokstäver som skrivs in i användarnamnet görs automatiskt om till stora bokstäver. Observera att det har stor betydelse om du använder stora eller små bokstäver i lösenordet. Tecknen måste vara desamma (stora eller små) nästa gång du vill logga in. Lösenordet lagras så att det inte går att läsa tillbaka på något sätt.

4.2.4 Hantering av transaktioner och möjligheter till loggning

För att upprätthålla integriteten då fler än en användare utnyttjar en delad databas, kan användningen delas in i logiska enheter kallade transaktioner. Om du skall använda transaktioner måste du skapa en transaktionsdatabank. Du kommer då att få frågan om filnamn och startstorlek på transaktionsdatabanken. Svaren skall vara av samma typ som då du skapade systemdatabanken.

Om du använder transaktioner kan du utnyttja ett antal möjligheter till loggning. Om en logdatabank existerar kommer alla utförda transaktioner att loggas. När det blir något fel på en disk eller att en databankfil blir förstörd kan logdatabanken användas tillsammans med en tidigare kopia av databanken för att återskapa denna.

Om du vill använda en logdatabank anger du namn och startstorlek på samma sätt som för systemdatabanken. Om du avstår från logdatabanken aktiveras inte möjligheten till loggning.

Observera att filnamnen för transaktions- och logdatabanken lagras i systemdatabanken med relation till de logiska databanknamnen TRANSDB och LOGDB. Du får inte använda samma transaktions- och logdatabanker i olika systemdatabanker.

Både TRANSDB och LOGDB kan senare definieras, definieras om eller tas bort med hjälpprogrammet 'dbadm'.

4.3 DBADM - Databasadministration

Att definiera databanker inkluderar även att definiera användare, databanker och användarnas åtkomstprivilegier. Hjälpprogrammet 'dbadm' kan användas för att göra dessa definitioner.

'dbadm' startas med kommandot dbadm. Efter det att programmet har startat måste du identifiera dig själv med användarnamn och lösenord. Om du har DBA (X) privilegier kommer det upp en meny med samtliga valmöjligheter annars kommer det upp en mindre meny.

dbadm tillval:

```
-V                    Ger versions signon
-t                    Avslutar med version signon
-l <user>/<password>  Automatisk login
```

Exempel på DBA meny:

```
***************** DATABASE ADMINISTRATION ***************
* 1. Define users  2. Define databanks  3. Define access*
* 4. Remove users  5. Remove databanks  6. Remove access*
* 7. Show   users  8. Show   databanks  9. Show   access*
*10. Handler info 11. User info        12. Restart      *
* 0. Exit                                               *
*********************************************************
Enter number:
```

För dina svar till de olika operationerna gäller följande:

Ett tomt svar (enbart return) betyder 'avsluta denna operation'. Du återväder då till den första frågan för operationen eller tillbaka till menyn om det var den första frågan.

Alla svar med små bokstäver görs automatiskt om till stora utom vid lösenordet.

4.3.2 Definiera, ta bort och visa databanker

Vid definition av databanker skall följande information ges:

Databank
Ett logiskt Mimer/DB-namn med upp till 8 bokstäver eller siffror med minst en inledande bokstav. Om du anger ett redan befintligt namn för du möjlighet att definiera om nedanstående:

Filnamn
Motsvarande operativsystemnamn på den sekundära lagringsfil som skall innehålla (eller innehåller) databanken. Namnet skall alltid vara det fullständiga filnamnet (pathname), d.v.s det skall börja med /, och får vara maximalt 64 tecken långt.

Anmärkning: Det måste vara ett ett-till-ett förhållande mellan filnamnet och databanken. Ingen kontroll utförs för att kontrollera detta eftersom operativsystemet tillåter olika namn för att indikera samma sekundära lagringsfil (länkade filer).

Storlek (Size)
Ett numeriskt värde större eller lika med noll. Värdet noll indikerar att den angivna filen redan existerar. Detta kan användas vid omdefiniering eller då en databank skall flyttas från ett enanvändarsystem till ett fleranvändarsystem. Förutom vid generell åtkomst B (se nedan) kommer filen att öppnas för att kontrollera om den existerar.

Om storlek > 0 kommer en ny fil att skapas och formatteras som en Mimer databank. Det angivna värdet används som erfordrad lagringsplats i Mimer/DB-sidor.

Anmärkning: Om det existerar en fil med det angivna namnet kommer den att förstöras.

Generell åtkomst
Generella åtkomstprivilegier anger vilka typer av operationer en användare får göra i/med en databank om inga individuella åtkomstprivilegier har definierats för användaren (se nedan). Fem typer av generella åtkostprivilegier existerar:

```
X - exklusiva privilegier. Alla typer av operationer är
    tillåtna inklusive datadefinitioner.

S - delade privilegier. Lägga till, ändra och hämta data
    är tillåtet

R - läsprivilegier.  Endast att hämta data är tillåtet
```

4.3.3 Definiera, ta bort och visa åtkomstprivilegier

När en databank definieras eller definieras om ges generella åtkomstprivilegier till databanken. För enskilda användare kan detta åsidosättas d.v.s om en användare ges egna åtkomstprivilegier för en databank gäller inte de generella åtkomstprivilegierna för denna användare.

Vid definition av åtkomstprivilegier kommer du att få frågorna:

Användarnamn (Username)
En tidigare definierad användare (se definiera användare).

Databank
En tidigare definierad databank (se definiera databanker).

Åtkomstprivilegier
Fyra typer av åtkomstprivilegier kan sättas upp för en användare och den avsedda databanken:

X - exklusiv eller ägar privilegier. Alla typer av operationer är tillåtna, inklusive datadefinitioner och bestämmande av åtkomstprivilegier för andra användare av databanken.

S - delade privilegier. Lägga till, ändra och hämta data är tillåtet.

R - Läsprivilegier. Endast hämtande av data är tillåtet.

P - privata privilegier. Inga operationer är tillåtna.

