

Tutorium SAS/AF

Grischa Pfister



Verwaltungstechnisches

- PPT & Demokatalog auf Homepage der KSFE
- Pause gegen 10:30 Uhr
- Fragen? Immer!

Was ist SAS/AF?

- Application Facility
- Programmierumgebung innerhalb SAS
- Grafische Benutzeroberflächen
- Batch-Processing
- Server-seitige Komponenten (WWW)

Woraus besteht SAS/AF?

- SAS Component Object Model (SCOM)
- SAS Component Language (SCL)
- Grafische Entwicklungsumgebung
 - Jede Menge Fenster
 - Testumgebung
 - Debugger
- Organisiert in Katalogen und Einträgen

Beispielapplikation

Beispielapplikation - Analyse

- Datenvisualisierung
 - Tabelle (editierbar)
 - Grafik
- Datenbearbeitung
 - Statistik
 - Export
- Interaktion
 - Oberfläche (Radio-Box, Pushbuttons, Combo-Boxes, Listbox)
 - Menü-Leiste
 - Icon-Leiste
 - Tastatur

Die Entwicklungsumgebung

Der SAS-Explorer

- Explorer ist zentrale Schaltstelle in V8
- Managen von Katalogen und Entries über Kontextmenü
 - Erzeugen
 - Löschen
 - Duplizieren
- Entry-types von AF:
CLASS, FRAME, INTRFACE, RESOURCE,
SCL
- Außerdem wichtig:
PMENU, SOURCE, TOOLBOX

Frame

- Grafische Benutzeroberflächen
 - Buttons, Bilder, Auswahllisten, ...
- Visualisierung von Daten
 - Tabellen, Grafiken, ...
- Interaktion mit dem Benutzer
 - Abfrage von Informationen für Operationen
 - Dateneingabe, ...

Erzeugen eines Frames

- Explorer in/auf Katalog:
RMB -> New -> FRAME
- 2 Fenster:
 - Links: Components-Fenster
zeigt die verfügbaren Klassen an
 - Rechts der eigentliche Frame
Container in dem die GUI entworfen wird

Arbeiten mit Komponenten

- Komponenten hinzufügen
 - Drag & Drop
 - Doppelklick in FRAME
 - Doppelklick in Components-Fenster
 - Copy & Paste aus Kontextmenü
 - Duplicate aus Kontextmenü
- Komponenten löschen
 - 1. Markieren 2. Löschen
 - Kontextmenü oder DEL

Arbeiten mit Komponenten

- Größe von Komponenten ändern
 - Schwarz hervorgehobene Anfasser
 - Direkt eingebbar
- Komponenten markieren
 - Mit Maus Rahmen aufziehen
 - Bei gedrückter STRG-Taste einzelne Komponenten mit Maus auswählen
 - Ausgerichtet wird an der Komponente mit dem hervorgehobenen Rahmen
- Komponenten verschieben

Arbeiten mit Komponenten

- Ausrichten von Komponenten
 - Zueinander
(Layout -> Align)
links, rechts, oben, unten,
(vertikal | horizontal) zentriert
 - In der Oberfläche
(View -> (Grid display | Grid snap)
Am Raster ausrichten

Das SAS Component Object Model (SCOM)

SAS Component Object Model

- Komponente = Klasse
- Klasse besitzt Eigenschaften
 - Attribute
 - Methoden
 - Events
 - Event Handlers
 - Interfaces
- Eigenschaften werden durch Metadaten beschrieben

Eigenschaften betrachten

- Properties-Fenster
 - Kontextmenü, Icon, View -> Properties Window
 - Links: Liste aller Objekte (incl. FRAME)
 - Rechts: Tabelle mit den jeweiligen Metadaten
 - Attribute
 - Festgelegter Typ: char, num, list, array, object
 - Nach Kategorien unterteilt
Appearance, Behaviour, Data, Drag & Drop, Help,
Misc, Model/View, Size & Location

