

ODS

Das Output Delivery System

Eine Einführung

Grischa Pfister

**Systematics Integration GmbH
Info Ware
Waldhofer Straße 11/5
69123 Heidelberg**

Grischa_pfister@info-ware.de

- Grischa Pfister

- Seit 1993 SAS aktiv

- 1999 Info Ware

- Ab 04/2003

iCASUS

- Arbeitsfelder

- Anwendungsentwicklung
AF, IntrNet, Java

- Data Warehouse

- Balanced Scorecard

- Schulungen
Base, AF, Java

- Zeitrahmen
 - 14:00 – 18:00 Uhr
 - Pausen ca. 15:15 – 15:30 und 16:45 – 17:00 Uhr
- Ablauf
 - Vortrag
 - Anwendungsbeispiele
- Fragen
 - Immer und jederzeit

Agenda

- Das Konzept
- Grundlegende Beispiele
- Output Objekte
- Der Output Kanal
- Ods Destinations Printer / PDF / RTF
- HTML Ausgabe
- Ods und Graphiken
- Ods Styles
- Ods und Reporting Prozeduren
- Table & Style Templates
- Neues in V9



- Haworth, Lauren E. (2001): Output Delivery System: The Basics, Cary, NC.
[Reihe Books by users]
- SAS OnlineDoc
 - Im Lieferumfang enthalten
 - Auch online verfügbar (mit Registrierung)
http://v8doc.sas.com/sod_register.html
<http://v8doc.sas.com/sashtml>
- support.sas.com
 - Communities -> Base SAS
- SUGI-Proceedings
 - <http://support.sas.com/usergroups/sugi/proceedings/>

Das Konzept

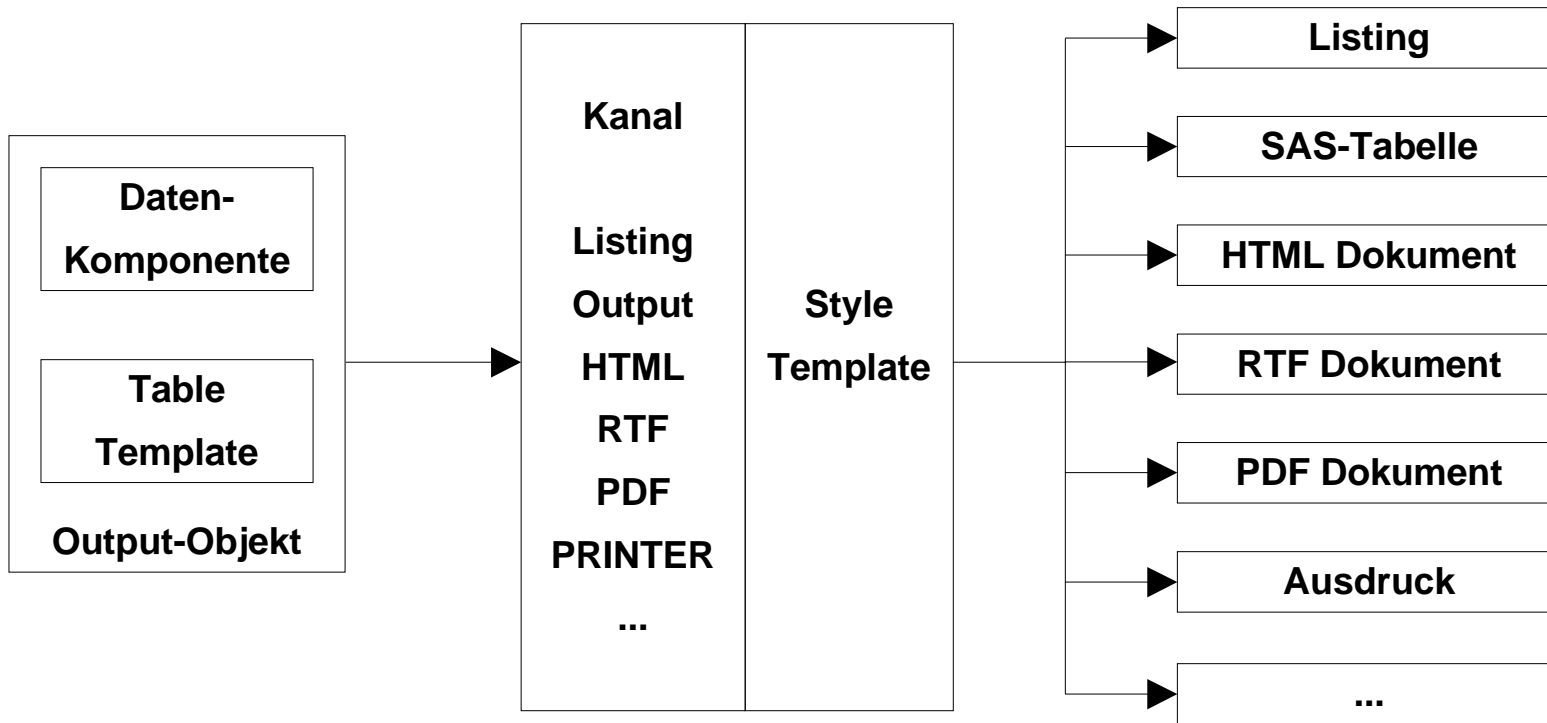
- Zwei Formen
 - Listing im Output-Fenster
 - Ausgabe-Tabellen
- Nachteile
 - Manche Prozeduren erzeugen keine Tabellen
 - Nicht alle Informationen in Tabellen
 - Statisch
 - Keine Proportional-Schriftarten
 - Keine echte Tabellenstruktur
 - Übernahme in Textverarbeitung, Tabellenkalkulation, Graphikprogramme schwer oder gar nicht möglich
 - Hoher Nachbearbeitungsaufwand

