

# **Dokumentation der Transformationssoftware GNTRANS\_NI**

**Stand: 27. Juni 2017, Version 1.4.4**

## Änderungshistorie

Version	Datum	Beschreibung der Version
1.4.4	20.06.2017	Dokumentation der Transformationssoftware GNTRANS_NI

## Inhaltsverzeichnis

<b>Abkürzungen</b> .....	<b>5</b>
<b>1 Einleitung</b> .....	<b>6</b>
1.1 Rahmenbedingungen.....	6
1.2 Amtliches Bezugssystem in Niedersachsen .....	6
1.3 Transformationssoftware für den Bezugssystemwechsel.....	7
1.3.1 Nutzungsbedingungen .....	7
1.3.2 Leistungsumfang von GNTRANS_NI .....	7
1.3.3 Das Transformationsmodell Niedersachsen .....	8
1.3.4 Höheninformationen für die Transformation .....	8
1.3.5 Eigenschaften der Transformation .....	9
<b>2 GNTRANS_NI</b> .....	<b>10</b>
2.1 Installation .....	10
2.2 Verzeichnisstruktur.....	14
2.3 GUI-Funktionalitäten .....	15
2.3.1 Einzelpunkte transformieren.....	15
2.3.2 Punkte aus einer Datei transformieren .....	17
2.4 CLI-Funktionalitäten .....	23
2.5 GNTRANS_NI API .....	25
2.5.1 Beschreibung .....	25
2.5.2 Installation und Einrichtung .....	25
2.5.3 Funktionsumfang.....	25
2.5.4 Fehlerwerte .....	29
2.5.5 Beispielprogramm .....	30
<b>3 Literatur</b> .....	<b>34</b>

## Abbildungsverzeichnis

Abbildung 1: Pakete der Transformationssoftware GNTRANS_NI .....	7
Abbildung 2: Ausdehnung des DGM-T im ETRS89/UTM (Geltungsbereich von GNTRANS_NI) .....	9
Abbildung 3: Metadaten zum DGM-T, thematisch gefärbt in Abhängigkeit der Datenquelle .....	9
Abbildung 4: Schritt 1 der Installation von GNTRANS_NI .....	10
Abbildung 5: Schritt 2 der Installation von GNTRANS_NI .....	10
Abbildung 6: Schritt 3 der Installation von GNTRANS_NI .....	11
Abbildung 7: Schritt 4 der Installation von GNTRANS_NI .....	11
Abbildung 8: Schritt 5 der Installation von GNTRANS_NI .....	12
Abbildung 9: Schritt 6 der Installation von GNTRANS_NI .....	12
Abbildung 10: Schritt 6 der Installation von GNTRANS_NI .....	13
Abbildung 11: Transformation von Einzelpunkten von DHDN/GK nach ETRS89/UTM mit WinGNTRANS_NI .....	16
Abbildung 12: Transformation von Einzelpunkten von ETRS89/UTM nach DHDN/GK mit WinGNTRANS_NI .....	16
Abbildung 13: Auswahl benachbarter Abbildungsstreifen bzw. -zonen.....	17
Abbildung 14: Datei auswählen .....	17
Abbildung 15: Punkte aus einer Datei mit WinGNTRANS_NI transformieren.....	18
Abbildung 16: Fehlermeldung bei Nichteinhaltung von Formatvorgaben.....	19
Abbildung 17: Fehlermeldung bei Dateien, deren Datensätze im Hinblick auf die Höhe der Punkte inhomogen sind.....	19
Abbildung 18: Hinweis auf den Umgang mit nutzerseitigen Höheninformationen .....	20
Abbildung 19: Warnung beim Anzeigen umfangreicher Dateien.....	20
Abbildung 20: Anzeige von Dateiinhalten am Beispiel der Eingabedatei „test.gkh“.....	21
Abbildung 21: Hinweis bei erfolgreicher Transformation .....	21
Abbildung 22: Hinweis auf Fehler bei der Transformation .....	22
Abbildung 23: Anzeige von Dateiinhalten am Beispiel der Ausgabedatei „test.utm“.....	22
Abbildung 24: Beispiel für den Inhalt einer Log-Datei .....	23

## Tabellenverzeichnis

Tabelle 1: Leistungsumfang der GNTRANS_NI-Pakete.....	8
Tabelle 2: Konsolenkommandos für GNTRANS_NI EXEC, Rücktransformation entsprechend .....	24
Tabelle 3: Funktionen der GNTRANS_NI API.....	25

## Abkürzungen

AAA	AFIS-ALKIS-ATKIS
AdV	Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland
AFIS	Amtliches Festpunktinformationssystem
ALKIS	Amtliches Liegenschaftskatasterinformationssystem
API	Application Programming Interface (engl.)
ASCII	American Standard Code for Information Interchange (engl.)
ATKIS	Amtliches Topographisch-Kartographisches Informationssystem
CLI	Command Line Interface (engl.)
CRS	Coordinate Reference System (engl.)
crsL	CRS der Lage
crsH	CRS der Höhe
DE	Deutschland
DGM	Digitales Geländemodell
DGM-T	Digitales Geländemodell für die Transformation
DHDN	Deutsches Hauptdreiecksnetz
DLL	Dynamic Link Library (engl.)
ETRS89	Europäisches Terrestrisches Referenzsystem zur Epoche 1989.0
EXEC	Executables
GDI	Geodateninfrastruktur
GeoInfoDok	Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens
GK	Gauß-Krüger
GUI	Graphical User Interface (engl.)
HS160	Höhenstatus 160
INSPIRE	Infrastructure for Spatial Information in Europe (engl.)
LS100	Lagestatus 100
LGLN	Landesamt für Geoinformation und Landentwicklung Niedersachsen
MD5	Message-Digest Algorithm 5
MSVC++	Microsoft Visual C++
NHN	Normalhöhen-Null
NI	Niedersachsen
sn	Streifennummer
SRTM	Shuttle Radar Topography Mission
USGS	U.S. Geological Survey
UTM	Universal Transversal Mercator (engl.)
VKV	Niedersächsische Vermessungs- und Katasterverwaltung
zn	Zonennummer

## 1 Einleitung

### 1.1 Rahmenbedingungen

Am 15. Mai 2007 ist die Richtlinie zur Schaffung einer Geodateninfrastruktur (GDI) in der Europäischen Gemeinschaft (Infrastructure for Spatial Information in Europe, INSPIRE) in Kraft getreten. Im Hinblick auf die Interoperabilität von Geodaten unterschiedlicher Quellen ist die Einrichtung eines staatenübergreifenden einheitlichen Raumbezugssystems grundlegend. Mit dem Europäischen Terrestrischen Referenzsystem 1989 (ETRS89) liegt ein solches genaues und genügend verdichtetes System vor. Für die Abbildung in die Ebene ist die Universale Transversale Mercator-Abbildung (UTM) bekannt und weit verbreitet.

In der Bundesrepublik Deutschland obliegt das Vermessungswesen den einzelnen Bundesländern. Die zuständigen Verwaltungen wirken in der Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV) zusammen. Die Einführung von ETRS89 in Verbindung mit der UTM-Abbildung als Lagebezugssystem ist von der AdV bereits in den Jahren 1991 und 1995 beschlossen worden.

Die flächendeckende Versorgung mit Geobasisdaten nimmt in Niedersachsen die Niedersächsische Vermessungs- und Katasterverwaltung (VKV) wahr (vgl. Niedersächsisches Gesetz über das amtliche Vermessungswesen vom 12. Dezember 2002). Sie hat den gesetzlichen Auftrag, ein Landesbezugssystem vorzuhalten sowie die Liegenschaften und die Topografie des Landesgebietes nachzuweisen. Betroffenen und Dritten stellt die VKV aus den Nachweisen Angaben des amtlichen Vermessungswesens und Standardpräsentationen bereit. Als Bestandteil der VKV wirkt der Geschäftsbereich Landesvermessung und Geobasisinformation im Landesamt für Geoinformation und Landentwicklung Niedersachsen (LGLN) an den genannten Aufgaben und insbesondere an der Vorhaltung eines Landesbezugssystems mit.

Den Wechsel des amtlichen Lagebezugssystems auf ETRS89/UTM führt die VKV in Verbindung mit der Umstellung ihrer Datenbestände auf das neue einheitliche Datenmodell des Amtlichen Festpunktinformationssystems (AFIS), des Amtlichen Liegenschaftskatasterinformationssystems (ALKIS) und des Amtlichen Topographisch-Kartographischen Informationssystems (ATKIS) durch. Dabei wird die notwendige Transformation der Datenbestände als Vorverarbeitungsschritt zur Migration nach AFIS- ALKIS-ATKIS (AAA) vollzogen. Das LGLN verwendet für die Transformation der AFIS- und der ALKIS- Daten die Software GNTRANS\_NI. Im Hinblick auf INSPIRE wird GNTRANS\_NI durch das LGLN allgemein zur Verfügung gestellt. Den geodatenhaltenden Stellen in Niedersachsen wird somit der Wechsel des Lagebezugssystems für eigene Geodaten ermöglicht.

### 1.2 Amtliches Bezugssystem in Niedersachsen

Amtliches niedersächsisches Lagebezugssystem ist bis zum ETRS89/UTM-Einführungserlass der Lagestatus 100 (LS100). Dabei handelt es sich um Koordinaten im Deutschen Hauptdreiecksnetz (DHDN), die mithilfe der Gauß-Krüger (GK)-Vorschriften in den drei Meridianstreifensystemen 2, 3 oder 4 abgebildet werden.

In der Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens (GeoInfoDok) werden für Koordinatenreferenzsysteme (CRS) Bezeichner mit folgendem Aufbau definiert (AdV 2009):

[Land]\_[geodätisches Datum]\_[Koordinatensystem]\_[Submerkmale des Koordinatensystems].

Die Submerkmale können beispielsweise den landesspezifischen Lagestatus umfassen. Demzufolge wird der niedersächsische LS100 wie folgt benannt:

DE\_DHDN\_3GK<sn>\_NI100.

Der Platzhalter <sn> beinhaltet die Nummer des GK-Meridianstreifens, so dass der LS100 durch DE\_DHDN\_3GK2\_NI100, DE\_DHDN\_3GK3\_NI100 und DE\_DHDN\_3GK4\_NI100 identifiziert werden kann. Das CRS ETRS89/UTM heißt im Sinne der Terminologie der GeoInfoDok ETRS89\_UTM<zn>.

Da Niedersachsen vollständig in der UTM-Zone mit der Zonennummer (<zn>) 32 liegt, wird das künftige Lagebezugssystem also mit ETRS89\_UTM32 angesprochen.

Der Bezeichner für ellipsoidische Koordinaten im DHDN ist DE\_DHDN\_Lat-Lon, für ellipsoidische Koordinaten im ETRS89 lautet er ETRS89\_Lat-Lon.

Dreidimensionale kartesische Koordinaten im ETRS89 werden mit ETRS89\_X-Y-Z angegeben.

CRS der Lage (<crsL>) können mit CRS der Höhe (<crsH>) wie folgt verknüpft werden: <crsL>\*<crsH>, z. B. DE\_DHDN\_3GK3\_NI100\*DE\_DHHN\_HS160 für Punkte, die Lagekoordinaten im LS100 und eine Höhe im Höhenstatus 160 (HS160) aufweisen.

### 1.3 Transformationssoftware für den Bezugssystemwechsel

#### 1.3.1 Nutzungsbedingungen

Bei der Nutzung von GNTRANS\_NI sind die entsprechenden Nutzungsbedingungen zu beachten, die unter [http://www.lgln.de/gntrans\\_ni](http://www.lgln.de/gntrans_ni) zur Verfügung gestellt werden.

#### 1.3.2 Leistungsumfang von GNTRANS\_NI

GNTRANS\_NI ist weder für Transformationen in den benachbarten Bundesländern vorgesehen noch für diese geeignet.

GNTRANS\_NI beruht auf der Software GNTRANS der Firma Geo++<sup>®</sup> Gesellschaft für satellitengestützte geodätische und navigatorische Technologien GmbH. Es ermöglicht eine landesweite Datumstransformation zwischen folgenden CRS der Lage:

- DE\_DHDN\_3GK2\_NI100 ↔ ETRS89\_UTM32,
- DE\_DHDN\_3GK3\_NI100 ↔ ETRS89\_UTM32,
- DE\_DHDN\_3GK4\_NI100 ↔ ETRS89\_UTM32.

GNTRANS\_NI enthält zwei Komponenten für windowsbasierte Computersysteme (s. Abbildung 1):

- GNTRANS\_NI EXEC kann als grafische Benutzeroberfläche (Graphical User Interface, GUI) oder über die Kommandozeile (Command Line Interface, CLI) genutzt werden;
- GNTRANS\_NI API beinhaltet eine Schnittstelle zur Anwendungsprogrammierung (Application Programming Interface, API) für eine flexible Einbindung in externe Programme; die API ist als dynamische Programmbibliothek (Dynamic Link Library, DLL) ausgestaltet.

Zielgruppen von GNTRANS\_NI EXEC sind Direktanwender, die Punktlisten im Format American Standard Code for Information Interchange (ASCII) oder Einzelpunkte transformieren möchten. Software-Entwickler benötigen hingegen für die individuelle Einbindung in externe Programme die GNTRANS\_NI API.

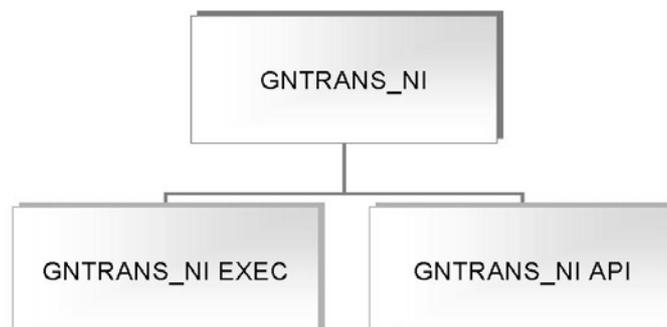


Abbildung 1: Pakete der Transformationssoftware GNTRANS\_NI

Die Pakete unterscheiden sich hinsichtlich der Unterstützung der möglichen CRS (s. Tabelle 1).

