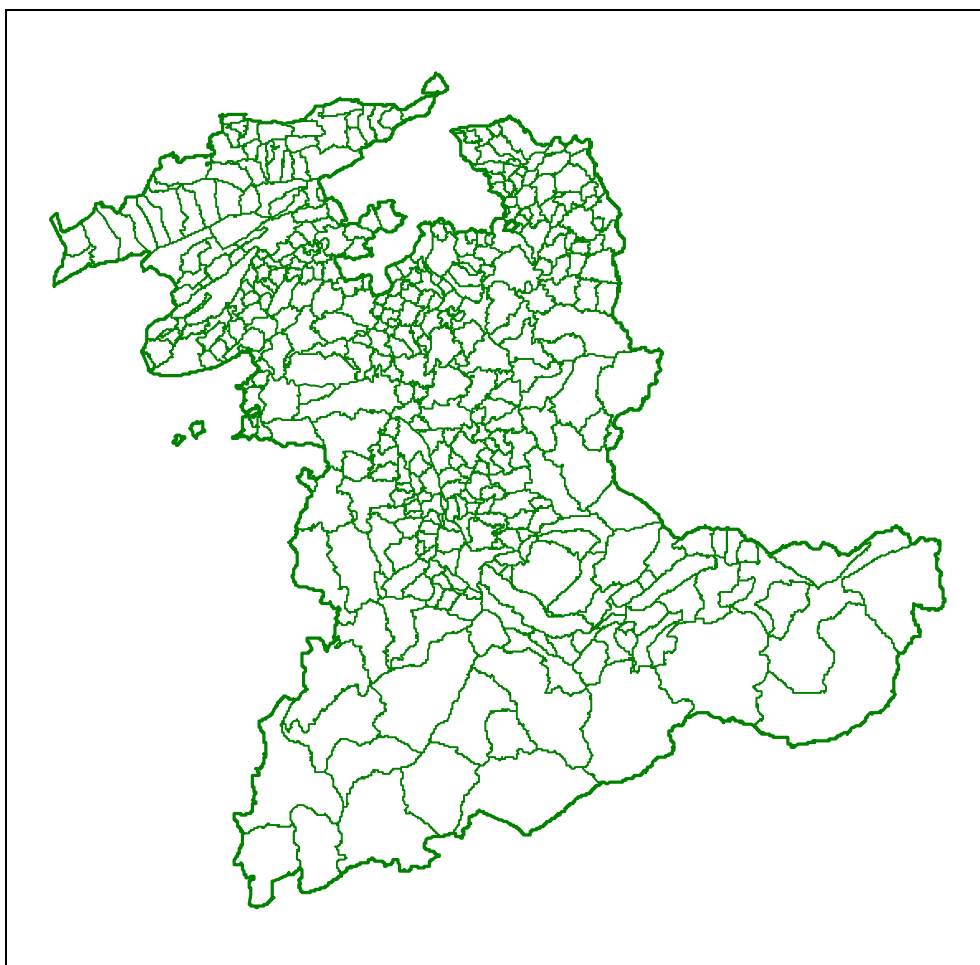


Technische Dokumentation für die ArcView-Erweiterung GEODAT-Tools

Version 1.0, vom 27.01.2000



Inhaltsverzeichnis

1	ALLGEMEINES ZUR APPLIKATION	3
1.1	BESCHREIBUNG DES FUNKTIONSUMFANGS	3
1.1.1	<i>Erweiterung der Benutzeroberfläche</i>	<i>3</i>
1.1.2	<i>Menü Ausschnitt</i>	<i>4</i>
1.1.3	<i>Menü Koordinaten suchen.....</i>	<i>5</i>
1.1.4	<i>Menü GEODAT-Karte erstellen (benutzerdefiniert).....</i>	<i>6</i>
1.2	BESONDERHEITEN UND ABHÄNGIGKEITEN	7
1.3	ENTWICKLUNGSPLATTFORM	8
1.4	HINWEISE ZUR ENTWICKLUNGSGESCHICHTE	8
2	DETAILLIERTE BESCHREIBUNG DER FUNKTIONSWEISE.....	9
2.1	AUSSCHNITT..	9
2.2	KOORDINATEN SUCHEN.....	14
2.3	GEODAT-KARTE ERSTELLEN (BENUTZERDEFINIERT).....	19
2.4	LISTE ALLER VERWENDETEN SKRIPTS	25
3	REFERENZEN.....	27

Verzeichnis der Abbildungen

ABB. 1:	MASKE ZUM MENÜ AUSSCHNITT	4
ABB. 2:	MODAL DIALOG ZUM SUCHEN EINES PUNKTES	5
ABB. 3:	MASKE ZUR ERSTELLUNG EINES KARTENLAYOUTS	6
ABB. 4:	INI-FILE DER GEODAT-TOOLS	7
ABB. 5:	BEISPIEL EINES DICTIONARY FÜR GEMEINDEN	10

1 Allgemeines zur Applikation

Die ArcView Erweiterung „Geodat-Tools“ dient in der Verwaltung des Kantons Bern als sogenannte Basis-Extension. Die Erweiterung stellt grundlegende Funktionalitäten zur Verfügung, die im Standardfunktionsumfang von ArcView nicht enthalten sind. Der Funktionsumfang befriedigt allgemeine Benutzerbedürfnisse und ist somit nicht auf die Anforderung einer Fachabteilung zugeschnitten.

1.1 Beschreibung des Funktionsumfangs

1.1.1 Erweiterung der Benutzeroberfläche

Durch das Laden der Erweiterung GEODAT-Tools wird die Menüleiste in der Benutzeroberfläche des View-Moduls um einen Eintrag erweitert. Alle Funktionalitäten der Erweiterung sind über diesen Menüeintrag „GEODAT“ und dessen Untermenüs zugänglich. Shortcuts oder ähnliches kommen nicht zur Anwendung.

Das Menü „GEODAT“ enthält folgende Einträge:

- Ausschnitt ...
- Koordinaten suchen
- GEODAT-Karte erstellen (benutzerdefiniert)
- WEA Karte M10, M25 mit Legende und Titelseite
- WEA Karte M1:25000

Bei der Wahl eines Menüeintrages wird in der Statuszeile eine kurze Erläuterungen zur Funktion angezeigt.

Die Applikation stellt im wesentlichen zwei Funktionalitäten zur Verfügung. Über die Menüs „Ausschnitt“ und „Koordinaten suchen“ kann der geographische Raum nach mehreren Kriterien eingegrenzt und im aktuellen View dargestellt werden. Über die Menüeinträge „GEODAT-Karte erstellen (benutzerdefiniert)“, „WEA Karte M10, M25 mit Legende und Titelseite“ und „WEA Karte M1:25000“ können ArcView-Layouts mit einem unterschiedlichem Erscheinungsbild erstellt werden.

Die Zusammenhänge der Menüeinträge, der Skripts, die durch diese Menüeinträge zur Ausführung gebracht werden, und der Resultate sind im sogenannten Funktionsplan graphisch dargestellt. Im Funktionsplan sind teilweise auch die Resultate (Kartenausdrucke) schematisch dargestellt, die über ein Menü erzeugt werden können. Die Darstellung der Resultate erfolgt nur dort, wo es für das Verständnis erforderlich ist.

Wird im vorliegenden Dokument auf Buttons in einer Maske verwiesen, so wird der Buttonname mit einem grauen Balken hinterlegt.

1.1.2 Menü Ausschnitt ...

Über das Menü wird eine Dialog-Designer Maske geöffnet, wie sie in Abb. 1 dargestellt ist. Je nach Wahl eines Radio-Buttons wird eine Auswahlbox mit einer Liste der Gemeinden oder einer Liste der PK25-Blattschnitte dargestellt. Wird der unterste Radio-Button gewählt **Rechteck-Ausschnitt**, wird ein Formular für die Eingabe eines Koordinatenpaares bereitgestellt.

Wird eine Gemeinde oder ein PK25-Blatt gewählt, wird die Gemeindefläche resp. das Kartenblatt formatfüllend im aktuellen View dargestellt.

