

Vorwort des Herausgebers

Der folgende Text stellt die *RoSy*-Software der Firma M.O.S.S., München vor. *RoSy* ist ein umfangreiches hybrides Graphik-System, mit dem Vektor-, Raster- und Textinformation verarbeitet werden kann. Es wird bei vielen Behörden und Unternehmen insbesondere zur Ersterfassung von GIS-Daten eingesetzt. Ein frühes Highlight waren und sind die Möglichkeiten zum Rastersnap und zur Mustererkennung. Die Rasterbearbeitung ist daher auch insgesamt in die Software der Firma SICAD eingebettet worden.

Die Einführung in *RoSy* soll einerseits kurz sein, andererseits aber möglichst viele Themen exemplarisch behandeln, damit Sie als der Leser eine Idee vom Leistungsumfang erhalten. Sie können dann gezielt sich vertiefen. Ein Grundverständnis der Datentechnik wird vorausgesetzt.

Karlsruhe, im Oktober 2001

Hans Kern

Notationen

Hervorhebung fett: **Variante**

Hervorhebung kursiv: *erlaubt*

Eingaben: >eingabe

Soweit nicht anders vorgeschrieben Schlüsselworte in Versalien und benutzerdefinierte Größen mit Kleinbuchstaben oder gemischt: IF [NOT] ERRORLEVEL Nummer Befehl

Ausgaben: ausgabe>

Querverweis auf Abschnitte dieses Textes: *Dateitypen mit Bearbeitungen versehen*

Produktnamen: *RoSy*

Querverweis auf Sachverhalte auf dem Bildschirm: *Eingabeaufforderung*

Bearbeitungspfade: *Start→Programme→Eingabeaufforderung*

Hinweise

Viele Dateien sind ziemlich umfangreich und werden hier häufig nur auszugsweise gezeigt; so finden Sie zum Beispiel die vollständige Datei unter colors.dat, den hier gezeigten Auszug unter colors_ausz.dat.

Vorwort des Herausgebers.....	5
Notationen	6
Hinweise	6
Kartographische Automation.....	11
Systemadministration für ein GIS-System am Beispiel von <i>RoSy</i>	11
Unterlagen zur Lehrveranstaltung	11
Fallstricke	11
Groß- und Kleinschreibung (Casesensitiveness)	11
Leerzeichen.....	11
/ oder \ bei Pfaden.....	11
Dateierweiterungen.....	12
End of Line	12
Vorhandene Dateien werden nicht gefunden.....	12
<i>WindowsNTWorkstation</i>	12
login.....	12
logout.....	13
Laufwerke.....	13
<i>WINDOWS NT</i> -Skripte	13
Kurzeinführung zu <i>WINDOWS NT</i> -Skripten	14
Graphische Oberfläche	16
Dateitypen mit Bearbeitungen versehen	18
Dateien mit Icon versehen	19
Environment	19
Das System <i>RoSy</i>	21
Start von <i>RoSy</i>	21
<i>RoSy</i> -Environment.....	21
Änderungen im <i>RoSy</i> -Environment.....	26
Wichtige <i>RoSy</i> -Dokumentationen und -Hilfen	26
Wichtige <i>RoSy</i> -Kommandos.....	27
Dateistruktur für Applikationen (Projekte).....	29

Anlage hierarchischer Ebenen	30
Update von Ebenen	31
Optionen-Dateien	32
Definition graphischer Attribute	33
Aspekte.....	34
Alte Aspekt-Dateien.....	35
Signaturen	36
Farben.....	36
Flächenfüllungen.....	37
Liniensignaturen.....	37
Linienstile.....	38
Linienstärken.....	39
Punktsignaturen.....	39
Eleganterer Einsatz von Optionendateien	40
Toolboxen	40
Systemprozeduren	45
OGS-Prozeduren	46
EASI-Prozeduren	46
Produktionsdateien	47
Ressourcen	50
Suchhierarchien für Dateien.....	50
Symbolbibliotheken	51
Benutzersperren.....	51
Templates	52
Druckoptionen.....	52
Editor-Buttonsperren.....	52
Passpunkte.....	52
Schriftarten.....	52
Textplatzhalter.....	52
X-Bitmaps	52



HANS F. KERN

Kartographische Automation

Systemadministration für ein GIS-System am Beispiel von *RoSy*

Unterlagen zur Lehrveranstaltung

Fallstricke

Bei der Arbeit mit *RoSy* an der *WindowsNTWorkstation* gibt es einige Fehlerquellen, die nicht so offensichtlich wie Schreibfehler sind und daher häufig zu Frust und Zeitverlust durch sehr langes Suchen führen.

Groß- und Kleinschreibung (Casesensitiveness)

Das *Microsoft*-Betriebssystem hat Grundlagen aus einer Zeit, als der Zeichensatz lediglich aus den normalen - das heißt im Englischen üblichen - lateinischen Großbuchstaben bestand. Es ist daher an einigen Stellen unerheblich, ob Sie für Pfade und Dateinamen Versalien oder gemischte Schrift verwenden. Es ist aber klug, Pfade und Dateinamen immer so zu schreiben, wie sie im *Explorer* angezeigt werden.

Leerzeichen

An vielen Stellen dienen Leerzeichen nur als Trennzeichen und zur typographischen Gliederung. Das dürfen Sie jedoch nicht ungeprüft voraussetzen; bei bat-Dateien zum Beispiel:

```
set pfa=C:\users  
set pfa = C:\users
```

Im ersten Fall ist die Variable "pfa" und ihr Wert ist "C:\users", im zweiten dagegen ist die Variable "pfa " und ihr Wert " C:\users". Dann dürfen Sie sich nicht wundern, wenn Pfadabfragen nicht klappen. Besonders übel ist, daß hinter users kein Leerzeichen stehen darf.

/ oder \ bei Pfaden

Unter *Unix* dient / zur Gliederung von Dateibäumen, *Microsoft* verwendet dafür \. Da Anwendungen möglichst unabhängig vom Betriebssystem sein sollen, entscheiden sich die Entwickler für eine der beiden Möglichkeiten und wandeln sie dann gegebenenfalls

intern in die andere um. Wenn Sie bei *MS-Word* auf *Einfügen* → *Feld...* → *EinfügenText* gehen, müssen Sie // verwenden! In *ftp* unter *Windows* gilt !/

Dateierweiterungen

Stellen Sie den *Explorer* so ein, daß Sie immer die Dateierweiterungen sehen; siehe *Ansicht* → *Optionen...* Es haben Studenten schon Stunden damit zugebracht, herauszufinden, warum die offensichtlich im richtigen Ordner mit dem richtigen Namen vorhandene Datei *forst.asp* vom Programm nicht gefunden wurde. Endlich hat der Dozent festgestellt, daß die Datei noch zusätzlich die Erweiterung *txt* (*forst.asp.txt*) trug.

End of Line

Die Betriebssysteme von *Mac*, *Microsoft* und *Unix* verwenden zur Markierung von Zeilenenden in reinen ASCII-Dateien unterschiedliche Kombinationen der nicht druckbaren Zeichen LF und CR. Beim Dateitransfer mit *ftp* zwischen den Betriebssystemen können Sie dann Probleme bekommen. *Mac*: LF CR; *PC*: CR LF, *Unix*: LF.

Manche Programme, zum Beispiel *OGS*, erwarten, daß das letzte Zeichen in einer Datei ein Zeilenende ist. Schließen Sie daher bitte alle Dateien mit einem *Return* ab.

Vorhandene Dateien werden nicht gefunden

Die Pfade zu Dateien können Sie absolut oder relativ angeben. Die absolute Angabe hat den Vorteil, daß Sie zum Beginn Ihrer Arbeit weniger Probleme haben, weil der absolute Pfad immer eindeutig ist. Relative Angaben sind aber eleganter, weil Sie so Ihre Applikation ohne Umdefinitionen in andere Dateibäume stellen können. Relative Pfade für *RoSy* bezeichnen den Pfad zu Dateien relativ zum Fenster, aus dem *RoSy* gestartet wurde.

WindowsNTWorkstation

login

Wir wollen annehmen, daß sich der Benutzer mit dem Benutzernamen *keha0001* und dem Kennwort *geheim* anmeldet. Es wird die Datei `\\NT-FH-02\NETLOGON\login.cmd` ausgeführt.

logout

Laufwerke

Das 1:-Laufwerk ist die lokale mit dem jeweiligen Arbeitsplatz verbundene Festplatte. An einem anderen Rechner haben Sie also andere Dateien auf 1:. Das n:-Laufwerk ist ein Netzwerklaufwerk, das jedem User individuell zugeordnet ist. Es liegt auf dem zentralen FH-Rechner unter \$HOME/.nt.

Nach dem Arbeiten sollten Sie also alle weiterhin gebrauchten Dateien von 1: auf n: kopieren und sich diese beim nächsten Arbeitsbeginn wieder holen.

Schreibzugriff besteht auf:

- Laufwerk e: Directory keha0001 (identisch mit 1:)
- Laufwerk 1: (lokale Festplatte)(identisch mit e:)
- Laufwerk m: (\$HOME auf dem Zentralrechner, von überall erreichbar)
- Laufwerk n: (\$HOME/.nt auf dem Zentralrechner, von überall erreichbar)

WINDOWS NT-Skripte

Wenn Sie die *WINDOWS NT*-Skriptsprache an Beispielen kennenlernen wollen, können Sie sich in d:\RoSy\Setup diese bat-Dateien ansehen.

- addappl.bat
- app.bat
- removeappl.bat
- rosy_enb.bat

Ein einfacheres Beispiel ist RoSy-Shell.bat auf v:\bin, mit der das *RoSy*-System gestartet wird:

```
REM RoSy Shell

If NOT EXIST 1:\RoSy MKDIR 1:\RoSy
If NOT EXIST n:\RoSy MKDIR n:\RoSy
If NOT EXIST n:\RoSy\hedirc.bat TOUCH n:\RoSy\hedirc.bat

1:
CD \
CD RoSy

c:\Programme\Exceed NT 6.0\exceed.exe

d:\RoSy\exe\rosyshell.exe N:\RoSy\hedirc.bat
```

Wir erkennen Kommentare (`rem`), bedingte Ausführung von Anweisungen (`if`), Erzeugung von Ordnern (`mkdir`), Erzeugung von Dateien (`touch`), Auswahl eines Laufwerks (1:), Auswahl eines Ordners (`cd`), Ausführung eines Programmes (`c:\Programme\Exceed NT 6.0\exceed.exe`) und Ausführung eines Programmes mit Parameterübergabe (`d:\RoSy\exe\rosysshell.exe N:\RoSy\hedirc.bat`).

