CHD

Title:

TELEDATA UTILITY PROGRAM

TELEREES

A/S REGNECENTRALEN

RC SYSTEM LIBRARY: FALKONERALLE 1 DK-2000 COPENHAGEN F

RCSL No: 21-T019

Edition: June 1978

Author: Ivan Larsen

Keywords:

RC8000, Teledata, database, recovery, reestablishing, reprocessing.

Abstract:

The manual describes the use of the program Telerees for reestablishing of a Teledata dabase. The safety/ recovery measures in the Teledata system are summarized.

Printed by A/S Regnecentralen, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

Contents Page

Preface 4

References 5

Introduction 6

2.1 Examples 7

2.2 Call 8

2.3 Recovery in Teledata version 3 10

2.3.1 Security measures 11

2.3.2 Recovery techniques 13

2.3.3 Administration of versions 14

2.4 Function 15

2.4.1 Program function 15

2.4.2 Error reactions 19

2.4.3 Result 20

2.5 Requirements 20

2.6 Output 20

2.9 Error messages 22

Preface

Version 1.1 is the first version of the Teledata reestablishing program Telerees. The program comprises the basic tools for a complete reestablishment of a Teledata version 3 database.

At present some optimizing measures are being implemented in the on-line system. Similar measures will be implemented in a later version of Telerees.

In this first version of the program, error situations (e.g. rejection of otherwise legal database operations) are managed in what may later turn out to be a too rigid way. Some rejected operations could possibly be accomplished in the on-line system. Similarly the restoration of text files may be omitted in some cases because the file positions are extracted from records in other files. The application of the restrictive solution is preferred until some experience with Teledata version 3 systems in service has been gained.

References

1.	Teleop	RCSL No: 21-T002
2.	Teledata Utility Programs	RCSL No: 21-T008
3.	The Teleadm Model	RCSL No: 21-T016
4.	SODA	RCSL No: 21-V019
5.	Connected Files System	RCSL No: 28-D5

2. Telerees

Telerees is intended for off-line reestablishment of a Teledata database. File types processed are:

cf	files
text	files
sq	files

Any of these file types may be selected in the call of the program.

The program scans the log file, starting at the checkpoint as specified in the status file, and moves the logged database record copies into a validated version of the database (safety copy). That is, all the modifications (insert, update and delete) performed since the latest checkpoint creation are repeated.

The program terminates when the updatings corresponding to the latest checkpoint creation are repeated.

The program terminates when the updatings corresponding to the latest finished transaction are performed.

Telerees requiries a valid db80, soda_ld database description file.
The log file may be located on disc or tape.

2. Telerees

2.1 Examples

Suppose a standard version 3 Teledata database for some reason must be reestablished. The database and soda_ld description is found in a file named 'descripfile', which is the default name of the binary database description file.

Notice that a valid version of the database must be loaded prior to the call.

Ex. 1

```
    telerees status.statusfilex ldsect.103.1 select.cf.sq
```

The program extracts the name and start position of the log file from a status file named 'statusfilex'. Information concerning the actual database is found in the description file ld_section 103.1. Records belonging to cf files and sq files (terminal work files) are selected when the log file is scanned, and written into the database.

2. Telerees

2.2 Call

The program is called by the fp command:

```
           ⎛                               ⎞ *
           ⎜ status.<filename1>            ⎟
           ⎜ descrip.<filename2>           ⎟
           ⎜ ldsect.<section>              ⎟
           ⎜        ⎧ no  ⎫                ⎟
           ⎜ tape.  ⎨     ⎬                ⎟
           ⎜        ⎩ yes ⎭                ⎟
           ⎜          ⎧ text ⎫ *           ⎟
           ⎜ select.  ⎨ sq   ⎬             ⎟
           ⎜          ⎩ cf   ⎭ 1           ⎟
telerees  <  oplog.<integer>               >
           ⎜           ⎧ no   ⎫            ⎟
           ⎜ testlog.  ⎨ yes  ⎬            ⎟
           ⎜           ⎩ only ⎭            ⎟
           ⎜        ⎧ no  ⎫                ⎟
           ⎜ test.  ⎨     ⎬                ⎟
           ⎜        ⎩ yes ⎭                ⎟
           ⎜                               ⎟
           ⎜ testout.<filename3>           ⎟
           ⎜                               ⎟
           ⎝                               ⎠ 1
```

where:

status. specifies the file <filename1> to be the current status file.
The default name of the status file is 'statusfile'.