- WYSIWYG
 - Alles was im OUTPUT-Fenster erscheint steht im ODS zur Verfügung
 - Alle Informationen aller Prozeduren
- Parallelisierung des Output
 - Ergebnisse werden an viele Kanäle gleichzeitig geliefert
- Gliederung des Output
 - Prozedur-Output wird in Objekte zerlegt
 - Zugriff auf jedes einzelne Objekt möglich

ODS Ausgabe-Kanäle

- Listing Output-Fenster
- Output SAS-Tabelle
- HTML Dokument für WWW
- PDF Dokument für Acrobat Reader
- RTF Dokument für Textverarbeitung
- Printer Systemdrucker
- PS Postscript
- Experimental
 - LateX
 - XML
 - Markup
 - ...

Funktionsweise des ODS



- Globale Statements
 - Voll batch-fähig
- Schalterprinzip
 - Kanal wird geöffnet
 - Output von Prozeduren/Data Steps
 - Kanal wird geschlossen
 - Ergebnis ist betrachtbar

Öffnen und Schließen von ODS-Kanälen

- Öffnen eines Kanals

- `Ods Kanal;`

- `Ods Kanal File = „pfad“;`

- `Ods Kanal output-objekt = SAS-Tabelle;`

- Schließen eines Kanals

- `Ods Kanal Close;`

- `Ods _all_ Close;`

- Grundprinzip

- Es muss mindestens ein Kanal geöffnet sein

- Ansonsten Warning im LOG-Fenster

Grundlegende Beispiele

- Ein-/Ausschalten
 - `Ods Listing;`
 - ...
 - `Ods Listing Close;`
- Beispiel01.sas

- Ein-/Ausschalten

- `Ods Html File = fileref | „pfad“;`

- ...

- `Ods Html close;`

- Beispiel02.sas

- Ein-/Ausschalten

- `Ods Pdf File = fileref | „pfad“;`

- ...

- `Ods Pdf Close;`

- Beispiel03.sas

- Ein-/Ausschalten

- `Ods Rtf File = fileref | „pfad“`

- ...

- `Ods Rtf Close;`

- Beispiel04.sas

- Ein-/Ausschalten

- `Ods Printer <Display>;`

- `Ods Ps File = fileref | „pfad“;`

- ...