Kurzeinführung zu *WINDOWS NT*-Skripten

Kommentare

`rem` am Zeilenanfang, gilt bis Zeilenende

Echo

```
echo Guten Tag!  
echo [on]|[off]  
echo.  
@echo off
```

Setzen von Variablen

```
set name=Victor Hugo.
```

Aufgabe: Geben Sie `set name` ein, geben Sie `set` ein. Was erkennen Sie?

Referenzieren von Variablen

```
echo Guten Tag, %name%!  
PATH=C:\BC5\BIN;%PATH%
```

Umlaute in echo

```
echo ae=„, AE=□ oe=“ OE=™ ue=□ UE=š sz=á
```

Geben Sie in der Eingabeaufforderung `echo ae=ä AE=Ä>tmp.bat` ein. Dann finden Sie in der Datei `tmp.bat` die Realisierung für die Umlaute.

Verzweigungen und Label

Die Möglichkeiten zur strukturierten Programmierung sind dürftig.

```
IF [NOT] ERRORLEVEL Nummer Befehl  
IF [NOT] Zeichenfolge1==Zeichenfolge2 Befehl  
IF [NOT] EXIST Dateiname Befehl  
IF DEFINED Variable Befehl
```

Beispiel:

```
if "%antwort%"=="Ja" goto JA  
if errorlevel 0 goto error0  
goto ende  
:JA  
echo Die Antwort war JA  
goto ende  
:error0  
echo "Errorlevel 0 aufgetreten"
```

```
:ende
```

Ausführen und Parameterübergabe

bat-, cmd-, com- und exe-Dateien können Sie ausführen durch Eingabe des Pfades und Dateinamens *ohne* Dateierweiterung. Dazu erstellen wir ein einfaches Beispiel als bat-Datei:

```
echo Ihr Vorname ist %2.  
echo Ihr Familienname ist %1.
```

Sie starten das Programm mit Parameterübergabe durch:

```
C:\my_project\my_batch Kern Hans
```

%0, %1, %2 beziehen sich auf den nullten, ersten und zweiten Parameter beim Aufruf eines Skripts, wobei der nullte Parameter der Name des Skripts selbst ist. Wenn der Parameter eine Datei bezeichnet, erhalten Sie mit %~d1 zum Beispiel das Laufwerk der Datei. Die Feinheiten finden Sie in der Eingabeaufforderung mit:

```
help call
```

%% zur Eingabe von %l

In einer bat-Datei steht:

```
ftype MOSS-Datei="C:\Programme\Windows NT\Zubehör\WORDPAD.EXE" "%*1"
```

Umleitung der Ein- und Ausgabe

Aufgabe: Wie übergeben Sie Eingabedateien an Programme? Wie speichern Sie die Ausgabe von Programmen?

Starten eines neuen Prozesses

Mit `start` öffnen Sie ein neues Fenster und lassen darin eine batch-Datei ablaufen.

Aufgabe: Was sind die Unterschiede von `call`, `cmd` und `start`? Geben Sie `cmd` ein, schauen Sie auf den Namen des Fensters, geben Sie `exit` ein.

Suchhierarchie

Wenn Sie ein Programm `my_program.exe` im Ordner `C:\my_project` angelegt haben, können Sie das Programm starten durch Eingabe von `C:\my_project\my_program`, also durch Eingabe des Pfades und des Dateinamens. Einfacher wäre es natürlich, wenn Sie lediglich den Dateinamen `my_program` ohne Pfad einzugeben hätten. (Wenn Sie von der Eingabeaufforderung `notepad` starten wollen, brauchen Sie ja auch nur `notepad` einzugeben!). Sie erreichen Ihr Ziel, indem Sie die Suchhierarchie für ausführbare Programme erweitern. Zunächst interessiert, wie die Suchhierarchie aktuell lautet:

```
>echo %PATH%  
C:\WINNT40\SYSTEM32;C:\WINNT40;>
```

Nun ändern Sie die Suchhierarchie und lassen sich das Ergebnis anzeigen:

```
>PATH=%PATH%;C:\my_project
```

```
>echo %PATH%  
C:\WINNT40\SYSTEM32;C:\WINNT40;C:\my_project;>
```

Hilfe

In der Eingabeaufforderung können Sie sich die Befehle mit `help` auflisten lassen; zum Beispiel `help start`. Auch mit `start /?` erhalten Sie oft Hilfe.

Graphische Oberfläche

Auf dem Bildschirm sehen Sie nach dem Login vier Icon:

- Icon Arbeitsplatz
- Icon Netzwerkumgebung
- Icon Papierkorb
- Icon Aktenkoffer

Wir wollen jetzt wissen, wo festgehalten ist, daß diese Icon gezeigt werden, und ob wir die textliche Beschreibung und das Icon ändern können.

Im *Explorer* sind unter Desktop die Icon Arbeitsplatz, Netzwerkumgebung und Papierkorb unter dem Typ Systemordner aufgeführt und in `C:\WINNT\Profiles\keha0001\Desktop` ist der Aktenkoffer als Verknüpfung aufgeführt.

Die textliche Beschreibung der Icon können Sie im *Explorer* ändern und beim Aktenkoffer können Sie auch das Icon im Aussehen ändern, da er eine Verknüpfung ist (siehe Dateien mit Icon versehen).

Klicken Sie einmal auf ein Icon und drücken Sie dann die rechte Maustaste. Sie sehen jetzt:

- beim Icon Arbeitsplatz
 - Öffnen
 - Explorer
 - Suchen...
 - -----
 - Netzlaufwerk verbinden...
 - Netzlaufwerk trennen...
 - -----
 - Verknüpfung erstellen
 - Umbenennen
 - -----
 - Eigenschaften
- beim Icon Netzwerkumgebung
 - Öffnen
 - Explorer
 - Computer suchen...



- -----
- Netzlaufwerk verbinden...
- Netzlaufwerk trennen...
- -----
- Verknüpfung erstellen
- Umbenennen
- -----
- Eigenschaften
- beim Icon Papierkorb
- Öffnen
- Explorer
- Papierkorb leeren
- -----
- Einfügen
- -----
- Verknüpfung erstellen
- -----
- Eigenschaften
- beim Icon Aktenkoffer
- Öffnen
- Explorer
- Alles aktualisieren
- -----
- Senden an
- -----
- Ausschneiden
- Kopieren
- -----
- Verknüpfung erstellen
- Löschen
- Umbenennen
- -----
- Eigenschaften

Mit den Icon sind also unterschiedliche Aktionen verbunden und diese sind in Gruppen eingeteilt. Beim Doppelklick auf ein Icon wird offensichtlich die zuoberst aufgeführte Aktion ausgeführt. Wie Sie die Aktionen ändern können, ist unter *Dateitypen mit typischen Bearbeitungen versehen* aufgeführt.

Beim Klick auf *Start* sehen Sie drei Bereiche, jeweils durch eine Linie getrennt.

- Ansys 5.6

- AutoCad
- Internet
- Maschinenelemente
- P T C
- RoSy
- Utils
- Viewer
- -----
- Programme ►
- Dokumente ►
- Einstellungen ►
- Suchen ►
- Hilfe
- Ausführen
- -----
- Beenden...

Die Einträge im ersten Bereich sind aufgeführt in V:\etc\Startmenü. Unter *Programme* sind die Einträge im ersten Bereich aufgeführt in C:\Winnt\files\keha0001\Startmenü\Programme, die im zweiten Bereich in C:\Winnt\Profiles\All Users\Startmenü\Programme.

Die Einträge in *Dokumente* finden Sie in C:\Winnt\Profiles\keha0001\Recent.

Dateitypen mit Bearbeitungen versehen

Die folgenden Dateitypen sind unter anderem bei der Arbeit mit *RoSy* wichtig: sgd- und dat-Dateien. Wir wollen erreichen, daß diese Dateien bei Doppelklick oder über die rechte Maustaste einfach mit typischen Bearbeitungen versehen werden.

.sgd-Dateien sollen im *Explorer* den Typ RoSy-Datei haben und das *RoSy*-Icon tragen. Bei Doppelklick sollen sie mit dem Skript v:\bin\RoSy-Shell.bat geöffnet werden.

.dat-Dateien sollen im *Explorer* den Typ Optionen-Datei haben und das MOSS-Icon tragen. Bei Doppelklick sollen sie mit wordpad geöffnet werden.

Aufgabe: txt-Dateien werden bei Doppelklick standard-mäßig mit notepad geöffnet. Ändern Sie es so, daß sie mit wordpad geöffnet werden!

Aufgabe: Da *RoSy* etliche Dateierweiterungen kennt und die Zuordnung der Bearbeitung in unserem Betriebsmodus von einer Sitzung zur nächsten verloren geht, ist es lästig, am Beginn einer Sitzung jeweils die Bearbeitungsprogramme zuzuordnen. Schreiben Sie ein Skript, das mit einem Aufruf diese Zuordnungen vornimmt!

Dateien mit Icon versehen

Ein Icon kann einer *Verknüpfung* individuell und einer *Dateierweiterung* global zugeordnet werden. exe-Dateien können bei der Erzeugung vom Programmierer ein oder mehrere Icons zugewiesen erhalten.

Icons müssen das bmp-Format haben, das Sie mit Paint erzeugen können, nicht jedoch mit dem RoSy-Bitmap-editor.

Ein Icon können Sie entweder zuweisen, indem Sie ein vorhandenes Icon übernehmen, oder eine bmp-Datei erstellen und zuweisen.

Für Ihre RoSy-Dateien können Sie Icon zum Beispiel von D:\RoSy\Setup\launch.exe oder LizView.exe übernehmen.

D:\RoSy\Xbm\icon32.xbm

D:\RoSy\App\inst\images\rosy_small.gif

D:\RoSy\App\BASE\Xbm\logo_moss.xbm

D:\RoSy\Online_Doku\Allgemein\images\moss.gif

D:\RoSy\Online_Doku\Allgemein\images\easi.gif

D:\RoSy\Online_Doku\Allgemein\images\rosy_klein.gif

.sgd-Dateien ansehen und nach DXF wandeln. (D:\RoSy\Exe\hedi.exe und sgd2dxf.exe)

.dat-Dateien bearbeiten.