Tabelle 1: Leistungsumfang der GNTRANS\_NI-Pakete

Unterstütztes CRS	GNTRANS_NI EXEC		GNTRANS_NI API
	GUI	CLI	
DE_DHDN_3GK<sn>_NI100	+	+	+
DE_DHDN_3GK3_NI100	+	+	+
DE_DHDN_3GK4_NI100	+	+	+
ETRS89_UTM32	+	+	+
DE_DHDN_Lat-Lon	-	-	+
ETRS89_Lat-Lon	-	+	+
ETRS89_X-Y-Z	-	-	± <sup>1</sup>

### 1.3.3 Das Transformationsmodell Niedersachsen

Das in GNTRANS\_NI implementierte amtliche Transformationsmodell Niedersachsen basiert auf Festpunkten aus dem AFIS. Sein räumlicher Geltungsbereich ist das Landesgebiet von Niedersachsen. Technisch wird dieser Geltungsbereich durch die Ausdehnung des DGM-T begrenzt (s. Abschnitt 1.3.4). Nur innerhalb des vom DGM-T abgedeckten Gebiets kann die Transformation von Lagekoordinaten durchgeführt werden. Dabei ist zu beachten, dass die Transformation innerhalb der Landesgebiets von Bremen zwar technisch möglich ist (s. Abbildung 2), aber das Transformationsmodell Niedersachsen nicht für Bremen gilt.

Qualitativ repräsentiert das Transformationsmodell Niedersachsen die amtlichen Festpunktfelder zum Zeitpunkt der Umstellung der Datenbestände des AFIS und des ALKIS auf ETRS89/UTM.

In technischer Hinsicht gilt das Transformationsmodell Niedersachsen nur bei Verwendung innerhalb der Software GNTRANS\_NI.

### 1.3.4 Höheninformationen für die Transformation

Für die Transformation werden Höheninformationen mit einer Genauigkeit von  $\pm 15$  m benötigt. Bei dieser Genauigkeit ist der Einfluss der Höhe auf die transformierten Lagekoordinaten geringer als 1 mm. Um eine entsprechende Genauigkeit und die Übereinstimmung mit den AFIS-/ALKIS-Geometrien im Bereich eines Millimeters auch für Dritte sicherzustellen, ist in GNTRANS\_NI ein Digitales Geländemodell für die Transformation (DGM-T) integriert. Es liefert innerhalb von Niedersachsen zu jedem Punkt auf der Geländeoberfläche die notwendige Höheninformation.

Im Hinblick auf das DGM-T ist zu beachten, dass jegliche Höheninformation des Nutzers im Berechnungsvorgang durch die DGM-T-Höhe ersetzt wird. Ausgegeben wird jedoch stets wieder die eingegebene Höhe. Damit geht einher, dass die mit GNTRANS\_NI erzielten Transformationsergebnisse bezüglich der Höhenlage nur für Punkte gelten, die auf der Erdoberfläche liegen bzw. maximal 15 m von dieser abweichen.

Das DGM-T ist innerhalb des niedersächsischen Landesgebiets und mindestens einen Kilometer über die Landesgrenze hinaus flächendeckend hinterlegt (s. Abbildung 2).

Im Bereich der Landfläche des Landes Niedersachsen liegen dem DGM-T Daten des Amtlichen Topographisch-Kartographischen Informationssystems (ATKIS) zugrunde. Im Gebiet der Nordsee und bei weiteren Landflächen außerhalb des Landes Niedersachsen sind Daten der Shuttle Radar Topography Mission (SRTM) des U.S. Geological Survey (USGS) ergänzt (USGS 2011). Entsprechende Informationen werden in einer Metadatendatei im Shapefile-Format („DGM-T\_Info\_V2.shp“) zum Download unter [http://www.lgln.de/gntrans\\_ni\\_bereitgestellt](http://www.lgln.de/gntrans_ni_bereitgestellt). Diese Metadatendatei enthält die Umringe der Einheiten des DGM-T im CRS ETRS89\_UTM32; Kartenmaterial ist nicht eingeschlossen. Zu jedem Umring sind die folgenden Informationen hinterlegt:

- 8-stellige Nummer der Einheit (Datenfeld „NBZ\_GK“),
- originäres CRS der Einheit (Datenfeld „NBZ\_OriginalCRS“) und
- Datenquelle (Datenfeld „DATASOURCE“).

Eine mögliche Visualisierung der Metadaten zum DGM-T zeigt Abbildung 3.

<sup>1</sup> Voraussetzung: Es liegt eine ellipsoidische Höhe vor (s. auch Hinweise zur Verwendung von Höheninformationen unter Abschnitt 1.3.4)

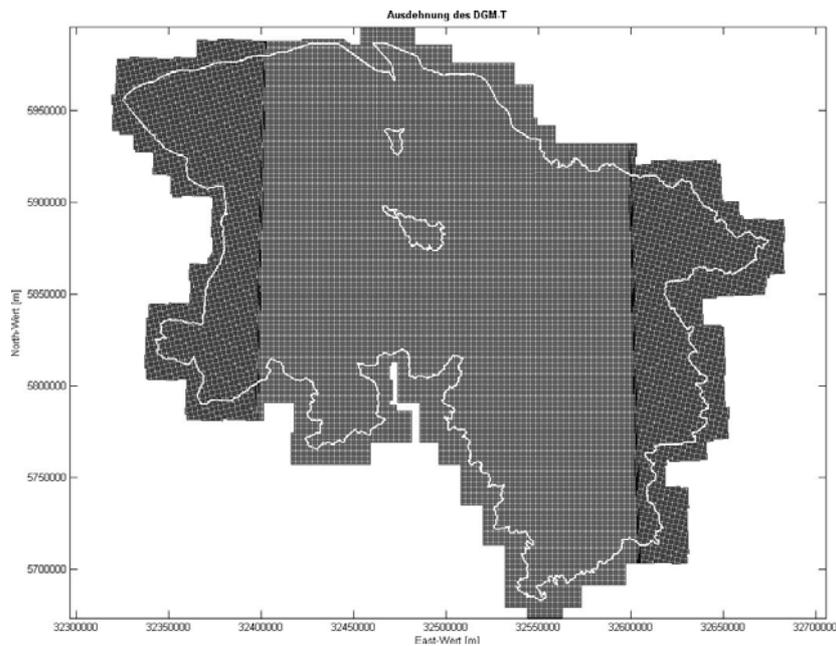


Abbildung 2: Ausdehnung des DGM-T im ETRS89/UTM (Geltungsbereich von GNTRANS\_NI)

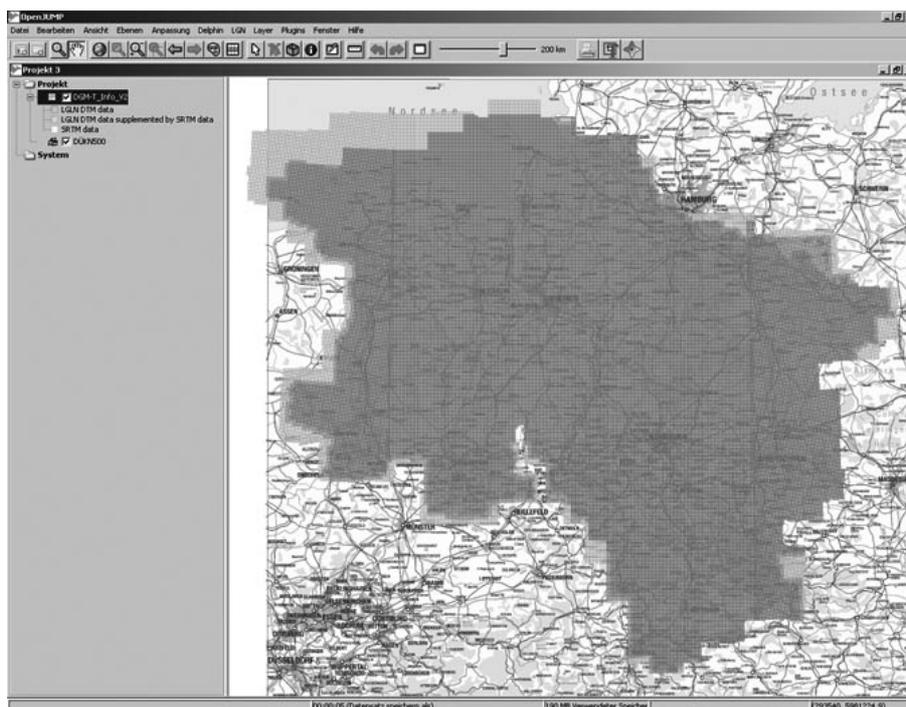


Abbildung 3: Metadaten zum DGM-T, thematisch gefärbt in Abhängigkeit der Datenquelle

### 1.3.5 Eigenschaften der Transformation

Das Transformationsmodell Niedersachsen ermöglicht über die gesamte Landesfläche die einheitliche, stetige, homogene, nachbarschaftstreue sowie in der Hin- und Rücktransformation eindeutige Datumstransformation zwischen den Lagebezugssystemen DE\_DHDN\_3GK<sn>\_NI100 und ETRS89\_UTM<zn>.

Die Genauigkeit der Transformation wird wesentlich von der Koordinatenqualität des Startsystems bestimmt. Das Transformationsmodell Niedersachsen bietet für den Wechsel des Lagebezugssystems eine Genauigkeit, die besser als 2 cm ist.

Die mit GNTRANS\_NI erzielten Ergebnisse gelten streng für Punkte auf der Erdoberfläche.

## 2 GNTRANS\_NI

### 2.1 Installation

Voraussetzung für die Installation von GNTRANS\_NI EXEC ist eine freie Festplattenkapazität von etwa 550 MB, vorzugsweise auf Partition „C:“ des Zielrechners. Für den Hauptspeicher ist eine Größe von 2 GB ausreichend. Die Sicherheitsrichtlinien des Zielsystems erfordern gegebenenfalls Administrationsrechte.

Der parallele Betrieb mit anderen Software-Produkten der Firma Geo++® führt zu keinen Konflikten, da durch das Setup keine Umgebungsvariablen gesetzt werden.

Die Installation von GNTRANS\_NI wurde unter Microsoft Windows 7 Enterprise Service Pack 1 32-/64-Bit getestet.

GNTRANS\_NI wird als ZIP-Archiv „GNTRANS\_NI\_V1-4-4\_setup.zip“ ausgeliefert. Das Archiv enthält das Installationsprogramm „GNTRANS\_NI\_V1-4-4\_setup.exe“ und ist zu entpacken.

Nach Ausführen der Datei „GNTRANS\_NI\_V1-4-4\_setup.exe“ durch Doppelklick, kann das Setup fortgesetzt bzw. abgebrochen werden (s. Abbildung 4).

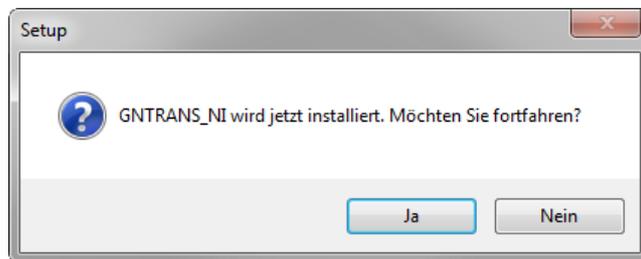


Abbildung 4: Schritt 1 der Installation von GNTRANS\_NI

Im nächsten Schritt werden die Nutzungsbedingungen angezeigt (s. Abbildung 5). Das Einverständnis zu diesen Nutzungsbedingungen ist zu bestätigen, bevor die Installation fortgesetzt werden kann.

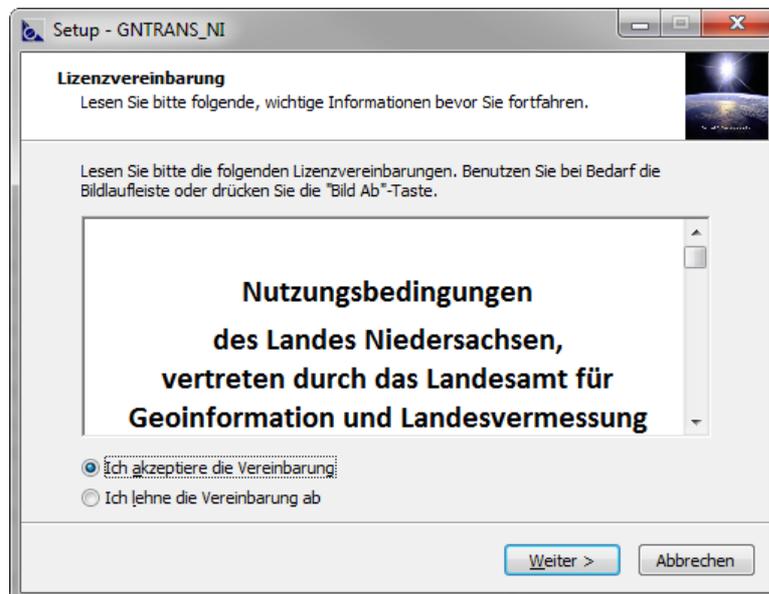


Abbildung 5: Schritt 2 der Installation von GNTRANS\_NI

Im nachfolgenden Dialog lässt sich der vorgegebene Standardordner anpassen (s. Abbildung 6).