- `Ods Printer/Ps Close;`

- Beispiel05.sas

Mehrere Kanäle gleichzeitig verwenden

- Unterschiedliche Kanäle

- `Ods Pdf File = fileref | „pfad“;`
- `Ods Html File = fileref | „pfad“;`
- ...
- `Ods _all_ Close;`

- Beispiel06.sas

Mehrere Kanäle gleichzeitig verwenden

- Gleiche Kanäle

- Ods Html (id=1) File = *fileref* | „*pfad*“;

- Ods Html (id=2) File = *fileref* | „*pfad*“;

- ...

- Ods Html (1) Close;

- ...

- Ods Html (2) Close

- Beispiel07.sas

- Tools -> Options -> Preferences
- Tabber „Results“
 - Ods Listing
 - Ergebnisse als HTML
 - Style = default
 - Web-Browser
 - ◆ Umstellen unter Tabber „Web“
 - Direkte Anzeige der Ergebnisse

Output Objekte

- Ods teilt Prozedur-Output in Objekte ein
 - Jedes Objekt ist identifizierbar
 - Jedes Objekt ist einzeln ansprechbar

- Anzahl der Output Objekte ist abhängig von
 - Prozedur
 - Prozedur-Optionen
 - By-Gruppen

- Identifizieren von Output Objekten
 - `Ods Trace On;`
 - ...
 - `Ods Trace Off;`
- Inhalt des Trace-Satzes
 - Name des Output Objektes
 - Label des Output Objektes
 - Name des verwendeten Table Templates
 - Pfad des Output Objekts
- Beispiel08.sas
- Results-Fenster zeigt Output Objekte an

Auswahl von Output Objekten

- Select-/Exclude-Liste
 - Enthält Auswahl von Output Objekten
 - Eine globale Liste
 - Eine Liste für jeden aktiven Kanal
- Anzeigen der Select-/Exclude-Liste
 - `Ods <Kanal> show;`
 - Dabei muss der Kanal muss aktiv sein
- Beispiel09.sas

Voreinstellungen der Auswahllisten

- Globale Select-/Exclude-Liste
 - Select All
- Alle Bericht-Kanäle
 - Select All
- Output-Kanal
 - Exclude All
- Beispiel11.sas

- Gültigkeitsdauer der Auswahl
 - Nachfolgende Prozedur
- Option Persist
 - macht Auswahl eines Objektes dauerhaft
 - muss mit neuem Statement überschrieben werden
 - `Ods <Kanal> (Select | Exclude)
ods-objekt(Persist);`
 - ...
 - `Ods <Kanal> Select All;`
- Beispiel12.sas

Output Kanal

- Ein-/Ausschalten

- `Ods Output ods-objekt = SAS-Tabelle ...;`

- ...

- `Ods Output Close;`

- Beispiel13.sas

- Verarbeitung von By-Gruppen

- Standardverhalten:
alle By-Gruppen in eine Tabelle
- Match_all-Option:

Ods Output

```
ods-objekt(Match_all<=mVar>) = SAS-Tabelle;
```

jede By-Gruppe in eigene Tabelle
automatische Benennung
mVar enthält Namen der erzeugten Tabellen

- Beispiel14.sas

- Verarbeiten mehrerer Prozeduren

- Match_all + Persist-Option

```
Ods Output
```

```
ods-objekt(Match_all<=mVar  
<Persist=run|proc>>) = SAS-Tabelle;
```

```
...
```

```
Ods Output Close;
```

- Erhält Benennungsschema

- Beispiel14a.sas

ODS Destinations Printer / PDF / RTF

- Grundeinstellung
 - Jede Prozedur beginnt neue Seite
 - Innerhalb der Prozedur selbst keine Eingriffsmöglichkeit
- Seitenumbrüche steuern
 - Startpage-Anweisung
 - `Ods Kanal Startpage = off;`
Schaltet automatischen Seitenumbruch nach Prozedur aus
 - `Ods Kanal Startpage = on;`
Schaltet automatischen Seitenumbruch nach Prozedur ein
 - `Ods Kanal Startpage = now;`
In Verbindung mit Ods Startpage off
erzwingt Seitenumbruch an bestimmter Stelle
- Beispiel15.sas

Allgemeines Layout

- Optionen für die Seitengestaltung

- `Title` Titel
- `Footnote` Fußnoten
- `[no]Center` linksbündig
- `[no]Number` keine Seitenzahlen
- `[no]Date` kein Datum
- `Orientation =
portrait|landscape` Hochformat|Querformat
- `paperSize = „Iso A4“ | (x,y)` Seitengröße
- `[left|right|top|
bottom]Margin = n` Randeinstellungen

- Funktionieren aber nicht immer mit allen Kanälen...