descrip. specifies that the file <filename2> contains the binary database description file. If no descrip command is given the default name 'descripfile' is used.

2. Telerees

ldsect. this command being the only command mandatory in the call specifies the ld section in the database description file.

<section>::= <sysdoc section number>.

tape. the tape.yes command may be used (logmode 3 only) to force the program to read the log file from tape. Otherwise the disc log automatically is selected.

select. if this command is given only the file types specified are restored. If no select command is given all types are selected.

oplog. for every <integer> transactions processed the running program displays a message on the console (for the impatient operator). Default an initial message only is displayed.

testlog. intended for test and debugging purposes. If testlog.yes or testlog.only are specified a listing of the contents of the log record heads is generated in a test area. With testlog.yes the listing is performed during the normal reestablishment. With testlog.only no reestablishment is performed, but status file and log label are checked and all log record heads are listed. For each log record 1 line of text is output.
Default testlog.no

2. Telerees

test. intended for test and debugging purposes. If test.yes is specified a trace of the database operations executed during the reestablishing is printed in the test area. If testlog also is specified 1 line is printed for each database operation, if not 2 lines are printed.
Default test.no.

testout. specifies <filename3> to be the name of the test area used for test and testlog. If no testout command is given and a test- or testlog command is given the document connected to zone out is used for test output. The test output being intermixed with messages from the program.

2.3 Recovery in Teledata version 3

This section summarizes the security measures and recovery tecniques used in the Teledata system. Notice that the term security used in this context only denotes security against data loss.

The types of errors considered are errors with inpact on the database consistency only That is, hard errors occurring during output operations and errors caused by media destruction. Reference are made to some results of experience with the old version of the Teledata system.

Further information concerning the log facility may be found in the manuals for Teleop, Telestatus and SODA.

2. Telerees

2.3.1 Security measures

The recovery of the database is based on:

- safety copies of the complete database (dump),
- the logfiles: reprocessing log and reestablishing log
- the status file
- creation of checkpoints.

No automatic generation of checkspoints or activation of error recovery (roll back) is provided.

The reprocessing log contains all the transactions received by the system and nothing else.

The reestablishing log contains the transactions received, accounting information etc. and copies of all the records stored in the database since open_day. These record copies are extracted by the reestablishing program and restored in the database.

The reestablishing log (hereafter called 'the log') may be located on disc or tape or both, controlled by the log mode. But whenever possible it should be placed on a disc, because a tape drive will slow the on-line system down.

The log file may grow to a size comparable to the database so for the sake of space, security and speed it should not be placed at the kit holding the database.

2. Telerees

A checkpoint comprises:

1. save (safety copy) and identification of a validated version of the database.
2. save of the current position on the logfile (see Telestatus)

An implicit checkpoint is always created at close_day/ open_day. Explicit checkpoints may be created during the daytime. At present it is necessary to close the system (Teleop), so in order to establish a frequency of checkpoint creation two factors must be considered:

a. close_down_time necessary for copying of the database disc to disc and restart of the system. Dump on tape will hardly be tolerated in the day-time. The close-down-time is independent of the hour of the day.

b. the time required for reestablishing and the probability for break down. The reestablishment time is proportional to the interval between ckeckpoints.

Experience with systems in use shows that a well running Teledata system should possess and average down-time caused by break down of less than 1 percent of the open time. In that case no checkpoints are created in the day-time (center policy).

After reestablishing a checkpoint always shoul be created.