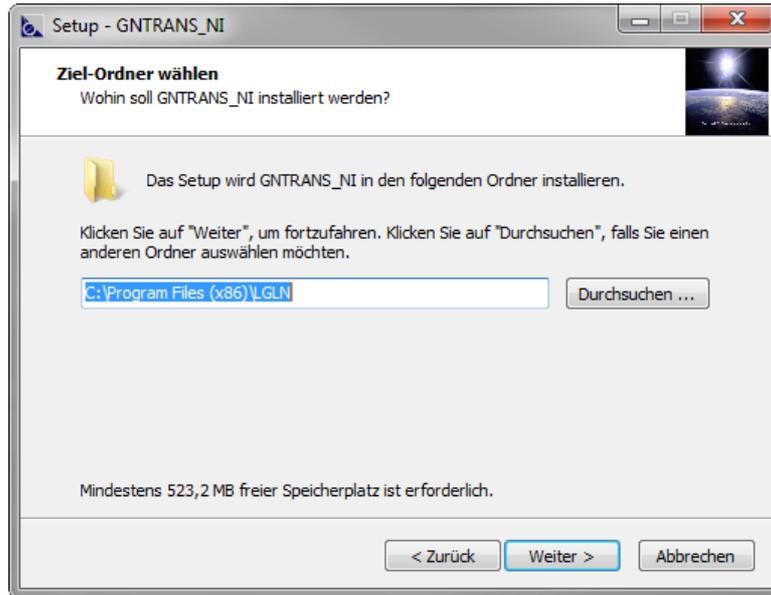


Abbildung 6: Schritt 3 der Installation von GNTTRANS\_NI

Der Name des Startmenü-Ordners zur Erstellung der Verknüpfungen kann angepasst werden (s. Abbildung 7).

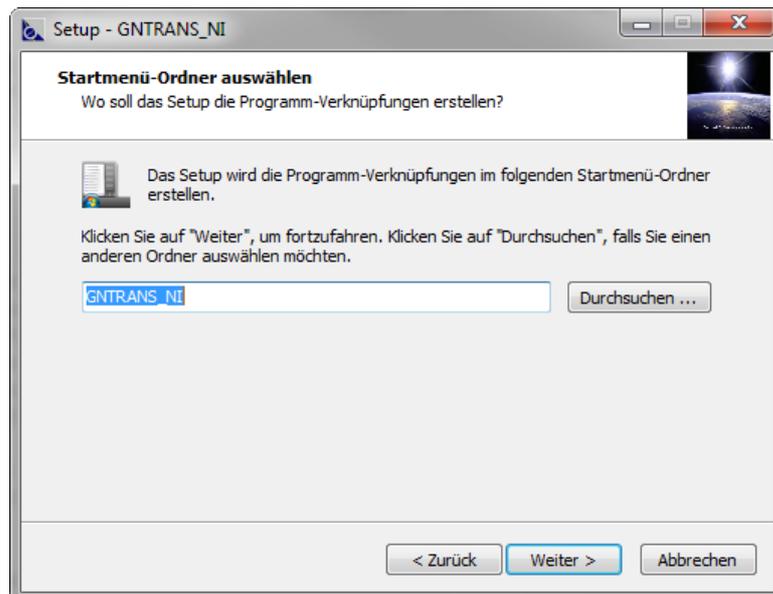


Abbildung 7: Schritt 4 der Installation von GNTTRANS\_NI

Optional wird eine Verknüpfung zur grafischen Benutzeroberfläche wingntrans\_ni.exe erstellt (s. Abbildung 8).

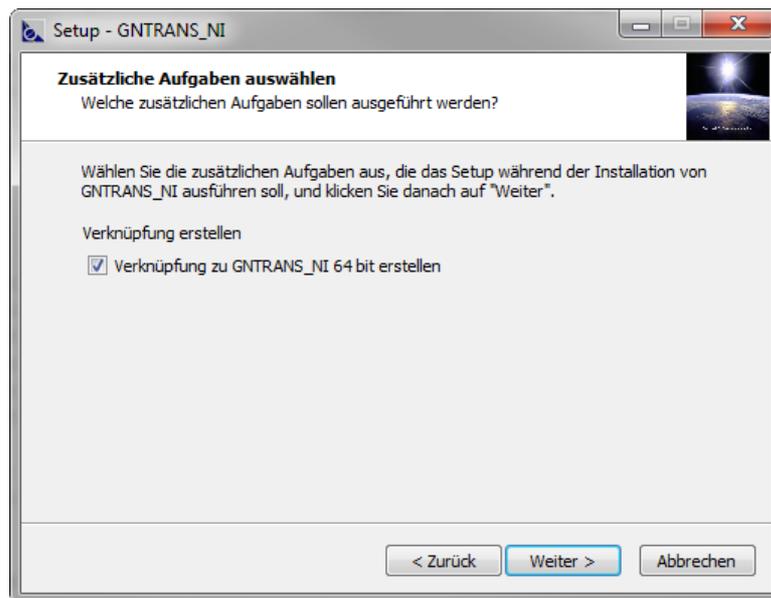


Abbildung 8: Schritt 5 der Installation von GNTRANS\_NI

Die getroffenen Einstellungen werden angezeigt (s. Abbildung 9).

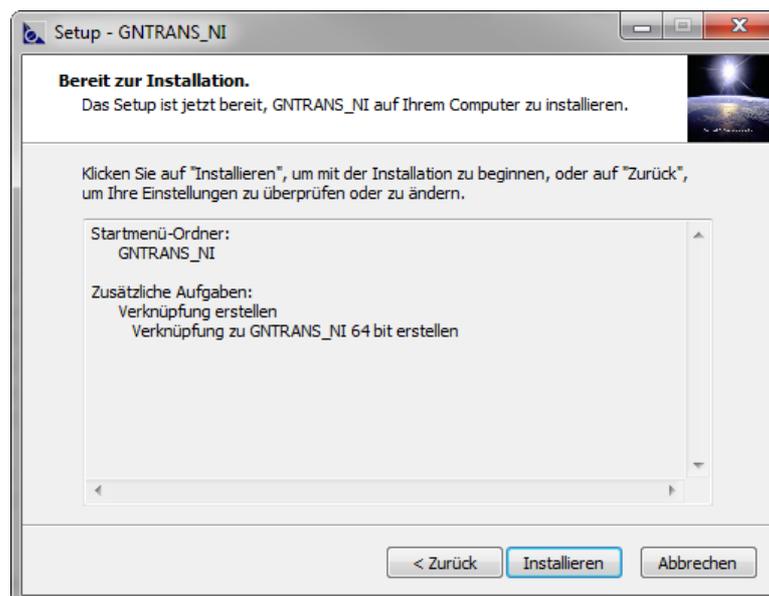


Abbildung 9: Schritt 6 der Installation von GNTRANS\_NI

Nach Abschluss des Setups von GNTRANS\_NI können bei Bedarf die Microsoft Visual C++ 2008 Redistributable Packages installiert werden. Sie enthalten die Laufzeitkomponenten der Visual C++-Bibliotheken, die erforderlich sind, um Anwendungen, die in Visual C++ entwickelt wurden, auf einem Computer auszuführen, auf dem kein Visual C++ 2008 installiert ist (s. Abbildung 10).

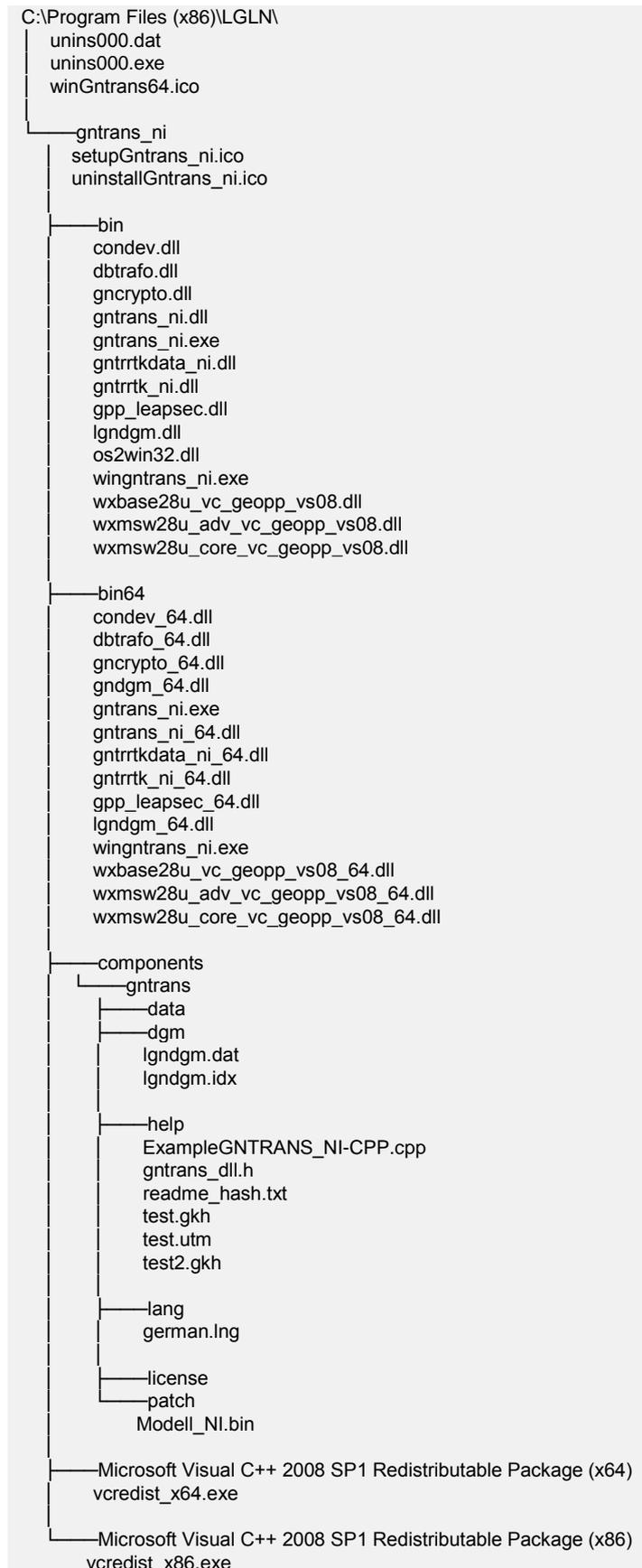


Abbildung 10: Schritt 6 der Installation von GNTRANS\_NI

Mit GNTRANS\_NI werden unter *Installationspfad*gntrans\_ni\components\gntrans\help\ in der Textdatei „readme\_hash.txt“ Message-Digest Algorithm (MD5)-Prüfsummen für das Transformationsmodell Niedersachsen („Modell\_NI.bin“) und für die Dateien des DGM-T („LGNDGM.dat“ und „LGNDGM.idx“) bereitgestellt. Es ist zu empfehlen, diese Prüfsummen für die genannten Dateien nach der Installation lokal zu ermitteln und mit den bereitgestellten Werten zu vergleichen. Bei Nicht-Übereinstimmung sind möglicherweise bei der Datenübertragung Fehler entstanden.

## 2.2 Verzeichnisstruktur

Nach der Installation im Standardverzeichnis „C:\Program Files (x86)\LGLN“ liegt die nachfolgend dargestellte Verzeichnisstruktur mit den angegebenen Dateien vor.



## 2.3 GUI-Funktionalitäten

Die grafische Benutzeroberfläche im Paket GNTRANS\_NI EXEC, auch WinGNTRANS\_NI genannt, ist eine Applikation zur grafischen Steuerung von GNTRANS\_NI. WinGNTRANS\_NI ermöglicht die Bedienung per Maussteuerung im Windows-Stil. Es ist für die Transformation von Einzelpunkten und ASCII-Punktlisten geeignet.

### 2.3.1 Einzelpunkte transformieren

Nach dem Start durch Ausführen der Datei „wingntrans\_ni.exe“ präsentiert sich WinGNTRANS\_NI wie in Abbildung 11. Auf der Eingabeseite können die Koordinaten (Rechts- und Hochwert) im Eingabesystem (DHDN) eingegeben werden.

Format und Projektion sind durch das ausgewählte Koordinatensystem festgelegt. Als Modell steht allein das Transformationsmodell Niedersachsen zur Verfügung.

Nach Mausklick auf die Schaltfläche „Transformieren“ werden die gesuchten Koordinaten im Ausgabesystem (ETRS89) auf der rechten Seite angezeigt (s. Abbildung 11). Für die Rücktransformation von Koordinaten ist links unter Eingabe ETRS89 und rechts unter Ausgabe DHDN auszuwählen (s. Abbildung 12). Die Koordinaten (East und North) sind wie zuvor einzugeben und zu transformieren.

Die für die Transformation der Lagekoordinaten notwendige Höheninformation wird mit entsprechender Genauigkeit aus dem DGM-T abgeleitet (vgl. Abschnitt 1.3.4). Es ist daher nicht notwendig, eigene Höhenangaben zu nutzen. Allerdings ist es nicht möglich, Höhen aus dem DGM-T im Ausgabesystem zu erhalten.