- Beispiel16.sas

- Title & Footnote

- Ähnlich SAS/GRAPH verwendbar

- ◆ `Font =` Schriftart
 - ◆ `Height =` Schriftgröße
 - ◆ `Color =` Farbe
 - ◆ `Justification =` Ausrichtung

- Beispiel17.sas

Inline-Formatierung

- Ods Escapechar = „Zeichen“;
Definiert Zeichen als Beginn einer Escape-Sequenz
- Standard Escape-Sequenzen
 - [arg]n [arg] Leerzeilen einfügen
 - S={ } Style-Attribute
 - {super text} hochgestellt
 - {sub text} tiefgestellt
 - [M Markierung (nur Printer)
 - -2n Einzug auf letzte Markierung]
 - R/[RTF|PDF|HTML]““ Roh-Text für eine Destination
- InlineFormatierung01.sas

- `Notoc` Kein Inhaltsverzeichnis
- `NoPdfNote` Keine Annotations
 - Style-Attribut „flyover“ wird in PDF Note umgewandelt
- `Columns = n` Ausgabe in mehrere Spalten
- `Text = „“` Einfügen von Text

- `V9`
 - `Compress = [0|9]`

- BodyTitle Titel im Text (buggy)
- Author = Angabe des Autors
- Title = Angabe des Titels
- RTF-spezifische Felder verwenden
 - Title/Footnote beginnt mit „{“, endet mit „}“
 - R/RTF „Anweisung“
 - `{Seite \field{*\fldinst PAGE } }`
- http://support.sas.com/rnd/base/topics/templateFAQ/Template_rtf.html
- <http://msdn.microsoft.com>
 - Nach „rtf specification“ suchen
- InlineFormatierung02.sas

- V9

- <http://support.sas.com/rnd/base/topics/odsprinter/new9.html>

- Ods Layout

- Image Scaling

- Weitere Inline-Formatierungen

- ◆ Z.B. freie Seitenumbruch-Steuerung

HTML Ausgabe

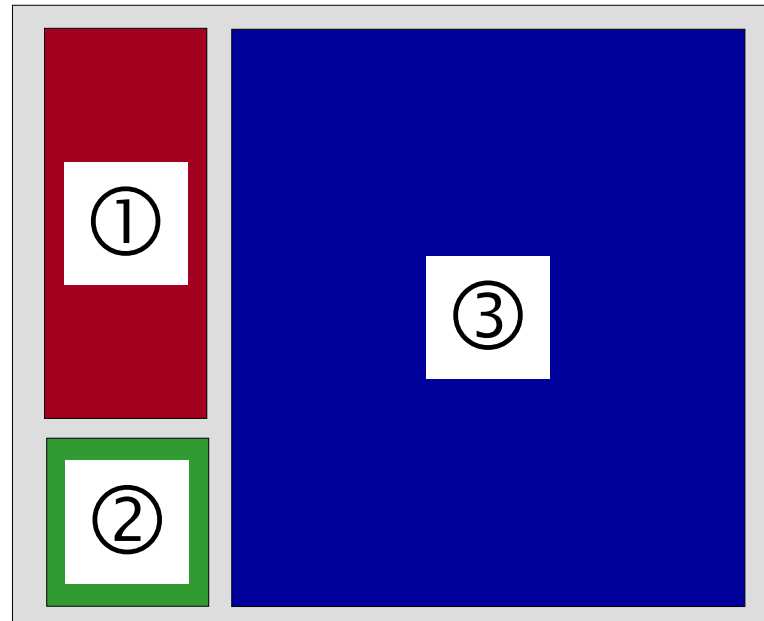
- Erstellen ganzer Web-Sites im Batch
 - Statische Reports als HTML
 - Automatische Erzeugung von Framesets
 - Verwendung von Cascading Style Sheets
 - „Verlinken“ verschiedener Seiten
 - Interaktive Graphiken
 - Graphiken mit Hyperlinks
 - Aufsplitten der Ergebnisse auf mehrere Dateien

- Was ist ein Frameset?

