

PROGRAMBESKRIVNING

GTRANS 3

EXPORTERADE SUBROUTINANROP (*Exported function calls*)

Hans Granegård 2000-02-23
National Land Survey of Sweden
LF-Geodesi

This description is in Swedish, but contains short comments in English.
Beskrivningen innehåller kortfattade kommentarer på engelska.

Innehåll:

- 1 Översikt över programkoncept (*Conceptual overview*)
- 2 Allmän beskrivning av DLL (*General DLL description*)
- 3 Subrutinbeskrivningar (*Subroutines*)

Bilagor med exempel på anrop. (*Appendices, sample calls*)

- 1 Visual Basic
- 2 Fortran
- 3 C
- 4 Pascal, Delphi

1 ÖVERSIKT ÖVER PROGRAMKONCEPT

Overview

*GTRANS can be used as a 'component' (DLL) in an application, and such applications may be developed freely, using this description. The **END USER** must obtain the program files (DLL:s) through the usual registration of the software GTRANS and payment of a registration fee (1000 SEK). The program files for GTRANS can therefore not become a part of another applications distribution files.*

When calling the DLL from another application, a full installation of GTRANS may not be necessary, this depends on what calls you make. For instance, if the predefined transformatic are not used, the GATESYS folder may not be necessary.

GTRANS can also be called with a command line syntax, this is described in the helpfile GTRANS3.HLP

GTRANS kan användas som en komponent (DLL) i ett annat programsystem. Sådan utveckling kan ske fritt. Användaren av ett sådant system måste dock skaffa sin installation av programvaran genom den vanliga nyttjanderätts-överenskommelsen, de som utvecklar sådana system får alltså inte distribuera GTRANS som en del av detta.

GTRANS32.DLL används av programmet GTRANS 3 under Windows NT el. 95
UNIX-libet LibGtrans.a är avsett för BankIR. Subrutinanropen har
Samma uppbyggnad i bägge fallen.

Beskrivningen i hjälpfilen GTRANS3.HLP för Gtrans 3 ger mer detaljerad information om koncept när det gäller transformationer och parameterfiler m.m.

SYSTEM

Coordinate systems are denoted by a NAME with up to 32 characters.
Koordinatsystem anges med ett namn som är en teckensträng på max 32 tecken

KFIL

Coordinate file in a certain syntax, where the SYSTEM name for the coordinates is supplied.
Koordinatfil enligt en viss syntax där SYSTEM-namnet anges

TRANSFORMATION, TF-FIL

Every transformation is defined by a parameterfile (TF-file), the predefined transformation files are part of the GTRANS installation.

Varje transformation som exekveras hämtas från en motsvarande parameterfil, denna TF-FIL kan sökas via tabeller i programinstallationen eller direkt som en sökväg till en fil.

INSTALLATION (*Installation folders*)
Installationen sker på katalogerna:

GATERUN (c:\program\gtrans3\bin)

Contains executable files and some other files supporting the application.

Innehåller exekverbara filer + div. andra filer

Ex: filen GATE-ERR.SWE som innehåller översättning av felkoder till klartext
KFORMAT-filer (.KF) är parameterfiler för hantering av olika filformat

GATESYS (c:\program\gtrans3\sys)

Contains TF-files and some other files concerning the coordinate systems and transformations, such as index tables, and geoid models (GRD-files)

Innehåller alla filer som definierar koordinattransformationer

- TF-filer är parameterfil som definierar en transformationsväg
- TRIAD-filer (.TR1,.TR2) är triangulationer för restfelsinterpolering
- GRID-filer (.GRD) är geoidhöjdsmodeller
- Tabellfiler (.GEO, .DEF) innehåller index för sökning av transformationer

GATEUFD

Userdefined objects.

Arkivkatalog är en användares katalog för egendefinierade transformationer och format. DEF-filer innehåller användardefinierade objekt

2 ALLMÄN BESKRIVNING AV DLL (LIB)

GTRANS32.DLL is developed using Microsoft Fortran Powerstation 4 (32-bit).

It can be invoked in several instances if necessary, and application calls have been tested using Visual Basic, Fortran, C++ and Pascal (Delphi). Any other language should be possible to use, if a correct interface is made which has the correct data types and calling convention (by reference in Fortran). Also the definition of string parameters cause some problems here.

This description contains a selection of the most useful subroutine interfaces, for the moment leaving some subroutines undescribed.

GTRANS32.DLL är en Fortran-dll skapad med Microsoft Fortran Powerstation 4 (32-bit)

Anrop sker enligt Fortransyntax med parametrar BY REFERENCE
Endast ett urval av de exporterade subrutinerna beskrivs här.