Om definitionen lyckas kommer du att få meddelandet <<<< Defined >>>> eller <<<< Redefined >>>>. Om den angivna användaren eller databanken inte definierats tidigare kommer du att få ett felmeddelande.

När du väljer att ta bort privilegier skall du ange avsett användarnamn och databank. När du har angett avsedda namn får du meddelandet <<<< Removed >>>> om kombinationen existerade. Om inte får du meddelandet <<<< not found >>>>.

När du väljer att visa privilegier kommer det upp en lista över de definierade privilegier som finns mellan användare och databanker.

4.3.5 Användarinformation (user info)

'user info' ger information om vilka användare som för tillfället använder mimerhanteraren.

Rubrikerna i kolumnerna har betydelsen:

```
No          Användarnummer i hanteraren
dnixusr     Användarnamn under D-NIX
mimusr      Användarnamn under Mimer (MIMER loginnamn)
program     Programnamn. Detta namn sätts med SETUSR rutine
            om inte blir det användartillämpningen.
            (Se Avsnitt 2.4 om hur man sätter
            programnamn med SETUSR )
procid      processidentitet för mimerprocessen
fetch       antal fetch1-anrop till hanteraren
insert      antal inser1-anrop till hanteraren
delete      antal delet1-anrop till hanteraren
update      antal updat1-anrop till hanteraren
db          antal öppnade databanker
time        tid du har varit inloggad
```

Exempel:

```
¤ dbadm -l dba/dba          (Automatisk inloggning används)

***************** DATABASE ADMINISTRATION ***************
* 1. Define users  2. Define databanks  3. Define access*
* 4. Remove users  5. Remove databanks  6. Remove access*
* 7. Show   users  8. Show   databanks  9. Show   access*
*10. Handler info 11. User info        12. Restart      *
* 0. Exit                                               *
*********************************************************
Enter number :11

No  dnixmim prog- proc- fetch insert delete update db time
    usr usr ram   id
-------------------------------------------------------
0000jt  DBA dbadm 08315 00003 00000  00000  00000  03 00:0
```

4.4.1 Backup på en databank

När du väljer att göra en backupkopia på en databank kommer du att få följande frågor:

```
FROM Original databank (max 8 chars) :
```

Ditt svar skall vara ett logiskt Mimer/DB-namn på den databank du vill göra en kopia av. En kontroll görs att den utvalda databanken finns definierad i din systemdatabank. Om den inte är det kommer du att få meddelandet < < < < Databank not defined > > > > och du kommer att återvända till menyn.

```
TO Backup databank (max 8 chars):
```

Här ger du ett logiskt Mimer/DB-namn på den databankfil som skall innehålla din kopia. Denna databank måste ha det generella åtkomstprivilegiet definierat som B (backup access). Om den inte har det kommer du att få meddelandet < < < < Backup databank not defined > > > >. Skriv in filnamnet om du vill definiera den:

Då har du möjlighet att skriva in ett operativsystemnamn på filen som skall innehålla backupkopian. Ett tomt svar (enbart RETURN) återför dig till menyn och ingen backupkopiering utförs.

Om dina svar accepteras kommer kopieringen att börja. Först kontrolleras att den utvalda databanken inte används av någon annan användare. Om den gör det får du meddelandet < < < < Databank occupied by others user > > > >. Andra möjliga fel visas med MIMER/DB's felmeddelanden.

När kopieringen är klar skapas en ny fil med det namn du angav som namn på kopian. Den nya filen kommer att få samma storlek som den databank du kopierade så se till att du har tillräckligt med ledigt utrymme på ditt skivminne innan du gör backupkopian.

Varning: Finns det en gammal fil med samma namn som den nyskapade kopian kommer den att skrivas över.

Förutom kopieringen kommer även alla loggade förändringar i den utvalda databanken att raderas från logdatabanken. Det görs även en kontroll att den gamla databanken överensstämmer med den nya kopian. Bitmap sidan(orna) skrivs in på nytt. Om det inte uppstår några fel kommer du att få meddelandet < < < < done > > > >.

Anmärkning: Backupkopieringen kan göras utan att det finns någon logdatabank, men då är det en ren kopiering fil-till-fil.

4.4.3 Kontrollera databank och ta bort loggade poster

Denna valmöjlighet kan du utnyttja om du vill göra backupkopian på din databank med något annat systemhjälpmedel. Om detta systemhjälpmedel tillåter kopiering av filer som är öppnade av andra användare måste du se till att den utvalda databanken inte används (t.ex genom att stoppa Mimer/DB fleranvändarsystemet). När du har använt denna valmöjlighet kommer alla loggade poster i den utvalda databanken att tas bort. D.v.s logdatabanken kommer efter detta enbart att innehålla förändringarna i databanken efter det att backupkopian gjorts. Dessa loggade poster kan sedan användas tillsammans med din backupkopia för att läsa tillbaka (restore) databanken.

När du väljer att kontrollera databank och ta bort loggade poster kommer du att få följande frågor:

```
Databank (max 8 chars):
```

Ditt svar skall vara ett logiskt Mimer/DB-namn på den databank där du vill ta bort alla loggade poster. Om databanken inte är definierad i din systemdatabank kommer du att få meddelandet < < < Databank not defined > > >. Annars kontrolleras att den utvalda databanken inte används av andra användare. Om den gör det kommer du att få meddelandet < < < Databank occupied by other user > > >.

Förutom att ta bort alla loggade poster för databanken görs även en kontroll av databanken och bitmap sidan(orna) skrivs in på nytt. Om inga fel upptäcks kommer du att få meddelandet < < < Done > > > när det är klart.

4.4.5 Vad loggas ?

När loggning används kopieras samtliga förändringar i databanken till ett separat logmedia. Detta media är en speciell databank, LOGDB, vilken måste definieras. Endast utförda operationer loggas. Loggning kräver även hantering av transaktioner och en egen transaktionsdatabank, TRANSDB.