Attribute und Kategorien

- Appearance
 - Layout des Objektes: Rahmen, Schriftart, ...
- Behaviour
 - Verhalten des Objektes: Sichtbar (Y|N), editierbar (Y|N), Kommando bei Auswahl, ...
- Data
 - Interaktiv veränderbare Werte: z.B. Text in einem Text-Entry

Attribute und Kategorien

- Drag & Drop
 - Regelt ob und wie Objekt D&D unterstützt
- Help
 - Hilfefunktionalität für Objekt
- Misc
 - Sammelsurium inc.: Name des Objektes und Art der Popmenu-Verarbeitung!

Attribute und Kategorien

- Model / View
 - Angaben zu Modelkomponente die Daten liefert
- Size & Location
 - Größe und Position des Objektes

Attribute ändern

- Überschreiben
 - z.B. Name, Label, ...
- Auswahl aus einer Liste
 - z.B. backgroundColor, buttonStyle, IconStyle, ...
- Auswahl/Eingabe mit Hilfe eines Editors
 - z.B. backgroundColor, icon, ...

Shared Attributes

- Gleiche Attribute verschiedener Objekte können gemeinsam bearbeitet werden
 - Selektieren
 - Attribut in Tabelle suchen
 - Z.B. height und width in Kategorie Size & Location

Klassenklassifizierung I

- Generell werden die Klassen in SAS/AF in zwei Kategorien aufgeteilt:

visuell und non-visuell

- Klassen mit visueller Repräsentation heißen Control
- Viele non-visuelle Klassen sind Models, die mit Controls zusammenarbeiten können

Klassenklassifizierung II

nach Funktion

- Auswahlhilfen
 - Radio box control
 - Check box control
 - Combo box control
 - List box control
 - Spin box control
- Aktion anstoßen
 - Pushbutton control
 - Desktop icon control

Klassenklassifikation II

nach Funktion

- Datenvisualisierung
 - Table viewer control
 - Form viewer control
 - Chart control
 - Histogram control
 - Map control
 - Pie control

Allgemein wichtige Attribute

- Name Objekt-Identifizier
- CommandOnClick SAS-Kommando(s)
- popMenuProcessing Kontextmenü (RMB)
- Editable veränderbar
- Visible sichtbar

Balken.frame

Pushbutton

- Label [Appearance]
- ButtonStyle [Appearance]
- IconStyle [Appearance]
- Icon [Appearance]

Der Testmodus

- Applikationen können in der Testumgebung überprüft werden
 - TESTAF-Kommando
 - Icon
 - Meldung „Frame does not have a program“ kann ignoriert werden
 - FRAME hat standardmäßig eine Kommandozeile

Frame

- BannerType [Appearance]
- BackgroundColor [Appearance]
- SCL-Entry [Source]

Chart Control

- dataSet [Data]
- xVariable [Data]
- yVariable [Data]

- title

Combo Box

- Items [Data]
- selectedItem [Data]
- selectedIndex [Data]

Kommunikation zwischen Objekten

- Direkt
 - Attribute linking
 - Drag & Drop
 - Model/viewer-Kommunikation
- Indirekt
 - Steuerung durch die FRAME-SCL

Attribute linking

- Die Attribute x_a und y_b von zwei Objekten A und B werden so miteinander in Verbindung gebracht, dass eine Veränderung des Attributes bei A automatisch eine Veränderung des Attributes bei B bewirkt
- Attribute linking wird immer bei der empfangenden Komponente eingerichtet!
In diesem Falle also bei B.

Beispiel

- Combo Box mit Farben
items=red, green, blue, yellow, cyan,
magenta
- Container Box
backgroundColor wird gelinkt auf den
selectedItem der Combo Box

Model/Viewer-Kommunikation

- Model-Viewer Paradigma
- Trennung zwischen Daten und Präsentation
- Model
 - Liefert Daten unabhängig von Darstellung
- Viewer
 - Zeigt Daten an unabhängig von der Speicherform
- Z.B. Tabellensicht / Record-Sicht (FSEDIT)
- Z.B. Variablenliste in ComboBox, Listbox, SpinBox,...