DATATYPER I DEKLARATIONER: (*Data types used*)

INTEGER = långt heltal (4 bytes)

CHARACTER*nn = fixlängd sträng nn bytes

DOUBLE PRECISION = flyttal dubbel precision (8 bytes)

arrayer anges med NAMN(nn) och antas vara deklarerade med fix storlek

Filenheter som refereras till är Fortan-units

De behöver vanligen inte användas annat än som heltal som förs över som parametrar. Sätts ev = 0 om de ej används

3 SUBRUTINBESKRIVNINGAR

1	GTRANS_INIT	initiera session.
2	GTRANS_OPEN	öppna tabeller, initiera vissa parametrar m.m.
3	GT_SETTRANS	initiera transformation
4	GT_EXETRANS	exekvera transformation
5	GTRANS_SHUT	stäng aktiv transformation
6	GT_SHUTGATE	stäng tabeller, avsluta session
7	GT_LISTSYSTEM	lista SYSTEM
8	GT_FINDSYSTEM	finn SYSTEM (entydigt namn) i tabell
9	GT_LISTTRANS	lista TF-filer
10	GT_NEWTRANS	lägg in ny transformation i tabell
11	GT_INVTRANS	sätt invers TF-fil
12	GT_KILLTRANS	radera transformation i tabell
13	GTRANS_MAKE	skapa ny tabell

3.1 GTRANS_INIT(minun, maxun)

FUNCTION: Initialization which always begins a DLL session.

Funktion: Initiera vissa parametrar m.m.

Inleder alla sessioner med GTRANS.DLL (avsluta med GT_SHUTGATE)

Parametrar:

In:

MINUN,MAXUN - (interval to search for free fortran units)

undre resp övre gräns för filenheter som GTRANS

tillåts använda. GTRANS söker lediga filenheter

i intervallet (minun,maxun)

Ut: inga parametrar.

Deklarationer:

INTEGER minun, maxun

3.2 GTRANS_OPEN(u_mode, ufd_mode, e_mode, errunit, ugatesys, ugateufd, ugaterun, ugateusr, ugatelng, err_code, err_str)

FUNCTION: Initialization where tables are opened and some global parameters are initialize

Funktion: Öppna tabeller m.m.

Inleder alla sessioner med GTRANS.DLL efter GTRANS_INIT

Parametrar:

In:

U_MODE användarmode
=0 öppna inga tabeller (no tables are opened)
=1 öppna endast tabeller på UGATESYS
=2 öppna endast tabeller på UGATEUFD
=3 öppna tabeller på både UGATESYS och UGATEUFD
UFD_MODE
=0 öppna tabell på GATEUFD för LÄSNING
=2 öppna tabell på GATEUFD för LÄSNING och SKRIVNING
E_MODE
=1 öppna felkodstabell (GATE-ERR.SWE) för översättning
av felkoder till klartext. (translation of error codes)
=0 öppna ej -"-
ERRUNIT
enhet för ev felutskrift, om >0 (kan vara konsol)
om den sätts =0 undertrycks felutskrifter
(i denna subrutin)
Globala systemvariabler, markerade med * behövs endast om EMODE=1:
UGATESYS namn på GATE-directory (GATESYS)
UGATEUFD namn på användares GATE-directory (GATEUFD)
UGATERUN * namn på directory för program+hjälppfiler(GATERUN)
UGATEUSR inloggad användar-id, behövs bara om skrivning i tabeller ska göras
UGATELNG * språk <'swe' ! 'eng'> (' ' = 'swe')
(eng has not been supported in this version)

Ut:

ERR_CODE felflagga, =0 om inget fel uppstår
ERR_STR felkod tolkad till klartext, om ERR_CODE>0

Deklarationer:

INTEGER u_mode, ufd_mode, e_mode, err_code, errunit
CHARACTER*256 u_gatesys, u_gateufd, u_gaterun, err_str
CHARACTER*8 u_gateusr
CHARACTER*4 u_gatelng

3.3 GT_SETTRANS(u_mode, tf_file, f_system, t_system, w_flag, errunit, err_lin,
err_code, err_str)

FUNCTION: Open an active transformation, this is closed after use by GTRANS_SHUT
Funktion: Initiera transformation, endast en transformation kan vara aktiv
Transformationen stängs med GTRANS_SHUT.