Det finns vissa begränsningar för operationer i en databank som loggas. Följande loggas inte:

- Datadefinitioner, d.v.s. definiera eller ta bort kolumner och definiera och ta bort index. (Mimer/QL kommandona: Define, Redefine eller Remove Table och Define eller Remove Index)

- All operationer utförda på en tabell som öppnats med privilegiet exklusivt-(x). När skapa nya rader eller ta bort alla rader utförs måste detta privilegium vara uppfyllt. (Mimer/QL kommandon: Copy/Load och Delete där Where-bisatsen utelämnats)

Anmärkning: Insert, delete och update operationer i en transaktion under x-tabell access görs direkt mot databanken och blir därför inte loggade. (Mimer/QL operationer efter Include:x)

- Databasdefinitioner, d.v.s alla ändringar som görs med hjälpprogrammet, DBADM. Dessa operationer görs mot systemdatabanken, SYSDB.

Anmärkning: Ändringar av databankdefinitionerna är speciellt känsliga då de loggade posterna är sammankopplade med databanken via de identitetsnummer som fås från systemdatabanken.

När någon av de ovanstående operationerna utförs på en databank som innehåller viktiga data rekommenderas att du gör en ny backupkopia av databanken. Detsamma gäller om ändringar görs utan transaktionshantering.

Exempel:

Efter det att du kört dbchk på systemdatabanken kan resultatfilen se ut så här:

```
Databank file : /usr/mimer/sysdb/sysdb

Time : 1987-03-10 20:48:56

Bitmap pages :          0

Root   pages :          1

Table: *TABDEF 00 Start: 2 Levels: 1 Keylen: 10 Reclen: 24

Table: *DBDEF  00 Start: 3 Levels: 1 Keylen:  8 Reclen: 77

Table: *USERDEF00 Start: 4 Levels: 1 Keylen:  8 Reclen: 21

Table: *ACCESS 00 Start: 5 Levels: 1 Keylen: 16 Reclen: 17
```

4.7 DBTAB

Dbtab är ett program för att interaktivt hantera mimertabeller. Programmet gör det möjligt att lista tabelldefinitioner, skapa nya tabeller och ta bort tabeller. Det kan även användas för att lägga till/ta bort sekundärindex.

Du startar programmet med att skriva kommandot dbtab. Dbtab-kommandot finns i systemfilen i biblioteket /usr/bin. När du startar programmet visar det följande:

```
1. List a table.
2. Define a new table.
3. Remove a table.
4. Define a secondary index
5. Remove a secondary index
0. Exit.
```

När du har valt något av ovanstående skall du ange vilken databank och tabell du vill arbeta med. Möjliga funktioner är:

1. Ta fram en tabelldefinition som beskriver typ, namn och längd på alla kolumner.

2. Interaktivt definiera en ny tabell d.v.s specificera kolumnerna i en ny tabell.

3. Ta bort en befintlig tabell från en databank.

4. Definiera ett sekundärt index till en kolumn i en tabell. Vilket resulterar i ett nytt binärt sökträd i den angivna kolumnen.

5. Ta bort ett sekundärt index för en kolumn i en tabell.

Exempel:

Exempel 1:

Lista tabellen DOC i databanken JECTST.

```
    Select: 1

List a table.
------------

Type ? for a list databanks/tables
and just <CR> to return to the main menu.
```

```
Now define remaining columns.

Column name (max 8): COL1
Type of column and size (C,I,F): C20
Column name (max 8):<CR>

Table defined as:

Databank: JECTST  Table: NEW

     (01)PRIMI    I 02   (primary)
     (02)PRIMC    C 20   (primary)
     (03)COL1     C 20


Verify: Is table ok (y/n) ? y
Done.
```

Exempel 3:

Ta bort en tabell NEW i databanken JECTST.

```
Manipulate mimer table(s).
==========================

  1. List a table.
  2. Define a new table.
  3. Remove a table.
  4. Define a secondary index.
  5. Remove a secondary index
  0. Exit.


    Select: 3

Remove an old table.
--------------------

 ( The table must be empty )


Databank name (max 8 chars): JECTST
Give name of old table: NEW

     (01)PRIMI    I 02   (primary)
     (02)PRIMC    C 20   (primary)
     (03)COL1     C 20

Is it ok to delete this table (y/n) ? y
Done.
```

```
Databank: JECTST  table: DOC

     (01)COL1   I 02   (primary)
     (02)COL2   C 20   (secondary)
     (03)COL3   F 04

Give secondary index to be removed: COL2
Done.
```

-s Returnera hanterarens status:

0 - Hanteraren är igång
1 - Hanteraren är inte igång
2 - Hanterarbiblioteket finns ej
3 - Hanteraren verkar loopa (hanteraren är igång men kan ej kontaktas med systemanrop).

-k Döda hanteraren (Kill). Följande returvärden kan förekomma:

0 - Hanteraren har stannats
1 - Hanteraren är inte igång, så inget har utförts av kommandot.
2 - Misslyckades med att stanna hanteraren.
-1 - Inga privilegier för att stanna hanteraren. (För att göra detta måste man vara su).

-f Som -k men hanteraren tas ner även om det finns inloggade användare.

-u Returnera hur många användare som är inloggade.

-r Återstarta hanteraren (Restart). Följande returvärden kan förekomma:

0 - Hanteraren har startats.
1 - Hanteraren är igång, så inget har utförts av kommandot.
2 - Uppstartsförsöket har misslyckats.
-1 - Inga privilegier för att återstarta hanteraren (För att göra detta måste man vara su).

-l <användare>/<Lösenord>
Detta kommando används för att logga in mot Mimer. Vissa kommandon kräver nämligen att man är inloggad (-g,-a). Denna användare behöver inte ha några privilegier alls på databanker i systemet. Returvärdet -1 ges om inloggningen ej lyckas.

-g <fil>
Generera en fil med filnamnen för alla databanker i systemet. Om fil ej ges används /tmp/dblog. Följande returvärden kan förekomma:

0 - Filen genererad utan problem
-2 - Man måste vara inloggad för detta kommando
-3 - Kan inte öppna sysdb.
-4 - Kan ej skapa <fil>.