Environment

Jedem Fenster ist ein Environment zugeordnet. Das sind Variablen, die in dem Fenster wichtig sind und die mit bestimmten Werten belegt sind.

Mit *Start* → *Programme* → *Eingabeaufforderung* öffnen wir ein Fenster zur Eingabe von Systemkommandos. Durch Eingabe des Befehls *env* (oder *set*) werden uns die gesetzten Environment-Variablen mit ihren aktuellen Werten aufgelistet. Das Ergebnis sieht dann ungefähr so aus:

```
!C:=C:\WINNT\Profiles\keha0001\Desktop
ACADSERVER=NT-RZ-01;NT-V-01
ANSYS56_DIR=D:\ANSYS56
ANSYS_SYSDIR=Intel
CLASSPATH=C:\Programme\Exceed NT 6.0
COMPUTERNAME=CAD2-01
COMSPEC=C:\WINNT\system32\cmd.exe
DISPLAY=cad2-01:0
HOMEDRIVE=Z:
HOMEPATH=\keha0001\.nt
```

```

HOMESHARE=\\NT-FH-01\HOMEUSER$

LOGONSERVER=\\NT-FH-02
NUMBER_OF_PROCESSORS=1
OS=Windows_NT
OS2LIBPATH=C:\WINNT\system32\os2\dll;

PATH=/cygdrive/c/Perl5/bin/:/cygdrive/c/WINNT/system32:/cygdrive/c/WINNT:/cyg
drive/c/bin:/bin;.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
POOL=CAD1
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 8 Stepping 6, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=0806
PROMPT=$P$G
P_SCHEMA=d:\ANSYS56\ac4\schema
SERVERNAME=NT-CAD-01
SYSTEMDRIVE=C:
SYSTEMROOT=C:\WINNT
TEMP=/cygdrive/c/TEMP
TMP=/cygdrive/c/TEMP
USERDOMAIN=LMD-FH
USERNAME=keha0001
USERPROFILE=C:\WINNT\Profiles\keha0001
WINDIR=C:\WINNT
TERM=cygwin

```

Wir sehen zum Beispiel, daß die Variable USERNAME den Wert keha0001 hat. Mit `set hase=Mein Name` ist Hase generieren wir eine neue Umgebungsvariable hase und belegen sie mit dem Wert Mein Name ist Hase. Mit `echo %hase%` können wir den Inhalt der Variablen hase sehen.

Aufgabe: Was machen?

- `set`
- `set hase`
- `set hase=<ENTER>` (Zwischen = und <ENTER> kein weiteres Zeichen!)

Die Environment-Variablen werden an die vom Fenster aus gestarteten Prozesse weiter vererbt. Davon überzeugen wir uns, indem wir eine weitere Eingabeaufforderung aus unserem Fenster öffnen durch Eingabe des Befehls `start`.

Das System RoSy

Start von RoSy

Unter *Start* gehen Sie zu *RoSy* und wählen *RoSy-Shell*. Es wird eine Shell geöffnet und es werden die Umgebungsvariablen für *RoSy* gesetzt. Zum Schluß wird die benutzereigene Datei `n:\RoSy\hedirc.bat` ausgeführt. Damit können Sie also Ihre RoSy-Umgebung individuell anpassen, indem Sie `n:\RoSy\hedirc.bat` entsprechend gestalten. Änderungen in `n:\RoSy\hedirc.bat` werden nur wirksam, wenn Sie eine neue RoSy-Shell öffnen (die alte können Sie dann schließen).

Inhalt der RoSy-Shell.bat auf `v:\bin`:

```
REM RoSy Shell

If NOT EXIST l:\RoSy MKDIR l:\RoSy
If NOT EXIST n:\RoSy MKDIR n:\RoSy
If NOT EXIST n:\RoSy\hedirc.bat TOUCH n:\RoSy\hedirc.bat

l:
CD \
CD RoSy

c:\Programme\Exceed NT 6.0\exceed.exe

d:\RoSy\exe\rosyshell.exe N:\RoSy\hedirc.bat
```

Details siehe insbesondere in der Datei `d:\RoSy\Setup\rosy_enb.bat`.

RoSy-Environment

Mit *Start*→*RoSy*→*RoSy-Shell* wird uns das Environment für RoSy-Programme bereitgestellt. Durch Eingabe des Befehls `env` (oder `set`) sehen wir wieder die Umgebungsvariablen. Es sind jetzt eine ganze Reihe von RoSy-Variablen mit ihren Werten hinzugekommen; `ROSY_COLORFILE` hat zum Beispiel den Wert `d:\RoSy\fonts\colors.dat`. `prenv` (print environment) ist eine RoSy-Prozedur, die nur die *gesetzten* RoSy-Variablen anzeigt. Das Ergebnis sieht etwa so aus:

```
----- RoSy-Environment -----
10: ROSY_APPLCMDPATH ..... n:\Forst\tbox\cmd
11: ROSY_APPLECMPTH ..... n:\Forst\tbox\ecm
12: ROSY_APPLGENPATH ..... d:\RoSy\Appl\BASE
13: ROSY_APPLNAME ..... Forst
```

```

15: ROSY_APPLPATH ..... n:\Forst\sgd

16: ROSY_APPLROOTPATH ..... n:
17: ROSY_ASPECTPATH ..... n:\Forst\asp
18: ROSY_ATTRIBPATH ..... n:\Forst\def
19: ROSY_BUTTONLOCKFILE ..... buttonlocks.dat -->
d:\RoSy\fonts\buttonlocks.dat
20: ROSY_CANVSIZFILE ..... canvassizes.dat -->
d:\RoSy\fonts\canvassizes.dat
21: ROSY_CMDPATH ..... n:\Forst\prod
22: ROSY_COLORFILE ..... n:\Forst\Fonts\colors.dat
24: ROSY_DIALSETTINGSFILE ... n:\Forst\res\dialsettings.res
25: ROSY_DIMENSIONSFILE ..... dimensions.dat -->
d:\RoSy\Appl\BASE\opt\dimensions.dat
26: ROSY_EASI ..... d:\RoSy\easi
27: ROSY_EASILIBPATH ..... d:\RoSy\Appl\BASE\Easi
30: ROSY_ECMPROCLISTFILE .... ecmproclist.dat -->
d:\RoSy\fonts\ecmproclist.dat
32: ROSY_ENVIRONFILE ..... rosyenv.dat --> d:\RoSy\setup\rosyenv.dat
33: ROSY_ENVIRONMENTXFILE .... rosyenv.txt --> d:\RoSy\setup\rosyenv.txt
34: ROSY_ERR ..... d:\RoSy\err
35: ROSY_EXE ..... d:\RoSy\exe
37: ROSY_FILLPATTERNFILE ... n:\Forst\Fonts\fillpattern.dat
38: ROSY_FONTS ..... d:\RoSy\fonts
40: ROSY_HELP ..... d:\RoSy\help
42: ROSY_IPARFILE ..... ipamsg.ger --> d:\RoSy\err\ipamsg.ger
44: ROSY_LANGUAGE ..... deutsch
47: ROSY_LINESIGNATFILE .... n:\Forst\Fonts\linesignats.dat
48: ROSY_LINESTYLEFILE ..... linestyles.dat -->
d:\RoSy\fonts\linestyles.dat
49: ROSY_LINEWIDTHFILE ..... n:\Forst\Fonts\linewidths.dat
54: ROSY_MSGFILE ..... rosymsg.ger --> d:\RoSy\err\rosymsg.ger
56: ROSY_MSGPREFIX ..... 0
58: ROSY_OCRPATH ..... n:\Forst\ocr
61: ROSY_PATH ..... d:\RoSy
62: ROSY_POINTSYMBOLFILE ... n:\Forst\Fonts\pointsymbols.dat
66: ROSY_PRINTER ..... 1
67: ROSY_PRINTERFILE ..... printers.dat --> d:\RoSy\fonts\printers.dat
68: ROSY_PRINTPROC ..... printproc.tcl --> d:\RoSy\exe\printproc.tcl
70: ROSY_RELEASE ..... 5.1.8
71: ROSY_RESOURCE ..... d:\RoSy\uid\GEO
74: ROSY_SETUP ..... d:\RoSy\setup
75: ROSY_SGDPATH ..... n:\Forst\Sgd
79: ROSY_SILENTMSG ..... 0
80: ROSY_SMLLOCKPATH ..... C:\TEMP

```

```

81: ROSY_SPECIALCHARSFILE ... specialchars.dat -->
d:\RoSy\fonts\specialchars.dat

82: ROSY_STANDARDSIZFILE .... standardsizes.dat -->
d:\RoSy\fonts\standardsizes.dat

85: ROSY_SYMLIBPATH ..... n:\Forst\sym
86: ROSY_SYSTEMDIR ..... win32
87: ROSY_TBOXLISTFILE ..... tboxlist.dat --> d:\RoSy\fonts\tboxlist.dat
88: ROSY_TEXTFILE ..... texte.dat --> d:\RoSy\fonts\texte.dat
89: ROSY_TEXTFONTSFILE ..... textfonts.dat --> d:\RoSy\fonts\textfonts.dat
93: ROSY_TMP ..... C:\TEMP
94: ROSY_TOOLBOXFILE ..... n:\Forst\tbox\sub\toolbox.dat
95: ROSY_TOOLBOXPATH ..... n:\Forst\tbox
97: ROSY_USERLOCKFILE ..... userlocks.dat --> d:\RoSy\fonts\userlocks.dat
99: ROSY_VERSION ..... GEO
100: ROSY_XBM ..... d:\RoSy\xbm
101: ROSY_XBMPATH ..... n:\Forst\tbox\xbm
103: TCL_LIBRARY ..... d:\RoSy\easi
104: TK_LIBRARY ..... d:\RoSy\easi\tk
105: UIDPATH ..... .\%U;d:\RoSy\uid\GEO\%U
107: VA_MAXWDWNR ..... 512
110: XBLANGPATH ..... .\%B;d:\RoSy\xbm\%B
115: XERRORDB ..... d:\RoSy\setup\XerrorDB
116: XKEYSYMDB ..... d:\RoSy\setup\xkeysymbd
117: XLOCALEDIR ..... d:\RoSy\setup\locale

```

Näheres zu den Environmentvariablen finden Sie in der Online-Hilfe "rhelp"

Wenn Sie das RoSy-Environment ändern (siehe unten), sollten Sie sich jeweils mit `prenv` vergewissern, daß die Änderungen wie von Ihnen beabsichtigt vorgenommen wurden. Näheres zu den Environmentvariablen finden Sie in der Online-Hilfe.