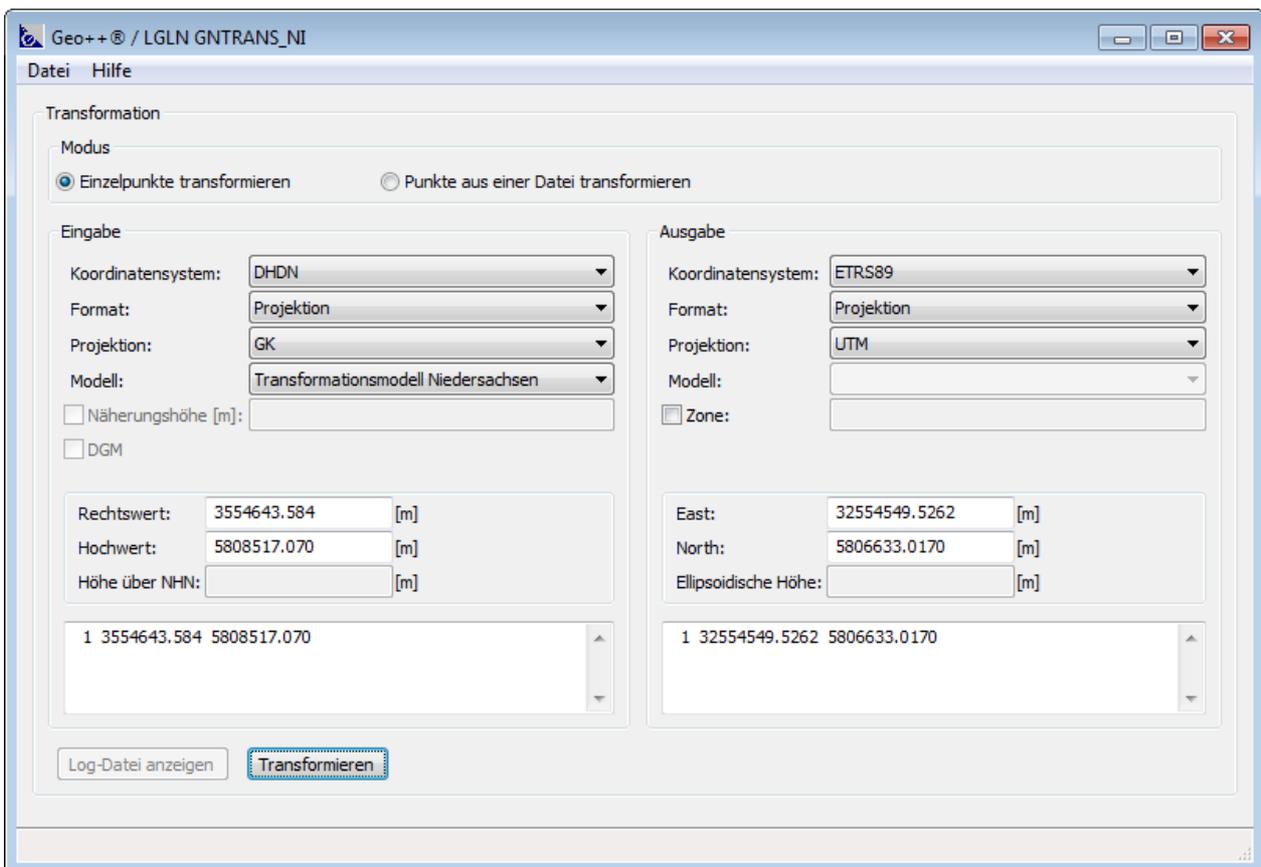


Abbildung 11: Transformation von Einzelpunkten von DHDN/GK nach ETRS89/UTM mit WinGNTRANS\_NI

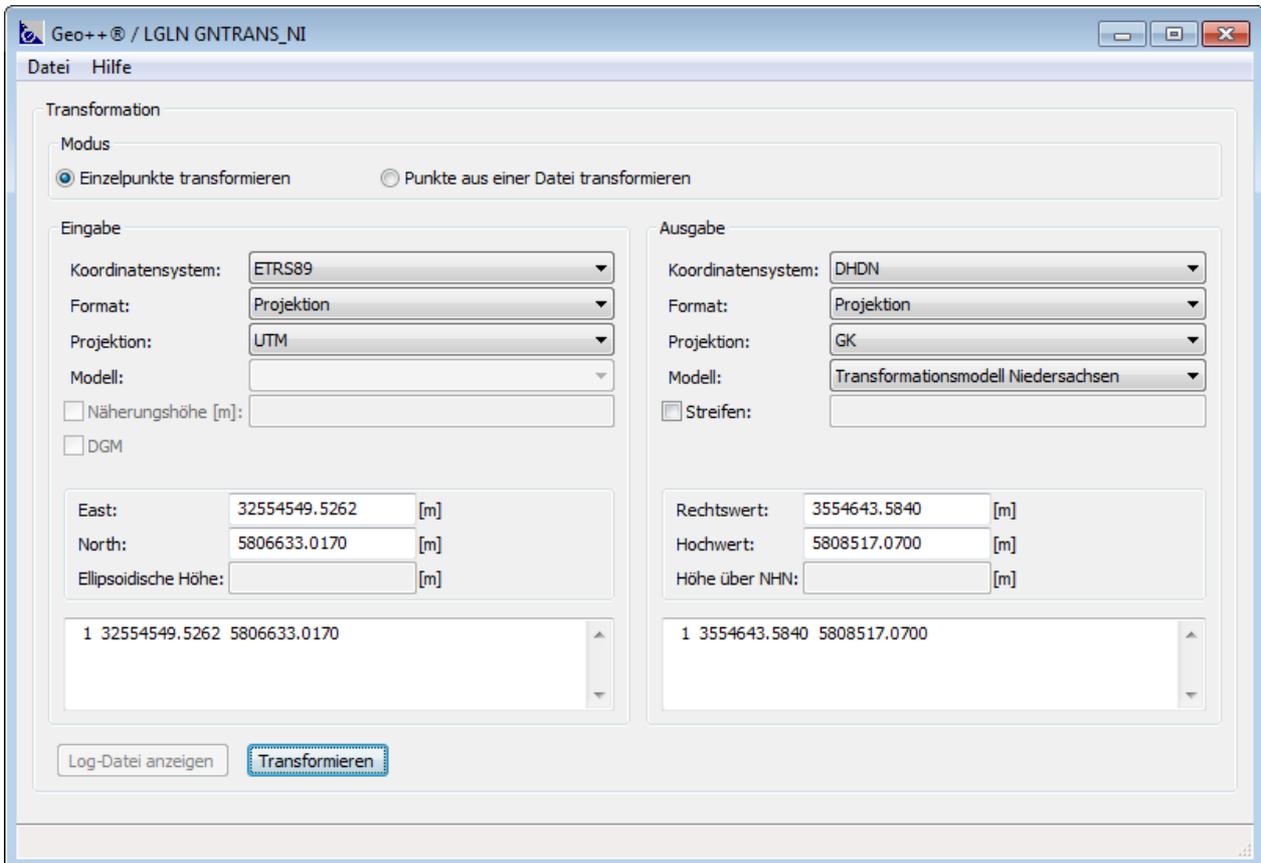


Abbildung 12: Transformation von Einzelpunkten von ETRS89/UTM nach DHDN/GK mit WinGNTRANS\_NI

Wie Abbildung 13 zeigt, können über das Eingabefeld „Streifen“ bzw. „Zone“ (benachbarte) Abbildungsstreifen bzw. -zonen angegeben werden. Ohne die Auswahl dieser Option wird die Streifen- bzw. Zonenzuordnung automatisch durchgeführt.

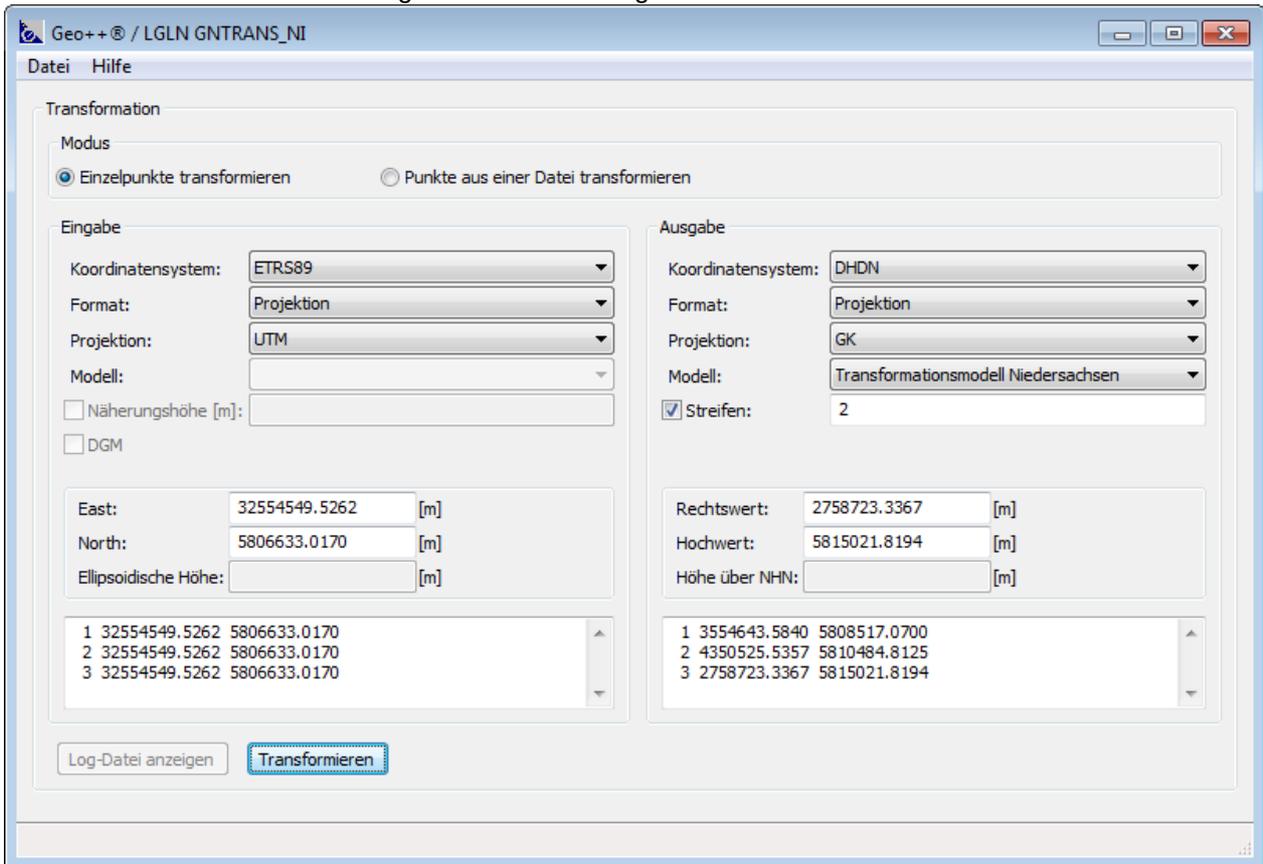


Abbildung 13: Auswahl benachbarter Abbildungsstreifen bzw. -zonen

### 2.3.2 Punkte aus einer Datei transformieren

In diesem Modus wird eine Eingabedatei (ASCII-kodiert) mit Koordinaten im Eingabesystem ausgewählt. Weitere Funktionalitäten wie die Auswahl des Eingabe- und des Ausgabekoordinatensystems stellen sich, wie unter Abschnitt 2.3.1 beschrieben, dar.

Die Eingabe- bzw. Ausgabedatei kann jeweils über die Schaltfläche „Durchsuchen...“ im Windows-Dateisystem ausgesucht werden (s. Abbildung 17 auf der Folgeseite). Dabei werden im Suchordner bei der Angabe des Dateitypen voreingestellt die Suffixe „gkh“ bzw. „utm“ angezeigt; über die Angabe von „Alle Dateien (\*.\*)“ können weitere Dateien beliebigen Typs gefunden werden (s. Abbildung 16).