2. Telerees

A save must comprise all files in the database including the terminal work files (wrkfilxx) and the textfiles (not reply areas). Notice that the system is not provided with an automatic administration system for the work- and text files.

2.3.2 Recovery techniques

According to the security measures outlined above 3 possibilities for restart after break down exist.

Reprocessing of all transactions processed since open-day. Teleop reads (off-line) the transactions from the reprocessing log exactly as they were received from the terminals. The technique is slow and situations not manageable off-line may arise. Should be used only if log- or status file are damaged.

Reestablishment of the database by restoring the logged record copies. The method is safe because the records to be stored are logged after check- and store operations in the on-line system. Reestablishing is an order of magnitude faster than reprocessing.

Immediate restart of Teleop is possible if it is assured that no database update operation was going on when the break down occurred. (transcur = transfin, see telestatus). Notice that logmode 3 is required.

2.3.3 Administration of versions

If it important to realize that exactly one verion of the database, one version of the status file and one version of the log file form an entity.

In order to make it possible to check the file versions before start of the reestablishment, version numbers must be maintained by the system. As version identification a 'shortcloc' value i used, but for the sake of simplicity let version denote an unambigous figure as used in the following proposal for a version administration model valid for disc log.

At closeday:

Save log- and status file on tape. Move the validated database files (cf-, wrk- and textfiles) to the safety locations on (another) disc.

Set version (current shortcloc) into catalog entries of all database files and of the corresponding safety copies and into the dbversion entry in the status file.

All database files may be dumped on tape.

At openday:

Check version consistency for all files. Create a status file version by setting 'shortcloc' into the date entry in the status file, thus identifying the status file by the start-up time.

The log file start addresses, transcur, transfin, logrecs and checkpoint are set to zero.

2. Telerees

Teleop moves dbversion and date from the status file to a label record of the log file thus completing the entity: database, log and status file.

At reestablishment:

för [illegible]

[illegible]

"dbversion"
<>
"date"
(i status file)

Check version of disc safety copies of the database against dbversion entry in the status file. Move the copies from the safety locations. Do not load a saved status file.

The reestablishing program checks the status file version and the dbversion entry in the status file against the information in the label record of the log, thus ensuring the entity: database, log and status file.

NY.

After a succesful reestablishment create a checkpoint.

start-up:

identify status file and start Teleop as in openday. Do not clear start addresses of log etc.

2.4 Function

This section summarizes the program function and outlines the error reactions and result.

2.4.1 Program function

The program may logically be divided into the following functions, some of which will be further explained.

2. Telerees

- read and check parameters
- read and check status file
- read and check log label

- initialize and connect zone buffer for database files
- scan log file and restore records
- update status file.

Status file

The basic information for control of the reestablishment is found in the status file.
The safemode must be 2 or 3. According to logmode (and a possible tape command) the position and the name of the log file are extracted. If the log file is on tape it is the name of the first tape. In that case the name will be send in the mount message to the console, when the logfile is positioned.

The contents of the entry 'logrecs', equal to the number of log records corresponding to the finished transactions, control the termination of the reestablishment.

In the normal case the current transaction no. (transcur) will be equal to the finished transaction no. (transfin) plus 1. If this is not the case a warning is displayed, but the program continues.
If the reestablishment is terminated properly transcur is set equal to transfin by telerees.

2. Telerees

Log file:

The log is read from the document specified in the status file. The first record is a label containing version information (see section 2.3.3) and, in the case of tape log, the tape name.

It is checked that the date (log version) and the dbversion in the log label is equal to the date (status version) and dbversion respectively in the status file. In case of version error the run is terminated, unless the program is called with testlog.only.

Initialization of buffers:

Buffers for the cf- and sq files are allocated and connected by the initialization procedures of the SODA system (initsoda1 and initsoda2). It is checked that the ld_version number carried in the log label record is the valid one.

The SODA procedures are not used for any database operation, but the files are closed by the procedure closesoda.

Restoring:

The log is read sequentially and records from cf-, sq and text files are processed according to file type.