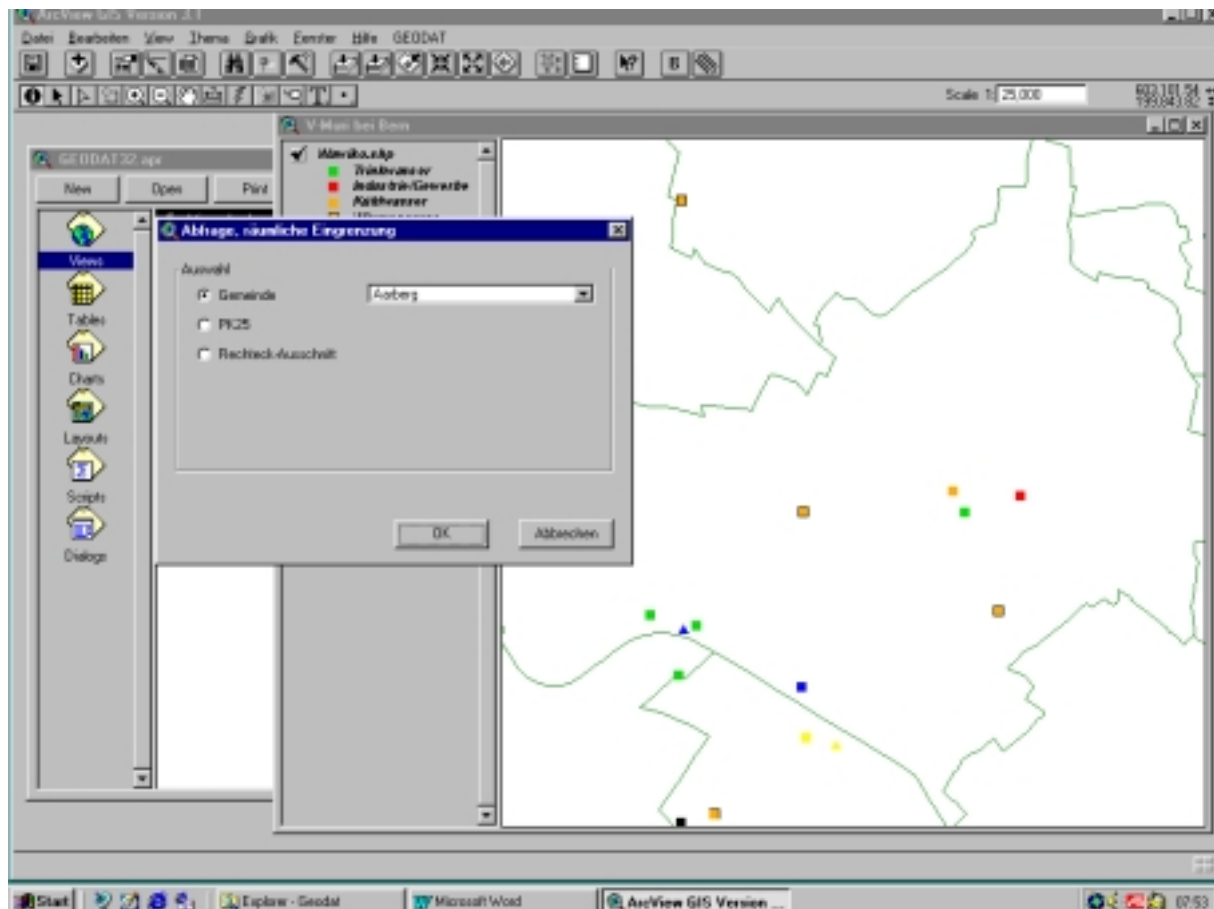


Abb. 1: Maske zum Menü Ausschnitt

Durch die Eingabe eines Koordinatenpaares – Button **Rechteck-Ausschnitt** –, wird das durch die Koordinaten aufgespannte Rechteck im View formatfüllend dargestellt und temporär mit einem roten Rahmen markiert.

1.1.3 Menü Koordinaten suchen

Es wird eine Dialog-Designer Maske geöffnet, wie sie in Abb. 2 zu sehen ist. Die Maske stellt zwei Funktionalitäten bereit:

- Werden in den beiden ersten Eingabezeilen die Koordinaten eines Punktes eingegeben, so wird der aktuelle View auf diesen Punkt zentriert und der gewünschte Masstab wird eingestellt.
- Wird der Button **Mit Maus** gewählt, so hat der Bearbeiter die Möglichkeit, mit der Maus einen Punkt im View zu markieren (Linke Maustaste). Hiernach wird der aktuelle View auf diesen Punkt zentriert und der gewählte Masstab, dritte Eingabezeile – wird eingestellt.

Bei beiden Funktionen wird der gewählte Punkt temporär mit einem roten Kreuz markiert, siehe Abbildung.

Die Eingabemaske ist ein sogenannter *modal dialog*, d.h. die Maske bleibt so lange offen, bis sie vom Bearbeiter explizit über den Button **Schliessen** oder über das Kreuz in der Windows-Titelleiste geschlossen wird.

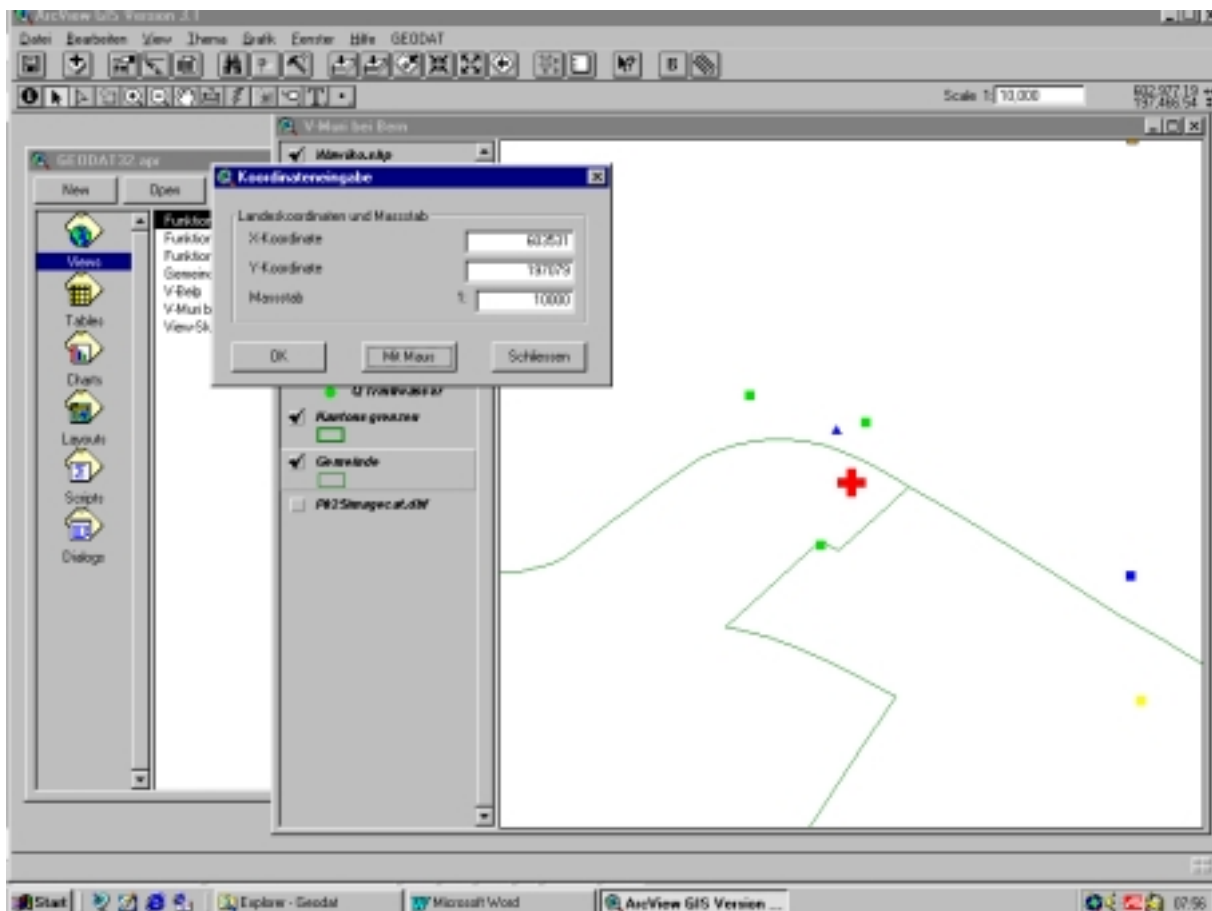


Abb. 2: Modal Dialog zum Suchen eines Punktes

1.1.4 Menü GEODAT-Karte erstellen (benutzerdefiniert)

Es wird eine Dialog-Designer Maske geöffnet, wie sie in Abb. 3 dargestellt ist. Die Maske enthält drei Control-Panels, über welche dem Bearbeiter folgende Funktionalitäten zur Verfügung gestellt werden:

- Wahl des Massstabs
- Wahl des Papierformates
- Eingabe des Legendentexts

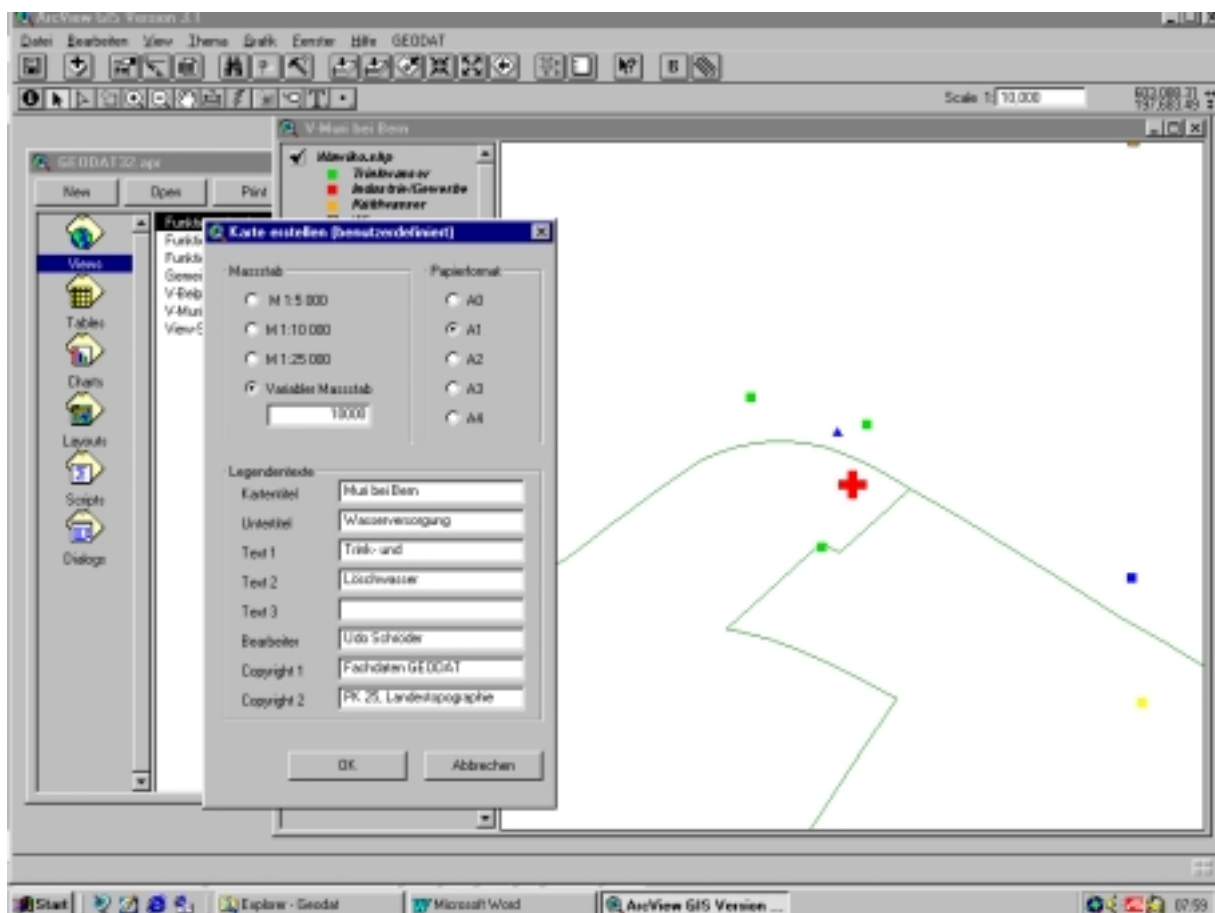


Abb. 3: Maske zur Erstellung eines Kartenlayouts

Im Kartenlayout wird jeweils der aktuelle Viewausschnitt dargestellt. Das Kartenlayout weist ein standardisiertes Erscheinungsbild auf hinsichtlich Position der Legende, der Logos, des Titels und des Views. Die Legendeninformaton wird am linken Kartenrand als Legendeleiste dargestellt.

Papierformat und Massstab stehen zueinander in Beziehung. Deshalb wird bei der Angabe von Papierformat oder Massstab geprüft, ob der aktuelle Viewausschnitt im Kartenlayout auch dargestellt werden kann. Führt diese Prüfung zu Konflikten wird dies dem Bearbeiter mitgeteilt, und er wird aufgefordert das Papierformat oder den Massstab zu korrigieren.

Alle Informationen der Maske (Format, Massstab, Text) werden dem View als Object-Tag zugewiesen und dienen bei einem erneuten Öffnen der Maske als Voreinstellung für die Maske.

1.2 Besonderheiten und Abhängigkeiten

Die benutzerspezifischen Masken der GEODAT-Tools werden mit dem Dialog-Designer erstellt. Die Extension Dialog-Designer ist Bestandteil von ArcView 3.1. Werden die GEODAT-Tools mit ArcView 3.0 betrieben, was technisch möglich ist, so ist der Dialog-Designer explizit zu installieren.

Die Extension GEODAT-Tools liest gewisse Informationen von einem INI-File. Ein solches INI-File ist in Abb. 4 beispielhaft dargestellt.

```
[GENERAL]
AnzahlBasisFiles=2
Massstab=10000
ZoomProzent=5

[LOGO-FILE]
Logo=D:\projekte\reseau\image\Geodat.tif
TitelLogo1=D:\projekte\reseau\image\Bern.tif
TitelLogo2=D:\projekte\reseau\image\wea-logo.tif
TitelLogo3=D:\projekte\reseau\image\oehe.tif

[DATENFILE1]
theFile=d:\Projekte\Reseau\shape\gde97.shp
theLegend=d:\Projekte\Reseau\shape\Gemeinde.avl
theName=Gemeinde
Sichtbar=1

[DATENFILE2]
theFile=d:\Projekte\Reseau\shape\netz25.shp
theLegend=d:\Projekte\Reseau\shape\netz25.avl
theName=PK25 Blattschnitte
Sichtbar=0

[KARTENLAYOUT]
LegBreite=4.5
RandOben=2
RandUnten=2
RandLinks=2
RandRechts=2
Amt=Wasser- und Energiewirtschaftsamt des Kantons Bern (WEA)
```

Abb. 4: INI-File der GEODAT-Tools

Detaillierte Erläuterungen zu den einzelnen Abschnitten (Sections) des INI-Files finden sich in den entsprechenden Kapiteln, in welchen auch die Skripts ausführlich beschrieben werden.

Das INI-File muss auf der Disk physisch im selben Verzeichnis wie die Extension vorhanden sein.

Die Extension ist so konzipiert, das die Extension selbst und das INI-File in einem speziellen Verzeichnis (\$USREXT) für benutzerspezifische Extensions gespeichert werden können, d.h. die Extension sucht Informationen des INI-Files zuerst in diesem speziellen Verzeichnis. Wird hier kein INI-File gefunden, so wird im Standardverzeichnis für ArcView-Extensions (\$AVEXT) gesucht.

Weiter Informationen zu dieser Thematik finden sich im ArcView [1] und AVENUE-Handbuch [2].

1.3 Entwicklungsplattform

Die Extension GEODAT-Tools wurde unter folgender Soft- und Hardware-konfiguration entwickelt.

Hardware:	Compaq PC, 333 MHZ, 64 MB Ram
Betriebssystem:	Win95, WinNT 4.0
ArcView:	ArcView 3.1, deutsche Version

1.4 Hinweise zur Entwicklungsgeschichte

Die Funktionalitäten der GEODAT-Tools wurden aus dem Projekt RESEAU, (RESEAU-Viewer) entwickelt. Weitere Informationen finden sich in den entsprechenden Dokumenten zum RESEAU-Viewer.

2 Detaillierte Beschreibung der Funktionsweise

In diesem Kapitel wird die Funktion, die über jedes der erwähnten Menüs zur Ausführung gebracht werden kann, detailliert erläutert. Nach einer Erläuterung der Funktionsweise und der softwaretechnischen Besonderheiten, wird jedes Skript und seine Funktionsweise detailliert beschrieben. Diese Beschreibung soll dazu dienen, sich einen schnellen Überblick über die Applikation zu verschaffen, ohne dass der Leser sich durch den Quellcode eines jeden Skripts durcharbeiten muss. Werden sehr detaillierte Informationen über eine Variable oder ein Objekt gesucht, so wird es weiterhin unerlässlich sein, den Quellcode direkt zu analysieren. Dies ist üblicherweise aber nur bei einer Modifikation oder einer Erweiterung der Applikation notwendig.

Es wird empfohlen, beim Lesen dieses Kapitel den Funktionsplan zur Hand zu haben. Somit ist die Verknüpfung der Skripts untereinander und die Übergabe von Parametern immer ersichtlich. Durch die Verknüpfung der Skripts, die über eine Menüeintrag zur Ausführung gebrachte werden entstehen sogenannte Funktionseinheiten. Der Begriff der Funktionseinheiten wird auch im Dokument „Anleitung zur Erstellung einer technischen Dokumentation“ verwendet.