Kommandoexempel:

1. En kommandofil för att skriva ut hanterarens status:

```
mimer -s
if test $? = 0 ; then echo Hanteraren är igång
                 else echo Hanteraren är ej startad.
fi
```

2. Uppstartskommando för att kontrollera samtliga databanker och köra ut de databanker som är ok på en streamer tape.

```
mimer -l backup/backup -a -R -T /dev/st0
```

2. Information om påloggade användare

Detta kommando används när man vill ha reda på vilka användare som kör mot hanteraren (och även vilka program som körs). Följande information ges kolumnvis i en tabell:

Nr	Det interna användarnummret i hanteraren
Dnixanv	Det användarnamn man loggat in under i D-NIX.
Mimeranv	Det användarnamn som används för inloggning mot mimer.
program	Namnet på programmet som körs. "User appl." betyder att en ej identifierad användarapplikation kör.
process	Processnummer för det mimerprogram som körs. Denna information gör att man kan identifiera processen m h a kommandot **ps -lx**
hämta	Antal hämtningar som gjorts från databaser
läggtill	Antal nya poster som lagts till
tabort	Antal poster som tagits bort
ändra	Antal uppdateringar som gjorts
db	Antal databanker som hålls öppna.
tid	Hur länge användaren varit inloggad.

9. Administration

Detta kommando används för att komma till en undermeny med möjlighet att starta olika administrationsprogram mm. För att komma till undermenyn måste man dock logga in.

De ovan nämnda kommandona kan köras av vilken användare som helst. De övriga kommandona däremot kräver att man är inloggad som su dvs "Super User" eller "root".

3. Kontrollera Transdb, Logdb

Detta kommando används då man snabbt vill kontrollera storleken av de speciella databankerna transdb och logdb. För att utföra det måste man dock först vara inloggad.

5. Kontrollera databanker

Tyvärr kan det inträffa att en eller flera databanker blir mer eller mindre sönderskriven. Av den orsaken är det viktigt att man har ett förfarande för att regelbundet ta backuper av databankerna i systemet, samt att man har en möjlighet att verifiera att en databank är riktig.

Observera dock att det enda som kan kontrolleras är om samtliga indexträd i databanken är riktiga. Att data i databanken är riktigt kan ej kontrolleras på detta sätt.

Val 5 ger en möjlighet till kontroll av en eller flera databanker enligt några olika möjligheter:

* Kontrollera samtliga databanker i systemet

* Kontrollera ett givet antal databanker som specificerats på en speciell fil för detta ändamål.

* Kontrollera en enstaka databanksfil.

Först när man givit val 5 får man frågan:

```
Vill du kontrollera samtliga databanker (j/n):
```

Observera att denna fråga bara kommer upp om hanteraren går. Om man här svarar j, får man logga in (om man ej redan är inloggad) så att programet får möjlighet att läsa vilka databanker som finns i systemet. Programet listar sedan i en tabell Databank, storlek och status, dvs om databanken är felaktig eller inte.

Om man svarar n, får man ytterligare en fråga:

```
Vill du kontrollera en enda databank (j/n) ?
```

Om du här svarar n, får du också ge namnet på den fil, från vilken den fysiska databanken skall läsas (en databank per rad).

För att kunna kontrollera databanksfilerna måste hanteraren vara stoppad. (Annars kan hanteraren ändra filen medan man kontrollerar den). Om inga användare är inloggade i detta läge dödas därför hanteraren omedelbart. Om användare är inloggade ges frågan:

```
X användare inloggade.
      Vill du stanna hanteraren ändå (j/n) ?
```

Om man ger n, avbryts kommandot här, annars stannas hanteraren trots att användare är inloggade. (Man kan med val 2 titta vilka de användare som är inloggade är).

Loggen från /usr/mimer/backup/LOG har utseendet

```
....
------------------------------------------------------------------------
Check databanks from MIMER programm:                    87-09-08 10:13:13
/usr/mimer/usrdb/fiadbbu - ERROR (see file /usr/mimer/backup/ERR).
/usr/mimer/usrdb/fiadb - ERROR (see file /usr/mimer/backup/ERR).
/usr/mimer/usrdb/hoteldb                    - ok.
/usr/mimer/usrdb/instdb                     - ok.
/usr/mimer/usrdb/jec                        - ok.
/usr/mimer/usrdb/jectst                     - ok.
/usr/mimer/sysdb/logdb                      - ok.
/usr/mimer/usrdb/procdb                     - ok.
/usr/mimer/usrdb/sizetst                    - ok.
/usr/mimer/sysdb/sysql                      - ok.
/usr/mimer/usrdb/tabtst                     - ok.
/usr/mimer/sysdb/transdb                    - ok.
/usr/mimer/usrdb/workdb - ERROR (see file /usr/mimer/backup/ERR).
/usr/mimer/usrdb/xxx - ERROR (see file /usr/mimer/backup/ERR).
```

6. Visa Hanterarstatus

Kommandot används för att ge information om: När hanteraren startats, vilket hanterarbiblioteket är, fysiska filnamnet för sysdb samt viss information om den interna 'cash' bufferten i hanteraren:

Exempel:

```
INFORMATION OM MIMERHANTERAREN

Hanterar bibliotek   : /mimerdbl
Hanteraren startad   : 870908:10.24
System databank      : /usr/mimer/sysdb/sysdb
Storlek på 'cash'-arean : 32768

Antal sidor i arean        : 61
Antal sidor för sortering  : 57
Antal sidbeställningar     : 0
Antal 'page faults'        : 0          ( 0.0%)
Antal sidor till skivminne : 0
Antal transaktioner        : 0
Antal utförda transaktioner: 0

Antal påloggade användare  : 0

                Tryck <CR> för att fortsätta...
```

4.8.2 Administrationsmenyn

För att administrera ett mimersystem finns ett antal små program som dbadm och dbtab. I alla dessa måste användarnamn och lösenord ges. För att slippa detta har en meny i Mimerprogrammet skapats med val för de olika funktionerna. Inloggning sker bara när man går till undermenyn för administration.