Parametrar:

In:

U_MODE 0-Aktuell katalog,1-Gtrans,2-Arkivkatalog
(U_MODE = 0 if not stored in table)
TF_FILE Namn på transformation (TF-fil, trådnamn om U_MODE=0)
(Full path if U_MODE=0)
W_FLAG sätt =1 för att initiera varningskoder
ERRUNIT enhet för ev felutskrift, om >0 (kan vara konsol)
om den sätts =0 undertrycks felutskrifter (i denna subrutin)

Ut:

F_SYSTEM Namn på från-system i TF-fil
T_SYSTEM Namn på till-system i TF-fil
ERR_LIN radnr i TF-fil när fel påträffats, om ERR_CODE<>0
ERR_CODE 0=OK, <>0=FEL
ERR_STR felmeddelande om ERR_CODE<>0

Deklarationer:

INTEGER u_mode, err_code, errunit, err_lin, w_flag
CHARACTER tf_file*1024, f_system*32, t_system*32, err_str*256

3.4 GT_EXETRANS(f_xn, f_je, f_zh, f_codim, t_xn, t_je, t_zh, t_codim, w_code,
err_code, err_str)

FUNCTION: Transform a point with active transformation

Funktion: Transformera en punkt enligt transformation som först har initierats med anrop till GT_SETTRANS

Parametrar:

In:

F_XN Ingående koordinat, X eller lat
Northing (m) or latitude (dec. degrees)
F_YE Ingående koordinat, Y eller long
Easting (m) or longitude (dec. degrees)
F_ZH Ingående Höjd eller Z
Height if F_CODIM=3
F_CODIM antal definierade element i ingående koordinatpost
1=endast höjd 2=X,Y 3=X,Y och H (Z)

Ut:

T_XN Utgående koordinat, X eller lat
T_YE Utgående koordinat, Y eller long
T_ZH Utgående Höjd eller Z
T_CODIM antal definierade element i utgående koordinatpost
1=endast höjd 2=X,Y 3=X,Y och H (Z)
W_CODE Varningskod, =1 om utanför tillämpningsområde, annars =0
ERR_CODE 0=OK, <>0=FEL
ERR_STR felmeddelande om ERR_CODE<>0

Deklarationer:

DOUBLE PRECISION f_xn, f_je, f_zh, t_xn, t_je, t_zh
INTEGER f_codim, t_codim, err_code, w_code
CHARACTER err_str*256

3.5 GTRANS_SHUT

FUNCTION: Close active transformation, no parameters.

Funktion: Stäng aktiv transformation

Parametrar: inga parametrar.

3.6 GT_SHUTGATE(u_mode, e_mode, err_code, err_str)

FUNCTION: Close session.

Funktion: Stäng tabeller och ev andra filer, avsluta session

Parametrar:

In:

U_MODE användarmode
 =1 stäng endast tabeller på GATESYS
 =2 stäng endast tabeller på GATEUFD (ARKIV)
 =3 stäng alla filer i GATE
 E_MODE se beskrivning av GTRANS_OPEN

Ut:
 ERR_CODE felflagga, 0=OK, <>0=FEL
 ERR_STR felmeddelande i klartext

Deklarationer:

INTEGER u_mode, e_mode, err_code
 CHARACTER err_str*256

3.7 GT_LISTSYSTEM(i_flag, n_mask, datum, n_stat, d_stat, std_stat, i_match, u_stat, rt_stat, system, u_mode, err_code, err_str)

FUNCTION: Listing of SYSTEM-names, according to N_MASK which can contain wildcards ()*

Funktion: Lista SYSTEM-namn i tabell enligt urvalskriterier. N_MASK med wildcard () kan resultera i flera träff. Flera namn hämtas då med upprepade anrop (se I_FLAG).*

Parametrar:

In:
 N_MASK Namn med el utan wildcard (*)
 DATUM Namn på datum för urval om D_STAT är satt
 N_STAT Urval på N_MASK, 0=NEJ(alla), 1=namn ev. wildcards
 D_STAT Urval på DATUM, 0=nej, 1=ja (entydigt namn)
 STD_STAT Internationella system, 0=nej, 1=ja
 U_STAT ARKIV (GATEUFD), 0=nej, 1=ja
 RT_STAT RT-system, 0=inga, 1-14=index, -1=alla

In/ut:
 I_FLAG 0=initiering (in), <1 =slut på poster (ut)
 >0=flera poster (ut), hämta post (in)

Ut:
 K_SYSTEM namn om I_FLAG
 I_MATCH antal funna system hittills
 ERR_CODE 0=OK, <>0=FEL
 ERR_STR felmeddelande om CODE<>0

Deklarationer:

INTEGER n_stat, d_stat, std_stat, u_stat, rt_stat, i_flag, err_code
 INTEGER u_mode, i_match
 CHARACTER k_system*32, n_mask*32, datum*32, err_str*256

3.8 GT_FINDSYSTEM(q_umode, q_system, o_system, o_datum, o_umode, o_comodel, o_htyp, sys_def, err_code, err_str)

FUNCTION: Find one unambiguous SYSTEM-name in table.