Grundlagen für die detaillierten Beschreibung der Skripts in diesem Kapitel sind die Informationen aus dem Programmkopf und die Blockkommentaren. Diese wurden softwaretechnisch mit der Applikation „AvDoku“ aus den Skripts extrahiert und anschliessend in ein Word-Dokument überführt.

Einträge im Programmkopf, die keine Information enthalten, wurden in der nachfolgenden Dokumentation entfernt. Dies dient vor allem der Lesbarkeit und erlaubt es, das Dokument vom Umfang her so gering wie möglich und somit übersichtlich zu gestalten, ohne das wesentliche Informationen fehlen.

2.1 Ausschnitt..

Über dieses Menü kann der geographische Raum nach mehreren Kriterien eingegrenzt und im aktuellen View dargestellt werden. Eingrenzungskriterien sind:

- die Gemeindefläche,
- die Blattschnitte der PK25 Blätter
- oder ein benutzerdefiniertes Rechteck.

Für die Eingrenzung des geographischen Raums über eine Gemeinde oder ein PK25 Kartenblatt wird in der Maske jeweils eine Auswahlliste bereitgestellt. Sowohl die Liste mit den Gemeindennamen als auch die Liste mit den PK25 Blattnamen wird über *Dictionaries* bereitgestellt. In den *Dictionaries*, die von den Skripts *Basis_GemeindeDict* und *Basis_BlattschnittDict* erzeugt werden, wird neben dem Namen, welcher in der Liste erscheint, auch der Extent der Gemeinde resp. des Kartenblattes geführt. Ein Auszug aus dem Dictionary für die Gemeinden im Kanton Bern ist in Abb. 5 dargestellt.

Durch diese Information muss softwaretechnisch kein *ZoomToExtent* auf dem Objekt gemacht werden. Es ist deshalb nicht notwendig, dass für das zoomen auf eine Gemeinde oder ein Kartenblatt ein Thema Gemeinde.shp oder Netz.shp geladen ist.

```
dctGemeinde = Dictionary.Make(421)
dctGemeinde.Empty

ye,      M10  M25  GdeNr  Gemeinde      x0,      y0,      xe,
dctGemeinde.Set("Aarberg",      {586207, 208431, 590077, 212306,"A2","A3",
301})
dctGemeinde.Set("Aarwangen",    {622258, 229800, 626867, 233531,"A2","A3",
321})
dctGemeinde.Set("Adelboden",    {603840, 140126, 614524, 153316,"AU","A2",
561})
dctGemeinde.Set("Aefligen",    {607290, 215082, 609431, 217666,"A3","A3",
401})
.
.
.
```

Abb. 5: Beispiel eines Dictionary für Gemeinden

Im Abschnitt [GENERAL] des INI-Files kann über das Steuerwort *ZoomProzent* ein Zoom-Faktor in Prozent, d.h. eine Zahl ≥ 0 festgelegt werden.

Erläuterung: Standardmässig wird in ArcView beim Zoomen auf ein Objekt der ViewExtent um 10% vergrössert. Über das Steuerwort *ZoomProzent* kann dieser Vergrösserungsfaktor für das Zoomen auf eine Gemeinde oder ein PK25 Blatt individuell festgelegt werden. Wird der Wert 0 eingegeben, so liegen die Gemeindegrenzen resp. die Blattränder genau auf dem Viewrand. Ein Wert von 10 entspricht der Standardvorgabe in ArcView, d.h. Vergrösserung des Viewauschnittes um 10%.

Die Eingrenzung des geographischen Raums durch ein benutzerdefinierende Rechteck erfolgt durch die Angabe von zwei Punkte in Landeskoordinaten. Zur Eingabe der Koordinaten der beiden Punkte werden dem Bearbeiter in der Maske Texteingabefelder zur Verfügung gestellt. Die Landeskoordinaten müssen dabei nicht sechsstellig eingegeben werden. Die Eingaben (Integer-Zahlen) in den Textfeldern werden programmintern immer auf sechs Stellen (Landeskoordinaten) umgerechnet.

Beispiel: Die Eingabe der Zahl 62 wird über eine Logarithmusfunktion in den Koordinatenwert 620000 umgerechnet.

Das vom Benutzer über die zwei Punkte definierte Rechteck wird im View temporär mit einem roten Rechteck (*GraphicShape*) markiert. Das *GraphicShape* wird nicht in die *GraphicList* des View geschrieben, was dazu führt, dass es nicht im Layout erscheint und bei Funktionen wie Zoomen und Verschieben gelöscht wird.

Für die Realisierung der Funktionseinheit sind acht Skripts erforderlich, siehe Funktionsplan im Anhang. Die Skripts werden nachfolgend auf der Grundlage ihrer Informationen im Programmkopf (Skriptheader) und in den Blockkommentaren dokumentiert.

Name: **Basis_MaskeAbfrage**
Version: 1.0
Projekt: res1.apr
Zweck: Dialog-Designer Maske für die Darstellung von Auswahllisten der Gemeindenamen und der PK25 Blattschnitte für den Kanton Bern. Die Listen für die Gemeinden und PK25 Blätter werden aus Dictionaries erstellt, d.h. die Gemeinden und Blattschnitte müssen nicht als Shape oder Coverage vorliegen. Die Maske stellt auch einen ControlPanel zur Verfügung, über welchen durch die Eingabe von zwei Punkten ein Rechteck festgelegt werden kann.
Plattform: ArcView 3.1
Programmaufrufe: Basis_MaskeAbfrageOK, Basis_DialogClose, Basis_RadButtClick, Basis_GemeindeDict
notwendige_GVar: _Objekte, _BasisList
Datum: 09.06.1998
Geschichte: Basis_MaskeDigiScreen ,
 28.10.98 Geändert auf SrcNames der Themen
 28.01.99 Auswahllisten als Listen der Keys der Dictionaries, diese können alphabetisch sortiert werden, was beim Request ReturnKeys nicht geht. Letzte Einstellung der RadioButtons wird als Objekt Tag am Menüeintrag gespeichert

Blockkommentare

10 --> Initialisieren, aktiven View holen'
 20 --> Erstellen eines Dictionary für die Gemeinde und die Blattschnitte der PK25-Blätter'
 25 --> ObjektTag mit den letzten Einstellungen der Maske lesen'
 30 --> Dialog erstellen
 40 --> Label Buttons für den Dialog festlegen'
 50 --> Control Panels für den Dialog festlegen'
 60 --> Je nach Einstellung (Wunsch) den Rahmen um die ComboBoxen unsichtbar machen '
 70 --> Radio Buttons im Dialog festlegen'
 80 --> Die ComboBoxen für den Dialog festlegen'
 90 --> Text Label's für den Dialog definieren'
 100 --> Text Line's für den Dialog definieren'
 110 --> Letzte Einstellungen der Maske lesen und als Vorgabe benutzen'
 120 --> Voreinstellungen, d.h. das Erscheinungsbild beim öffnen des Dialogs festlegen'
 130 --> Radiobuttons auswerten
 140 --> Dialog-Maske öffnen '

Name: **Basis_GemeindeDict**
Version: 1.0
Projekt: Geodat31.apr
Zweck: Erstellen eines Dictionary mit den Eigenschaften der Gemeinden im Kanton Bern, einschliesslich der grossen Seen. Das Dictionary enthält neben dem Gemeindenamen und der Gemeindenummer, auch die Ausdehnung der Gemeinden in Landeskoordinaten. In zwei weiteren Spalten werden die Plottgrössen (AO, A1, A2, A3) für die Massstäbe 1:10000 und 1:25000 geführt.
Plattform: ArcView 3.1
Rueckgabewerte: dctGemeinde
Datum: 25.03.1999
Geschichte: 26.03.99 x1 von Bätterkinden von Hand um 1 cm nach Westen verschoben

Name: **Basis_SetViewUnits**
Version: 1.0
Projekt: punktsetzen.apr
Zweck: Falls für den aktuellen View keine Karten- und Abstandseinheiten gesetzt sind, werden diese Eigenschaften im aktuellen View auf Meter gesetzt.
Plattform: ArcView 3.1
Argumente: theView
Datum: 24.08.1998

Name: **Basis_RadButtClick**
Version: 1.0
Projekt: res1.apr
Zweck: Click-Skript an einem Radio-Button, mit welchem die Darstellung eines weiteren Controls gesteuert wird. Könnte auch als Listener-Broadcaster Beziehung realisiert werden.
Plattform: ArcView 3.1
SELF: aDialog
Datum: 09.06.1998
Geschichte: 23.03.1999 die Koordinaten für die Position der Controlpanels in eine Liste schreiben, um die Positionierung bei Modifikationen zu vereinfachen.