Menyn har följande utseende:

```
DATUM: 87-09-08          D-MENY              PROGRAM: mimer
ANVÄNDARE: JEC
                         MIMER

                Mimer Databas Administration

  Administrationsprogram                      Mimer/QL
  1. Databasadministration (dbadm)            8. QL
  2. Backup hantering (dbbru)
  3. Tabellhantering (dbtab)                  Systemkommandon
  4. Exportera/Importera (expimp)             9. D-NIX
  5. Minifrågespråk (MQ)                     10. Lista filer
  6. Visa versioner                          11. Visa innehåll i fil
  7. Kontrollera Mimermoduler                12. DE editor
                                              0. Tillbaka till huvudmeny

       'ESC <' för Huvudmeny            Ctrl-W för att avsluta
```

1.-5., 8. Mimermoduler

Dessa val används för att starta olika Mimermoduler. Ingen inloggning behöver ske. Då modulen avslutas kommer man tillbaka till menyn.

6. Visa versioner

Detta kommando används för att fråga vissa administrationsprogram vad de har för versionsnummer.

```
VISA VERSIONER (DB)

Mimer DBADM v.3.2.8(.1)     (861020)
Mimer DBBRU v.3.2.8(.1)    (861023)
Mimer EXPIMP v.3.2.8(.1)   (861111)
DBTAB v.3.2.8(.2) (870409)
```

4.9 Dbcopy

Hjälpprogram för att kopiera tabell-definitioner och tabell-innehåll från en databank till en annan. Kan användas för att rädda en databank där någon tabell blivit fördärvad, eller för att definiera upp tabeller i en ny databas.

Normalt förutsäts att både tabell-def och data flyttas. Om (vid -d) en rad redan finns, sker update. Default-aktionerna kan ändras med parametrar.

Utrymme finns för att göra projektioner för max 502 byte, vilket är maximal radlängd med block a 2 kbyte. Man kan gör max 10 selectioner. Maximal selectionslängd är 19 tecken.

PARAMETRAR

```
-d    definiera inte tabellen i den nya databasen
      (redan definierad t.ex. med expimp)
-c    kopiera inte tabell-innehåll
-u    gör ingen update om kollitioner.
-l    inloggning med <user>/<password>
-p    skapa ingen loggfil 'dbcopy.log'
-s    gör ingen kopiering utan räkna poster
      i angiven databank
-V    versionsnummer
```

Efter inloggning mot Mimer anger man från databank samt till databank. Databankerna skall finnas definierade. Därefter fås följande fråga för varje tabell i databanken:

```
    Doit with <table>? (Yes/No/Part/Rest/Help/TAB/Quit)

Yes  Hela tabellen kopieras

No   Tabellen kopieras ej

Part Del av tabellen kopieras.
     Detta val ger möjlighet att sätta upp sökvillkor.
     LOP kan vara F = first (första villkor)
                  A = and   ( och )
            eller O = or    ( eller )

              ! avslutar villkoren.

     Column: Ange önskad kolumn
     ROP : önskat villkor ( GE, LE ....)
     Value: Ange sökvärdet
Rest Resten av databanken kopieras.
```

5. Backup

5.2 DBBRU backup

Hjälpprogrammet 'dbbru' kan användas för att skapa backupkopior på skivminne. Om backupkopian lagras på samma fysiska skivminne som originalet kan du förlora båda databankerna vid skivminneshaveri. Om 'dbbru' används skall backupkopian på databanken lagras på ett annat skivminne.

5.4 Backuprutinen

Filen /usr/mimer/backup/backup innehåller shellrutiner som kan användas för backup av mimer databanker eller som ett exempel på hur du skriver egna rutiner.

Om du gör backup på systemet varje natt så kan du lägga till kommandon för att ta ner hanteraren och kontrollera databankerna i början på din backuprutin. I slutet lägger du till ett kommando för att kopiera den tomma logdatabanken till logdb.

Backuprutinen utförs enligt följande:

1. Om db1m-hanteraren går skall den stoppas.

2. Kontrollera databankerna med mdbchk.

3. Kontrollera att det inte var några fel i databankerna. Finns det en fil /usr/mimer/backup/ERR ?

4. Gör backup med tar.

5. Starta hanteraren igen.

Anmärkning: Transaktionsdatabanken, transdb, kan rensas genom att kopiera

```
/usr/mimer/sysdb/transdb.empty
```

till

```
/usr/mimer/sysdb/transdb
```

Detta måste utföras då hanteraren inte går.

För att använda backup-rutinen /usr/mimer/backup/backup ska du göra en kopia av filen och placera denna i biblioteket /usr/adm.

Dessutom måste följande shell variabler vara initialiserade enligt dina behov.

SELECT Filerna och/eller biblioteken som skall säkerhetskopieras.

DBFILES Databanksfilerna i Mimer version 3 som skall kontrolleras med **mdbchk**.