Funktion: Finn SYSTEM (entydigt namn) i tabell

Parametrar:

In:
 Q_UMODE 1=GATESYS, 2=ARKIV, 3=båda

Q_SYSTEM namn på system

Ut:

O_SYSTEM namn på system i tabell
O_DATUM namn på DATUM
O_UMODE 1=GATESYS,2=ARKIV,0=ej funnet
O_COMODEL 1=PLAN,2=LATLONG,3=CART,4=POL2
O_HTYP 1=NORMAL,2=ORTO,3=ELLH
SYS_DEF textbuffert med dataserier för SYSTEM
ERR_CODE felkod
ERR_STR felmeddelande om ERR_CODE<>0

Deklarationer:

INTEGER q_umode, o_umode, err_code, o_comodel, o_htyp
CHARACTER q_system*32, o_system*32, o_datum*32, err_str*256
CHARACTER sys_def*2048

3.9 GT_LISTTRANS(umode, tmode, imode, nmask, fmask, tmask, iflag, imatch,
name, oinvers, ofsystem, otsystem, tf_text, adminfo, trainfo,
code, errstr)

FUNCTION: Listing of Transformation-files in tables, according to NMASK,FMASK,TMASK which can contain wildcards ()*

Funktion: Lista objekt i TFFIL-tabeller
N_MASK med wildcard (*) kan resultera i flera träff.
Flera namn hämtas då med upprepade anrop (se I_FLAG).

Parametrar:

In:

UMODE 1-GATESYS,2-GATEUFD,3-Båda

TMODE:

Om TMODE = 1 :

NMASK, FMASK och TMASK kan innehålla wildcard som styr urval.
Sök i tab TFFIL el UTFFIL beroende på UMODE. Om IMATCH=+-1 så är det entydig träff.
Om IMODE=1 och MATCH=+-1 så öppnas TF-filen och läses in i LIST

Om TMODE = 2 (endast för UMODE=1 f.n.) :

sök i tabell COTRANS med FSYSTEM och TSYSTEM som entydig nyckel. Om posten hittas sätts NAME=namn och IMATCH=-1 och p.s.s. som i TMODE=1 för IMATCH=1.

IMODE Se TMODE

NMASK Namn med el utan wildcard (*)

FMASK Urval på Från-SYSTEM med el utan wildcard (*)

TMASK Urval på Till-SYSTEM med el utan wildcard (*)

In/ut:

IFLAG 0 =initiering (in), slut på poster (ut)
-1 =flera poster (ut), hämta post (in)

Ut:

IMATCH 0=ingen matchande,-1=STANDARD,1=entydig,2=flera
NAME Namn på TF-fil
OINVERS Namn på INVERS TF-fil
OFSYSTEM Namn på Från-SYSTEM
OTSYSTEM Namn på Till-SYSTEM
TF_TEXT TF-filens text
ADMINFO Om IMATCH=1 så innehåller
ADMINFO administrativ information :
<PROT><USERID><DATE>, där

```

<PROT> = A ! P; A=kan ändras (om rättighet)
          P=ovillkorligt skrivskyddad
<USERID> = användarid på max 8 tecken
<DATE> = ÅÅÅÅMMDDttmmss, 14 tecken
<STATUS> 'R' om användaren ej har rättigheter
TRAINFO -
  trainfo (1) = koordinatmodell FRÅN
          (2) = htyp FRÅN
          (3) = degflag FRÅN (=1 för grader, annars=0)
          (4) = pekar till aktuell ELLIPSOID i ILIST
          (5) = pekar till ellipsoidparametrar i DLIST
          (6) = pekar till aktuell GEOID i ILIST
          (7) = aktuellt PROJNR (iproj)
  trainfo(11-20) = motsv. för TILL
  trainfo(21...) innehåller div. statistik om transformationen
          (21) odefinierad f.n.
          (22) antal överbestämn. för (21), <0 betyder odef.
          (23) odefinierad f.n.
          (24) antal överbestämn. för (23), <0 betyder odef.
          (25) NOPER = totalt antal satta operatorer
          (26) NFORM = antal FORMEL el. TRIAD
          (27) NRESID >0 om RESIDUAL given
          (28) NSYST = antal def. SYSTEM inkl FSYSTEM & TSYSTEM
          (29) INVFLAG      =-1 => INVERS kan göras (automatiskt)
                   = 0 => INVERS kan (ev.) göras
                   =+1 => INVERS kan ej göras
          (30) EXTERN = antal externa ref. (filenhet för baser)

CODE 0=OK, <>0=FEL
ERRSTR felmeddelande om CODE<>0