- Konzept in HTML
- Zeigt gleichzeitig mehrere HTML-Dateien an
- Unterteilt die Anzeige in verschiedene Bereiche (Frames)

- Mit Ods möglich

- ① Inhaltsverzeichnis
- ② Seitenverzeichnis
- ③ Hauptseite



- Erstellen eines Frameset mit ODS

- Ods Html Path = „verzeichnis“ (Url = none)
Body = „name1.html“
Contents = „name2.html“
Page = „name3.html“
Frame = „name4.html“;

- Path: Pfad unter dem alle Dateien angelegt werden

- (Url = none): Hyperlinks werden relativ angelegt

- Body: Hauptseite

- Contents: Inhaltsverzeichnis

- Page: Seitenverzeichnis

- Frame: Html-Seite, die den Frameset enthält

- Beispiel18.sas

Unterstützung von Cascading Style Sheets

- Hintergrund

- HTML: tag-gesteuerte Layoutanweisungen, ASCII
- Dateien sollen möglichst klein sein
- Wie wird Layout definiert?
 - ◆ Für jedes Element einzeln
 - ◆ Definition von Klassen (Cascading Style Sheet)

- Verschiedene Möglichkeiten CSS zu verwenden

- `Ods Html ... Css;`
eingebettetes Stylesheet
- `Ods Html ... Stylesheet = „name.css“;`
CSS wird erstellt
- `Ods Html ... Stylesheet = (Url=„“);`
vorhandenes Stylesheet wird benutzt

- Beispiel19.sas



- Aufsplitten des Output

- HTML-Dateien sollten möglichst klein sein

- Grundeinstellung: aller Prozedur-Output landet in einer Datei

- NewFile-Option

- `Ods Html ... NewFile = Output | Page | Proc ;`

- Erstellt neue Dateien für

- ◆ Jedes Output Objekt
 - ◆ Jede logische Seite
 - ◆ Jede Prozedurausgabe

- Beispiel20.sas

- Eingriffsmöglichkeiten in das Inhaltsverzeichnis
 - `Ods Proclabel „text“;`
Ändert die Hauptüberschrift
 - Option `Contents = „text“` bei den Prozeduren Print, Report und Tabulate (hier auch als Option im Table-Statement)
Ändert die Unterüberschrift(en)
 - `Ods Noptitle;`
Unterdrückt den Prozedurspezifischen Titel
- Beispiel21.sas

- HTML direkt verwenden
 - Titel und Fußnoten dürfen HTML-Code enthalten
 - 1. Zeichen muss „<“ sein!!
- Beispiel22.sas
- Corporate Design nutzen
 - **Notop**
Keine HTML-Startsequenz
 - **Nobot**
Keine HTML-Schlusssequenz
- Beispiel23.sas

Ods und Graphiken

- Rtf

- Scheint momentan nicht korrekt zu funktionieren -> Installationsproblem?
- TTFs gehen verloren
- -> Graphiken getrennt erstellen
- Devices wmf, emf, sasemf

- Beispiel24.sas

- Pdf

- Vortrag von Heinrich Stürzel auf der KSFE 2002

- HTML

- Pixel-Graphiken in GIF- bzw. JPEG-Format
- Werden als Referenz in Seite integriert ()
- `Ods Html Gpath = „verzeichnis“ (Url=none);`

setzt das Verzeichnis, in dem Graphiken gespeichert werden

- SAS/GRAPH Treiber

- JPEG
- GIF
 - ◆ `Goptions`
 - `xPixels = n yPixels = n`
 - `[no]Transparency`

- Beispiel25.sas

- Spezielle SAS/GRAPH Treiber

- ActiveX

- Java

- Interaktive Graphiken mit diversen Möglichkeiten wie

- ◆ Anzeige von Werten

- ◆ Rotieren

- Strg + LMB

- ◆ Verschieben

- Alt + LMB

- ◆ Zoomen

- Shift + LMB

- ◆ Selektion

- Strg + Shift + LMB

- ◆ Umstellen des Graphiktyps

- Kontextmenü

- ◆ ...