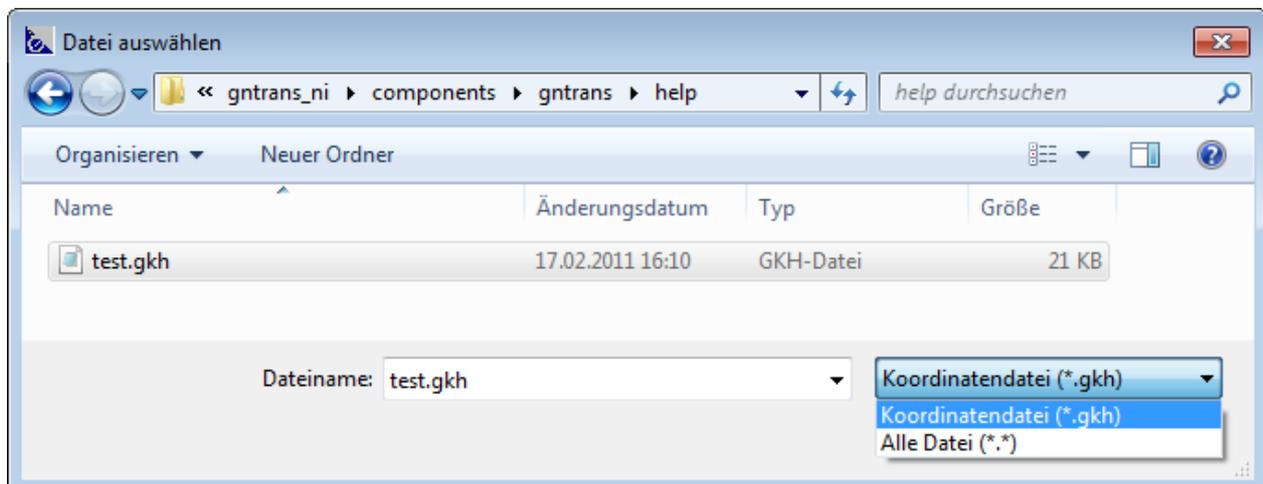


Abbildung 14: Datei auswählen

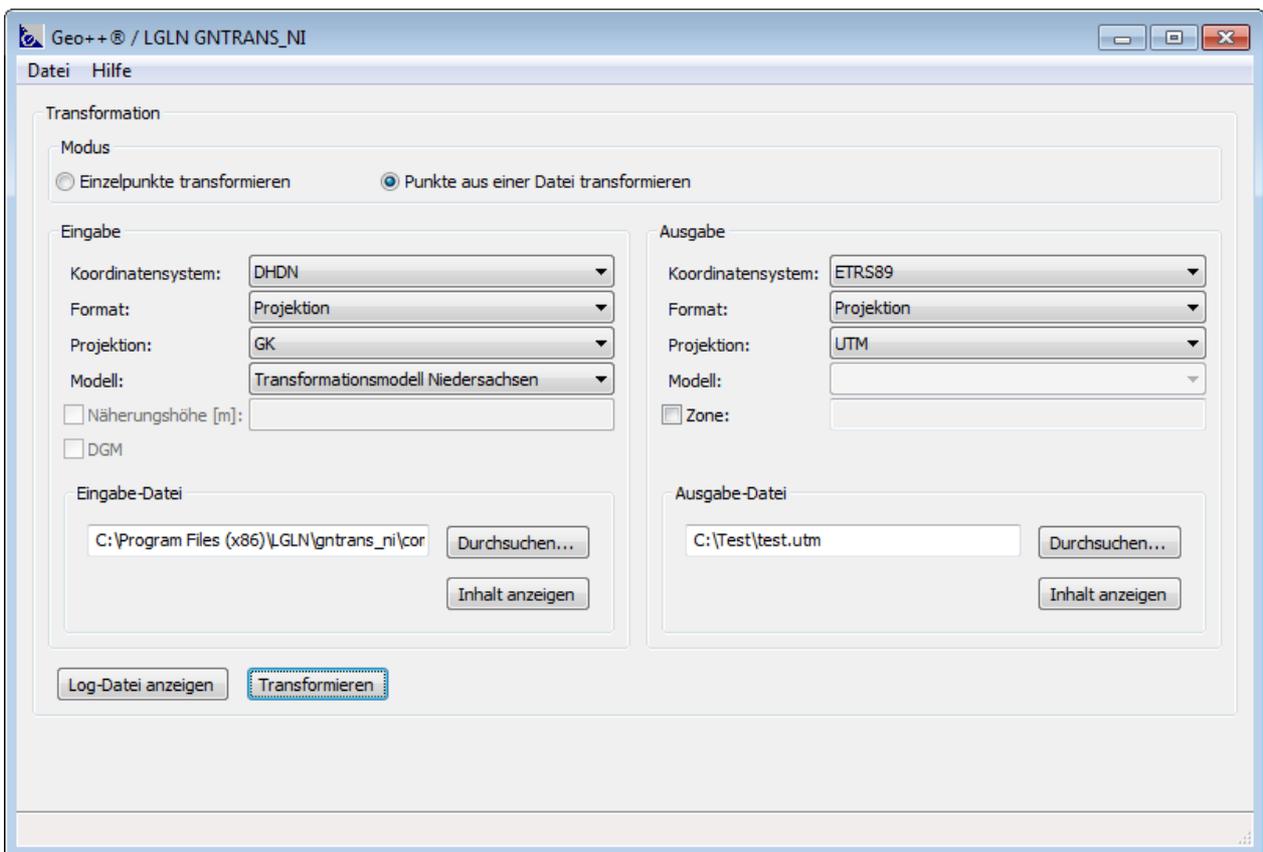


Abbildung 15: Punkte aus einer Datei mit WinGTRANS\_NI transformieren

Die Eingabedatei muss generell aus drei oder vier Spalten bestehen, deren Inhalte in der Zeile frei verteilt sein können. Beispielsweise sind die nachfolgenden drei Dateiausschnitte gleichwertig transformierbar.

1. Dateiausschnitt:

1	3593500.000	5724000.000	248.800
2	3593500.000	5725000.000	249.800
3	3593500.000	5726000.000	250.000
RS4	3593500.000	5727000.000	325.300
5	3593500.000	5728000.000	398.600
		35570001000	3593500.000
7	3593500.000	5730000.000	498.700
8	3593500.000	5731000.000	429.800
###9	3593500.000	5732000.000	682.000
10	3593500.000	5733000.000	575.400
11	3593500.000	5734000.000	412.500
12	3593500.000	5735000.000	356.300

2. Dateiausschnitt:

1	3593500.000	5724000.000	248.800
2	3593500.000	5725000.000	249.800
3	3593500.000	5726000.000	250.000
RS4	3593500.000	5727000.000	325.300
5	3593500.000	5728000.000	398.600
35570001000	3593500.000	5729000.000	400.300
7	3593500.000	5730000.000	498.700
8	3593500.000	5731000.000	429.800
###9	3593500.000	5732000.000	682.000
10	3593500.000	5733000.000	575.400
11	3593500.000	5734000.000	412.500
12	3593500.000	5735000.000	356.300

### 3. Dateiausschnitt:

1	3593500.000	5724000.000
2	3593500.000	5725000.000
3	3593500.000	5726000.000
RS4	3593500.000	5727000.000
5	3593500.000	5728000.000
35570001000	3593500.000	5729000.000
7	3593500.000	5730000.000
8	3593500.000	5731000.000
###9	3593500.000	5732000.000
10	3593500.000	5733000.000
11	3593500.000	5734000.000
12	3593500.000	5735000.000

Die Reihenfolge Punktnummer, Rechtswert, Hochwert, [Höhe] ist einzuhalten. Im Falle der optionalen Höhenangabe muss diese in sämtlichen Zeilen den Lagekoordinaten folgen. Werden diese Formatvorschriften nicht beachtet, erscheint bei Nutzung der Schaltflächen „Inhalt anzeigen“ eine Fehlermeldung (s. Abbildung 16).

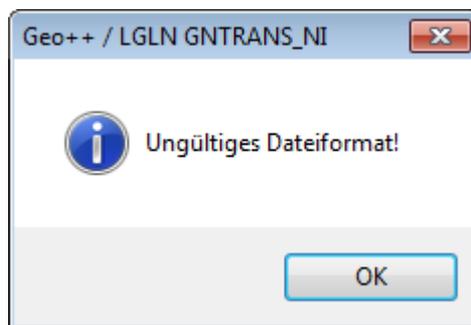


Abbildung 16: Fehlermeldung bei Nichteinhaltung von Formatvorgaben

Sofern die Eingabedatei sowohl Punkte mit und ohne Höhenwert enthält, werden die Punkte nur bis zum ersten Punkt transformiert, dessen Datenstruktur von den vorhergehenden abweicht. Zusätzlich erscheint die in Abbildung 17 dargestellte Fehlermeldung.



Abbildung 17: Fehlermeldung bei Dateien, deren Datensätze im Hinblick auf die Höhe der Punkte inhomogen sind

Bei Einhaltung der genannten Formatanforderungen kann über „Inhalt anzeigen“ der Inhalt der Eingabedatei vor der Transformation betrachtet werden. Liegen durchgehend Höhenangaben vor, so wird hierzu ein Hinweis geliefert und die Anzeige auf die Lagekoordinaten beschränkt (s. Abbildung 18).

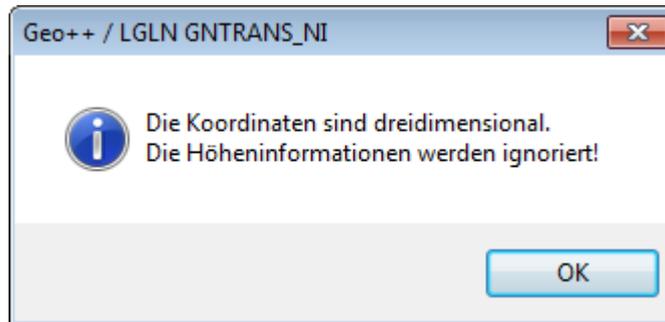


Abbildung 18: Hinweis auf den Umgang mit nutzerseitigen Höheninformationen

Wenn die anzuzeigende Datei eine gewisse Größe überschreitet, wird dem Anwender empfohlen, einen effizienteren Editor zur Anzeige des Dateiinhalts zu verwenden (s. Abbildung 19).

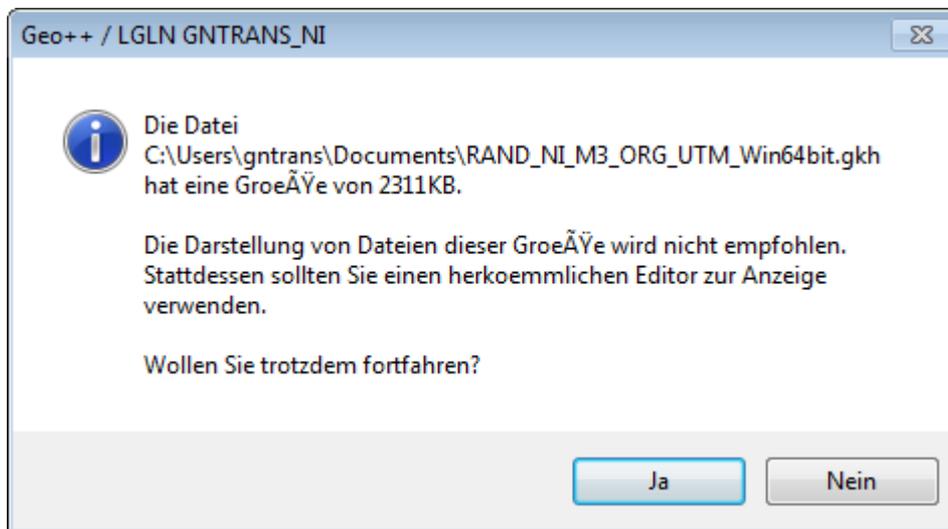
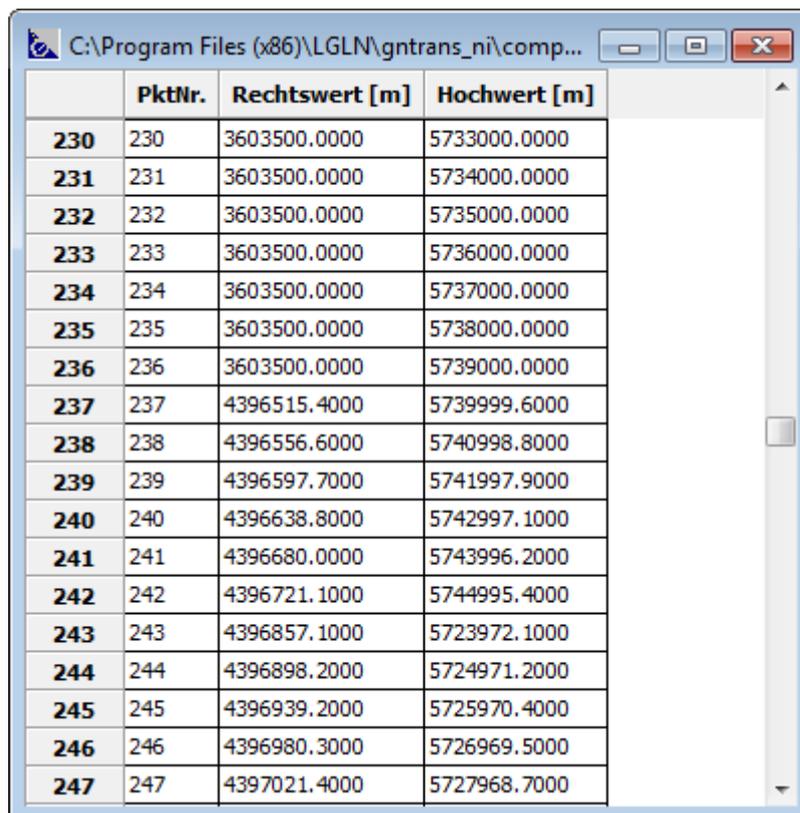


Abbildung 19: Warnung beim Anzeigen umfangreicher Dateien



The screenshot shows a window titled "C:\Program Files (x86)\LGLN\gntrans\_ni\comp...". Inside the window is a table with four columns: "PktNr.", "Rechtswert [m]", and "Hochwert [m]". The table contains 18 rows of data, with the first column containing bolded numbers from 230 to 247. The second column contains point numbers from 230 to 247. The third and fourth columns contain coordinate values in meters.

	PktNr.	Rechtswert [m]	Hochwert [m]
<b>230</b>	230	3603500.0000	5733000.0000
<b>231</b>	231	3603500.0000	5734000.0000
<b>232</b>	232	3603500.0000	5735000.0000
<b>233</b>	233	3603500.0000	5736000.0000
<b>234</b>	234	3603500.0000	5737000.0000
<b>235</b>	235	3603500.0000	5738000.0000
<b>236</b>	236	3603500.0000	5739000.0000
<b>237</b>	237	4396515.4000	5739999.6000
<b>238</b>	238	4396556.6000	5740998.8000
<b>239</b>	239	4396597.7000	5741997.9000
<b>240</b>	240	4396638.8000	5742997.1000
<b>241</b>	241	4396680.0000	5743996.2000
<b>242</b>	242	4396721.1000	5744995.4000
<b>243</b>	243	4396857.1000	5723972.1000
<b>244</b>	244	4396898.2000	5724971.2000
<b>245</b>	245	4396939.2000	5725970.4000
<b>246</b>	246	4396980.3000	5726969.5000
<b>247</b>	247	4397021.4000	5727968.7000

Abbildung 20: Anzeige von Dateiinhalt am Beispiel der Eingabedatei „test.gkh“

Das Beispiel in Abbildung 20 verdeutlicht, dass es grundsätzlich möglich ist, Koordinaten aus verschiedenen Abbildungsstreifen bzw. -zonen in einer Eingangsdatei vorzuhalten und zu transformieren.