```

Deklarationer:

```

INTEGER      umode, tmode, imode, iflag, imatch, code, trainfo(64)
CHARACTER    nmask*32, fmask*32, tmask*32, name*32, oinvers*32
CHARACTER    ofsystem*32, otsystem*32, errstr*256, adminfo*32
CHARACTER    tf_text*2048

```

3.10 GT_NEWTRANS(u_mode, t_mode, tf_file, err_code, err_str)

```

FUNCTION:    Insert new Transformation-file into table.
Funktion:    Lägg in TF-filen TF_FILE i tabell
            TF_FILE förutsätts redan ligga på resp filkatalog

```

Parametrar:

```

In:         U_MODE      1 => GATESYS, 2 => GATEUFD (ARKIV)

            T_MODE      1 => tf_file lagras i tabell
                       2 => tf_file sätts som STANDARD (bara för u_mode=1)
                       3 => kombinera 1 & 2

            TF_FILE      Namn på TF-fil

```

Ut:

```

ERR_CODE    0=OK, <>0=FEL
ERR_STR      felmeddelande om ERR_CODE<>0

```

Deklarationer:

```

INTEGER      u_mode, t_mode, err_code
CHARACTER    tf_file*32, err_str*256

```


3.11 GT_INVTRANS(u_mode, tf_file, tf_invers, err_code, err_str)

FUNCTION: Set the TF-file TF_INVERS as the inverse of TF_FILE in table.
Funktion: Sätt TF-filen TF_INVERS som invers till TF-filen TF_FILE i tabell
Båda TF-filerna förutsätts redan ligga i tabellen.
Posten TF_INVERS i tab TFFIL/UTFFIL antas existera, detta namn läggs in
i fältet 'invers' för posten TF_FILE som också antas existera .
Omvänt läggs också TF_FILE in som invers för posten TF_INVERS.

För att ta bort en invers ur tabellen anropas GT_INVTRANS med
TF_INVERS satt till ' '

Parametrar:

In:
U_MODE 1 - GATESYS, 2 - GATEUFD (ARKIV)
TF_FILE Namn på TF-fil
TF_INVERS Namn på INVERS TF-fil

Ut:
ERR_CODE 0=OK, <>0=FEL
ERR_STR felmeddelande om ERR_CODE<>0

Deklarationer:

INTEGER u_mode, err_code
CHARACTER tf_file*32, tf_invers*32, err_str*256

3.12 GT_KILLTRANS(mode, tf_file, err_code, err_str)

FUNCTION: Remove the TF-file TF_FILE from table.
Funktion: Ta bort TFFIL ur användartabell (ARKIV)
Posten TF_FILE i tabell UTFFIL (ARKIV) raderas (delete-markeras).
Om en invers till TF_FILE finns inlagd,
så kommer fältet 'invers' (=TF_FILE) i denna post
att sättas blankt, men själva posten (INVERS) ligger kvar.

Parametrar:

In:
MODE
Om MODE=0 görs ingenting med själva TF-filen TF_FILE,
om MODE=1 så raderas filen TF_FILE fysiskt.

Ut:
TF_FILE filnamn
ERR_CODE 0=OK, <>0=FEL
ERR_STR felmeddelande om ERR_CODE<>0

Deklarationer:

INTEGER mode, err_code
CHARACTER tf_file*1024, err_str*256

3.13 GTRANS_MAKE(ufd, u_mode, err_unit, err_code, err_str)

FUNCTION: Create new table.

Funktion: Skapa ett nytt GATE-system med tabeller på
directory GATESYS/GATEUFD. Föregås av GTRANS_OPEN.
Kontrollerar om tabeller finns.
I varje tabell läggs 1:a post in + ev. standardposter

Parametrar:

In:

UFD katalog för tabell (GATESYS el GATEUFD)

U_MODE användarmode

=1 skapa tabeller på GATESYS, endast systemadministratör

=2 skapa tabeller på GATEUFD (ARKIV) för användare

ERR_UNIT enhet för ev felutskrift, kan sättas=0

Ut:

ERR_CODE felflagga

ERR_STR felkod tolkad till klartext, om ERR_CODE>0

Deklarationer:

INTEGER u_mode, err_unit, err_code

CHARACTER*256 ufd, err_str


```

emode = 0
errunit = 0
ugatesys = "c:\program\gtrans3\sys"
ugateufd = "c:\program\gtrans3\Arkiv"
ugaterun = "c:\program\gtrans3\bin"
ugateusr = "Jag" ' User name
ugatelng = "swe" ' User language

'      Initialization
Call GTRANS_OPEN(umode, ufd_mode, emode, errunit, ugatesys, ugateufd, ugaterun, ugateusr,
ugatelng, err_code, err_str)

'      Pick a transformation, this must be known by name. A dialogue where the user can navigate
'      to this, by using SYSTEM names, can be implemented, as in the GTRANS main program.
'      Välj en transformation
f_system = " "
t_system = " "
tf_file = "R9025W5W.TF"
w_flag = 0

'      Alternatively you can call an arbitrary TF-file like:
'      umode = 0
'      tf_file = "c:\userdata\my_own.tf"

'      Opening of current transformation.
Call GT_SETTRANS(umode, tf_file, f_system, t_system, w_flag, errunit, err_lin, err_code,
err_str)

Print #1, tf_file
Print #1, f_system
Print #1, "till"
Print #1, t_system

'      Define the coordinate data and execute, the data would normally be supplied some other way
'      from the calling application.
'      Sätt indata och transformera
f_xn = 6500000#      ' Northing in the Swedish system RT 90 2.5 gon V
f_ye = 1400000#      ' Easting in the Swedish system RT 90 2.5 gon V
f_zh = 0#           ' Height is set to zero, is not actually used here.
f_codim = 2         ' Number of coordinates (X and Y), if Z is also used it is set to 3.

Print #1, "ingående koordinater:"
Print #1, f_xn, f_ye

'      Execution
Call GT_EXETRANS(f_xn, f_ye, f_zh, f_codim, t_xn, t_ye, t_zh, t_codim, w_code, err_code,
err_str)

'      Write output of transformed coordinates
'      Skriv transformerade data, avrundat till 3 dec: x=6498839.092 y=1530749.846
Print #1, "transformerade koordinater:"
Print #1, t_xn, t_ye

'      Close transformation
'      Stäng
Call GTRANS_SHUT
'      Close session
Call GT_SHUTGATE(umode, emode, err_code, err_str)

Close #1

End Sub

```



```

C      umode = 0
C      tf_file = 'c:\userdata\my_own.tf'

      Call GT_SETTRANS(umode, tf_file, f_system, t_system, w_flag,
&                      errunit, err_lin, err_code, err_str)

      write (7,*) tf_file
      write (7,*) f_system
      write (7,*) "till"
      write (7,*) t_system

C      Sätt indata och transformera
      f_xn = 6500000.d0
      f_ye = 1400000.d0
      f_zh = 0.d0
      f_codim = 2
      write (7,*) 'ingående koordinater:'
      write (7,*) f_xn, f_ye

      Call GT_EXETRANS(f_xn, f_ye, f_zh, f_codim, t_xn, t_ye, t_zh,
&                      t_codim, w_code, err_code, err_str)

C      Skriv transformerade data, avrundat till 3 dec: x=6498839.092 y=1530749.846
      write (7,*) 'transformerade koordinater:'
      write (7,*) t_xn, t_ye

C      Stäng
      Close (7)
      Call GTRANS_SHUT
      Call GT_SHUTGATE(umode, emode, err_code, err_str)

      End