RMFILES Temporärfiler som skall tas bort.

```
DBCHKDIR="/usr/mimer/backup"
BACKUPLOG="¤LOGDIR/backup.LOG"
WRITELOG="¤LOGDIR/write.LOG"
READLOG="¤LOGDIR/read.LOG"
ERRORLOG="¤LOGDIR/error.LOG"
RMFILELOG="¤LOGDIR/rmfile.LOG"
RMTMP="/tmp/rmtmp¤¤"
MOTD="/etc/motd"

SYSID=`uname -n ö tr Äa-zÅ ÄA-ZÅ`
TMPFIRST="/tmp/"¤SYSID"_first¤¤"
TMPLAST="/tmp/"¤SYSID"_last¤¤"

DEVICE="/dev/st0"
BLOCK="300"
VOLSIZE="60000"
#
# Test mode
# ---------
if ¤TEST
then
        SELECT="usr/mimer/sysdb/sysdb"
        RMFILES=""
        DEVICE="tmp/st0"
        rm ¤LOGDIR/*.LOG
fi
#
# Reset files
# -----------
if Ä -r ¤MOTD".org" Å
then
    cp ¤MOTD".org" ¤MOTD
else
    cp ¤MOTD ¤MOTD".org"
fi
rm -f ¤ERRORLOG
rm -f ¤DBCHKDIR/ERR
>¤WRITELOG
>¤READLOG
echo "`date` Start backup." >¤BACKUPLOG
#
# Create label files
# ------------------
echo "START backup på maskin ¤SYSID `date`" >¤TMPFIRST
echo "SLUT  backup på maskin ¤SYSID `date`" >¤TMPLAST
#
# Stop MIMER handler if running
# -----------------------------
```

```
    STAT=¤?
    if Ä ¤STAT -eq 0 Å
    then
        echo "`date` Backup klar och OK." >>¤BACKUPLOG
        echo "`date` Backup klar och OK." >>¤WRITELOG
    else
        echo "`date` FEL nr ¤STAT vid backup." >>¤BACKUPLO
        echo "`date` FEL nr ¤STAT vid backup." >>¤ERRORLOG
    fi
#
# Readcheck backup
# ----------------
    echo "`date` Kontroll start." >>¤BACKUPLOG
    echo "`date` Kontroll start." >>¤READLOG
    tar tvfbsS ¤DEVICE ¤BLOCK >>¤READLOG 2>&1
    STAT=¤?
    if Ä ¤STAT -eq 0 Å
    then
        echo "`date` Kontroll klar och OK." >>¤BACKUPLOG
        echo "`date` Kontroll klar och OK." >>¤READLOG
    else
        echo "`date` FEL nr ¤STAT vid kontroll." Ö
              >>¤BACKUPLOG
        echo "`date` FEL nr ¤STAT vid kontroll." Ö
              >>¤ERRORLOG
    fi
#
# Check if first and last label is present
# ----------------------------------------
    if Ä `grep -c ¤TMPFIRST ¤READLOG` = "0" Å
    then
        echo "`date` START-etikett saknas på backupen." Ö
             >>¤BACKUPLOG
        echo "`date` START-etikett saknas på backupen." Ö
             >>¤ERRORLOG
    fi
    if Ä `grep -c ¤TMPLAST ¤READLOG` = "0" Å
    then
        echo "`date` SLUT -etikett saknas på backupen." Ö
             >>¤BACKUPLOG
        echo "`date` SLUT -etikett saknas på backupen." Ö
             >>¤ERRORLOG
    else
        set `grep ¤TMPLAST ¤WRITELOG`
        echo "`date` Storlek på backupen : ¤3 Ö
             ("¤VOLSIZE"K)" >>¤BACKUPLOG
    fi
#
```

```
then
    echo "Ö007KONTAKTA SYSTEMANSVARIG OMEDELBART !!" Ö
          >>¤ERRORLOG
    mv ¤MOTD ¤MOTD"-"
    cp ¤ERRORLOG ¤MOTD
else
    if Ä "¤RMFILES" != "" Å
    then
        echo "`date` Start borttagning av filer." Ö
             >>¤BACKUPLOG
        echo "BORTTAGNA FILER `date`" >¤RMFILELOG
        echo "Fri-dagar    : ¤SAVEDAYS" >>¤RMFILELOG
        echo "Sökta filer : ¤RMFILES" >>¤RMFILELOG
        for RMFILE in ¤RMFILES
        do
           find / -name ¤RMFILE -atime ¤SAVEDAYS -exec
           rm ää Ö;  -print >>¤RMTMP 2>&1
        done
        if Ä "`cat ¤RMTMP`" != "" Å
        then
            cat ¤RMTMP >>¤RMFILELOG
            rm ¤RMTMP
        else
            echo "Inga filer hittades." >>¤RMFILELOG
        fi
        echo "`date` Slut borttagning av filer." Ö
              >>¤BACKUPLOG
    fi
fi
#
# Cleanup
# -------
rm ¤TMPFIRST ¤TMPLAST
echo "`date` Backup slutförd." >>¤BACKUPLOG
```

6. Att använda Mimer/QL

6.2 Exekvera D-NIX kommandon från QL

Det är möjligt att exekvera D-NIX kommandon från QL genom att:

a) Skriva QL> COMmand;

Du får nu upp en shell-prompt ¤ och kan exekvera D-NIX kommandon som vanligt.

Om du vill återgå till QL, trycker du bara <CR> på en tom rad.

b) Skriva QL> COMmand 'D-NIXcommand';

detta kommer att exekvera ett D-NIX kommando och sedan återgå till QL.

Anmärkning: Det bästa sättet att exekvera flera shell-kommandon i ett undershell är genom att använda:

```
QL> COM 'sh';
```

Exempel:

```
QL> COM 'ps lax';     ( För att få process status).
QL> COM 'l ö print'; ( Skriver ut filerna i det
                        aktuella biblioteket ).
```

6.4 Buffertar i QL

De interna buffertarna i QL kan kontrolleras med

```
QL> sho buf;
```

Om det uppstår problem med utrymme för tabellerna, försök då att undvika att öppna kompletta databanker. Öppna istället bara de tabeller från databanken som behövs. Detta sparar inte enbart utrymme utan även tid.

Exempel:

```
QL> inc dbl(name,adress);
```

7. Att använda Mimer/SH

7.2 Filer och terminaltyper

Ett antal filer behövs för att köra SH program:

/usr/mimer/sh/shfiles/shterm	-	Terminaldefinitionsfilen som används då programmen körs.
/usr/mimer/sh/shfiles/shlang	-	Språkfil som används av kompilatorn.
/usr/mimer/sh/bild/sheobj	-	Bildobjektfil som används av editorn.