Einrichten der M/V-Verbindung

- 1. Von Hand
 - Model auf Frame ziehen
 - In Model/View-Kategorie bei Viewer das Model auswählen
- 2. Automatisch
 - Model auf Frame ziehen und auf Viewer dropfen

Variable List Model

- DataSet [Data]
- typeFilter [Data]

Main.frame

Table Viewer Control

- Keine Angabe der SAS-Tabelle möglich
- Arbeitet mit SAS Data Set Model
zusammen
- Model [Model/View]
- attachedInterface [Model/View]

SAS Data Set Model

- Table [Data]
- Where [Data]
- EditMode [Data]
- columnOrder [Data]

Radio Box Control

- BorderTitle [Appearance]
- Orientation [Appearance]

Die SAS Component Language (SCL)

Aufgaben der SCL

- Ablaufsteuerung in Frames
- Eigenständige Programme ohne visuelle Oberfläche
- Definition von Klassen und/oder Methoden
- ...

Grundstruktur

- Verarbeitung erfolgt in Blöcken
- Jeder Block beginnt mit einem LABEL:
und endet mit einem RETURN;
- Blöcke können mit (LINK | GOTO)
LABEL angesprungen werden
- Standard-Labels:
INIT – MAIN – TERM
– Werden automatisch durchlaufen

Beispiele

- Sc1001
 - Standardlabel
- Sc1002
 - Standardlabel + LINK
- Sc1003
 - Standardlabel + GOTO

Grundstruktur für Frames

- INIT
 - Wird vor dem Bildschirmaufbau abgearbeitet
- MAIN
 - Nach Auswahl eines Controls
- TERM
 - Vor Schließen des Frames
- Component-Name
 - Für jede vorhandene Komponente kann ein Label-Block eingeführt werden, er wird bei Auswahl der Komponente vor MAIN abgearbeitet
- Sc1004.frame
 - Standardlabel + Objekt-Label im Frame

Sprachelemente

Variablentypen

- Primitive
 - Char [mit Längenangabe, default=200]
 - Num [Länge ist immer 8]
- Reference
 - List
 - generic object
 - specific object
 - [werden als numerische Identifier verwaltet]

Sprachelemente

Datendeklaration

- DECLARE-Statement

```
DCL type<(length)> var < = initial value>,  
      type<(length)> var < = initial value>;
```

- Ohne oder mit Initialwert

```
DCL num i k anzObs,
```

```
      char (20) name = ‚Pfister‘ vorname = Grischa‘
```

```
;
```

- sc1005

Sprachelemente

konditionelle Verarbeitung

- IF THEN ELSE
 - Bekannt aus dem Daten-Schritt ;-)
- SELECT
 - SELECT (variable);
 - WHEN (ausprägung-1) aktion;
 - WHEN (ausprägung-2);
 - aktion-1;
 - aktion-n;
 - OTHERWISE standardverarbeitung;
 - END;

Sprachelemente

Schleifenkonstrukte

- Inkrementelles DO
DO i=start TO end <BY step>;
END;
- DO WHILE
DO WHILE (Bedingung nicht erfüllt);
END;
- DO UNTIL
DO UNTIL (Bedingung erfüllt);
END;
- DO UNTIL-Schleifen werden am Schleifenende geprüft!

Sprachelemente

Funktionen

- Arbeiten mit Strings
 - String = substring(var,start<,stop>);
 - String = scan(var,n,'delimiter');
 - String = upcase(string)
 - String = compress(var<,'c'>);
 - String = tranwrd(var,what,with);

Sprachelemente

Funktionen

- Interaktion mit dem Benutzer
 - cSelection = messageBox()
 - Rc = fileDialog()
 - cSelection = openSasFileDialog()
 - cSelection = saveSasFileDialog()
 - Rc = popmenu()
- scl006

Sprachelemente

Funktionen

- Funktionen können unterschiedlich verwendet werden
 - Wert = funktion();
Funktion gibt einen Wert zurück
 - Rc = funktion();
Funktion tut etwas und gibt zurück, ob das ganze erfolgreich war
 - IF (funktion()) THEN ...