```

#Bilaga 3, Anrop i C, testad i Microsoft Visual C++ och Borland C++
#3 Sample application developed in C, does the same as sample 1 and 2

```
/* Dlltest.cpp */

#include <stdio.h>
#include <windows.h>
#include <string.h>

typedef void (__stdcall *lpGTRANS_INIT)(long*,long*);

typedef void (__stdcall *lpGTRANS_OPEN)
(long*,long*,long*,long*,char*,char*,char*,char*,char*,long*,char*);

typedef void (__stdcall *lpGT_SETTRANS)
(long*,char*,char*,char*,long*,long*,long*,long*,char*);

typedef void (__stdcall *lpGT_EXETRANS)
(double*,double*,double*,long*,double*,double*,double*,long*,long*,long*,char*);

typedef void (__stdcall *lpGTRANS_SHUT)(void);

typedef void (__stdcall *lpGT_SHUTGATE)(long*, long*, long*, char*);

lpGTRANS_INIT GTRANS_INIT;
lpGTRANS_OPEN GTRANS_OPEN;
lpGT_SETTRANS GT_SETTRANS;
lpGT_EXETRANS GT_EXETRANS;
lpGTRANS_SHUT GTRANS_SHUT;
lpGT_SHUTGATE GT_SHUTGATE;

void strFortran(char* ,char* ,int); // Fyll ut sträng med blanka

void main(void)
{
    HINSTANCE hLibrary;
    FILE *outfile;
    char outfilename[20];
    long minun, maxun, errunit, umode, ufd_mode, emode;
    long w_flag, w_code, err_code, err_lin;
    long f_codim, t_codim;
    double f_xn, f_yn, f_zh;
    double t_xn, t_yn, t_zh;
    char ugatesys[257], ugateufd[257], ugaterun[257];
    char err_str[257], ugateusr[9], ugatelng[5];
    char tf_file[1025], f_system[33], t_system[33];

    int len4=4;
    int len8=8;
    int len32=32;
    int len256=256;
    int len1024=1024;

    hLibrary = LoadLibrary("gtrans32.dll");
    GTRANS_INIT = (lpGTRANS_INIT) GetProcAddress(hLibrary, "GTRANS_INIT");
    GTRANS_OPEN = (lpGTRANS_OPEN) GetProcAddress(hLibrary, "GTRANS_OPEN");
    GT_SETTRANS = (lpGT_SETTRANS) GetProcAddress(hLibrary, "GT_SETTRANS");
    GT_EXETRANS = (lpGT_EXETRANS) GetProcAddress(hLibrary, "GT_EXETRANS");
    GTRANS_SHUT = (lpGTRANS_SHUT) GetProcAddress(hLibrary, "GTRANS_SHUT");
    GT_SHUTGATE = (lpGT_SHUTGATE) GetProcAddress(hLibrary, "GT_SHUTGATE");

    /* Öppna tillfällig resultatfil */
    strcpy( outfilename, "c:\\temp\\dll_c.txt");
    if((outfile = fopen(outfilename, "w")) != NULL)
    {
        fprintf(outfile, "C-dlltest\n");
        fprintf( outfile, " \n");
    }
}
```

```

/* Fortran units kan t.ex. börja med 8, övre gräns 70 är väl tilltagen */
minun = 8;
maxun = 70;

GTRANS_INIT(&minun, &maxun);

/* Initiera för ej förlagrad transformation */
umode = 0;
ufd_mode = 0;
emode = 0;
errunit = 0;
strFortran(ugatesys, " ", len256);
strFortran(ugateufd, " ", len256);
strFortran(ugaterun, " ", len256);
strFortran(ugateusr, "Jag", len8);
strFortran(ugatelng, "swe", len4);
strFortran(err_str, " ", len256);

GTRANS_OPEN(&umode, &ufd_mode, &emode, &errunit, ugatesys,
            ugateufd, ugaterun, ugateusr,
            ugatelng, &err_code, err_str);

/* Välj en transformation */
strFortran(tf_file, "c:\\program\\gtrans3\\sys\\R9025W5W.TF", len1024);
strFortran(f_system, " ", len32);
strFortran(t_system, " ", len32);
strFortran(err_str, " ", len256);
w_flag = 0;

GT_SETTRANS(&umode, tf_file, f_system, t_system, &w_flag,
            &errunit, &err_lin, &err_code, err_str);

    fprintf(outfile,tf_file);
    fprintf(outfile,"\n");
    fprintf(outfile,f_system);
    fprintf(outfile,"\n");
    fprintf(outfile,"till\n");
    fprintf(outfile,t_system);
    fprintf(outfile,"\n");

/* Sätt indata och transformera */
f_xn = 6500000.0;
f_ye = 1400000.0;
f_zh = 0.0;
f_codim = 2;
fprintf( outfile,"ingående koordinater:\n");
fprintf( outfile, "%f %f\n", f_xn, f_ye );
strFortran(err_str, " ", len256);

GT_EXETRANS(&f_xn, &f_ye, &f_zh, &f_codim, &t_xn, &t_ye, &t_zh,
            &t_codim, &w_code, &err_code, err_str);

/* Skriv transformerade data, avrundat till 3 dec: x=6498839.092 y=1530749.846 */
fprintf( outfile,"transformerade koordinater:\n");
fprintf( outfile, "%f %f\n", t_xn, t_ye );

/* Stäng */
GTRANS_SHUT();
strFortran(err_str, " ", len256);
GT_SHUTGATE(&umode, &emode, &err_code, err_str);
fclose (outfile);
FreeLibrary(hLibrary);
    return;}
}

```



```
void strFortran(char* strFor, char* strSource, int declared_length)
{
    /* Fyll ut sträng med blanka till fix (Fortran-deklarerad) längd */
    /* Pad with blank spaces to Fortran fix-length */

    int i, istart;

    strcpy (strFor, strSource);
    istart = strlen(strFor);
    for ( i = istart; i < declared_length; i++ )
        { strcat( strFor, " " ); }
}
```

#Bilaga 4, Anrop i Pascal, testad i Borland Delphi
#4 Sample application developed in Pascal, does the same as previous samples

Översatt till Pascal av Jan Flaage

dlltest.dpr

```
program dlltest;

uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1};

{$R *.RES}

begin
  Application.Initialize;
  Application.Title := 'dlltest';
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

Unit1.pas

```
unit Unit1;