2. Telerees

Records from text- and sq files (wrk) are extracted from the log and always stored directly at the address (segment, halfword) as specified in the head of the log record.

Records from cf files are processed according to record type and operation.

Master records.

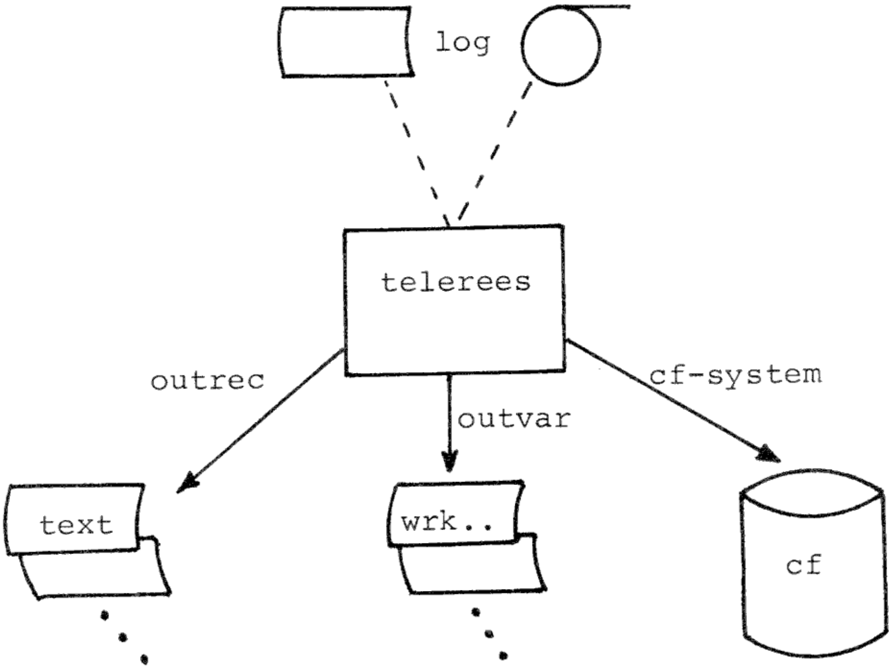
Insert: insert complete record from log
Update: get record (key from log), move contents from log and put.
Delete: get record (key from log) and delete

List records.

Insert: insert complete reoord from log after positioning in chain.
Update: get record directly (recno from log), move contents from log an put.
Delete: get record after positioning in chain, check recno and delete.

2. Telerees

fig. 2.4.1



2.4.2 Error reactions

After check of status file and log label in principle two types of errors may be detected.

Hard errors caused by alarms from the cf-system (unreadable files or chains) will cause the program to terminate.

Restore errors denote rejections of otherwise legal database operations (e.g. insertions). If this type of error is detected for cf files, the run is terminated. (This solution may be to rigid, because som restoration could be accomplished by on-line transactions later).

2. Telerees

Breaks received during writing in text- and sq files caused by lack of resources or by nonexisting files will not terminate the program. These files may be restored separately in a second run.

2.4.3 Result

Errors preventing a complete reestablishing will affect fp 'errorbits'.
Harderrors and restore errors wil cause ok.no.

2.5 Requirements

Size:	as large as possible.
Area:	min 2 + number of files accessed.

2.6 Output

After the start message the following output will appear on current output during a normal run.

2. Telerees

After read and check of status file:

```
Statusfile: <filename1>                       version=<date>
            logmode = <logmode>               logrecs=<logrecs>
            transfin = <transfin>
            logfile: <filename>
               point: { discseg=<segment   > dischw=<halfword   > }
                      { tapefile=<file  >    block=<block  >      }
            dbvers = <dbversion>
```

After read and check of log label:

```
Logfile :    <filename>                         version=<log date>
             records expected <no. of records to read>
```

During reestablishing (according to oplog command):

```
from telerees: <n> transactions finished
               time elapsed: cpu=<seconds> real=<seconds>
```

After termination:

```
telerees, termination {not} ok, warnings=<integer>
  transactions counted = <no of finished transaction in processed
                          section of log>
  log records counted  = <no of records processed>
  last log record no   = <record no from latest log record head>
  last log transac no  = <transac no from latest log record head>
```

2. Telerees

2.9 Error messages

Telerees may generate errormessages during execution of the functions outlined in section 2.4.1. The format and contents of the messages are shown below. Some knowledge of the cf system is required.