- Beispiel26.sas

- Erweiterung der Graphik-Prozeduren
 - Graphiken mit Hyperlinks versehen
 - Option `Html = variable` in den verschiedenen Statements
- Schritte zum Erstellen von gelinkten Graphiken
 - 1. Tabelle mit Drillvariable erzeugen
`variable = „href = name.html“;`
 - 2. Graphik erzeugen
 - 3. Die Seiten erzeugen, auf die verwiesen wird
- Beispiel27.sas

ODS Styles

- Abstrakte Beschreibung von ODS-Elementen
 - Dokument
 - Inhaltsverzeichnis
 - Seitenverzeichnis
- Layoutangaben
 - Farben
 - Schriftarten (Größe, Gestaltung)
 - Tabellengestaltung (Abstände, Ränder)
 - Beschriftung (z.B. des Inhaltsverzeichnisses)
- Werden immer verwendet
 - HTML -> default
 - RTF -> rtf
 - PDF -> printer
 - PRINTER/PS -> printer

- Mitgelieferte Styles

- Default

- D3d

- Brick

- statDoc

- Minimal

- sasWeb

- barrettsBlue

- ...

- Tools -> Options -> Preferences -> Results Tabber

- Verwendung von Styles

- `Ods Kanal style = name;`

- Beispiel28.sas

ODS und Reporting Prozeduren

Proc Print, Proc Report, Proc Tabulate

- Neue Style-Option

- `style = {style-anweisung ...};`

- `style(element) = {style-anweisung...};`

- Überschreibt Layoutanweisungen des gewählten Styles

- Style Anweisungen (Auswahl)

- `Background =` Farbe

- `Foreground =` Farbe

- `borderColor =` Farbe

- Schriftart spezifisch

- ◆ `Font_face=` Schriftart

- ◆ `Font_size=` Schriftgröße

- ◆ `Font_weight=` Schrift-Gewichtung (normal/fett)

- ◆ `Font_style=` Darstellung (italic=kursiv)



→ Zellenlayout

- ◆ Cellheight= Zellenhöhe
- ◆ Cellwidth= Zellenbreite
- ◆ Cellpadding= Randabstand innerhalb Zelle
- ◆ Cellspacing= Abstand zwischen Zellen
- ◆ Just= horizontale Ausrichtung
- ◆ Vjust= vertikale Ausrichtung

→ Graphische Elemente

- ◆ BackgroundImage= Hintergrundbild
- ◆ PreImage= Bild vor Objekt
- ◆ PostImage= Bild nach Objekt

→ HTML spezifisch

- ◆ PreHtml=
- ◆ PostHtml=
- ◆ Url=
- ◆ Flyover=

- ODS & Proc Print

- Output in Elemente unterteilt

- Jedes Element kann einzeln formatiert werden

- Style-Angaben für

- ◆ Gesamt-Tabelle
 - ◆ Spaltenüberschriften
 - ◆ Obs-Spalte
 - ◆ Datenbereich
 - ◆ Einzelne Spalte
 - ◆ Einzelne Zelle (Ampel-Funktion)

- Achtung bei Formaten:

- ODS verarbeitet formatierte Werte !!!