Blockkommentare
10 --> Initialisieren'
20 --> Radiobuttons auswerten'

Name: **Basis_DialogClose**
Version: 1.0
Projekt: res1.apr
Zweck: Skript zum Schliessen eines Dialogs
Plattform: ArcView 3.1
SELF: aDialog
Datum: 09.06.1998

Name: **Basis_ReadINIValues**
Version: 1.0
Projekt: testini.apr
Zweck: Lesen von Informationen aus einem INI File
Plattform: ArcView 3.1
SELF: SELF
Argumente: strINIFile, strSection, strKey
Rueckgabewerte: strValue
Datum: 21.11.1998
Geschichte: Von der ArcView Home Page geholt
Probleme: Die Länge der Zeichenkette strValue ist auf 100 Zeichen begrenzt

Blockkommentare

10 --> Übergabeparameter lesen und in Variablen speichern'
20 --> Prüfen ob Windows läuft, da das Konzept auf einem WIN32 API call basiert'
30 --> Das Betriebssystem indentifizieren, und analysieren wo es installiert ist'
40 --> Eine leere Stringvariable erstellen für die Übergabe der Information '
50 --> Einen Pfad zum kernel32.dll aufbauen und ein DLL-Objekt erstellen'
60 --> Die DLL Prozedur erstellen'
70 --> INI-File lesen und die gesuchte Variable in der entsprechenden Sektion suchen'
80 --> Den gefunden String als Stringvariable zurückgeben'

Name: **Basis_ZoomToRect**
Version: 1.0
Projekt: Geodat31.apr
Zweck: Erstellen eines Rechtecks, Zoom auf das Rechteck und temporäres markieren des Rechtecks durch einen roten Rahmen. Die Koordinatenangaben des Benutzers werden auf 6 Stellen (Landeskoordinaten) umgerechnet.
Plattform: ArcView 3.1
Argumente: numXlu, numYlu , numXro, numYro
Datum: 23.03.1999

Blockkommentare

10 --> Eingabe des Punktes links unten und des Punktes rechts oben'
20 --> Anzahl der eingegebenen Koordinatenstellen analysieren'
30 --> Polygonsymbol für den temporären roten Rahmen erstellen'
40 --> Umrechnung der Benutzereingaben (Koordinaten) auf Landeskoordinaten '
50 --> Aktiven View und Display holen'
60 --> Aus den Koordiantenangaben ein Rechteck erstellen und temporär im View zeichnen'

2.2 Koordinaten suchen

Das Menü öffnet eine Dialog-Designer Maske (*Modal-Dialog*), siehe Abb. 2. Mit der Maske kann durch die Angabe von Landeskoordinaten ein Punkt definiert werden, der im Zentrum des aktuellen Views dargestellt wird. Die Eingabe des Punktes kann auf zweierlei Arten erfolgen:

- durch die Eingabe von Landeskoordinaten in die Textzeilen,
- oder durch die Selektion eines Standortes mit der Maus.

Für diese an sich banalen Funktionseinheit sind insgesamt 10 Skripts erforderlich. Dies ist darin begründet, dass die Leistungsfähigkeit der Funktion weit über das hinausgeht, was man auf den ersten Blick erwartet. Die Maske, welche sich über den Menüeintrag dem Benutzer öffnet, ist ein sogenannter *Modal-Dialog*. Dies bringt folgende Vorteile mit sich:

- Die Maske bleibt so lange offen, bis sie explizit über den Button **Schliessen** oder über das Kreuz in der Windows-Titelleiste geschlossen wird.
- Während die Maske offen ist, können andere Funktionen im ArcView ausgeführt werden. So können z.B. Themen geladen/gelöscht oder ein neuer View geöffnet werden. Die Informationen in der Maske sind dann auch auf den neuen View anwendbar, sofern dieser aktiviert wurde.

Über Texteingabefelder kann in der Maske ein Paar von Landeskoordinaten und ein benutzerdefinierter Massstab eingegeben werden. Ein Standardwert für den Massstab wird aus der Section [GENERAL] unter dem Steuerwort *Massstab* vom INI-File gelesen, Abb. 4. Der Wert kann vom Bearbeiter in der Maske verändert werden und bleibt für die Laufzeit des ArcView-Projektes erhalten. Diese temporäre Massstabs-Information wird als ObjectTag der Maske zugewiesen.

Wie bereits angesprochen kann der Bearbeiter über den Button **Mit Maus** interaktiv - mit der Maus - einen Punkt im aktuellen View selektieren. Der View wird auf den gewählten Punkt zentriert und der gewünschte Massstab wird dem aktuellen View zugewiesen. Zudem werden die Koordinaten des gewählten Punktes in der Maske angezeigt.

Die Möglichkeit, mit der Maus einen Punkt im View zu selektieren, kann nur über ein Tool realisiert werden. Um die erwähnte Funktion bereitzustellen, wurde das Tool-Menü um ein unsichtbares Tool erweitert.

Für einen *Modal-Dialog* muss sichergestellt werden, dass er vom Benutzer nicht mehrfach geöffnet werden kann. Im vorliegenden Fall wird dies vom Skript *Basis_MaskeNeu*, zu Beginn der Bearbeitung geprüft. Ist die Maske schon offen, wird die Bearbeitung, d.h. die Erstellung einer neuen Maske, abgebrochen.

Um unerwünschte Effekte beim öffnen eines Projektes zu vermeiden, ist sicherzustellen, dass beim Speichern des Projektes kein *Modal-Dialog* mehr offen ist. Diese Funktion wird von einem *ProjectSaveScript* wahrgenommen.

Nachfolgend werden die zur Ausführung der Funktionseinheit erforderlichen zehn Skripts beschrieben. Bei der Betrachtung der Skripts sollte der Funktionsplan vorliegen.

Name: **Basis_MaskeNeu**
Version: 1.0
Projekt: punktsetzen.apr
Zweck: Es wird eine Dialog-Designer Maske für die Eingabe eines Koordinatenpaares und eines Massstabs bereitgestellt. Es kann im aktuellen View auf den vorgegebenen Punkt gezoomt werden, d.h der Punkt wird mit einem temporären Symbol markiert und in der Mitte des View-Ausschnitts dargestellt. Beim Zoom auf den Punkt wird dem aktuellen View ein benutzerdefinierter Massstab zugewiesen. Ein Standardwert für den Massstab wird vom INI-File gelesen. Wird der Massstab verändert, so bleibt die Änderung für die Laufzeit des Programms aktiv, der Massstab wird als ObjectTag dem Dialog zugewiesen. Der Bearbeiter kann auch interaktiv - mit der Maus - einen Punkt im View wählen. Die Koordinaten des Punktes werden in der Maske angezeigt, und der gewählte Punkt wird in der Mitte des Views dargestellt. Der View, der Dialog und der Massstab werden in die Liste _KoorList geschrieben und sind so jederzeit abrufbar.
Plattform: ArcView 3.1
Programmaufrufe: Basis_MaskeNeuOK, Basis_MausSetzen, Basis_DialogClose, Basis_SetViewUnits, Basis_aTexLin1Update, Basis_aTexLin2Update
Datum: 24.08.1998
Geschichte: 28.01.99 Scale aus _KoorList lesen
 23.03.1999 Prüfen ob der Dialog schon offen ist. Wenn ja kann er nicht noch einmal geöffnet werden.
 Durch diese Modifikation kann die Maske auch über das x in der Windows-Titelleiste verlassen werden.