Dot-Syntax und Attribute

- Jedes Attribut eines Objektes kann über DOT-Syntax angesprochen werden
- Setzen von Attributen
 - `objekt.attribut = wert`
- Abfragen von Attributen
 - `Wert = objekt.attribut`
- `Sc1007.frame`

DOT-Syntax und Methoden

- Objekt-Methoden werden über DOT-Syntax aufgerufen

Objekt._methode(<par₁,...par_n>);

- Objekt-Methoden können ebenfalls einen Rückgabewert liefern

rc = objekt._methode(<par₁,...par_n>);

- Scl008.frame

Main.Frame

Dateiauswahl ergänzen

Sprachelemente

Slisten

- Slisten sind eines der wichtigsten Konzepte in AF
- Objekte verwalten ihre Daten in Slisten
- Teile der Funktionen & Methoden erwarten Slisten als Parameter und/oder geben sie als Ergebnis zurück

Sprachelemente

Slisten

- Verwaltung beliebig vieler Items
- Jeder Item ist durch seine Position gekennzeichnet
- Jeder Item kann zusätzlich einen Namen haben
- Jeder Item kann einen anderen Typ haben
- Erlaubte Typen: num, char, list, object

Sprachelemente

Slisten

- Erzeugen von Slisten
 - DECLARE-Statement
 - DCL list lsDaten = {};
 - DCL list lsDaten = {,Pfister','Grischa',2.1};
 - DCL list lsDaten =
{name=,Pfister',vorname=,Grischa',wert=2.1};
 - Makelist()-Funktion
 - lsDaten = makelist();

Sprachelemente

Slisten

- Löschen von Slisten
 - Jede Sliste die erzeugt wurde, muss auch wieder gelöscht werden !!!!!
 - Das gilt insbesondere für Listen, die andere Listen enthalten !!!!!

DCL list ls = {1,2,3,{,a','b','c'}};

Rc = dellist(ls);



Rc = dellist(ls, 'y'); * rekursives Löschen !!!;



Sprachelemente

Slisten

- Um einen Item aus der Sliste abzufragen wird benötigt

- Position & Typ

- $(c | n | l | o) = \text{getitem}(c | n | l | o)(\text{liste}, \text{pos});$

- Name & Typ

- $(c | n | l | o) = \text{getnitem}(c | n | l | o)(\text{liste}, \text{name});$

Sprachelemente

Slisten

- Um einen Item in eine Sliste zu schreiben wird benötigt
 - Position & Typ
 - `rc = setitem(c | n | l | o)(liste,wert,pos)`
 - [überschreibt einen bestehenden Wert]
 - Name & Typ
 - `rc = setnitem(c | n | l | o)(liste,wert,name)`
 - [überschreibt einen bestehenden Wert]

Sprachelemente

Slisten

- Um einen neuen Wert in eine Sliste zu schreiben und gleichzeitig zu benennen
 - Typ
 - `Rc = insert(c | n | l | o)(list,wert,-1,name);`
 - Mit `-1` wird an die Liste angehängt
- scl009

Sprachelemente

Slisten

- Was kann man sonst mit Slisten machen?
 - Kopieren
 - Sortieren
 - Speichern & Laden (Entry-Type SLIST)
 - Items suchen & löschen
 - Ins LOG ausgeben [call putlist(liste,titel,incr)]
 - Popup-Menü

Interaktion mit dem Benutzer

Interaktion

- Über Bildschirmobjekte
- Über Menü-Leiste
- Über Icon-Leiste

- Menü- und Icon-Leiste verwenden Kommandos, die
 - Automatisch verarbeitet werden (SAS-Kommandos)
 - In der SCL abgefangen und verarbeitet werden