Terminaltypen läses in från variabeln TERM. Den kan sättas med:

```
¤ TERM=vt100
¤ export TERM
```

Terminaltyperna som känns igen av Mimerbiblioteket är:

```
comex
vt52
vt100
tandberg 2115
tandberg 2215
vc404
vc414
adm3a
twist
```

7.4 Tillämpningar vid länkning

Biblioteket som innehåller alla Mimer/SH tillämpningsrutiner är /usr/-lib/libMsh.a, refererad till av '-lMsh' i länkningen.

Om du inte har några kontrollrutiner, kan en standardversion av 'coupling' filen, användas från biblioteket.

Exempel:

```
¤ f77 apll.f -lMsh -o app          (Inga kontrollrutiner)
¤ cc appl.c bild.o -lMsh -o app (Kontrollrutiner i
                                  bild.o. tillämpningsp-
                                  rogram i appl.c)
```

```
      while ( readall(MENY)== 0)
       ä
          getsh2_("MENY:VAL ",&val,"I");
          switch(val)
            ä
                 case 1:  prompt();
                          break;

                 case 2:  check();
                          break;

                 case 3:  edit();
                          break;

                 case 4:  endsh2_();
                          exit(0);

                 default: break;
              å
           clinit(MENY); /*Clear screen and rewrite MENY*/
        å
     endsh2_();
å

 prompt()

  ä
     char datumÄ20Å;

     iniwrp(PROMPT);

     for(;;)
      ä
         clrfld(PROMPT); /* Clear all fields */
         mdrpdt_(datum); /* Read date ... */
         putsh2_("PROMPT:REG     ",&datumÄ2Å,"C");
                                /* ... and write to
                                   picturefield */
         wrfsh2_(PROMPT,"REG     ");

         if ( readall(PROMPT) == 2)
           ä
             /* Break char entered */
             rmpsh2_(PROMPT); /* Remove picture */
             return(0);
           å
      å
  å
```

```
/*-------------------------------------------------
 *
 *  Example of a C/MimerSH-interface.
 *
 *-------------------------------------------------
 */


int statusÄ4Å;

shinit(p)

char *p;

ä
   int term=0;
   inish2_(&term,p);
å

clrscr()

ä
  int i0=0;

  /*  Clear the whole screen */

  clssh2_("WH",&i0,&i0,&i0,&i0);
å

iniwrp(p);

  ä
     /*  Clear screen
         Init picture p and clear all fields */

     clrscr();
     inpsh2_(p);
     inash2_(p);
     wrpsh2_(p);
     clfsh2_(p,"* ");
  å

clinit(p);

  ä
     clrscr(); wrpsh2_(p);
     clfsh2_(p,"* ");
  ä
```

Exempel 2:

Detta är ett exempel på en kontrollrutin skriven för programmet i föregående exempel. Kontrollrutinen anropas när rutinen CHECK exekveras.

```
/*-------------------------------------------------
 *
 *   Mimer/SH CHECK ROUTINE
 *
 *     The data entered into the field to be checked is
 *     a date of the form 850529
 *
 *-------------------------------------------------
 */

int dmÄl3Å = ä 0,31,29,31,30,31,30,31,31,30,31,30,31å;

dd(ioa,len,stat)

int *len,*stat;
char *ioa;

ä
  int mm,dd;

  *stat = 0;
  mm = mdrcic(&ioaÅ2Ä,2);  dd = mdrcic(&ioaÄ4Å,2);

  if ( (mm<0)öö(mm>12) ) ä

          /*  Error in field. Write message and
              return stat = -1 to tell SH to reread
              the field. */

                          msg(1,"Wrong Month.");
                          *stat = -1;
                          return(1);
                        å
  if ( (dd<1)öö(dd>dmÄmmÅ)) ä
                        msg(1,"Wrong day in month.");
                          *stat= -1;
                          return(1);
                        å

 å

msg(nr,p)
```

8. Att använda Mimer/PG

8.2 Databanker som används av Mimer/PG

Alla Mimer/PG användare måste ha en privat databank med samma namn som sitt Mimer användarnamn. Användaren skall ha X-åtkomstprivilegier till sin egen databank.

Alla Mimer/PG felmeddelanden och hjälptexter finns i databanken SYSPG vilken fysiskt finns i

```
/usr/mimer/pg/syspg
```

Alla användare måste ha R-åtkomstprivilegier till SYSPG databanken.

8.4 Skapa program

När du laddat ditt program och interaktivt kontrollerat dess funktion, är det dags att skapa ett Fortran/Cobol program.

Det är möjligt att generera kod för olika datortyper, men vanligtvis genereras koden för den egna datorn, i det här fallet DS90. Du skriver bara:

```
PG> SET LAN FORTRAN DS_90;
PG> GENERATE TEST;
```

Här är TEST ditt program.

Om programmet är en undermodul skriver du istället:

```
PG> GENERATE TEST NOMAIN;
```

När genereringen är klar är du tillbaka i pg-kommandofilen nämnd i föregående avsnitt. Du måste svara på några frågor:

```
1) Is this a main module (Y/N) ?
```

Om du svarar 'Y' läggs ett huvudavsnitt in i början på den genererade koden.

```
2) Do you want to compile the fortran file (Y/N) ?
```

Om du svarar 'Y' skall du ange namnet på den exekverbara fil som skall genereras vid länkningen.

```
3) Give couplingfile and subroutines:
```

Här skriver du namnen på de objektfiler som skall inkluderas i länkningen. Om du har någon kontrollrutin måste du även ange namnet på 'coupling' objektfilen. Om du inte har några CALL-satser eller kontrollrutiner i ditt program trycker du bara på <CR>.

8.6 Vanliga problem

I nuvarande version av PG finns det ett känt problem:

* Om du använder 'Ö' i dina namn på databanker, tabeller eller fält, kommer du att få ett syntaxfel i F77 kompilatorn.