Blockkommentare

10 --> Prüfen ob der Dialog schon offen ist, falls ja wird er nicht noch einmal geöffnet'
 20 --> Prüfen ob der Dialog vom View-Modul aufgerufen wurde'
 30 --> Dialog erstellen'
 40 --> Label Button's für den Dialog definieren'
 50 --> Control Panel's für den Dialog definieren'
 60 --> Text Label's für den Dialog definieren'
 70 --> Text Line's für den Dialog definieren'
 80 --> Voreinstellungen für den Massstab aus der letzten Benutzung der Maske berücksichtigen'
 90 --> Abfragen der Bildschirmauflösung und Positionierung der Maske rechts unten'
 100 --> Dialog öffnen '
 110 --> View und Dialog in die Liste der Globalvariablen schreiben'

Name: **Basis_DialogClose**
Version: 1.0
Projekt: res1.apr
Zweck: Skript zum Schliessen eines Dialogs
Plattform: ArcView 3.1
SELF: aDialog
Datum: 09.06.1998

Name: **Basis_SetViewUnits**
Version: 1.0
Projekt: punktsetzen.apr
Zweck: Falls für den aktuellen View keine Karten- und Abstandseinheiten gesetzt sind, werden diese Eigenschaften im aktuellen View auf Meter gesetzt.
Plattform: ArcView 3.1
Argumente: theView
Datum: 24.08.1998

Name: **Basis_aTexLin1Update**
Version: 1.0
Projekt: punktsetzen.apr
Zweck: Update-Skript der Eingabezeile aTexLin1 (X-Koordinate). Der aktuelle (sichtbare) View, in dem die Funktionalität ausgeführt werden soll, wird gefunden da der Dialog und der View als ObjectTag dem Menü zugewiesen wurde. Die X-Koordinate des gewählten Punktes wird in die Text-Line geschrieben.
Plattform: ArcView 3.1
Argumente: aDialog (ObjTag), thePoint (ObjTag)
notwendige_GVar: keine
Datum: 1 5.09.1998

Name: **Basis_aTexLin2Update**
Version: 1.0
Projekt: punktsetzen.apr
Zweck: Update-Skript der Eingabezeile aTexLin2 (Y-Koordinate). Der aktuelle (sichtbare) View, in dem die Funktionalität ausgeführt werden soll, wird gefunden da der Dialog und der View als ObjectTag dem Menü zugewiesen wurde. Die Y-Koordinate des gewählten Punktes wird in die Text-Line geschrieben.
Plattform: ArcView 3.1
Argumente: aDialog (ObjTag), thePoint (ObjTag)
notwendige_GVar: keine
Datum: 15.09.1998

Name: **Basis_MaskeNeuOK**
Version: 1.0
Projekt: punktsetzen.apr
Zweck: Die benutzerspezifischen Eingaben (Kantonsbereich, Massstab) werden auf ihre Gültigkeit überprüft.
Plattform: ArcView 3.1
SELF: GetDialog
Argumente: aBGx, aBGy, strScale
Programmaufrufe: Basis_ZoomAndPan
Datum: 24.08.1998

Geschichte: 29.10.98 Prüfen der Beträge der eingegebenen Koordinaten über den Log(10) und errechnen der tatsächlichen Landeskoordinaten
28.01.99 Letzte Einstellung der Maske als ObjectTag speichern

Blockkommentare

10 --> Initialisieren und eine Dictionary für die Umrechnung der Koordinaten erstellen'
20 --> Die Eingaben der Dialog-Designer Maske auswerten'
30 --> Anzahl der eingegebenen Stellen analysieren'
40 --> Umrechnung der Eingaben in Landeskoordinaten '
50 --> Überprüfen der Eingaben auf Gültigkeit für den Kanton Bern, sehr rudimentäre Angabe der Koordinaten (ist deaktiviert)
60 --> Prüfen ob der Massstabsbetrag positiv ist'
60 --> Den gewählten Punkt in der Mitte des View darstellen und den vorgegebenen Massstab einstellen'

Name: **Basis_ZoomAndPan**
Version: 1.0
Projekt: punktsetzen.apr
Zweck: Stellt den angegebenen Punkt in der Mitte des View dar und markiert den Standort temporär mit einem roten Symbol. Der gewünschte Massstab wird für den aktuellen View eingestellt. Der temporäre Marker verschwindet beim Zoomen oder Verschieben. Er wird auch nicht in einem Layout dargestellt.
Plattform: ArcView 3.1
Argumente: numBGx, numBGy, numScale
notwendige_GVar: keine
Datum: 24.08.1998

Blockkommentare

10 --> Übergabeparameter lesen und den Massstab in die _KoorList schreiben'
20 --> Den aktuellen View aus der _KoorList holen'
40 --> Aus den Eingaben der Koordinaten einen Punktobjekt erstellen'
50 --> Lesen des Marker-Symbols aus der _KoorList'
60 --> View verschieben und auf den angegeben Punkt im vorgegeben Massstab zoomen '
70 --> Zeichnen des temporären Markers'

Name: **Basis_MausSetzen**
Version: 1.0
Projekt: punktsetzen.apr
Zweck: Sucht das für den Bearbeiter unsichtbare Tool (Red Diamond, unsichtbar) im Toolbar. Das Tool mit dem Namen des Click Skripts Basis_MausSetzenOK wird selektiert.
Plattform: ArcView 3.1
SELF: GetDialog
Argumente: aTexLin3 (theScale)
Programmaufrufe: Basis_MausSetzenOK
notwendige_GVar: _KoorList
Datum: 24.08.1998

Blockkommentare

10 --> Initialisieren und Benutzereingaben (Massstab) der Dialog-Maske auswerten'
20 --> Selektieren des unsichtbaren Buttons in der Tool-Leiste'

Name: **Basis_MausSetzenOK**
Version: 1.0
Projekt: punktsetzen.apr
Zweck: Der Benutzer kann im aktuellen View einen Punkt interaktiv mit der Maus wählen (linke Maustaste). Der Punkt wird im Zentrum des Views dargestellt und mit einem temporären Marker gekennzeichnet. Die Koordinateninformationen des gewählten Punktes werden als ObjectTag dem Dialog hinzugefügt. Damit ist der Punkt auch in den Update-Skripts der Text-Lines bekannt.
Plattform: ArcView 3.1
Programmaufrufe: Basis_SetViewUnits
notwendige_GVar: _KoorList
Datum: 24.08.1998

Blockkommentare

10 --> Den View und das Display aus der _KoorList holen und prüfen'
20 --> Prüfen ob es sich um einen neuen View handelt'
30 --> Analyse des Punktes, den der Anwender mit der Maus gewählt hat'
40 --> Den Cursor für die Selektion des Punktes wieder zurücksetzen'
50 --> Das unsichtbare Tool wieder deselektieren'
60 --> Den Punkt nur setzen und zeichnen, falls die Maske noch offen ist'

2.3 GEODAT-Karte erstellen (benutzerdefiniert)

Über diesen Menüeintrag kann der Benutzer ein standardisiertes Kartenlayout erstellen, siehe Funktionsplan. Das Layout ist standardisiert hinsichtlich des gesamten Erscheinungsbildes der Karte (Position des ViewFrames, der Legende und des Logos). Das Kartenlayout wird nicht als Template gespeichert, sondern softwaretechnisch im Skript *Basis_PlotMake* erstellt. Hieraus ergeben sich zahlreiche Vorteile, die ein Template nicht bietet.

Das Menü stellt dem Bearbeiter eine Dialog-Designer Maske zur Verfügung, in welcher er den Massstab und das Papierformat für das Kartenlayout wählen kann, siehe Abb. 3. Feste Massstäbe können über RadioButtons definiert werden, variable Massstäbe über die Benutzereingabe in einer Textzeile.

Im Kartenlayout dargestellt wird immer der sichtbare Ausschnitt (ViewExtent) des aktuellen Views. Um diese Anforderung zu erfüllen, müssen Papierformat und Massstab immer auf gegenseitige Verträglichkeit geprüft werden (Skript: *Basis_CheckFormat*). Um diese Funktionalität bereitzustellen, ohne dass der aktuelle View verändert wird, erfolgt die Erstellung der Layouts auf einer Kopie (Clone) des aktuellen View. Diese Kopie erhält den Namen Layout-View und wird nach Erstellung der Layouts gelöscht.

Im dritten ControlPanel der Maske können Legendeninformationen wie Titel, allgemeiner Text, Name des Bearbeiters und Copyrights eingegeben werden.

Werden die Eingaben in der Maske mit OK bestätigt, so wird ein Layout mit dem Namen „Karte-GEODAT“ erstellt. Existiert bereits ein Layout mit diesem Namen wird der Bearbeiter gefragt, ob er das Layout überschreiben will oder einen neuen Namen verwenden möchte (neues Kartenlayout erstellen).

Das Layout „Karte-GEODAT“ ist nicht *live-linked* mit dem View, aus welchem es erstellt wurde (Standardvorgabe von ArcView). Diese Verbindung müsste der Bearbeiter wieder herstellen, falls es gewünscht ist. Der Layout muss mit dem geclonten View „Layout-View“ *live-linked* werden.

Die benutzerdefinierten Einstellungen, die in der Maske getätigt werden, werden dem Menüeintrag als ObjectTag zugeordnet. Beim erneuten Öffnen der Maske dienen diese Informationen als Voreinstellung. Dieser ObjectTag ist nur für die Laufzeit der ArcView Sitzung gültig.

Die Legendentexte werden in einer Liste geführt und dem aktuellen View als ObjectTag zugeordnet. Hierdurch stehen sie permanent zur Verfügung und werden mit einem ArcView-Projekt abgespeichert.