Wenn Sie wissen wollen, welche RoSy-Variablen insgesamt *erlaubt* sind, verwenden Sie die RoSy-Prozedur `crenv` (create environment) mit Eingabe von `a`. Mit `crenv` können Sie auch RoSy-Variablen setzen. Zum Setzen gehen wir jedoch wie unten beschrieben vor, da wir das Environment automatisiert erzeugen wollen. Sie erhalten etwa dieses Ergebnis:

```

----- RoSy-Environment -----
10: ROSY_APPLCMDPATH .....
11: ROSY_APPLECMPATH .....
12: ROSY_APPLGENPATH ..... d:\RoSy\Appl\BASE
13: ROSY_APPLNAME ..... Forst
14: ROSY_APPLNAMELIST .....
15: ROSY_APPLPATH .....

```

16: ROSY_APPLROOTPATH n:

17: ROSY_ASPECTPATH

18: ROSY_ATTRIBPATH

19: ROSY_BUTTONLOCKFILE buttonlocks.dat

20: ROSY_CANVSIZFILE canvassizes.dat

21: ROSY_CMDPATH

22: ROSY_COLORFILE n:\Forst\Fonts\colors1.dat

23: ROSY_CVIMPATH

24: ROSY_DIALSETTINGSFILE ... dialsettings.res

25: ROSY_DIMENSIONSFILE dimensions.dat

26: ROSY_EASI d:\RoSy\easi

27: ROSY_EASILIBPATH d:\RoSy\Appl\BASE\Easi

28: ROSY_EASIPROTOCOL

29: ROSY_ECMPROCLIST

30: ROSY_ECMPROCLISTFILE ... ecmproclist.dat

31: ROSY_ECMPATH

32: ROSY_ENVIRONFILE rosyenv.dat

33: ROSY_ENVIRONTXTFILE rosyenv.txt

34: ROSY_ERR d:\RoSy\err

35: ROSY_EXE d:\RoSy\exe

36: ROSY_EXEPATH

37: ROSY_FILLPATTERNFILE ... fillpattern.dat

38: ROSY_FONTS d:\RoSy\fonts

39: ROSY_HEDIPROTOCOL

40: ROSY_HELP d:\RoSy\help

41: ROSY_HELPPATH

42: ROSY_IPARFILE iparmsg.ger

44: ROSY_LANGUAGE deutsch

45: ROSY_LICENCEFILE

46: ROSY_LICFILE

47: ROSY_LINESIGNATFILE linesignats.dat

48: ROSY_LINESTYLEFILE linestyles.dat

49: ROSY_LINEWIDTHFILE linewidthths.dat

50: ROSY_LOADASPECTPATH

51: ROSY_LOADATTRIBPATH

52: ROSY_LOADRESFILE

54: ROSY_MSGFILE rosymsg.ger

56: ROSY_MSGPREFIX 0

58: ROSY_OCRPATH

60: ROSY_OPTPATH

61: ROSY_PATH d:\RoSy

62: ROSY_POINTSMBOLFILE ... pointsymbols.dat

64: ROSY_PPDEFPATH

66: ROSY_PRINTER 1

67: ROSY_PRINTERFILE printers.dat

68: ROSY_PRINTPROC printproc.tcl
70: ROSY_RELEASE 5.1.8
71: ROSY_RESOURCE d:\RoSy\uid\GEO

72: ROSY_RESOURCEPATH
73: ROSY_SAVERESFILE
74: ROSY_SETUP d:\RoSy\setup
75: ROSY_SGDPATH n:\Forst\Sgd
76: ROSY_SGDPRECISION
77: ROSY_SICAD
78: ROSY_SILENTERR
79: ROSY_SILENTMSG 0
80: ROSY_SMLLOCKPATH C:\TEMP
81: ROSY_SPECIALCHARSFILE ... specialchars.dat
82: ROSY_STANDARDSIZEFILE ... standardsizes.dat
83: ROSY_SUBPATH
84: ROSY_SYMLIBFILE
85: ROSY_SYMLIBPATH
86: ROSY_SYSTEMDIR win32
87: ROSY_TBOXLISTFILE tboxlist.dat
88: ROSY_TEXTFILE texte.dat
89: ROSY_TEXTFONTSFILE textfonts.dat
91: ROSY_TIFFPATH
92: ROSY_TKAPPLNAME
93: ROSY_TMP C:\TEMP
94: ROSY_TOOLBOXFILE
95: ROSY_TOOLBOXPATH
96: ROSY_TXTPATH
97: ROSY_USERLOCKFILE userlocks.dat
98: ROSY_USERROOTPATH
99: ROSY_VERSION GEO
100: ROSY_XBM d:\RoSy\xbm
101: ROSY_XBMPATH
102: SGDSYMLIB
103: TCL_LIBRARY d:\RoSy\easi
104: TK_LIBRARY d:\RoSy\easi\tk
105: UIDPATH\%U;d:\RoSy\uid\GEO\%U
107: VA_MAXWDWNR 512
110: XEMLANGPATH\%B;d:\RoSy\xbm\%B
115: XERRORDB d:\RoSy\setup\XerrorDB
116: XKEYSYMDB d:\RoSy\setup\xkeysymbd
117: XLOCALEDIR d:\RoSy\setup\locale
118: XNLSPATH

Änderungen im RoSy-Environment

Beim erstmaligen Ausführen von *Start→RoSy→RoSy-Shell* wird geprüft, ob auf dem Laufwerk N: die Directory RoSy existiert. Wenn nicht, wird sie angelegt. Wenn ja, wird die Datei hedirc.bat, falls vorhanden, ausgeführt (siehe oben).

Damit haben wir eine Methode, um Änderungen im RoSy-Environment automatisiert vorzunehmen. Alle notwendigen Änderungen nehmen wir in dieser Datei hedirc.bat vor. Wenn RoSy-Variablen Pfade bezeichnen, werden wir diese Pfade umdefinieren. Wenn RoSy-Variablen dagegen Dateien bezeichnen, werden wir diese Dateien, die default-mäßig vorhanden sind, in einen Ordner, für den wir Schreibrechte haben, kopieren und dann unseren Bedürfnissen anpassen. Ein erster Anfang könnte so aussehen:

```
rem hedirc.bat

rem Die default-Variablen von RoSy werden geändert.

set ROSY_APPNAME=Forst
set ROSY_APPLROOTPATH=n:
set ROSY_SGDPATH=%ROSY_APPLROOTPATH%\%ROSY_APPNAME%\Sgd
set
ROSY_POINTSYMBOLFILE=%ROSY_APPLROOTPATH%\%ROSY_APPNAME%\Fonts\pointsymbols.d
at
```

Ihre Aufgabe ist es nun, das RoSy-Environment so einzustellen, daß nach dem Start von *RoSy* möglichst wenig noch zu tun ist. Es werden die richtigen graphischen Attribute und Icon verwendet und wenn Sie eine aspn-, dat-, def- oder sgd-Datei Ihrer Applikation laden wollen, werden schon die vorhandenen angezeigt. Wenn Sie also Aspekte, Ebenen, Toolboxen oder Karten einlesen wollen, soll der richtige Pfad schon gesetzt sein.

Wichtige RoSy-Dokumentationen und -Hilfen

Mit *Start/RoSy/Online-Hilfe/Referenzhandbuch* erhalten Sie als HTML-Dokument Informationen zu:

- rhelp RoSy-Files (für Fortgeschrittene)
- shelp SGD-Grundlagen (Einführung)
- s2help Translator-Programme
- rifhelp Raster-Programme
- ehelp EASI-Befehle
- emhelp EASI-AR-Befehle
- Basis-Applikation
- EASI-Referenz

- Tcl 8.0-Befehle
- Tk 8.0-Befehle
- Tcl/Tk-Packages
- Externe TIFF-Dokumentation

Unter Start/RoSy/Online-Hilfe/PDF-Editorhandbuch finden Sie die Acrobat-Version des Editorhandbuchs. Und schließlich können Sie aus einer Eingabeaufforderung die folgenden - allerdings an der FH nur schlecht lesbaren - Hilfen aufrufen:

- ehelphilfe für die EASI-Befehle
- emhelphilfe für die EASI-Methoden-Befehle
- rhelphilfe für RoSy-Files
- rifhelphilfe für die Raster-Programme
- s2helphilfe für die SGD-Konverter-Programme
- shelphilfe für die SGD-Grundlagen
- tclhelphilfe zu Tcl
- tkhelphilfe zu Tk

Wichtige *RoSy*-Kommandos

Diese Kommandos sind Shell-Skripte und werden aus der RoSy-Shell aufgerufen.