Nachdem die Eingabe- und die Ausgabedatei festgelegt worden sind, wird die Transformation durch Betätigen der Schaltfläche „Transformieren“ ausgelöst. Der Transformationserfolg wird mit einem entsprechenden Hinweis dokumentiert (s. Abbildung 21). Sind in der Eingangsdatei Koordinaten außerhalb des DGM-T und damit außerhalb von Niedersachsen vorhanden, wird die Meldung in Abbildung 22 ausgegeben.

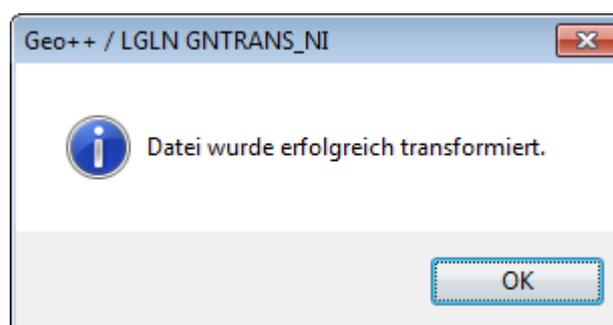


Abbildung 21: Hinweis bei erfolgreicher Transformation

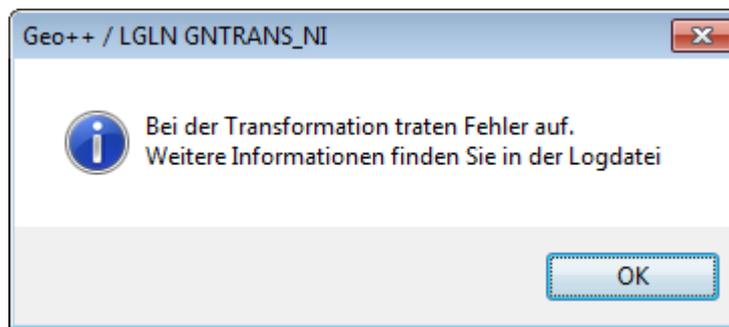


Abbildung 22: Hinweis auf Fehler bei der Transformation

Nach der Transformation können die neu berechneten Koordinaten im Ausgabesystem über „Inhalt anzeigen“ betrachtet werden (s. Abbildung 23).

	PktNr.	East [m]	North [m]
<b>230</b>	230	32603385.3957	5731145.3321
<b>231</b>	231	32603385.4107	5732144.9371
<b>232</b>	232	32603385.4262	5733144.5422
<b>233</b>	233	32603385.4432	5734144.1486
<b>234</b>	234	32603385.4599	5735143.7512
<b>235</b>	235	32603385.4769	5736143.3547
<b>236</b>	236	32603385.4939	5737142.9577
<b>237</b>	237	32603385.4376	5738142.4751
<b>238</b>	238	32603385.5155	5739142.1233
<b>239</b>	239	32603385.4911	5740141.6647
<b>240</b>	240	32603385.4573	5741141.3054
<b>241</b>	241	32603385.5234	5742140.8530
<b>242</b>	242	32603385.4799	5743140.4947
<b>243</b>	243	32604384.8179	5722148.8748
<b>244</b>	244	32604384.8767	5723148.4199
<b>245</b>	245	32604384.8265	5724148.0674
<b>246</b>	246	32604384.8759	5725147.6173
<b>247</b>	247	32604384.9167	5726147.2675

Abbildung 23: Anzeige von Dateiinhalt am Beispiel der Ausgabedatei „test.utm“

Das Ergebnis wird in der Ausgabedatei im ASCII-Format abgespeichert. Die Höhen der Eingabedatei werden unverändert in die Ausgabedatei übertragen, damit bereits vorhandene physikalische Höhen im neuen Lagebezugssystem weiter genutzt werden können.

*Ausschnitt aus der Ausgabedatei „test.utm“:*

1	32593389.2731	5722149.0173	248.8000
2	32593389.2867	5723148.6179	249.8000
3	32593389.2982	5724148.2197	250.0000
4	32593389.3089	5725147.8246	325.3000
5	32593389.3200	5726147.4290	398.6000
6	32593389.3305	5727147.0303	400.3000
7	32593389.3424	5728146.6334	498.7000
8	32593389.3560	5729146.2337	429.8000
9	32593389.3708	5730145.8423	682.0000
10	32593389.3869	5731145.4410	575.4000
11	32593389.4071	5732145.0359	412.5000
12	32593389.4263	5733144.6349	356.3000

Neben der Ausgabedatei legt WinGNTRANS\_NI eine Ereignisprotokolldatei (Log-Datei) an (s. Abbildung 24). Diese wird unter Windows 7 im Pfad „C:\Users\gntrans\geopp\gntrans\log“ (Unterordner [gntrans] entspricht dem angemeldeten Benutzer) bereitgestellt. Die Log-Datei trägt, tageweise separiert, folgende Namensstruktur: „Gntrans\_NI\_JAHR-MM-TT.log“.

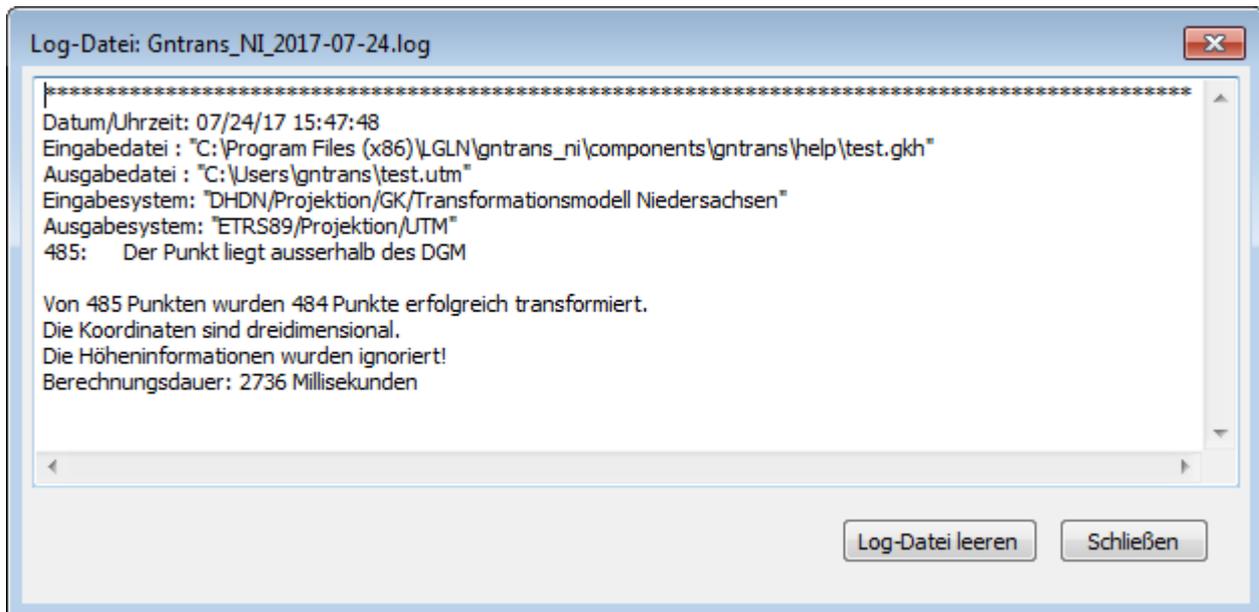


Abbildung 24: Beispiel für den Inhalt einer Log-Datei

## 2.4 CLI-Funktionalitäten

GNTRANS\_NI kann in einer zweiten Applikation durch Aufruf an der Eingabekonsole oder in Stapelverarbeitung aus einer so genannten Batch-Datei heraus gestartet werden. Koordinaten können damit aus einer Datei heraus, nicht jedoch durch einzelne Eingabe transformiert werden.

Nach Aufruf von „GNTRANS\_NI“ auf der Kommandozeile der Windows 7-Eingabeaufforderung mit dem Befehl „gntrans\_ni -?“ erscheint nachfolgende Ausgabe.

```
GNTRANS_NI Version Jun 20 2017 13:11:06
C:\Program Files (x86)\LGLN\gntrans_ni\bin64\gntrans_ni.exe -?
GNTRANS_NI - Zur Transformation von Koordinaten im DHDN und ETRS89
Revision 1.4.3.0 Copyright (c) 2001 - 2017 Geo++ GmbH
LGLN-Standard-Modus ist aktiviert.

Befehlszeile:
  GNTRANS_NI -t tran [-l land] [Optionen] [infile [outfile]]
Argumente:
  -t tran  : Benutze Transformation tran
  -l land  : Benutze Bundesland-Kennung land
  infile   : input file [stdin]
  outfile  : output file [stdout]
Optionen:
  -llh    : Eingabe und Ausgabe von ellipsoidischen Koordinaten
            (nur ETRS89)
  -st wert: Setze Abbildungsstreifen bzw. -zone auf Wert wert
  -2D     : Aktiviere Hoehenverschneidung
  +S      : aktiviere Stochastik-Ausgabe
  -?      : Ausgabe dieser Online-Hilfe
  -q      : Deaktiviere Ausgaben auf Fehler-Kanal
  -time   : Aktiviere Ausgabe der Rechenzeit
  -s      : Zeige alle Laender und Patches

Transformationen:
  ETLS: ETRS89 nach DHDN/GK
  LSET: DHDN/GK nach ETRS89

Bundesland-Kennungen:
  NISA7P_P53: Transformationsmodell Niedersachsen

Beispiele:
  Transformation einer Datei von ETRS89/UTM nach DHDN/GK:
  GNTRANS_NI -l NISA7P_P53 -t ETLS -2D < Eingabe-Datei > Ausgabe-Datei
```

Demnach gestaltet sich der Aufruf für das in Abschnitt 2.3.2 dokumentierte Transformationsbeispiel auf der Eingabekonzole wie folgt (inkl. Ausgabe der Rechenzeit):

```
GNTRANS_NI -l NISA7P_P53 -t LSET -time <test.gkh >test.utm 2>log.lst
```

Das Berechnungsergebnis wird über die Standardausgabe in die Datei „test.utm“ geschrieben. Weiterhin erscheint über den Fehler-Kanal (2) nachstehende Ausgabe in der Datei „log.lst“. Eine weitere Log-Datei wird nicht angelegt.

*Inhalt der Log-Datei:*

```
GNTRANS_NI Version Jun 20 2017 13:11:06
GNTRANS_NI.exe -l NISA7P_P53 -t LSET -time
09:36:18: GNTRANS-Modell initialisiert.
09:36:18: Standard 2D-Modus von GNTRANS_NI aktiviert.
09:36:18: 3D Eingabe: Koordinaten werden mit DGM-Hoehe transformiert.
09:36:18: Ausgabe ist 3D mit unveraenderten Eingangshoehen.
09:36:19: Warnung: Koordinatenwerte von Punkt 485 liegen ausserhalb des DGM
09:36:19: Zeit: 984 Millisekunden
09:36:19: Normales Ende
```

Das Ergebnis wird in der Ausgabedatei im ASCII-Format abgespeichert. Wie bei WinGNTRANS\_NI werden bei dieser Applikation die Höhen aus der Eingabedatei unverändert an die Koordinaten im Ausgabesystem angeschrieben. Somit behalten DHDN/GK-Punktlisten mit physikalischen Höhen über Normalhöhen-Null (NHN) diese nach der Transformation nach ETRS89/UTM bei.

*Inhalt der Ausgabedatei „test.utm“ (Auszug):*

1	32593389.2731	5722149.0173	248.8000
2	32593389.2867	5723148.6179	249.8000
3	32593389.2982	5724148.2197	250.0000
4	32593389.3089	5725147.8246	325.3000
5	32593389.3200	5726147.4290	398.6000
6	32593389.3305	5727147.0303	400.3000
7	32593389.3424	5728146.6334	498.7000
8	32593389.3560	5729146.2337	429.8000
9	32593389.3708	5730145.8423	682.0000
10	32593389.3869	5731145.4410	575.4000
11	32593389.4071	5732145.0359	412.5000
12	32593389.4263	5733144.6349	356.3000
13	32593389.4441	5734144.2383	378.4000
14	32593389.4633	5735143.8435	505.8000
15	32593389.4834	5736143.4468	606.8000
16	32593389.5020	5737143.0470	560.3000

Die nachfolgende Tabelle fasst die grundsätzliche Aufrufsyntax von GNTRANS\_NI für die Transformation in das Zielsystem ETRS89 zusammen. Dabei ist zu beachten, dass bei Setzen der Option „-llh“ die ellipsoidischen Koordinaten im ETRS89 in Grad, Minuten und Sekunden (jeweils durch ein Leerzeichen getrennt) ausgegeben werden bzw. einzugeben sind.

**Tabelle 2: Konsolenkommandos für GNTRANS\_NI EXEC, Rücktransformation entsprechend**

Startsystem	Zielsystem	Aufruf
DE_DHDN_3GK<sn>_NI100	ETRS89_UTM<zn>	gntrans_ni -l NISA7P_P53 -t LSET -2D
DE_DHDN_3GK<sn>_NI100	ETRS89_Lat-Lon	gntrans_ni -l NISA7P_P53 -t LSET -2D -llh

## 2.5 GNTRANS\_NI API

### 2.5.1 Beschreibung

Die GNTRANS\_NI API ist auf windowsbasierten Computersystemen einsetzbar. Sie dient der individuellen funktionellen Integration in externe Programme. Für diesen Zweck besteht sie aus

- den erforderlichen dynamischen Programmbibliotheken (u. a. „gntrans\_ni.dll“),
- einer Header-Datei („gntrans\_dll.h“),
- dem Transformationsmodell Niedersachsen („Modell\_NI.bin“) und
- dem DGM-T („LGNDGM.dat“ und „LGNDGM.idx“).