Einige Informationen zum Erscheinungsbild des Layouts werden aus dem INI-File gelesen (Abb. 4). Die Informationen stehen in der Section [KARTENLAYOUT] und betreffen:

- Die Legendenbreite. Als Minimum wird vom System eine Breite von 4.5 cm gesetzt.
- Die Breite der Ränder. Diese ist teilweise abhängig von den Druckereigenschaften
- Die Bezeichnung einer Amtsstelle oder Fachabteilung. (Wird als Schriftzug unterhalb der Legende dargestellt).

Nachfolgend werden die einzelnen Skripts detailliert erläutert.

Name: **Basis_KarteMain**
Version: 1.0
Projekt: Geodat31.apr
Zweck: Übergeordnetes Skript für die Erstellung von benutzerspezifischen Karten. Das Skript ermöglicht es den Layout-View, d.h. die geclonte Version des aktuellen View nach der Erstellung des Layout zu löschen. Des weiteren ist es möglich das Layout nach dessen Erstellung automatisch zu öffnen.
Plattform: ArcView 3.1
Programmaufrufe: Basis_MaskeKarte
erzeugte_GVar: _BasisFlagKarte
Datum: 24.03.1999
Probleme: Der Name des Layouts wird hardcoded gesetzt

Blockkommentare

10 --> Flag für die Entscheidung ob das Layout geöffnet werden soll. '
 20 --> Das Layout „Karte-GEODAT“ öffnen wenn ein Layout erfolgreich erstellt wurde'
 30 --> Den Layout-View löschen'

Name: **Basis_MaskeKarte**
Version: 1.0
Projekt: res1.apr
Zweck: Dialog-Designer Maske zum Erstellen einer benutzerdefinierten Karte. Als Kartenausschnitt wird der sichtbare ViewExtent des aktuellen Views genommen. Massstab und Papierformat sind frei wählbar. Die Legendenbeschriftung kann in geringem Umfang individuell gestaltet werden. Hierfür werden dem Bearbeiter Textlines zur Verfügung gestellt. Die Länge der Textlines ist von der Legendenbreite abhängig. Die Legendenbreite kann über das INI-File verändert werden.
Plattform: ArcView 3.1
Programmaufrufe: Basis_BtnA0-A4, Basis_DialogClose, Basis_MaskeKarteOK, Basis_BtnMassstab, Basis_SetViewUnits
notwendige_GVar: _BasisList
Datum: 12.06.1998
Geschichte: Lineg
 28.01.99 Letzte Einstellung der Maske als ObjectTag am Menüeintrag speichern

Blockkommentare

05 --> Formatangaben für die Legendenbreite vom INI-File lesen'
 10 --> Initialisieren und den aktiven View suchen '
 20 --> Leere Liste für Legendenbeschriftung erstellen '

30 --> Prüfen ob vom View Modul aufgerufen wurde und ein View aktiv ist'
 40 --> Legendenbeschriftung abfragen, diese ist als ObjectTag einem View zugeordnet'
 50 --> Dialog aufbauen '
 55 --> ObjectTag mit den letzten Einstellungen der Maske lesen'
 60 --> LabelButtons des Dialogs festlegen'
 70 --> ControlPanels des Dialogs festlegen'
 80 --> Rahmen um ComboBoxen nach Bedarf unsichtbar machen '
 90 --> TextLabels des Dialogs definieren'
 100 --> Sichtbaren Viewausschnitt ermitteln und Massstab abfragen'
 110 --> RadioButtons des Dialogs festlegen'
 120 --> Die Länge der Textlines aufgrund der Legendenbreite und der Schriftgrösse berechnen'
 130 --> Die TextLines für den Dialog festlegen'
 140 --> Einstellungen der letzten Maske beim erneuten öffnen berücksichtigen'
 150 --> Den Massstab für die Prüfung von Papierformat und View-Extent klonen'
 160 --> Dialog öffnen'

Name: **Basis_BtnA0-A4**
Version: 1.0
Projekt: res1.apr
Zweck: Testen ob der gewählte View-Ausschnitt als Layout auf das gewählte Papierformat mit dem gewünschten Massstab passt. Die Papiergrößen sind in einem Dictionary definiert.
Plattform: ArcView 3.1
SELF: GetDialog
notwendige_GVar: _Ausschnitt, _BasisList
Datum: 26.06.1998
Geschichte: Globalvariable _theVScale ersetzt
Probleme: Die Definition der Papierformate sollte zentral erfolgen.

Blockkommentare

10 --> Eingaben der Dialog-Designer analysieren'
 20 --> Papiergrößen definieren und in ein Dictionary schreiben'
 30 --> Hilfsvariablen initialisieren '
 40 --> Papiergrösse für das gewählte Format festlegen'
 50 --> Extent des gewählten Viewausschnitts festlegen'

Name: **Basis_BtnMassstab**
Version: 1.0
Projekt: res1.apr
Zweck: Click-Skript um die Verträglichkeit des gewählten Papierformates und des Massstabs zu behandeln. Bei der Wahl des Papierformates und des Massstabes ist immer zu prüfen, ob der sichtbare Ausschnitt des Views beim gewählten Papierformat auch darstellbar ist.
Plattform: ArcView 3.1
SELF: GetDialog
Programmaufrufe: Basis_CheckFormat
notwendige_GVar: _BasisList
Datum: 26.06.1998
Geschichte: Globalvariable _theVScale ersetzt

Blockkommentare

- 10 --> In der Dialog-Designer Maske den vom Benutzer gewählten Button (Massstab) suchen'
- 20 --> Prüfen ob das Skript vom View-Modul aufgerufen wurde '
- 30 --> Auswertung der Eingabe wenn ein variabler Massstabs eingegeben wurde
- 40 --> Analyse durchführen wenn feste Massstäbe über die RadioButtons gewählt wurden'
- 50 --> Den neuen Massstab im geclonten Layout-View setzen'

Name: **Basis_CheckFormat**
Version: 1.0
Projekt: res1.apr
Zweck: Prüfen ob der gewählte Massstab und die Papiergrösse zusammenpassen. Massstab und Papiergrösse stehen in einer Beziehung zueinander
Plattform: ArcView 3.1
Argumente: aDialog, theScale
Rueckgabewerte: bInFlag
notwendige_GVar: _Ausschnitt
Datum: 26.06.1998

Blockkommentare

- 10 --> Papiergrössen definieren'
- 20 --> Papiergrösse zum gewählten Format festlegen'
- 30 --> Extent des gewählten Viewausschnitts analysieren, der geplottet werden soll'

Name: **Basis_SetViewUnits**
Version: 1.0
Projekt: punktsetzen.apr
Zweck: Falls für den aktuellen View keine Karten- und Abstandseinheiten gesetzt sind, werden diese Eigenschaften im aktuellen View auf Meter gesetzt.
Plattform: ArcView 3.1
Argumente: theView
Datum: 24.08.1998

Name: **Basis_DialogClose**
Version: 1.0
Projekt: res1.apr
Zweck: Skript zum Schliessen eines Dialogs
Plattform: ArcView 3.1
SELF: aDialog
Datum: 09.06.1998

Name: **Basis_MaskeKarteOK**
Version: 1.0
Projekt: argis.apr
Zweck: OK Skript für die Dialog-Designer Maske zur Erstellung von benutzerdefinierten Kartenlayouts.
Plattform: ArcView 3.1
SELF: GetDialog
Programmaufrufe: Basis_PlotMake, Basis_CheckFormat
erzeugte_GVar: _BasisFlagKarte, _theLView
Datum: 28.10.1998
Geschichte: 28.10.98 Neuordnung der Legendenbeschriftung

Blockkommentare

10 --> RadioButtons der rufenden Dialog-Designer Maske analysieren '
 20 --> Den gewählten Massstab abfragen und prüfen '
 30 --> Legendenbeschriftung aus den Textlines abfragen und als ObjektTag dem View zuweisen '
 35 --> Masstab und Papierformat für das Kartenlayout noch einmal prüfen '
 40 --> Dialog schliessen '
 50 --> Den zu druckenden View-Ausschnitt festlegen und Zoom auf den Mittelpunkt '
 60 --> Den alten Layout-View löschen '
 70 --> Für die Ausgabe im Layout eine Kopie des View erstellen '
 80 --> Aus dem gewählten Papierformat das zu rufende Skript ermitteln '
 90 --> View Fenster des geklonten Views schliessen und Layout-View löschen '
 100 --> Aktuell gewählte Einstellungen als ObjectTag dem Menüeintrag zuordnen '