- `prenv` (print environment) listet die gesetzten RoSy-Umgebungsvariablen auf.
- `bitmap` startet einen Bitmap-Editor, mit dem wir Icons, Raster-Punktsignaturen und alle xmp-Dateien von *RoSy* erzeugen können.
- `crenv` (create environment) bietet die Möglichkeit, RoSy-Variablen zu setzen. Sie können sich ansehen, welche RoSy-Variablen es insgesamt gibt (leistet `prenv` nicht). Die so gesetzten RoSy-Variablen sind aber beim nächsten Start einer RoSy-Shell verloren, daher ist es einfacher, die RoSy-Variablen in der Datei `hedirc.bat` zu speichern. Denn `hedirc.bat` wird bei jedem Start einer RoSy-Shell durchlaufen.
- `app name` setzt die Applikation `name`; das bedeutet, daß die üblicherweise vom Namen abhängenden Pfade und Dateien gesetzt werden. Wenn Sie sich also an das von M.O.S.S. vorgegebene Schema halten und zum Beispiel für Ihre Applikation `forst` die Datei mit den Farbfestlegungen `colors.dat` nennen und in den richtigen Ordner legen, brauchen Sie nur `app forst` einzugeben und alle Pfade sind richtig gesetzt. Im folgenden Auszug aus der Datei `rosy.enb_csh` sehen Sie, wie `app` die Pfade und Dateien setzt.

```
setenv ROSY_VERSION "Geo"
setenv ROSY_LANGUAGE "deutsch"
setenv ROSY_PATH "/moss adm"
setenv ROSY_SETUP "$ROSY_PATH/setup"
setenv ROSY_APPLROOTPATH "/usr/APPLICATIONS"
setenv ROSY_APPLPATH "$ROSY_APPLROOTPATH/$ROSY_APPLNAME"
setenv ROSY_APPLCMDPATH "$ROSY_APPLPATH/tbox/cmd"
```

```

setenv ROSY_APPLECMPPATH "$ROSY_APPLPATH/tbox/ecm"

setenv ROSY_BUTTONLOCKFILE "buttonlocks.dat"
setenv ROSY_CMDPATH "$ROSY_APPLPATH/prod"
setenv ROSY_COLORFILE "colors.dat"
setenv ROSY_DIALSETTINGSFILE "dialsettings.res"
setenv ROSY_DIMENSIONSFILE "dimensions.dat"
setenv ROSY_EASI "$ROSY_PATH/easi"
setenv ROSY_ECMPROCLISTFILE "ecmproclist.dat"
setenv ROSY_ENVIRONFILE "rosyenv.dat"
setenv ROSY_ENVIRONTXTFILE "rosyenv.txt"
setenv ROSY_ERR "$ROSY_PATH/err"
setenv ROSY_EXE "$ROSY_PATH/exe"
setenv ROSY_FILLPATTERNFILE "fillpattern.dat"
setenv ROSY_FONTS "$ROSY_PATH/fonts"
setenv ROSY_HELP "$ROSY_PATH/help"
setenv ROSY_IPARFILE "ipamsg.${msg_lang}"
setenv ROSY_LICENSEFILE "$ROSY_PATH/setup/rosylic_${myhostname}.dat"
setenv ROSY_LICFILE "/moss_adm/setup/license_indy.dat"
setenv ROSY_LINESIGNATFILE "linesignats.dat"
setenv ROSY_LINESTYLEFILE "linestyles.dat"
setenv ROSY_LINEWIDTHFILE "linewidths.dat"
setenv ROSY_MSGFILE "rosymsg.${msg_lang}"
setenv ROSY_MSGPREFIX "0"
setenv ROSY_OCRPATH "$ROSY_APPLPATH/ocr"
setenv ROSY_POINTSYMBOLFILE "pointsymbols.dat"
setenv ROSY_PRINTER "1"
setenv ROSY_PRINTERFILE "/moss_adm/fonts/printers.dat"
setenv ROSY_PRINTPROC "printproc"
setenv ROSY_RELEASE "5.0.0g"
setenv ROSY_RESOURCE "$ROSY_PATH/uid/${uid_ver}${uid_lang}"
setenv ROSY_SGDPATH "$ROSY_APPLPATH"
setenv ROSY_SPECIALCHARSFILE "specialchars.dat"
setenv ROSY_STANDARDSIZEFILE "standardsizes.dat"
setenv ROSY_TEXTFILE "texte.dat"
setenv ROSY_TEXTFONTFILE "textfonts.dat"
setenv ROSY_TMP "/usr/tmp"
setenv ROSY_TOOLBOXFILE "$ROSY_APPLPATH/tbox/sub/toolbox.dat"
setenv ROSY_TOOLBOXPATH "$ROSY_APPLPATH/tbox"
setenv ROSY_USERLOCKFILE "userlocks.dat"
setenv ROSY_XBM "$ROSY_PATH/xbm"
setenv ROSY_XBMPATH "$ROSY_TOOLBOXPATH/xbm"
setenv TCL_LIBRARY "$ROSY_PATH/easi"
setenv TK_LIBRARY "$ROSY_PATH/easi/tk"
setenv UIDPATH
"$ROSY_PATH/uid/${uid_ver}${uid_lang}/%U:$ROSY_PATH/uid/rosy${uid_lang}/%U"

```

```
setenv VA_MAXWDWNR "128"
```

```
setenv XAPPLRESDIR "$ROSY_PATH/uid/${uid_ver}${uid_lang}"  
setenv XBLANGPATH "$ROSY_PATH/xml/%B"
```

Dateistruktur für Applikationen (Projekte)

Dateibaum für die Applikation forst

Dateibaum für die Applikation forst
forst

```
\---asp  
  \---*.asp, *.aspn          (Aspektdateien)  
\---def  
  \---*.def                  (Attributtabellendefinitionsdateien)  
\---demo  
  \---*.sgd                  (Demo-Dateien)  
\---easi  
  \---*.ecp, *.ecm, *.tcl    (EASI-Dateien)  
\---fonts  
  \---*.dat                  (Environment-Dateien)  
\---help  
  \---*.hlp                  (Hilfe-Dateien)  
\---ocr  
  \---*.pef, *.fnt, *.rld    (OCR-Dateien)  
\---opt  
  \---*.dat                  (Optionendateien)  
\---pp  
  \---*.pp                   (Passpunktdateien)  
\---prod  
  \---*.bat, *.rsh           (Shell-Skriptdateien)  
\---res  
  \---*.res                  (Ressourcendateien)  
\---sgd  
  \---*.sgd                  (Karten)  
\---sym  
  \---*.sgs                  (Symbolbibliotheken)  
\---tbox  
  \---cmd  
    \---*.cmd                (OGS-Prozedurdateien)  
  \---ecm  
    \---*.ecm                (Easi-Prozedurdateien)  
  \---sub  
    \---*.dat                (Toolboxdateien)
```

```

    \---xbrn
        \---*.xbrn          (Bitmapdateien)
\---tif
    \---*.tif              (Scan-Dateien)
\---xbrn
    \---*.xbrn, *.gif      (Bitmapdateien)

```

Anlage hierarchischer Ebenen

Die Ebenen können Sie in *RoSy* hierarchisch anordnen. In der Regel ist ja die Zahl der Ebenen sehr groß. So haben Sie mit einer hierarchischen Struktur einen besseren Überblick als mit einer flachen Struktur, die wir daher hier nicht behandeln. In unserem Beispiel ist NULL die Hierarchie-Wurzel, von der die Äste *grenzen*, *wege* und *gewaesser* abgehen; *etumsgrenze*, *abteilgrenze*, *maschweg*, *fussweg* und *wasserflaeche* sind die Blätter.

NULL

```

\---grenzen
    \---etumsgrenze
    \---abteilgrenze
\---wege
    \---maschweg
    \---fussweg
\---gewaesser
    \---wasserflaeche

```

Realisieren können Sie diese Struktur in einer *def*-Datei mit den Befehlen *CH* (Create Hierarchy) und *CA* (Create Attribute). Mit *CA* wird eine Datenbanktabelle angelegt. Die Tabelle bekommt hier den Namen *forst_attribute*, sie hat nur eine Spalte, *ebenen*, und die Werte der Spalte haben den Datentyp *forst_hierarchie*. Die Ebenen-Indizes sind aufsteigend sortiert und sollten genügend große Lücken haben, damit Sie eventuell erforderliche weitere Ebenen noch an die richtige Stelle einfügen können.

```

CH forst_hierarchie forst_enumeration USER (
NULL  grenzen          2000,
      grenzen  etumsgrenze  2010,
      grenzen  abteilgrenze 2020,
NULL  wege            3000,
      wege    maschweg    3010,
      wege    fussweg     3020,

```

```

NULL   gewaesser           4000,
      gewaesser wasserflaeche 4010
    )
CA forst_attribute S (
  ebenen HIERAR 1 forst_hierarchie KEY
)

```

Achten Sie bitte genau auf die Syntax! Das Fehlen von Leerzeichen vor "(" führt zu schwer erkennbaren Fehlern!

Update von Ebenen

Die Ebenen werden mit den Karten fest verbunden. Wenn dann also Änderungen notwendig werden, können Sie nicht einfach die obige def-Datei ändern und erneut einlesen. Jedes Update erfordert vielmehr eine eigene def-Datei. Die Hierarchie nach dem Update soll so sein:

```

NULL
\---grenzen
  \---etumgrenze
  \---abteलगrenze
\---wege
  \---maschweg
  \---fussweg
  \---fuellmasch
\---gewaesser
  \---wasserflaeche
\---signaturen
  \---laubbaum
  \---diensthuette

```

Bei den Wegen kommt die Ebene fuellmasch dazu. Außerdem wird der neue Zweig signaturen eingefügt. Daher brauchen Sie das folgende Update (Update Hierarchy):

```

UH forst_hierarchie forst_enumeration USER (
  wege      fuellmasch      3030,
NULL  signaturen           5000,
  signaturen laubbaum      5010,
)

```

signaturen diensthuette 5020

)

Optionen-Dateien

Optionen-Dateien werden von *RoSy* häufig verwendet. Sie dienen zum Beispiel dazu, um Aspekte, Farben und Signaturen zu definieren. Ihre Struktur ist einheitlich und sie wird deshalb hier vorgestellt. Diese Struktur kennen im übrigen auch andere Systeme; siehe zum Beispiel die ini-Dateien bei Windows-Betriebssystemen.

Eine Optionendatei besteht aus Varianten. Jede Variante wiederum hat Items. Eine Variante wird durch eine Nummer oder durch einen in eckigen Klammern stehenden Namen eingeleitet:

12: oder [Header]

Die Items haben die Struktur: itemname:itemwert

Als Trennzeichen sind in beliebiger Zahl zur typographischen Gestaltung erlaubt :, =, Blank, Tab

Sind zur Eingabe des Itemwertes mehrere Zeilen erforderlich, wird mit \ ein Zeilenwechsel realisiert.

Kommentare beginnen mit ! und gelten bis zum Ende der Zeile. Sicher ist die Verwendung von ! aber nur am Anfang der Zeile. In Toolboxdateien führt ! als Kommentaranfang hinter Items oft zu Fehlern.

Z.B. werden die im *RoSy* verwendeten Farben in einer Optionendatei gespeichert. Die Datei könnte so aussehen:

```
[Header]
X11: white
EPSF1: 255
HPGL2: 255 255 255

01:
X11=black
EPSF1=0
HPGL2=0==0:::0

02:
X11: grey80
EPSF1: 192
HPGL2: 192\
      192 192
```

Die Datei hat die numerischen Varianten 1 und 2 und die Namensvariante Header. Jede Variante hat die Items X11, EPSF1 und HPGL2.