Darüber hinaus sind der GNTRANS\_NI API auch folgende Dateien beigelegt:

- das in Abschnitt 2.5.5 abgedruckte Beispielprogramm („ExampleGNTRANS\_NI-CPP.cpp“) und
- zu Testzwecken eine exemplarische Eingabedatei („test.gkh“) mit Koordinaten in den CRS DE\_DHDN\_3GK3\_NI100 und DE\_DHDN\_3GK4\_NI100 und die zugehörige Ausgabedatei („test.utm“) mit den transformierten Koordinaten im CRS ETRS89\_UTM32.

### 2.5.2 Installation und Einrichtung

Bei der Programmierung ist darauf zu achten, dass der verwendete Compiler die Header-Datei finden kann. Dies geschieht in der Regel über die Angabe einer entsprechenden Compiler-Option.

Die Einbindung der GNTRANS\_NI-DLL erfolgt explizit. Bei der expliziten Verknüpfung werden die Funktionen der GNTRANS\_NI-DLL zur Laufzeit geladen. Das Programmierbeispiel in Abschnitt 2.5.5 ist derart angelegt.

Durch die Installation von GNTRANS\_NI werden keine Umgebungsvariablen gesetzt, daher ist es in eigenen Programmen notwendig, vor dem Laden der GNTRANS\_NI-DLL in das Verzeichnis der Bibliothek zu wechseln, damit die GNTRANS\_NI-DLL auf alle erforderlichen Programmkomponenten zugreifen kann.

**Codebeispiel:**

```
SetCurrentDirectory("C:\\Program Files (x86)\\LGLN\\gntrans_ni\\bin64");
hdll = LoadLibrary("C:\\Program Files (x86)\\LGLN\\gntrans_ni\\bin64\\gntrans_ni_64.dll");
```

### 2.5.3 Funktionsumfang

Um Aufrufe der GNTRANS\_NI API in einer Hochsprache realisieren zu können, wird die Header-Datei „gntrans\_dll.h“ mitgeliefert.

Wie in Abschnitt 1.3.2 gesagt, ist GNTRANS\_NI lediglich für die Transformation zwischen den CRS DE\_DHDN\_3GK<sn>\_NI100 und ETRS89\_UTM32 vorgesehen. In der Header-Datei ist vermerkt, dass ETRS89-Koordinaten darüber hinaus auch in kartesischer und ellipsoidischer Form verarbeitet werden können. Dementgegen ist die Ein- und Ausgabe kartesischer ETRS89-Koordinaten bei der GNTRANS\_NI API jedoch nicht zugelassen.

In Tabelle 3 sind die Funktionen der GNTRANS\_NI API zusammengestellt. Neben der aufgeführten Funktionsdeklaration sind alle Eingabeparameter erläutert (vgl. auch die Angaben in der Header-Datei). Zusätzliche Hinweise finden sich gegebenenfalls in den Bemerkungen wieder.

Tabelle 3: Funktionen der GNTRANS\_NI API

Funktionsname Kurzbeschreibung	Funktionsdeklaration Eingabeparameter	Bemerkungen
GetDllVersion Liefert die Versionsnummer der DLL	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GetDllVersion(void);	
GnTransGetErrorMsg Gibt die Fehlermeldung zu einem numerischen GnTrans Fehlercode zurück.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransGetErrorMsg( int retVal, char* retValString, int retValStringLength);	

<b>Funktionsname</b> <b>Kurzbeschreibung</b>	<b>Funktionsdeklaration</b> <b>Eingabeparameter</b>
<b>Bemerkungen</b>	
gntrans_get_version_ Liefert einen Versionsstring der DLL. Dient vor Allem zur Identifizierung in Geo++ Programmen.	GNTRANS_DLL_API const char * CALLING_CONVENTION_GNTRANS_DLL gntrans_get_version_(void);
GnTransGetLGNDgmDLLVersion Gibt die Version der lgndgm.dll zurück, die von GNTRANS geladen wird.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransGetLGNDgmDLLVersion(void);
GnTransGetLGNDgmVersion Gibt die Version des LGN DGM zurück, das von GNTRANS geladen wird.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransGetLGNDgmVersion(void); ----- Voraussetzung für eine Einbindung von GnTransGetLGNDgmVersion ist, dass der „Schalter“ MAKE_NISA_GNTRANS gesetzt worden ist. Dies geschieht im Rahmen der Präprozessor-Anweisungen vor der Einbindung der Header-Datei „gntrans_dll.h“ (#define MAKE_NISA_GNTRANS).
GnTransInit Initialisiert ein Gntrans Handle und muss vor der Benutzung eines Gntrans Handles einmal aufgerufen werden.	GNTRANS_DLL_API GNTRANS_HDL CALLING_CONVENTION_GNTRANS_DLL GnTransInit(int* ret);
GnTransInit2DTo3D Initialisiert ein Gntrans Handle und muss vor der Benutzung eines Gntrans Handles einmal aufgerufen werden, anstatt. <b>Diese Funktion arbeitet nur bei entsprechender Freischaltung der AED-SICAD/LGN Option.</b>	GNTRANS_DLL_API GNTRANS_HDL CALLING_CONVENTION_GNTRANS_DLL GnTransInit2DTo3D(int* ret); ----- <b>Diese Funktion ist nicht verfügbar</b>
GnTransDestroy Der Speicher der internen Struktur eines Gntrans Handles kann mit dieser Funktion freigegeben werden.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransDestroy(GNTRANS_HDL pGNTRANS);
GetNPossibleStates Funktion liefert Anzahl verfügbarer Bundesländer, die in gntrans.dll implementiert sind.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GetNPossibleStates(GNTRANS_HDL pGNTRANS, int* CountStates);
GetPossibleStates Funktion liefert Informationen zu den verfügbaren Bundesländern.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GetPossibleStates( GNTRANS_HDL pGNTRANS, SStateInfo* States);
GetPossibleStatesEx Liefert erweiterte Informationen zu den verfügbaren Bundesländern.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GetPossibleStatesEx( GNTRANS_HDL pGNTRANS, SStateInfoEx* States);
GetPossibleStatesEx2 Liefert erweiterte Informationen zu den verfügbaren Bundesländern.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GetPossibleStatesEx2( GNTRANS_HDL pGNTRANS, SStateInfoEx2* States);
GnTransGiveOverViewStates Gibt auf dem Ausgabekanal stdout die verfügbaren Systeme aus.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransGiveOverViewStates( GNTRANS_HDL pGNTRANS);
GnTransFile Dient zur Transformation von Dateien. Die Steuerung erfolgt über die Parameter analog zu den Optionen im Programm GNTRANS.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransFile( GNTRANS_HDL pGNTRANS, const char* land, const char* tran, int nletflag, int nhflag, double nh, const char* dateiin, const char* dateiout);
GnTransFileEx Dient zur Transformation von Dateien. Die Steuerung erfolgt über die Parameter analog zu den Optionen im Programm GNTRANS. Im Gegensatz zu GnTransFile() kann innerhalb dieser Funktion ein bestimmter Meridianstreifen vorgegeben werden.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransFileEx( GNTRANS_HDL pGNTRANS, const char* land, const char* tran, int nletflag, int nhflag, double nh, const char* dateiin, const char* dateiout, int ht,

Funktionsname Kurzbeschreibung	Funktionsdeklaration Eingabeparameter
	<pre>int strip);</pre>
<p>GnTransStruct Transformiert einen Eingabedatensatz und schreibt das Ergebnis in einen Ausgabedatensatz.</p>	<pre>GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransStruct(     GNTRANS_HDL pGNTRANS,     const char* land,     const char* tran,     int nletflag,     int nhflag,     double nh,     const koord* koordin,     koord* koordout);</pre>
<p>GnTransStructEx Transformiert wie <i>GnTransStruct</i> einen Datensatz. Hier erfolgt jedoch zusätzlich eine manuelle Wahl des Hauptmeridians.</p>	<pre>GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransStructEx(     GNTRANS_HDL pGNTRANS,     const char* land,     const char* tran,     int nletflag,     int nhflag,     double nh,     const koord* koordin,     koord* koordout,     int ht,     int strip);</pre>
<p>GnTransStructInit Initialisiert die DLL-internen Variablen für eine bestimmte Transformation, als Voraussetzung für die Verwendung von <i>GnTransStructEx2</i>).</p>	<pre>GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransStructInit(     GNTRANS_HDL pGNTRANS,     const char* land, const char* transformation,     int nletflag, int nhflag, double nh,     int ht, int strip);</pre>
<p>GnTransStructEx2 Transformiert eine Eingabestruktur mit zuvor initialisierten (<i>GnTransStructInit</i>()) Parametern. Kann beliebig oft ausgeführt werden, da sich die Transformationseigenschaften nicht ändern.</p>	<pre>GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransStructEx2(     GNTRANS_HDL pGNTRANS,     const koord* in,     koord* out);</pre> <p>-----</p> <p><b>Aus Performancegründen empfiehlt sich immer die Verwendung von GnTransStructEx2 in Verbindung mit GnTransInit (statt GnTransStruct oder GnTransStructEx).</b></p>
<p>GnTransStructxyz2llh Führt eine Umformung von geozentrisch-kartesischen in ellipsoidische Koordinaten durch (keine Datumstransformation).</p>	<pre>GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransStructxyz2llh(     const koord* in,     koord* out,     const char* ell);</pre> <p>-----</p> <p>Ein Einsatz dieser Funktion kann sinnvoll sein, wenn dreidimensional kartesische ETRS89-Koordinaten vorliegen. Da GNTRANS_NI die direkte Transformation solcher Koordinaten in das Landessystem nicht unterstützt, kann zuvor eine Koordinatenumformung in ellipsoidische ETRS89-Koordinaten durchgeführt werden. Dabei ist dann zu beachten, dass bei einer anschließenden Datumstransformation nicht mehr die originäre Höheninformation, sondern die Höhe aus dem internen DGM für die Berechnungen verwendet wird. Ferner ist zu bedenken, dass die Unterstützung der Transformation zwischen ellipsoidischen ETRS89- und DE_DHDN_3GK&lt;sn&gt;_NI100-Koordinaten nicht abschließend gewährleistet wird (s. Ausführungen unter <i>GnTransStruct</i>). In dem Fall dann eine weitere Umformung der ellipsoidischen in abgebildete Koordinaten notwendig (s. <i>GnTransStructllh2xyz</i>).</p>
<p>GnTransStructllh2xyz Führt eine Umformung von ellipsoidischen in geozentrisch-kartesische Koordinaten durch (keine Datumstransformation).</p>	<pre>GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransStructllh2xyz(     const koord* in,     koord* out,     const char* ell);</pre> <p>-----</p> <ul style="list-style-type: none"> <li>• s. <i>GnTransStructxyz2llh</i></li> <li>• Für die Verwendung dieser Funktion ist die Kenntnis über die ellipsoidische Höhe unabdingbar. Zu bedenken ist in diesem Zusammenhang, dass GNTRANS_NI bei der Ausgabe ellipsoidischer Koordinaten <u>keine</u> ellipsoidische Höhe ausgibt.</li> </ul>
<p>GnTransInitSystem Initialisiert ein internes System Handle für Koordinatenumformungen. Muss einmal vor einer</p>	<pre>GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransInitSystem(     const char* ell, const char* proj, int pjc_strip,     int force_strip, double lat_0, double lon_0, double x_0,</pre>

Funktionsname Kurzbeschreibung	Funktionsdeklaration Eingabeparameter ----- Bemerkungen
Umformungsfunktion gerufen werden. Danach können beliebig viele Umformungen mit dem eingestellten System durchgeführt werden.	double y_0, double pjc_scl);
GnTransStruct1lh2RHhEx Formt Koordinaten von geografisch/ellipsoidisch in Projektionssystem um. Vorher muss einmal GnTransInitSystem() aufgerufen werden, um Ellipsoid- und Projektionsparameter zu setzen.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransStruct1lh2RHhEx( const koord* koordin, koord* koordout); ----- siehe GnTransStructxyz21lh
GnTransStructRHh21lhEx Formt Koordinaten vom Projektionssystem in geografisch/ellipsoidisch um. Vorher muss einmal GnTransInitSystem() aufgerufen werden, um Ellipsoid- und Projektionsparameter zu setzen.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransStructRHh21lhEx( const koord* koordin, koord* koordout); ----- siehe GnTransStructxyz21lh
GnTransStructxyz21lhEx Formt Koordinaten von geozentrisch-kartesisch in geografisch/ellipsoidisch um. Vorher muss einmal GnTransInitSystem() aufgerufen werden, um Ellipsoidparameter zu setzen.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransStructxyz21lhEx( const koord* koordin, koord* koordout); ----- siehe GnTransStructxyz21lh
GnTransStruct1lh2xyzEx Formt Koordinaten von geografisch/ellipsoidisch in geozentrisch-kartesisch um. Vorher muss einmal GnTransInitSystem() aufgerufen werden, um Ellipsoidparameter zu setzen.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GnTransStruct1lh2xyzEx( const koord* koordin, koord* koordout); ----- siehe GnTransStructxyz21lh
GetPatchVersion Liefert die Version des Patches, die über den Patch Generator eingegeben wurde.	GNTRANS_DLL_API int CALLING_CONVENTION_GNTRANS_DLL GetPatchVersion( GNTRANS_HDL pGNTRANS, const char* PatchName);

## 2.5.4 Fehlerwerte

In Tabelle 4 sind die Fehlerwerte (Returncodes) dokumentiert, die bei der Benutzung der Funktionen der GNTRANS API auftreten können.