Interaktion

- Das CONTROL-Statement
 - Ändert das Standard-Verhalten des Frames
 - CONTROL always;
MAIN wird durchlaufen, wenn ein unbekanntes Kommando abgegeben wurde
- Abfangen eines Kommandos
 - String = word(1<, 'u');
 - Call nextcmd();
- Int001.frame

Interaktion

- Menü-Leiste
- Wird mit PROC PMENU erstellt

- Syntax:

```
PROC PMENU cat = lib.cat;
```

```
  Menu name;
```

```
    Item i-name menu = submenu;
```

```
  Submenu submenu;
```

```
    Item i-name selection = sel mnemonic = ,c‘;
```

```
  Selection sel ,Kommando(s)‘;
```

```
run;quit;
```

- Int002.frame

Interaktion

- Icon-Leiste
- Mit Hilfe von TOOLBOX-Editor erstellen
- int002

Main.frame

Erweitern um Menü-Leiste
und Icon-Leiste

Interaktion mit SAS/BASE

Interaktion mit SAS/BASE

- Aus AF heraus kann BASE-Code ausgeführt werden (nicht i.d. Testumgebung)
 - SUBMIT-Blöcke
SUBMIT;
ENDSUBMIT;
- Variablenübergabe von SCL in BASE-Code
 - Durch Voranstellen eines „&“ vor den SCL-Variablenamen

Interaktion mit SAS/BASE

- Der Code aus SUBMIT-Blöcken wird in einem Buffer gesammelt, bis er mit einem „SUBMIT continue“ an den SAS-Prozessor übergeben wird
- Wenn reines SQL verwendet wird, kann der SQL-Prozessor direkt angesprochen werden via SUBMIT sql <continue> (auch in Testumgebung)

Sprachelemente

Interaktion mit SAS/BASE

- Makrovariablen
 - Abfragen von Werten
 - Char = symget(,m-var');
 - Num = symgetn(,m-var');
 - Setzen von Werten
 - Call symput(,m-var',wert);

Erweitern von Main.frame

Export für

HTML

PDF

Aufrufen von anderen Frames

- Call `display()`-Routine
- Kann Parameter übergeben
- Syntax:
`call display(<lib.cat.>entry.type,par1,...parn);`

Erweitern von Main.frame

Aufruf von Balken.Frame

Prozedur Means zusammensetzen

- 1. Abfrage der Parameter über Means1.frame
 - Analysevariablen
 - 1 BY-Variable
 - > Sliste zur Verwaltung
- 2. Wenn BY-Variable genannt
 - Sortieren nach BY-Variable
 - SQL-View
- 3. Prozeduraufruf zusammensetzen
- 4. Ergebnis anzeigen in Means2.frame