Variante Header ist in der üblichen typographischen Gestaltung dargestellt, indem : und Blanks als Trennzeichen verwendet werden.

In Variante 1 werden in eher unüblicher Weise die Trennzeichen : und = eingesetzt. In Variante 2 erstreckt sich der Itemwert zum Item HPGL2 über zwei Zeilen.

Die Varianten 0 und Header betreffen häufig globale und default-Einstellungen.

Definition graphischer Attribute

In *RoSy* können graphische Attribute nicht individuell an Kartenelemente vergeben werden. Sie werden vielmehr den Geometrietypen von Ebenen zugeordnet. So können zum Beispiel auf der Ebene diensthuette Punktsignaturen (die Signatur für Diensthütte) als auch Texte (dh) liegen, die mit unterschiedlichen graphischen Attributen versehen werden. Wenn Sie nachvollziehen wollen, welche graphischen Attribute einem Geometriotyp einer Ebene zugeordnet sind, müssen Sie in der def-Datei den Ebenenname entnehmen, dann in der aspn-Datei die Vereinbarungen für die Geometrietypen dieser Ebene suchen und schließlich in den dat-Dateien die zugeordneten Attribute und Verweise auf xbm-Dateien finden. Die Struktur ist so:

Ebenen (def-Datei)

```
\---Aspekte (aspn-Datei)
  \---colors.dat
  \---fillpattern.dat
    \---xbm-Datei
  \---linesignats.dat
    \---xbm-Datei
  \---linestyles.dat
    \---xbm-Datei
  \---linewidths.dat
  \---pointsymbols.dat
    \---xbm-Datei
```

Aspekte

Eine Aspekt-Datei dient dazu, den Ebenen einer Applikation graphische Attribute - zum Beispiel Farben, Linienarten und Signaturen - zuzuordnen. Hier folgt eine Aspektdatei zu den oben vereinbarten Ebenen:

```
[Header]
ASPECTNAME:   forstaspekte
ATTRIBUTELIST: forst_attribute.ebenen
ERROR:        on
WARNING:      off
INFO:         off
```

```
10:
ATTRIBUTENR:  1
ATTRVALLIST:  etunggrenze
DRAWOPTLIST:  \
  elemColor:  3 \
  lineSignat: 150
```

```
20:
ATTRIBUTENR:  1
ATTRVALLIST:  abteilgrenze
DRAWOPTLIST:  \
  elemColor:  1 \
  lineWidth:  18 \
  lineSignat: 164
```

```
30:
ATTRIBUTENR:  1
ATTRVALLIST:  maschweg
DRAWOPTLIST:  \
  elemColor:  1 \
  lineWidth:  18 \
  lineSignat: 181
```

```
40:
ATTRIBUTENR:  1
ATTRVALLIST:  fussweg
DRAWOPTLIST:  \
  elemColor:  1 \
  lineWidth:  25 \
  lineSignat: 185
```

```
50:
ATTRIBUTENR:  1
ATTRVALLIST:  fuellmasch
```

```

DRAWOPTLIST: \
    elemColor: 0 \
    lineWidth: 32 \
    lineSignat: 328

60:
ATTRIBUTENR: 1
ATTRVALLIST: wasserflaeche
DRAWOPTLIST: \
    elemColor: 1 \
    fillPattern:150 \
    borderWidth:18

70:
ATTRIBUTENR: 1
ATTRVALLIST: laubbaum
DRAWOPTLIST: \
    elemColor: 1 \
    lineWidth: 18 \
    opaque: 0 \
    pointSymbol:180

80:
ATTRIBUTENR: 1
ATTRVALLIST: diensthuette
DRAWOPTLIST: \
    elemColor: 1 \
    opaque: 0 \
    pointSymbol:170

```

Unter dem Item Header können Sie globale default-Zeichenattribute vereinbaren, unter den übrigen individuelle. Default-Zeichenattribute: BibSymbolStyleDef., BibSymbolWidthDef., BorderWidthDef., ColorDef., DisplayPriorityDef., FillPatternDef., LineSignatDef., LineStyleDef., LineWidthDef., OpaqueDef., PointSymbolDef., PointSymbolWidthDef., PriorityDef., WidthUnitDef.. Individuelle Zeichenattribute: BibSymbolStyleDef., BibSymbolWidthDef., BorderWidth:, Color:, DisplayPriority:, FillPattern:, LineSignat:, LineStyle:, LineWidth:, Opaque:, PointSymbol:, PointSymbolWidth:, Priority:, WidthUnit:.

Alte Aspekt-Dateien

Das Format der Aspekt-Dateien ist erweitert worden. Daher müssen alte Aspekt-Dateien umgewandelt werden. Das erreichen Sie, wenn Sie aus der Eingabeaufforderung im Ordner mit den alten Aspektdateien folgendes starten:

easish d:\RoSy\Easi\asp2aspn.ecm

Signaturen

Wenn Sie eine Applikation neu einrichten müssen, entsteht gleich zu Beginn die Forderung, neue Punkt-, Linien- und Flächensignaturen mit bestimmten Farben, Flächenfüllungen, Linienstilen und Linienstärken zu gestalten. Die default-Dateien dazu finden Sie in colors.dat, fillpattern.dat, linesignats.dat, linestyles.dat, linewidths.dat und pointsymbols.dat. Diese Dateien haben die Struktur von Optionendateien, wie sie schon oben dargestellt wurde.

Farben

Die Farbvereinbarungen werden in der Optionendatei colors.dat gespeichert. Die zulässigen Items für verschiedene Ausgabegeräte sind X11, EPSF1, EPSF2, HPGL2, GDI, SQDCI und SGDRGB. Im Beispiel werden die Farben unter einem Namen (X11) oder in RGB-Komponenten gespeichert.

```
00:  
X11:  white  
EPSF1: 255  
HPGL2: 255 255 255  
GDI:  255 255 255
```

```
01:  
X11:  black  
EPSF1: 0  
HPGL2: 0 0 0  
GDI:  0 0 0
```

```
02:  
X11:  grey80  
EPSF1: 192  
HPGL2: 192 192 192  
GDI:  192 192 192
```

```
03:  
X11:  red  
EPSF1: 255 0 0  
HPGL2: 255 0 0  
GDI:  255 0 0
```

Flächenfüllungen

Flächenfüllungen sind in fillpattern.dat gespeichert. Zulässige Items sind XBM, DEF und BITMAP. Unter XBM ist der absolute oder relative Pfad einer Bitmap-Datei angegeben, die zur Darstellung auf dem Bildschirm verwendet wird. BITMAP definiert eine 16*16 Bitmap zur Rasterfüllung auf dem Ausgabegerät. Für DEF sind diese Methoden zugelassen: deflist, solid, hatched, crosshatched, soffhatched, symbol, shading und bitmap.

```
00:

! Wasserflaeche
150:
XBM: hatch04.xbm
DEF: deflist 151 4
BITMAP: FC3F F81F F18F E3C7 C7E3 8FF1 1FF8 3FFC 3FFC 1FF8 8FF1 C7E3 E3C7
F18F F81F FC3F

151:
XBM: hatch05.xbm
DEF: soffhatched 1.0 0 0.2 1
BITMAP: 0101 8080 0000 0000 1010 0808 0000 0000 0101 8080 0000 0000 1010
0808 0000 0000
```

Im Beispiel verweist deflist auf das Item 151, das in der Farbe 4 darzustellen ist. Die Parameter für soffhatched sind hier Abstand, Winkel, Linienstärke und Skalierung.

Liniensignaturen

Die Datei für Liniensignaturen ist linesignats.dat. Als Items sind bekannt: DEF mit CONSTR, LINE, AREA, FREE (siehe Punktensignaturen), COLOR und XBM (siehe Flächenfüllungen). DEF und CONSTR enthalten die Methoden und Nebenbedingungen zur Erzeugung der Liniensignatur. Als Methoden (und Nebenbedingungen) sind erlaubt: deflist, periodic (balance, setend, snapvert), dash (fillend, fillgap, lengthend, nogapend, snapvert), sdash (fillend, fillgap, lengthend, nogapend, snapvert), vertex, relhalo, abs-halo, firstline, lastline und firstlast.

```
! Eigentumsgrenze
150:
DEF: deflist 151 0 0 152 0 0

151:
DEF: dash 0.5 20.0 0.0

152:
```

```

DEF: vertxaxis 0 1 0

LINE: 0.2 s -0.5 0 ci 0 0
FREE: s -0.5 0 ci 0 0

! Abteilungsgrenze
164:
DEF: deflist 0 1 0.0 165 0 0.0

165:
DEF: periodic 10.0 5.65
AREA: s 0.65 0.0 ci 0.0 0.0

! Maschinenweg
181:
DEF: deflist 0 1 -0.25 183 0 0.25

183:
DEF: dash 0.18 3.0 1.0

! Fussweg
185:
DEF: deflist 186 0 0.0

186:
DEF: dash 0.25 2.0 1.0

! Teilweise befestigter Weg und Maschinenweg
328:
DEF: deflist 329 0 0.0
329:
DEF: dash 0.32 10.0 0.0

```

Mit deflist wird die Liniensignatur aus anderen Items aufgebaut.

Liniestile

Die Liniestile sind in linestyles.dat gespeichert. Items sind DEF, XBM. (siehe Flächenfüllungen) und SM (SICAD-spezifisch).

```

00:
XBM: line00.xbm

01:
DEF: 1 3 5 5

```



```
s -1.0 2.8 d 1.5 4.0 d 4.0 2.8
```

```
! Nadelbaum  
180:  
LINE: 0.12 s -0.75 0.0 d 0.0 3.0 d 0.75 0.0 d 2.0 0.0  
  
! Laubbaum  
182:  
LINE: 0.12 s -0.5 0.0 ca 0.0 0.8 d 0.5 0 d 1.5 0
```

Der erste Parameter von LINE ist die Strichstärke, im übrigen sind LINE, AREA und FREE gleich aufgebaut. Sie bestehen jeweils aus Folgen von fahrbehl, x-wert und y-wert. Es gibt diese Fahrbehle: s (start), d (draw), ca (circle arc), cac (circle arc counterclockwise) und ci (circle). Zusätzlich gibt es alle Fahrbehle unter Verwendung relativer Koordinaten: rs, rd, rca, rcac und rci.