Name: **Basis_PlotMake**
Version: 1.0
Projekt: res1.apr
Zweck: Erstellen eines benutzerdefinierten Kartenlayouts für die Papierformate A0 bis A4 . Die Breite der Legende wird im INI-File definiert. Alle Angaben über die Breite und Höhe der Rahmen im Layout beziehen sich auf die linke untere Ecke des Layouts in Layouteinheiten. Das Logo und Informationen zu einer Dienststelle werden ebenfalls aus dem INI-File gelesen.
Plattform: ArcView 3.1
SELF: aPaper
Argumente: strPaperSize, IstLegendenTexte
Programmaufrufe: Basis_LayoutFormat, Basis_Legende, Basis_ReadINIValues
Annahmen: Das LogoFile wird vom INI-File gelesen
notwendige_GVar: _BasisFonts, _theVLayout
Datum: 27.06.1998
Geschichte: Aus Lineg und NBkat Projekt abgeleitet, theVView geklont
 28.10.98 Reihenfolge der Legendenbeschriftung geändert.
 03.11.98 Massstäbe < 4000 berücksichtigt für ScaleBar
 24.03.1999 Schreiben des M für Massstab hardcoded gesetzt
 Copyrightzeichen für die Legende hinzugefügt

Blockkommentare

05 --> Schriftart und Grösse für die Textsymbole definieren '
 10 --> Initialisieren und Übergabeparameter auswerten '
 20 --> Masseinheiten festlegen, Umrechnung von Inch in Zentimeter '
 30 --> Text- und Flächensymbole aus den Globalvariablen laden '

40 --> Prüfen ob vom View Modul aufgerufen wurde und ein View aktiv ist'
 50 --> Existiert ein altes Layoutdokument wird dieses gelöscht und eine Neues erstellt'
 60 --> Grösse des Layoutobjektes festlegen und ViewExtent abfragen'
 70 --> Informationen zur Grösse der Ränder und zur Legende vom INI-File lesen'
 80 --> Die Grösse des Views und der Rahmen im Layout setzen '
 90 --> the LView clonen um die Objektorientierung aufzuheben und den ViewFrame erstellen'
 100 --> Rahmen für die Legendenkästen erstellen '
 110 --> Logo-File lesen und im Layout darstellen'
 120 --> Massstabsleiste erstellen'
 130 --> Länge und Unterteilung der Massstabsleiste aufgrund des aktuellen Massstabs berechnen '
 140 --> Ist der Massstab < 4000 dann wird der ScaleBar auf 50 Meter gesetzt'
 150 --> Die Höhe des Legendenrahmens festlegen '
 160 --> Den Legend-Frame für die im View sichtbaren Themen erstellen'
 170 --> Die Beschriftungstexte für die Legende auswerten und dem Layout hinzufügen'
 180 --> Copyright Zeichen erstellen, wenn der entsprechende Textstring nicht leer ist'
 190 --> Das Kartenlayout erstellen und dem Projekt hinzufügen '
 200 --> Die Grösse der Legendenbeschriftung nachträglich verändern'

Name: **Basis_LayoutFormat**
Version: 1.0
Projekt: res1.apr
Zweck: Einstellung von Layouteigenschaften wie Papierformat, Auflösung und Orientierung des Layouts
Plattform: ArcView 3.1
Argumente: strPapierFormat
notwendige_GVar: _theVLayout
Datum: 26.06.1998

Blockkommentare

10 --> Initialisieren und Übergabeparameter auswerten'
 20 --> Papiergrösse des Layouts aufgrund der Benutzerangaben festlegen'
 30 --> Die Layoutorientierung auf Querformat ändern '
 40 --> Displayeigenschaften festlegen '

Name: **Basis_Legende**
Version: 1.0
Projekt: basis.apr
Zweck: Nachträgliche Aufarbeitung der formellen Darstellung der Legende im Layout (primär Text- und Symbolgrösse). Die Schriftzüge der Legenden werden auf eine 10er Schrift (Arial) gesetzt. Die Punktsymbole werden ebenfalls auf 10pt gesetzt. Linien- und Polygonsymbole werden nicht bearbeitet.
Plattform: ArcView 3.1
Argumente: strLayoutName
Annahmen: Ein Layout-Dokument mit einem nicht leeren LegendFrame
Datum: 16.10.1998
Geschichte: ArcView Sample Library, Layout.SimplifyLegendByTheme

Blockkommentare

- 10 --> Schriftart, Grösse und Farbe definieren'
- 20 --> Das angegebenen Layout suchen und den LegendFrame selektieren'
- 30 --> LegendFrame noch einmal suchen und aktivieren'
- 40 --> Den View und die Legende suchen, die zum Layout gehören'
- 50 --> Den LegendFrame vereinfachen, d.h. die Gruppierung auflösen'
- 60 --> Graphiken differenzieren nach Text und Symbolen'
- 70 --> Die Textsymbole auf die gewünschte Grösse ändern'
- 80 --> Die Punktsymbole auf vorgegebene Grösse stellen'
- 90 --> Den live-link wieder aufgeben'

Name:	Basis_ReadINIValues
Version:	1.0
Projekt:	testini.apr
Zweck:	Lesen von Informationen aus einem INI File
Plattform:	ArcView 3.1
SELF:	SELF
Argumente:	strINIFile, strSection, strKey
Rueckgabewerte:	strValue
Datum:	21.11.1998
Geschichte:	Von der ArcView Home Page
Probleme:	Die Länge des Strings strValue ist auf 100 Zeichen begrenzt

Blockkommentare

- 10 --> Übergabeparameter lesen und in Variablen speichern'
- 20 --> Prüfen ob Windows läuft, da das Konzept auf einem WIN32 API call basiert'
- 30 --> Das Betriebssystem indentifizieren, und anlyisieren wo es installiert ist'
- 40 --> Eine leere Stringvariable erstellen, für die Übergabe der Information '
- 50 --> Einen Pfad zum kernel32.dll aufbauen und ein DLL-Objekt erstellen'
- 60 --> Die DLL Prozedur erstellen'
- 70 --> INI-File lesen und die gesuchte Variable in der entsprechenden Sektion suchen'
- 80 --> Den gefunden String als Variable zurückgeben'

2.4 Liste aller verwendeten Skripts

In der nachfolgenden Liste sind alle Skripts, die von der Extension verwendet werden, in alphabetischer Reihenfolge aufgeführt. Werden die Skripts im Funktionsplan mit der nachfolgenden Liste verglichen, so stellt man fest, dass im Funktionsplan mehr Skripts aufgeführt sind. Dies kommt daher, dass einige Skripts, wie z.B. *Basis_SetViewUnits* von mehreren Funktionseinheiten verwendet werden.

```

Basis_aTexLin1Update
Basis_aTexLin2Update
Basis_Beschriftung
Basis_BlattschnittDict
Basis_BtnA0-A4
Basis_BtnMassstab
Basis_ButtonUpdate
Basis_CheckField
Basis_CheckFormat
Basis_DialogClose

```

Basis_DrawKmNetz
Basis_GemeindeDict
Basis_GetList
Basis_GetNullPunkte
Basis_Hilfe
Basis_ImageLegendMake
Basis_Init
Basis_Karte
Basis_KarteMain
Basis_LayoutFalz
Basis_LayoutFormat
Basis_LayoutFormatM10

Basis_LayoutFormatM25k
Basis_LayoutLegendeM25
Basis_LayoutLegendeSeite
Basis_LayoutMakeRect
Basis_LayoutMakeSchnitt
Basis_LayoutSelectName
Basis_LayoutTitelLogo
Basis_LayoutTitelM25
Basis_LayoutTitelSeite
Basis_LayoutViewFrameMake
Basis_Legende
Basis_ListThemes
Basis_LoadFiles
Basis_MaskeAbfrage
Basis_MaskeAbfrageOK
Basis_MaskeKarte
Basis_MaskeKarteM10
Basis_MaskeKarteM10OK
Basis_MaskeKarteM25k
Basis_MaskeKarteM25OK
Basis_MaskeKarteOK
Basis_MaskeNeu
Basis_MaskeNeuOK
Basis_MausSetzen
Basis_MausSetzenOK
Basis_PlotMake
Basis_PlotMakeM10
Basis_PlotMakeM25k
Basis_PrintLayouts
Basis_PrintSingleLayout
Basis_ProjectSaveScript
Basis_RadButtClick
Basis_ReadINIValues
Basis_SelectRecord
Basis_SetViewUnits
Basis_StartupScript
Basis_TextSymbolMake
Basis_ThemeUpDate
Basis_UpdateGemeide
Basis_ViewClipThemes
Basis_ViewClipThemeUnionFTAB
Basis_ViewClipThemeUnionGraphics
Basis_ZoomAndPan
Basis_ZoomToRect
ExtensionMake
My Extension Install
My Extension Uninstall

3 Referenzen

[1] ESRI: Avenue, 1996

[2] ESRI: Using ArcView GIS, 1996

Anhang

Quellcode aller Skripts