Main.frame erweitern

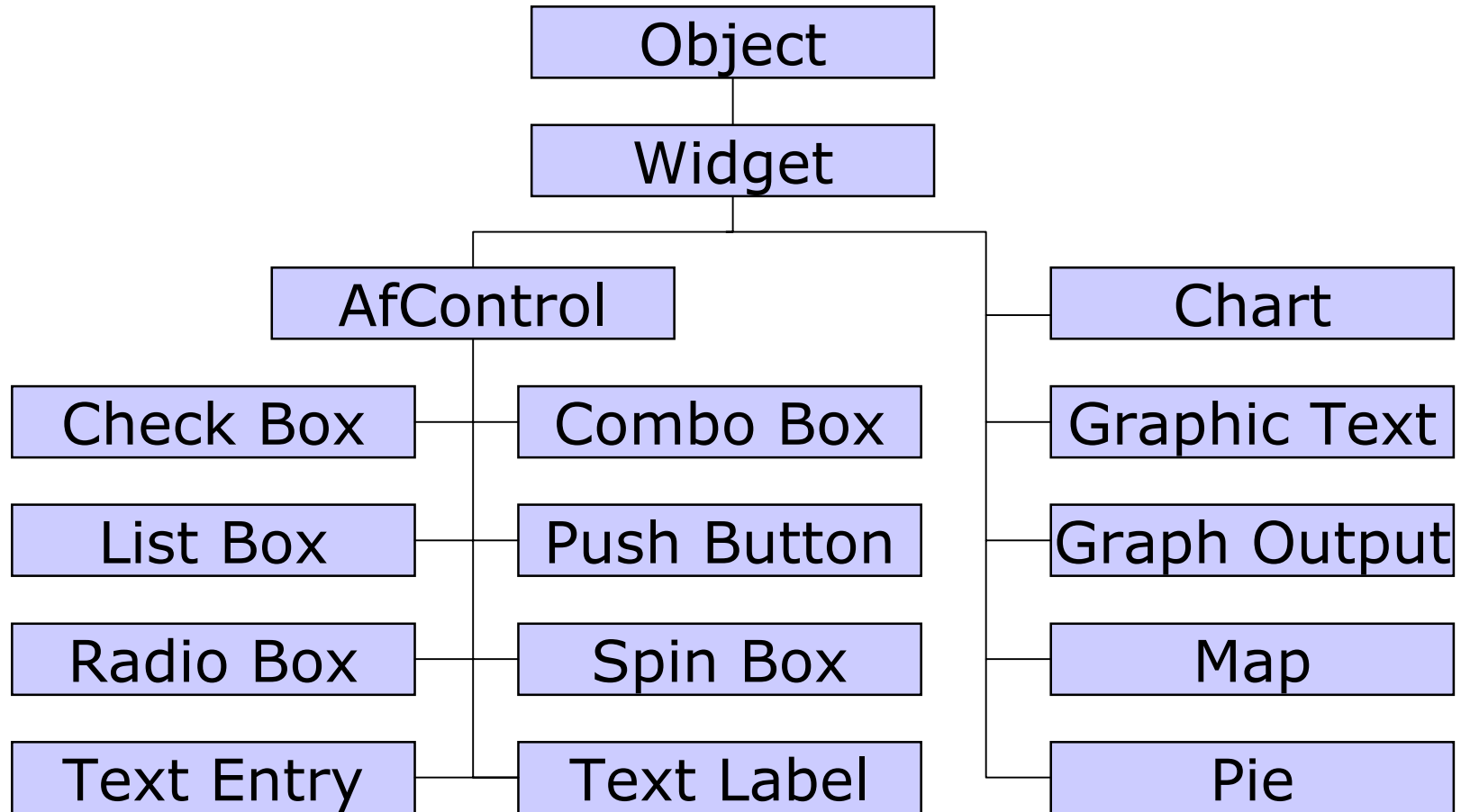
Aufruf von PROC MEANS

Klassen erstellen

Begriffe

- Klasse
 - Menge von Daten und von Operationen
 - Abstrakte Definition
- Objekt
 - Konkrete Instanz einer Klasse

AF-Klassenhierarchie



...

Klasse erstellen interaktiv

- Im Properties-Fenster Objekt auswählen
 - RMB save as (CLASS oder SCL)
 - Name vergeben
- Klassen-Editor
 - Parent-class angeben
 - Attribute ändern
 - Methoden überschreiben

Klasse erstellen – SCL

- SCL-Entry erstellen
- CLASS- & ENDCLASS-Statement ergänzen
- SYNTAX:
CLASS lib.cat.ent extends lib.cat.ent;
Attribute
Methoden
...
ENDCLASS;
- Mit SAVECLASS speichern

Eigene Klassen verwenden

- Von Explorer in Frame ziehen
- Neue Resource erstellen

Resource

- Klassenbibliothek
- Jeder Frame hat mindestens 1 Resource
- Enthält Kopien aller registrierten Klassen
- Wird bei Aufruf des Frames komplett geladen
- Bearbeitung über Resource-Editor
- Resource muß in Registry eingetragen werden (oder über Components-Fenster angezogen werden)