8.8 Exempel på en PG session

```
¤ mimpg

            ============================
            M I M E R   P G  O N  D S 9 0
            ============================

            Date to day : 1987-03-11

            PG generated: 1986-10-01

    Wait.... Loading Mimer PG

                    ********************
                    * M I M E R / P G *
                    *    v.3.2.12      *
                    ********************

Username: jec
Password:

PG> load file test;

INC REG;
DES TAB FLYT;

REG     FLYT    PRIMT* C4
                INT    I4
                FLYT   F4

PROGRAM TEST=GET FLYT.*;
...

PG> exec test;

PRIM INT        FLYT
ett      1       1.53E-11
två      2      -5.68E7

2 Row(s) found!

PG> lang fortran ds_90;
PG> generate test;
Wait...
Fortran code is being generated.
Fortran code is being written.
```

A. Fortran kompilatorn

Nedanstående exempel gäller för Fortran 77 under D-NIX 5.12. För information om Fortran 77 under D-NIX 5.2, se Fortran handboken.

1. Startparametrar

`-c`	Ingen länkning. Skapar bara en objektfil (.o)
`-o utfil`	Den exekverbara filen från länkningen kommer att kallas 'utfil' och inte 'a.out' som standard.
`-V`	Kommentarer i alla pass skrivs ut.
`-w`	Skriver inte ut några varningar.
`-w66`	Undertrycker alla F66 kompatibilitetsvarningar.
`-S`	Tar inte bort assemblerkällfilen (.s)
`-u`	Standardtypen på en variabel definieras inte.
`-I2`	Gör alla heltal 2 bytes istället för 4 bytes.
`-C`	Kontrollerar matrisgränser under körningen.
`-Ntddd`	Allokerar mer utrymme till tabeller. Används när 'front end' ger fatala kompileringsfel. t är q,x,c,n (för equivalence,external name, control-structure and name table) ddd är det nya antalet element i tabellen.

2. Fortranenheter

5 - Standard input
6 - Standard output
x - filenhet öppnad i en OPEN-sats.

3. Starta kompilatorn

En källfil i FORTRAN måste ha extension .f. Om inte -c används kommer länkaren att anropas när alla filer är kompilerade.

Exempel:

```
f77 -V pt.f coup.o -o pt >comp.list 2>&1
```

B. C-kompilatorn

Cc är C-kompilatorn. Argument vars namn slutar med '.c' är C källprogram. Argument vars namn slutar med '.s' är assembler källkod och assembleras. Utfilen i båda fallen blir av typen '.o'.

Inparametrarna är:

`-c`	`Undertrycker laddningsfasen.`
`-O`	`Startar objektkodoptimeraren.`
`-S`	`Ger assembler källfiler.`
`-w`	`Undertrycker varningar.`
`-o fil`	`Skapar en utfil kallad 'fil'.`
`-ldir`	`Använder biblioteket /usr/lib/lib'dir'.a vid länkning.`

Det finns även ett antal tillval som kan användas vid länkning (av laddaren):

`-x`	`Spara inte lokala symboler. (Kan användas om du får overflow i symboltabellen).`
`-n`	`Separera kod och data.`
`-s`	`Strip the executable file.`

C. Mimer Basic

Det är möjligt att hantera data i Mimertabeller från D-BASIC V. För en beskrivning se 'D-BASIC V Manual' kapitel 9.

Sats	Beskrivning
MIMER BEGIN	Starta en Mimer session.
MIMER OPEN	Öppna en databank/tabell.
MIMER GETFIRST	Hämta första raden.
MIMER GETNEXT	Hämta nästa rad med avseende på select condition.
MIMER WRITE	Lägg till en ny rad.
MIMER UPDATE	Uppdatera en rad.
MIMER DELETE	Ta bort en rad.
MIMER TRANSACTION	Starta en Mimer transaktion.
MIMER COMIT	Slutför transaktionen.
MIMER ABORT	Avbryt transaktionen.

Exempel:

```
100 MIMER OPEN "USER     ",Pword¤
170 MIMER OPEN "DB1.car" as file 1 "regnr","model",
    "color","year"
190 MIMER GETFIRST #1,"regnr" EQ regnr¤,
    reg¤,model¤,color¤,year¤
```

D. Vanliga problem

Kan inte få kontakt med Mimerhanteraren

Har hanteraren startat ?

Använd ps lx för att kontrollera att processen 'db1m' går. Om inte starta hanteraren med 'startdb'.

Kan inte skapa tabeller med 'dbtab'

Har du X privilegier för den utvalda databanken ?

Kontrollera med 'dbadm'. Om inte definiera X privilegier.

Kan inte logga in till Mimer

1. Går hanteraren ? (ps lx)

2. Har systemdatabanken skapats ? (sdbgen) (Anmärkning: Att köra 'sdbgen' förstör den gamla systemdatabanken)

3. Är systemdatabanken korrekt ? (kör 'dbchk' på /usr/mimer/sysdb/sysdb)

4. Är användaren definierad ? (dbadm)

Utvärdering av dokumentation:

Mimer 3.2 Användarhandbok (A)

Vi är mycket intresserade av vad Du tycker om dokumentationen och hur Du tycker att den skulle kunna förbättras. Vi vore därför mycket tacksamma om Du använde några minuter av Din tid för att besvara frågorna nedan och returnera detta papper till oss.

1. Är det lätt att hitta den information Du söker? Är det någon information Du behöver som Du inte kan hitta?

_____

2. Är texten lättläst och lätt att förstå? Ange gärna sida och stycke om Du har något exempel på svårförståeligt avsnitt.

_____

3. Är uppbyggnaden av dokumentationen logisk? Bör några avsnitt byta plats, flyttas eller liknande?

_____

4. Finns det tillräckligt många exempel och beskriver de rätt delar av informationen?

_____

5. Hur tycker Du att dokumentationen skulle kunna förbättras, ge gärna exempel?

_____

_____

_____

_____

_____

Namn:_____ Företag: _____
(Frivilliga uppgifter)

På baksidan av detta formulär finns vår adress förtryckt. Vik formuläret på mitten och häfta eller tejpa ihop det så kan det skickas som det är med posten.

Tack för hjälpen!