Parameter reading

```
         ***telerees params.: { illegal structure          }
                              { unknown keyword or subkey  }
                              { <identifier>               }
```

Syntax eror.

In any other case the first line of the messages has the format:

```
         ***telerees,   <explanatory text>
```

Hereafter only the <explanatory text> part will be shown.

Parameter check

```
         no ldsection specified
```

The ldsect command is mandatory.

```
         testarea does not exist   <lookup result>
```

2. Telerees

A nonexisting file is specified in the testout command.

Check of status file

```
    statusfile does not exist
```

```
    warning: transcur <> transfin + 1     <transcur>
```

See section 2.4.1.

```
    logmode inconsistent
```

The logmode is not 1, 2 or 3, or tape command given for logmode 2.

```
    illegal safemode                      <safemode>
```

Must be 2 or 3.

```
    stop after status check
```

Run terminated.

Check of log label

```
    log file does not exist
```

```
    label record not found        <log record type>
```

The type of the first log record do not correspond to a label record.

2. Telerees

```
log version <> status version          <log.date> <status.date>
```

```
dversion(log) <> dbversion(status)     <log.dbvers> <status.dbvers>
```

See section 2.3.3.

```
tape name   <> log name
```

The name in the label record is not the log file name.

```
stop after label check
```

Initialization

```
illegal result from { initsoda1     <result>}
                    { initsoda2             }
```

Consult the SODA manual.

Database restoration

These messages are extended with selected information from the head of the log record, displayed with the format:

```
loghead:    recno            type           setno          sdpos
            <log record no>  <record type>  <soda set>     <position>
             user             term           trans          file name
            <user no>        <terminal no>  <transac no>   <db file>
```

2. Telerees

```
log ordering inconsistent <record number> <expected>
```

The <record number> from the log head was different from the <expected> one. Maybe because a position outside the log is read.

```
illegal record type                <record type>
```

```
illegal file type                  <file type>
```

Illegal types specified in log head.

```
break in text rees      <param>  <cause>
```

```
break in sq rees        <param>  <cause>
```

See section 2.4.2.

```
create/reserve sq   <createresult*10 + reserveresult>
```

Area process for sq file could not be created or reserved.

Master records

```
insert master, allready in db
```

```
insert master, extension failed <resultcf> <segments>
```

```
insert master, failed after extension     <resultcf>
```

```
update master, no found                   <resultcf>
```

2. Telerees

```
update master, put failed                    <resultcf>
```

```
delete master, not found                     <resultcf>
```

```
delete master, failed                        <resultcf>
```

List records

For list records resultcf and 'recnocf' from the record copy in the log always are displayed on a separate line.

```
insert list, owner not found   <ownerset> <primary chain>
```

```
insert list, position not found <primary chain> <position reached>
```

The record could not be positioned in the primary chain.

```
insert list, extension failed      <primary chain> <segments>
```

```
insert list, failed after extension <primary chain> <icmode>
```

```
connect list, owner not found   0   <secondary chain>
```

A secondary owner could not be accessed.

```
update list, not found                  <primary chain> <recnocf>
```

Record could not be accessed via recnocf. Recnocf could be illegal.

```
update list, put failed                 <primary chain>
```

2. Telerees

```
delete list, owner not found <ownerset> <primary chain>
```

```
delete list, position not found <primary chain> <reached position>
```

The record could not be located in the primary chain.

```
delete list, actual recno not expected <primary chain> <recnocf>
```

The recnocf for the located record was different from the recnocf in the logged record.

```
delete list, failed     <primary chain>
```