- Beispiel29.sas

- ODS & Proc Report

- Globale Angaben als Optionen in Prozeduraufruf

- ◆ Header
 - ◆ Column

- Spaltenbezogene Angaben als Optionen im Define-Block

- ◆ Header
 - ◆ Column

- Beispiel30.sas

- ODS & Proc Tabulate

- Verwenden der Style-Option in verschiedenen Statements
- Class
- Classlev
- Var
- Keyword
- Box-Option
- Table-Statement

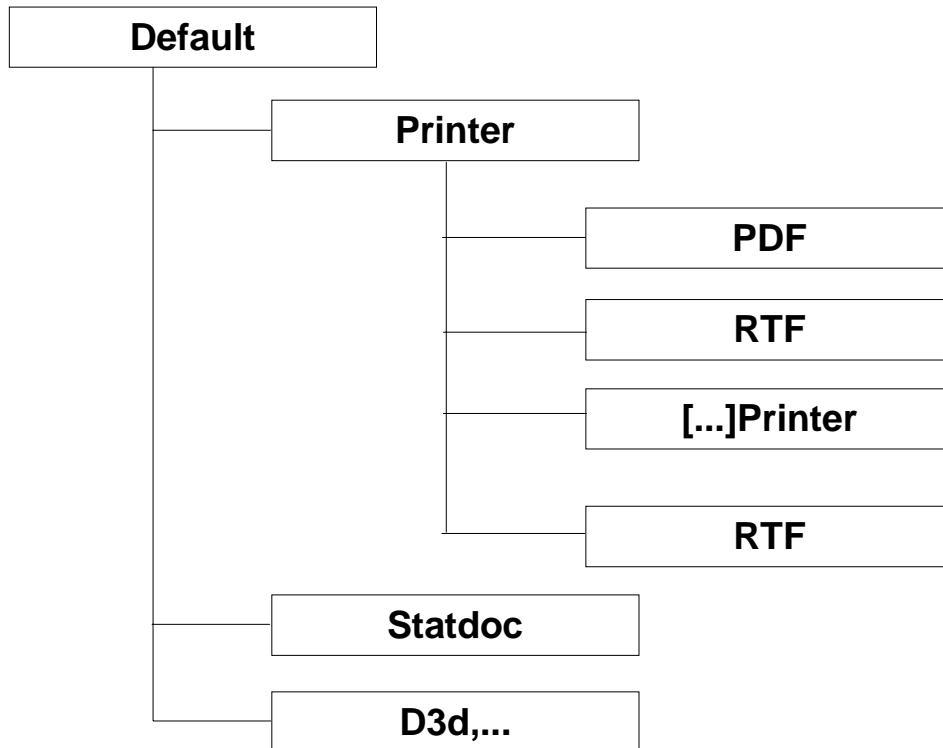
- Beispiel31.sas

Table & Style Templates

- Table Template
 - Grund-Layout
 - Aufteilung in Spalten
 - Spaltenbeschriftung
 - Zahlenformatierung
 - Kann auch Style-Anweisungen enthalten

- Style Template
 - Layout der Elemente
 - Abhängig vom Kanal

- Objekt-orientierter Ansatz



- Verwaltung von Table & Style Templates
 - Ods Template Viewer
 - ◆ Kommando OdsTemplate
 - ◆ Results-Fenster
 - Hierarchische Organisation in MT itemstore
 - ◆ Defaults: Sashelp.Tmplmst (schreibgeschützt)
- Eigene: Sasuser.Templat
 - Anlegen eigener itemstores mit Proc Template
 - Ods Suchpfad
 - ◆ `Ods Path Show;`
 - ◆ `Ods Path libref.itemstore(modus) ...;`
modus = read | update | write

- Table Templates verändern
 - Beispiel Proc Univariate: Extreme Observations
 - Beschriftung eindeutschen
 - Wird in Sasuser.Templat gespeichert

- Beispiel32.sas

- Style Templates anpassen
 - Neuen Style ableiten
 - Gewünschte Änderungen vornehmen
- Aufbau des Styles Default
 - Definition von Schriftarten
 - Definition von Farben
 - Definition der einzelnen Elemente
 - Hierarchie in OnlineDoc beschrieben
 - Base SAS Software
 - > Guide to the SAS Output Delivery System
 - > Reference
 - > The TEMPLATE Procedure
 - > Concepts
- Beispiel33.sas