Eleganterer Einsatz von Optionendateien

Am Beginn Ihrer Arbeiten mit *RoSy* nutzen Sie sicher Optionendateien derart, daß Sie die default-Dateien in Ihre Applikation kopieren und Ihren eigenen Bedürfnissen anpassen. Längerfristig hat das aber Nachteile. Sie oder Ihr Nachfolger wissen bald nicht mehr, welche Varianten tatsächlich von Ihnen modifiziert wurden und welche nur Kopien der Firmware sind. Sobald M.O.S.S. eine neue Version mit verbesserten Optionendateien schickt, müssen Sie Ihre eigenen Optionendateien umständlich nachführen.

Am Beispiel der Punktsignaturen wollen wir zeigen, wie wir diesen Nachteil vermeiden können. `ROSY_POINTSYMBOLFILE` zeigt auf eine Datei, die jetzt nur eine Zeile enthält, nämlich das Metakommando `#filemerge`:

```
#filemerge $ROSY_FONTS/pointsymbols.dat fonts/pointsymbols_new.dat
```

`$ROSY_FONTS` zeigt auf die default-Dateien und `pointsymbols_new.dat` enthält Ihre Erweiterungen der Punktsignaturen und *RoSy* wird aus dem Parent-Ordner von `fonts` gestartet.

Toolboxen

In modernen Programmen können Sie häufig genutzte, identische Arbeitsabläufe auf Makros und Buttons legen. In *RoSy* können Sie eigene Toolboxen schaffen, die diesen Zweck erfüllen und mit denen Sie die mit *RoSy* gegebene defaultmäßige Mensch-Maschine-Schnittstelle auf die spezifischen Anforderungen Ihrer Applikation ausrichten.

Eine einfache Toolbox, die nur Button enthält, sieht zum Beispiel so aus:

```

0:
UEBERSCHR:      Demo Toolbox
ANNROW:         1
ABSPOS:         0,0
ROWCOL:         8,2
COLOR:          0,1
ANNBALLON:      on
DEFXEM:         default.xbm
CLOSEBT:        on
HILFE:          Toolbox als Demo-Beispiel

1:
UEBERSCHR:      Funktionen
NEWROW:

2:
FIELD:          Button
XEM:            sys.xbm
SYS:            tbox/sub/sys.dat
COLOR:          20,1
HILFE:          Aufruf eines Skripts
ERKLAER:        Skript

3:
XEM:            ogs.xbm
CMD:            ogs.cmd
ACT:            KP_Divide KP_Add
COLOR:          20,1
HILFE:          Aufruf einer OGS-Prozedur
ERKLAER:        OGS-Prozedur

4:
XEM:            easi.xbm
ECM:            easi.ecm
COLOR:          20,1
HILFE:          Aufruf einer EASI-Prozedur
ERKLAER:        EASI-Prozedur

5:
XEM:            res.xbm
RES:            res.dat
RESLIS:         attrTable forst_attribute \
                attrGeoTyp Points \
                attrField ebenen=laubbaum \
                attrGeoTyp Other \

```

```

                                attrField ebenen=fuellmasch

COLOR:                20,1
HILFE:                Aufruf einer Ressourcen-Datei \

und Setzen von Ressourcen
ERKLAER:              Ressourcen

6:
UEBERSCHR:           Neue Tbox

7:
XBM:                  sub.xbm
SUBTBOX              sub.dat
COLOR:                18,1
HILFE:                Oeffnen einer Subtoolbox
ERKLAER:              Subtoolbox

8:
XBM:                  new.xbm
NEWTOBOX             new.dat
COLOR:                18,1
HILFE:                Oeffnen einer neuen Toolbox
ERKLAER:              Neue Toolbox

9:
XBM:                  men.xbm
MENTBOX             men.dat
COLOR:                18,1
HILFE:                Oeffnen einer Menue-Toolbox
ERKLAER:              Menue-Toolbox

```

Die Button sind so mit Funktionalitäten versehen: In Variante 2 wird mit SYS ein System-Skript aufgerufen, Variante 3 besorgt mit OGS den Aufruf einer OGS-Prozedur und löst mit ACT zwei Editor-Aktionen aus, eine Easi-Prozedur wird mit ECM in Variante 4 angestoßen, in Variante 5 werden Editor-Ressourcen über eine Ressourcendatei mit RES und über eine Ressourcenliste mit RESLIS gesetzt. Mit den Varianten 7, 8 und 9 werden abhängige Toolboxes geöffnet.

Die wichtigsten Items in Variante 0 sind: UEBERSCHR: Name der Toolbox, ABSPOS: Absolute Position in Pixel, ROWCOL: Zeilen * Spalten, ANNROW: Anzahl der Zeilen für Erklärungen, ANNBALLON: Annotation Ballon, CLOSEBT: Schließen-Button generieren, DEFXBM: Default Icon, COLOR: Hintergrund, Vordergrund, HILFE: Hilfetext in Kombination mit F1. Die Items in Variante 0 sorgen für globale Voreinstellungen, die von den übrigen Items überschrieben werden können.

Die wichtigsten Items in Varianten ungleich 0 sind: UEBERSCHR: Text zur Gliederung, NEWROW: Neue Zeile, FIELD: mit den zur Zeit möglichen Werten Label, Button, Switch, Text, List, RadioBox, ComboBox und Scale, XBM: Icon-Datei, COLOR: Vordergrund, Hintergrund, ACT: Editor-Aktion, SYS: System-Skript-Datei ausführen, OGS: OGS-Skript-Datei ausführen, ECM: Easi-Skript-Datei ausführen, ECMLIS: Easi-Befehlsliste, RES: Ressourcen-Datei ausführen, RESLIS: Ressourcenliste, ERKLAER: Text für Annotation, CLOSE: Toolbox schließen, HILFE: Hilfetext in Kombination mit F1, SUBTBOX: Parent-Toolbox wird ersetzt, NEWTBOX: Unabhängige Toolbox erscheint, MENTBOX: Menütoolbox.

Subtoolbox, Newtoolbox und Menütoolbox haben den gleichen Aufbau wie oben. Der Unterschied liegt nur in der Art des Aufrufs.

In der folgenden Subtoolbox sind knappe Beispiele für alle übrigen Toolboxfelder aufgeführt. Sie erkennen damit die generellen Strukturen und die Möglichkeiten. Das Setzen und Abfragen der Toolboxfelder müssen Sie in EASI programmieren.

```

0:
UEBERSCHR:      Subtoolbox
ANNROW:         1
ROWCOL:         4,1
COLOR:          2,3
DEFXBM:         default.xml
HILFE:          Submenue-Toolbox

1:
UEBERSCHR:      Das ist ein Label-Field

2:
FIELD:          Label
NAME:           Label-Name
BESCHR:         Label-Field
COLOR:          18,1
HILFE:          Hilfe zu Label
ERKLAER:        Erkl\344rung zum Label-Field

3:
UEBERSCHR:      Das ist ein Switch-Field

4:
FIELD:          Switch
NAME:           Switch-Name
BESCHR:         Switch-Field
COLOR:          18,1
ROWCOL:         0, 3
HILFE:          Hilfe zu Switch

```

ERKLAER: Erkl\344rung zum Switch-Field

5:
UEBERSCHR: Das ist ein Text-Field

6:
FIELD: Text
NAME: Text-Name
BESCHR: Erste Zeile \
Zweite Zeile
COLOR: 18,1
ROWCOL: 3
SLIDER: hor ver
HILFE: Hilfe zu Text
ERKLAER: Erkl\344rung zum Text-Field

7:
UEBERSCHR: Das ist ein List-Field

8:
FIELD: List
NAME: List-Name
BESCHR: {Erstes Element} \
{Zweites Element}
COLOR: 18,1
ROWCOL: 1, 3
HILFE: Hilfe zu List
ERKLAER: Erkl\344rung zum List-Field

9:
UEBERSCHR: Das ist ein RadioBox-Field

10:
FIELD: RadioBox
NAME: RadioBox-Name
BESCHR: Rb1 Rb2 Rb3
COLOR: 18,1
ROWCOL: 0, 3
HILFE: Hilfe zu RadioBox
ERKLAER: Erkl\344rung zum RadioBox-Field

11:
UEBERSCHR: Das ist ein ComboBox-Field

12:

```

FIELD:          ComboBox
NAME:           ComboBox-Name
BESCHR:         {Erste Vorlage} \
                {Zweite Vorlage}
COLOR:          18,1
ROWCOL:         3, 5
HILFE:          Hilfe zu ComboBox
ERKLAER:        Erkl\344rung zum ComboBox-Field

13:
UEBERSCHR:      Das ist ein Scale-Field

14:
FIELD:          Scale
NAME:           Scale-Name
BESCHR:         Scale-Field
COLOR:          18,1
ROWCOL:         0, 100, 5
CONTROL:        show
HILFE:          Hilfe zu Scale
ERKLAER:        Erkl\344rung zum Scale-Field

15:
UEBERSCHR:      Annotationen

```

Systemprozeduren

Systemprozeduren werden aus Toolboxen mit dem Item SYS als bat-Dateien aufgerufen. Damit gelangen Sie aus *RoSy* auf die Betriebssystemebene.

```

@echo off
color 1f
echo Sie haben soeben das Skript %0 gestartet.
echo Sie sind angemeldet als %USERNAME%.
echo Sie arbeiten am Rechner %COMPUTERNAME% im Domain %USERDOMAIN%.
echo Und jetzt wird ein neues Fenster geöffnert, in dem notepad
echo mit dieser Datei geöffnert wird.
start notepad %0
pause
echo haben Sie die Farbänderung gesehen?
color 0f
echo sys-Skript wurde ausgeführt.
echo Echo-Ausgabe erscheint in dem Fenster, von dem hedi gestartet wurde.