Tabelle 4: Fehlerwerte

Fehlerwert	Beschreibung
-99	Es ist ein unvorhergesehener Fehler aufgetreten
0	Die Transformation war erfolgreich
1	Gntrans Handle nicht initialisiert / Konnte EingabeDatei nicht oeffnen / System nicht initialisiert
2	Konnte AusgabeDatei nicht oeffnen
3	Fehler beim Lesen der Eingabedatei
4	Gntrans Handle ist nicht initialisiert
5	GnTransStructInit did not run succesfully
6	Fehler in der Länderangabe.
7	Fehler in der Transformationsangabe.
9	Checksum Fehler in Patch Datei
10	Invalid SStateInfo struct version
11	No states possible (Es stehen keine Bundesländer zur Verfügung)
20	ungueltige Projektion in proj (Der Projektionsname konnte nicht gefunden werden.)
21	ungueltiges Ellipsoid in ell (Der Ellipsoidname konnte nicht gefunden werden.)
22	System nicht mit GnTransInitSystem() initialisiert
98	Benutzung ist nicht erlaubt --> außerhalb des lizenzierten Wertebereiches (z.B. NISA)
99	Benutzung ist nicht erlaubt --> DongleSchutz
101 – 199	Der Wertebereich der Eingangswerte ist fehlerhaft.
991	“sorry, you are not allowed to use this program please check your dongle no GNTRANS license available“
992	Es ist ein Fehler beim Lesen der Datei „autostate.bin“ aufgetreten.
993	Die Datei „autostate.bin“ existiert nicht.
1335	Das Bundesland existiert bereits.

## 2.5.5 Beispielprogramm

```
//=====
// Name      : ExampleGNTRANS_NI-CPP.cpp
// Copyright : Landesamt für Geoinformation und Landentwicklung Niedersachsen
// Datum     : 17.02.2011
// Description: Implementierungsbeispiel für die GNTRANS_NI API
//            mit expliziter Verknüpfung der GNTRANS_NI-DLL
//            C++, Ansi-style
//=====
// Einbinden von Header-Dateien
#include <windows.h>
#include <iostream>
using namespace std;
#include <iomanip>

// Definitionen
#define MAKE_NISA_GNTRANS

// Einbinden der GNTRANS_NI-Header-Datei
#include <gntrans_dll.h>

// Hauptprogramm
int main(int argc, char* argv[])
{
    // Definition der Funktionen der DLL
    typedef int (*GetDllVersion)();
    typedef int (*GnTransGetLGNDgmVersion)();
    typedef const char* (*gntrans_get_version_>();
    typedef GNTRANSHDL (*GnTransInit)(int*);
    typedef int (*GnTransDestroy)(GNTRANSHDL);
    typedef int (*GetNPossibleStates)(GNTRANSHDL, int*);
    typedef int (*GetPossibleStatesEx)(GNTRANSHDL, SStateInfoEx*);
    typedef int (*GnTransStructInit)(GNTRANSHDL, char*, char*, int, int, double,
                                     int, int);
    typedef int (*GnTransStructEx2)(GNTRANSHDL, koord*, koord*);
    typedef int (*GnTransFileEx)(GNTRANSHDL, char*, char*, int, int, double,
                                  char*, char*, int, int);

    // Deklaration von Funktionszeigern
    GetDllVersion pGetDllVersion;
    GnTransGetLGNDgmVersion pGnTransGetLGNDgmVersion;
    gntrans_get_version_ pgntrans_get_version_;
    GnTransInit pGnTransInit;
    GnTransDestroy pGnTransDestroy;
    GetNPossibleStates pGetNPossibleStates;
    GetPossibleStatesEx pGetPossibleStatesEx;
    GnTransStructInit pGnTransStructInit;
    GnTransStructEx2 pGnTransStructEx2;
    GnTransFileEx pGnTransFileEx;

    // Initialisierung des Rückgabewertes
    int ret = 0;

    // dynamische Einbindung der DLL
    HINSTANCE hdll; // Handle für DLL
    hdll = LoadLibrary("gntrans_ni.dll");

    if(hdll!=NULL)
    {
        // Funktionszeiger auf die Funktionen abrufen
        pGetDllVersion = (GetDllVersion)GetProcAddress(hdll, "GetDllVersion");
        pGnTransGetLGNDgmVersion = (GnTransGetLGNDgmVersion)GetProcAddress(hdll,
            "GnTransGetLGNDgmVersion");
        pgntrans_get_version_ = (gntrans_get_version_)GetProcAddress(hdll ,
            "gntrans_get_version_");
        pGnTransInit = (GnTransInit)GetProcAddress(hdll, "GnTransInit");
        pGnTransDestroy = (GnTransDestroy)GetProcAddress(hdll, "GnTransDestroy");
        pGetNPossibleStates = (GetNPossibleStates)GetProcAddress(hdll,
            "GetNPossibleStates");
        pGetPossibleStatesEx = (GetPossibleStatesEx)GetProcAddress(hdll,
            "GetPossibleStatesEx");
    }
}
```

```
pGnTransStructInit = (GnTransStructInit)GetProcAddress(hdll,
    "GnTransStructInit");
pGnTransStructEx2 = (GnTransStructEx2)GetProcAddress(hdll,
    "GnTransStructEx2");
pGnTransFileEx = (GnTransFileEx)GetProcAddress(hdll, "GnTransFileEx");

/**
 * Ausgabe allgemeiner Informationen
 */

// Ausgabe der Version der gntrans.dll
if(pGetDllVersion) {
    cout << "Die Version der gntrans_ni.dll ist "
        << pGetDllVersion() << ".\n";
} else {
    cerr << "Die Funktion GetDllVersion steht nicht zur Verfuegung.\n";
}

// Ausgabe der Version der LGNDGM-DLL
if(pGnTransGetLGNDgmVersion) {
    cout << "Die Version der LGNDGM-DLL ist "
        << pGnTransGetLGNDgmVersion() << ".\n";
} else {
    cerr << "Die Funktion GnTransGetLGNDgmVersion "
        << "steht nicht zur Verfuegung.\n";
}

// Versionsstring der DLL
if(pgntrans_get_version_) {
    cout << "Versionsstring der gntrans_ni.dll: "
        << pgntrans_get_version_() << "\n\n";
} else {
    cerr << "Die Funktion gntrans_get_version_ steht nicht zur "
        << "Verfuegung.\n\n";
}

// Initialisierung der DLL:
if(pGnTransInit) {
    GNTRANSHDL pGntrans = pGnTransInit(&ret);
    cerr << "Initialisierung der gntrans_ni.dll; Rueckgabewert: "
        << ret << "\n";
    if(ret==0) {

        //Anzahl implementierter Transformationen
        int countStates;
        if(pGetNPossibleStates) {
            ret = pGetNPossibleStates(pGntrans, &countStates);
            if(ret==0) {
                cout << "\nAnzahl gegebener Transformationsmoeglichkeiten: "
                    << countStates << "\n";

                //erweiterte Informationen
                SStateInfoEx *statesEx =
                    (SStateInfoEx *)malloc(countStates*sizeof(SStateInfoEx));
                if(pGetPossibleStatesEx) {
                    ret = pGetPossibleStatesEx(pGntrans, statesEx);
                    if(ret==0) {
                        cout << "\n*** Erweiterte Informationen zu den "
                            << "Transformationen ***";
                        for(int j=0; j<countStates; j++) {
                            cout << "\nVersion: "
                                << statesEx[j].Version << "\n"
                                << "Bezeichnung der Transformation: "
                                << statesEx[j].Name << "\n"
                                << "ISO-bezogene Bezeichnung: "
                                << statesEx[j].ISO2C << "\n"
                                << "Kuerzel des Patches: "
                                << statesEx[j].Shrt << "\n"
                                << "Bezeichnungen der Koordinatenkomponenten: "
                                << statesEx[j].Comp[0] << ", "
                                << statesEx[j].Comp[1] << ", "
                                << statesEx[j].Comp[2] << "\n"
                                << "Kommentar zum Bundesland: "
                                << statesEx[j].Info << "\n"

```

```
        << "Abbildung: "  
        << statesEx[j].Proj << "\n"  
        << "Verdichtungsstufe: "  
        << statesEx[j].Patch << "\n";  
    }  
    } else {  
        cerr << "\nFehler bei der erweiterten Ausgabe "  
        << "von Informationen zu den verfügbaren "  
        << "Transformationen; Rueckgabewert: " << ret << "\n";  
    }  
    } else {  
        cerr << "\nDie Funktion GetPossibleStatesEx steht nicht "  
        << "zur Verfügung."  
        << "\n";  
    }  
    } else {  
        cerr << "\nFehler bei der Ermittlung der Anzahl verfügbarer "  
        << "Transformationen; Rueckgabewert: " << ret << "\n";  
    }  
    } else {  
        cerr << "Die Funktion GetNPossibleStates steht nicht zur "  
        << "Verfügung.\n";  
    }  
    }  
    /**  
    * Transformation mit GnTransStructEx2  
    */  
  
    //Füllen der Steuerungsstrukturen und Variablen:  
    koord Xin = {{3554643.584, 5808517.070, -999999999.0},  
                {0.001, 0.001, 0.001},  
                {0.000, 0.000, 0.000}};  
    koord Xout = {{0.000, 0.000, 0.000},  
                 {0.000, 0.000, 0.000},  
                 {0.000, 0.000, 0.000}};  
    char land[] = "NISA7P_P53";  
    char tran[] = "LSET";  
    int nlflag = 2;  
    int nhflag = 0;  
    double nh = 0.;  
    int ht = 0;  
    int strip = 0;  
  
    //Initialisierung der Transformation mit GnTransStructEx2:  
    if(pGnTransStructInit&&pGnTransStructEx2) {  
        cout << "\n*** Transformation mit GnTransStructEx2 ***\n";  
        ret = pGnTransStructInit(pGntrans, land, tran, nlflag, nhflag, nh,  
                                ht, strip);  
        if(ret!=0) {  
            cerr << "Fehler bei der Initialisierung der Transformation für"  
            << " GnTransStructEx2; Rueckgabewert: " << ret << "\n";  
        } else {  
            for(int i = 0; i<5; i++) {  
  
                //Durchführen der Transformation:  
                ret = pGnTransStructEx2(pGntrans, &Xin, &Xout);  
                if(ret==0) {  
                    cerr << "Transformation mit GnTransStructEx2 war "  
                    << "erfolgreich; Rueckgabewert: " << ret << ".\n";  
                    cout.precision(4);  
                    cout << "Die transformierten Koordinaten lauten (ENh):\n"  
                    << Xout.xyz[0] << " " << Xout.xyz[1] << " " <<  
                    Xout.xyz[2] << "\n";  
                    Xin.xyz[0]+=0.1;  
                    Xin.xyz[1]+=0.1;  
                } else {  
                    cerr << "Fehler bei der Transformation mit "  
                    << "GnTransStructEx2; Rueckgabewert: " << ret << "\n";  
                }  
            }  
        }  
    } else {  
        cerr << "\nDie Funktion GnTransStructInit oder GnTransStructEx2 "  
        << "steht nicht zur Verfügung.\n";  
    }  
}
```

```
}

/**
 * Transformation mit GnTransFileEx
 */

//Füllen der Steuerungsstrukturen und Variablen:
strcpy(tran,"LSET");
char dateiin[] = "test.gkh";
char dateiout[] = "test.utm";
//Transformation mit GnTransFileEx
if(pGnTransFileEx) {
    cout << "\n*** Transformation mit GnTransFileEx ***\n";
    ret = pGnTransFileEx(pGntrans, land, tran, nlflag, nhflag, nh,
        dateiin, dateiout, ht, strip);

    if(ret==0) {
        cerr << "Transformation mit GnTransFileEx war erfolgreich; "
            << "Rueckgabewert: " << ret << "\n";
        cout << "Die Ergebnisse wurden in folgende Datei geschrieben: "
            << dateiout << ".\n";
    } else {
        cerr << "Fehler bei der Transformation mittels GnTransFileEx; "
            << "Rueckgabewert: " << ret << "\n";
    }
} else {
    cerr << "\nDie Funktion GnTransFileEx steht nicht zur "
        << "Verfuegung.\n";
}

//Freigabe der Systemressourcen
ret = pGnTransDestroy(pGntrans);
FreeLibrary(hdll);
cerr << "\nGNTRANS_NI wird normal beendet.\n";
return ret;

} else {
    cerr << "\nFehler bei der Initialisierung der DLL. "
        << "Programmabbruch!\n";
    FreeLibrary(hdll);
    return 9998;
}
} else {
    cerr << "Die Funktion GnTransInit steht nicht zur Verfuegung. "
        << "Programmabbruch!\n";
    FreeLibrary(hdll);
    return 9999;
}
} else {
    cerr << "Die GNTRANS_NI-DLL steht nicht zur Verfuegung.\n";
    return 10000;
}
}
```

### 3 Literatur

AdV: Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens (GeoInfoDok), Kapitel 7.1 – Koordinatenreferenzsysteme für AFIS-ALKIS-ATKIS, Version 6.0.1, Stand: 01.07.2009

Kreitlow, Stefanie ; Brettschneider, Andrea ; Jahn, Cord-Hinrich ; Feldmann-Westendorff, Uwe: ETRS89/UTM – Der Bezugssystemwechsel und die Auswirkungen auf die Geodatenutzung. In: Kartographische Nachrichten. Fachzeitschrift für Geoinformation und Visualisierung 60 (2010), Heft 4, S. 179–187

Nutzungsbedingungen des Landes Niedersachsen, vertreten durch das Landesamt für Geoinformation und Landesvermessung Niedersachsen, Landesbetrieb Landesvermessung und Geobasisinformation, Podbielskistraße 331, 30659 Hannover für die Transformationssoftware GNTRANS\_NI vom 01.08.2017

Niedersächsisches Gesetz über das amtliche Vermessungswesen (NVerMG) vom 12. Dezember 2002

Richtlinie 2007/2/EG des Europäischen Parlaments und des Rates vom 14. März 2007 zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft (INSPIRE)

USGS: EarthExplorer. URL: <http://earthexplorer.usgs.gov> (Abruf am 21. Februar 2011)