{$O-}           // Slå av optimering

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  Plint = ^longint;
  Pdouble = ^double;
  Char4 = array[0..3] of char; Pchar4 = ^Char4;
  Char8 = array[0..7] of char; Pchar8 = ^Char8;
  Char32 = array[0..31] of char; Pchar32 = ^Char32;
  Char256 = array[0..255] of char; Pchar256 = ^Char256;
  Char1024 = array[0..1023] of char; Pchar1024 = ^Char1024;

type
  TForm1 = class(TForm)
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

procedure Gtrans_init(minun, maxun: Plint
  ); stdcall; external 'GTRANS32.DLL' name 'GTRANS_INIT';

procedure Gtrans_open(
  umode, ufd_mode, e_mode, errunit: Plint;
  ugetesys, ugateufd, ugaterun: PChar;
  ugateusr: PChar;
  ugatelng: PChar;
  err_code: Plint;
  err_str: PChar256
  ); stdcall; external 'GTRANS32.DLL' name 'GTRANS_OPEN';
```

```

procedure Gt_settrans(
    u_mode: Plint;
    tf_file: PChar1024;
    f_system, t_system: Pchar32;
    w_flag, errunit, err_lin, err_code: Plint;
    err_str: Pchar256
); stdcall; external 'GTRANS32.DLL' name 'GT_SETTRANS';

procedure Gt_exetrans(
    f_xn, f_ye, f_zh: Pdouble; f_codim: Plint;
    t_xn, t_ye, t_zh: Pdouble;
    t_codim, w_code, err_code: Plint;
    err_str: PChar256
); stdcall; external 'GTRANS32.DLL' name 'GT_EXETRANS';

procedure Gtrans_shut; stdcall; external 'GTRANS32.DLL' name 'GTRANS_SHUT';

procedure Gt_shutgate(
    u_mode, e_mode, err_code: Plint;
    err_str: PChar256
); stdcall; external 'GTRANS32.DLL' name 'GT_SHUTGATE';

var
    Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
var
    F: TextFile;

    minun, maxun, errunit: Longint;
    umode, ufd_mode, emode: Longint;

    u gatesys, ugateufd, ugaterun: Char256;
    ugateusr: Char8;
    ugatelng: Char4;
    tf_file: Char1024;
    f_system, t_system: Char32;
    err_str: Char256;

    w_flag, w_code, err_code, err_lin: Longint;

    f_xn, f_ye, f_zh: Double;
    t_xn, t_ye, t_zh: Double;
    f_codim, t_codim: Longint;

begin
    // Öppna tillfällig resultatfil
    AssignFile(F, 'c:\temp\dll_pas.txt');
    Rewrite(F);
    Writeln(F, 'Delphi DLL-test');
    Writeln(F);

    // Fortran units kan t.ex. börja med 8, övre gräns 70 är väl tilltagen
    minun := 8;
    maxun := 70;

    Gtrans_init(@minun, @maxun);

    // Initiera för ej förlagrade transformationer
    umode := 0;

```

```

ufd_mode := 0;
emode := 0;
errunit := 0;
ugatesys := 'c:\program\gtrans3\sys';
ugateufd := 'c:\program\gtrans3\Arkiv';
ugaterun := 'c:\program\gtrans3\bin';
ugateusr := 'Jag';
ugatelng := 'swe';

Gtrans_open( @umode, @ufd_mode, @emode, @errunit,
             @ugatesys, @ugateufd, @ugaterun,
             @ugateusr, @ugatelng,
             @err_code, @err_str);

// Välj en transformation
f_system := ' '; // blankställ, längd 32
t_system := ' ';
tf_file := 'c:\Program\Gtrans3\sys\R9025W5W.TF';
w_flag := 0;
errunit := 0;

Gt_settrans(@umode, @tf_file, @f_system, @t_system, @w_flag, @errunit,
            @err_lin, @err_code, @err_str);

Writeln(F, tf_file);
Writeln(F, f_system);
Writeln(F, 'till');
Writeln(F, t_system);

// Sätt indata och transformera
f_xn := 6500000.0;
f_ye := 1400000.0;
f_zh := 0.0;
f_codim := 2;

Writeln(F, 'ingående koordinater: ');
Writeln(F, f_xn, f_ye);

Gt_exetrans(@f_xn, @f_ye, @f_zh, @f_codim,
            @t_xn, @t_ye, @t_zh, @t_codim,
            @w_code, @err_code, @err_str);

// Skriv transformerade data, avrundat till 3 dec: x=6498839.092 y=1530749.846
Writeln(F, 'transformerade koordinater: ');
Writeln(F, t_xn, t_ye);

// Stäng
Gtrans_shut;
Gt_shutgate(@umode, @emode, @err_code, @err_str);
CloseFile(F);

end;

end.

```