```

```
echo sgddatei=%ROSY_APPLROOTPATH%\%ROSY_APPLNAME%\sgd\demo_area.cmd.sgd%
echo.
```

OGS-Prozeduren

Die objektbasierte graphische Sprache *OGS* stellt insbesondere Verfahren zur Mustererkennung zur Verfügung. Ihre Prozeduren werden mit dem Item CMD gestartet. Im folgenden Beispiel wird sie zum Setzen von Editor-Ressourcen verwendet.

```
PRINT ('Start der Prozedur')
LOADEXTERN (NAME = gif)
SETGIFVAL( attrTable = 'my_attribute')
SETGIFVAL( attrGeoTyp = 'Points')
SETGIFVAL( attrField = 'my_field=blaue')
SETGIFVAL( attrTable = 'my_attribute')
SETGIFVAL( attrGeoTyp = 'Lines')
SETGIFVAL( attrField = 'my_field=leer')

!showGrid wird gesetzt und ausgeführt!
SETGIFVAL( showGrid = 'False')
SETGIFVAL( fontName = 'DIN 6776 B Vertical')
PRINT ('Ende der Prozedur')
QUIT
```

EASI-Prozeduren

Die im erweiterbaren Applikations-Sprach-Interpreter EASI geschriebenen Prozeduren werden aus den Toolboxen mit dem Item ECM aufgerufen. EASI ist eine M.O.S.S.-Erweiterung von Tcl, mit der Sie den weitesten Zugriff in *RoSy* realisieren können. Hier verwenden wir EASI nur wieder zum Setzen von Editor-Ressourcen.

```
set FD_TAB "my_attribute"
set EBENE "my_field"

ecmset_resource attrTable $FD_TAB
ecmset_resource attrGeoTyp Points
ecmset_resource attrField $EBENE=leer

ecmset_resource attrTable $FD_TAB
ecmset_resource attrGeoTyp Lines
ecmset_resource attrField $EBENE=lilane
```

```
#showGrid wird gesetzt, aber nicht ausgeführt. Wie führen Sie aus?
```

```
ecmset_resource showGrid 1  
ecmset_resource fontName "Helvetica"
```

```
#So wird ausgeführt!  
ecmset_resource requestRedraw True
```

Produktionsdateien

Produktionsdateien dienen zur Erledigung häufig wiederkehrender Aufgaben, die keine Interaktion mit einem Bearbeiter erfordern. Hier folgt ein Beispiel zur Filterung und Vektorisierung von Tiff-Dateien.

```
# TIFF-Format in RLN-Format umwandeln  
  
tif2rln $1  
  
# RLN-Format filtern. Hier kuerzer: TIFF-Format nach RLN-Format filtern  
  
filt $1.rln -filter DICK1 -filter HOBELN -filter SCHLITZ \  
          -filter BEREINIGEN -MaxExtent 6 \  
          -filter DUENN1 -filter SCHLITZ -filter HOBELN -filter HOBELN \  
          -filter BEREINIGEN  
  
# Mittellinienenvektorisierung, in SGD-Format umwandeln  
  
ml $1  
  
geo2sgd $1 -RasterFilter off  
  
# Zwischenformate loeschen  
  
/bin/rm -f $1.rln $1.geo $1.ladr
```

```
EINFÜGENTEXT          "C:\\users\\Ka\\simple\\prod\\start1_ausz"      \\  
FORMATVERBINDEN  
  
@echo off  
rem set projekt=rok  
rem set projekt=rokl  
set projekt=forst  
rem set projekt=stadtw
```

```

rem Auffaedeln strichlierter Linien, vorlaefig. rok(-) rok1(-) forst(+-)
stadtw(--)
rem set PRODFILE=dash.cmd

rem Finden von Flaechen dreier Groessenklassen. rok(+)
rem set PRODFILE=demo_area.cmd

rem Elemente in einer Box finden. rok(+)
rem set PRODFILE=demo_box.cmd

rem Kreise bestimmter Groesse finden und ersetzen. forst(+)
set PRODFILE=demo_circle_mod.cmd

rem Clippen im Rasterbild, anwenden auf Datei rok_pass.sgd. rok1(+) forst(--)
rem set PRODFILE=demo_clip_raster.cmd

rem Clippen im Vektorbild, anwenden auf Datei rok_pass.sgd. rok1(-)
rem set PRODFILE=demo_clip_vector.cmd

rem Auffaedeln strichlierter Linien zu Kurven, vorlaefig. rok(-)
rem set PRODFILE=demo_curve.cmd

rem Texterkennung, Font-Datei notwendig. rok(-)
rem set PRODFILE=demo_font.cmd

rem Finden und Schliessen von Luecken. rok(+)
rem set PRODFILE=demo_gap_mod.cmd

rem Umgang mit Grid, anwenden auf Datei rok_pass.sgd, vorlaefig.

rem Schraffuren finden. rok(+)
rem set PRODFILE=demo_hatch.cmd

rem Infos ueber OGS-Objekte ausgeben. rok(+)
spiel. Nulltes Bei-
rem set PRODFILE=demo_info.cmd

rem Infos ueber OGS-Objekte ausgeben, nicht so wichtig. rok(+)
rem set PRODFILE=demo_infol.cmd

rem Inselflaechen finden, vorlaefig. rok(+)
rem set PRODFILE=demo_islearea.cmd

rem Elemente mit Lasso finden. rok(+)

```

```

rem set PRODFILE=demo_lasso.cmd

rem Elemente nach Laenge auswaehlen. rok(+)
rem set PRODFILE=demo_length.cmd
Erstes Beispiel.

rem Knoten, Linien und duenne Linien trennen. rok(+)
rem set PRODFILE=demo_nodes_etc.cmd

rem Knoten und Linien trennen. rok(+)
rem set PRODFILE=demo_roh.cmd
Zweites Bei-
spiel.

rem Elemente nach Groessenklassen auswaehlen. rok(+)
rem set PRODFILE=demo_size.cmd

rem Flaechen finden und durch Signatur ersetzen. rok(+)
rem set PRODFILE=demo_stone_mod.cmd

rem Ziemlich gerade Linien suchen und durch Geraden ersetzen. rok(+)
rem set PRODFILE=demo_stra_mod.cmd

rem Ziemlich gerade Linien suchen. rok(+)
rem set PRODFILE=demo_straight.cmd

rem Linien nach Linienbreite klassifizieren. rok(+)
rem set PRODFILE=demo_width.cmd

rem Entfernung von sehr kurzen Linien. rok(+)
rem set PRODFILE=hair.cmd

set PRODPATH=N:\demo\tbox\cmd
set DEFPPATH=N:\demo\def
set DATAPATH=N:\demo\sgd
set TMPFILE=%PRODFILE%.sgd
set ROSY_ASPECTFILE=N:\demo\asp\demo.asp

if %projekt%==rok set DATAFILE=rok_filt_ml_mr.sgd
if %projekt%==rokl set DATAFILE=rok_pass.sgd
if %projekt%==forst set DATAFILE=blum8.sgd
if %projekt%==stadtw set DATAFILE=stadtw.sgd

rem Remove old file

rem del %DATAPATH%\%TMPFILE%

```

```
rem Copy file

copy %DATAPATH%\%DATAFILE% %DATAPATH%\%TMPFILE%

rem Call productionline

ogs %DATAPATH%\%TMPFILE% -q -info -p %PRODPATH%\%PRODFILE%
```

Ressourcen

Wenn *RoSy* in der Firmenkonfiguration gestartet wird, fallen gewisse Voreinstellungen auf. So ist zum Beispiel in *Optionen* → *Zeichenhilfe* *Punktsymbole vektoriell zeichnen* nicht gesetzt, während in *Optionen* → *Elementfilter* der Auswahlfilter *Raster* gesetzt ist. Alle Voreinstellungen werden in der Optionendatei *dialsettings.res* vorgenommen. Wenn Sie sie für Ihre Zwecke anpassen wollen, kopieren Sie sich die default-Datei in Ihre Applikation und setzen *ROSY_DIALSETTINGSFILE* entsprechend. Um die obigen Voreinstellungen zu ändern, müssen Sie *drawSymbols* auf 1 und *numbRaster* auf 0 setzen. (siehe *rhelph* Ressourcennamensliste)

Aufgabe: Ändern Sie *dialsettings.res* so, daß in *Extras* → *Ebenenstruktur* ... Punkte und Linien auf verschiedene Ebenen gesetzt sind.

Suchhierarchien für Dateien

Bei der Systemadministration müssen Sie häufig Dateinamen angeben, damit die Datei gefunden werden kann. In der Datei *toolbox.dat* haben Sie zum Beispiel ein Icon für Ihre Toolbox in der folgenden Form angegeben:

XBM: *easi.xbm*

Die Datei *easi.xbm* können Sie, wie hier, durch einen relativen oder absoluten Pfad angeben. Ein absoluter Pfad wäre zum Beispiel:

n:/RoSy/forst/tbox/xbm/easi.xbm

Die absolute Pfadangabe hat den Vorteil, daß Sie sicher wissen, wo die Datei stehen muss, aber den erheblichen Nachteil, daß bei einer Migration Ihrer Applikation auf einen anderen Rechner ziemlich sicher alle Pfade überarbeitet werden müssen. Bei relativer Pfadangabe können Sie diesen Aufwand vermeiden.

RoSy sucht die Datei *easi.xbm* nacheinander in folgenden Ordnern:

- Pfad, von dem aus RoSy gestartet wurde

- \$ROSY_TOOLBOXPATH/xbm
- \$ROSY_APPLGENPATH/tbox/xbm
- \$ROSY_APPLPATH/xbm
- \$ROSY_APPLGENPATH/xbm
- \$ROSY_XBMPATH
- \$ROSY_XBM

Diese Suchhierarchie ist für jeden Dateityp fest in RoSy implementiert (siehe rhelp).

Symbolbibliotheken

Benutzersperren

Bei der Systemadministration ist es oft erforderlich, die unerwünschte, unabsichtliche oder unerlaubte Änderung von Dateien oder bestimmter Inhalte durch besondere Mechanismen zu schützen. Insbesondere Veränderungen in großen GIS-Dateien können zu erheblichem Schaden führen. Daher können Sie bei RoSy bestimmte Änderungen für bestimmte User mit Hilfe der Optionen-Datei userlocks.dat sperren. Die RoSy-Environmentvariable dazu ist ROSY_USERLOCKFILE. Hier ein selbsterklärendes Beispiel:

```
1:
USER: kern
HEDI: notasp notras notworld notogs
```

notasp, notattr, noteasi, notedit, notfulpp, notpp, notogs, notras, notsm1, notsym, notworld sind die Einschränkungen, die definiert werden können (siehe rhelp).

Templates

Druckoptionen

Editor-Buttonsperren

Passpunkte

Schriftarten

Textplatzhalter

X-Bitmaps