- Data Step Reporting & Table Templates

- Mit Table Templates Schablonen entwerfen

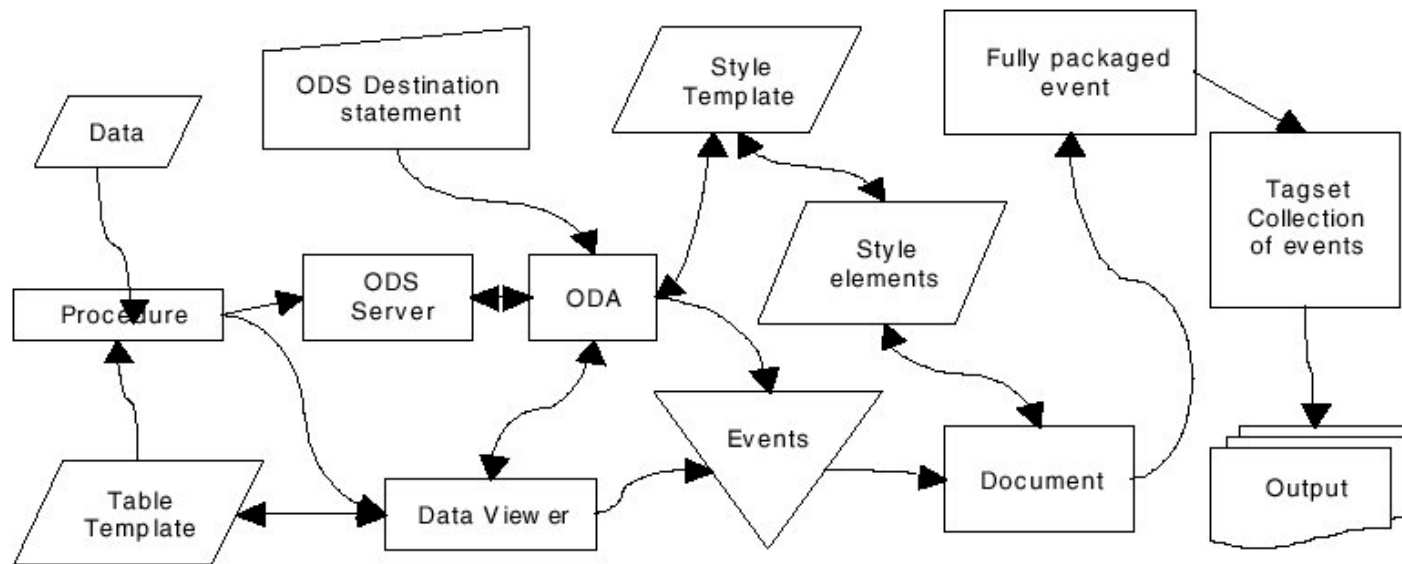
- ◆ Spaltenlayout
 - ◆ Beschriftung
 - ◆ Style-Anweisungen
 - ◆ Ampelfunktion mit Hilfe von
 - Cellstyle-Anweisung
 - Style-Anweisung + Expression-Funktion
 - Style-Anweisung + Format
 - ◆ Berechnete Spalten
 - ◆ Verändern von Werten

- Beispiel34.sas – Beispiel38.sas

Neues in V9

Ods Markup

- Neue Destination
- Stellt Event-Modell zur Verfügung



- Event = Ereignis
 - Start & finish ODER stateless
 - Mit `Trigger event` können eigene Events aufgerufen werden
- Verhalten in Tagsets zusammengefasst
- Z.B. Tagset Csv
- Tagsets01.sas

Tagsets als Informationsquelle

- Informationen über Style-Elemente anzeigen
 - Style_popup
 - Style_display
 - namedHtml
- Informationen über Events anzeigen
 - Event_map
- Volle Freiheit für Benutzer aber hohe Komplexität
- <http://support.sas.com/rnd/base/topics/odstags/>
- Eric Gebhart: MARKUP: The Power of Choice and Change. Sugi 27, Advanced Tutorials
- Tagsets02.sas

- <http://support.sas.com/rnd/base/topics/odsdocument/>
- Experimentell in V8.2, produktiv in V9
- Speichern und Wiedergeben von Output-Objekten
- Z.B. Ändern der Reihenfolge
- Wird als Membertype Itemstore gespeichert
- **Ods Document**
`Name = <Lib.>Itemstore(mode);`

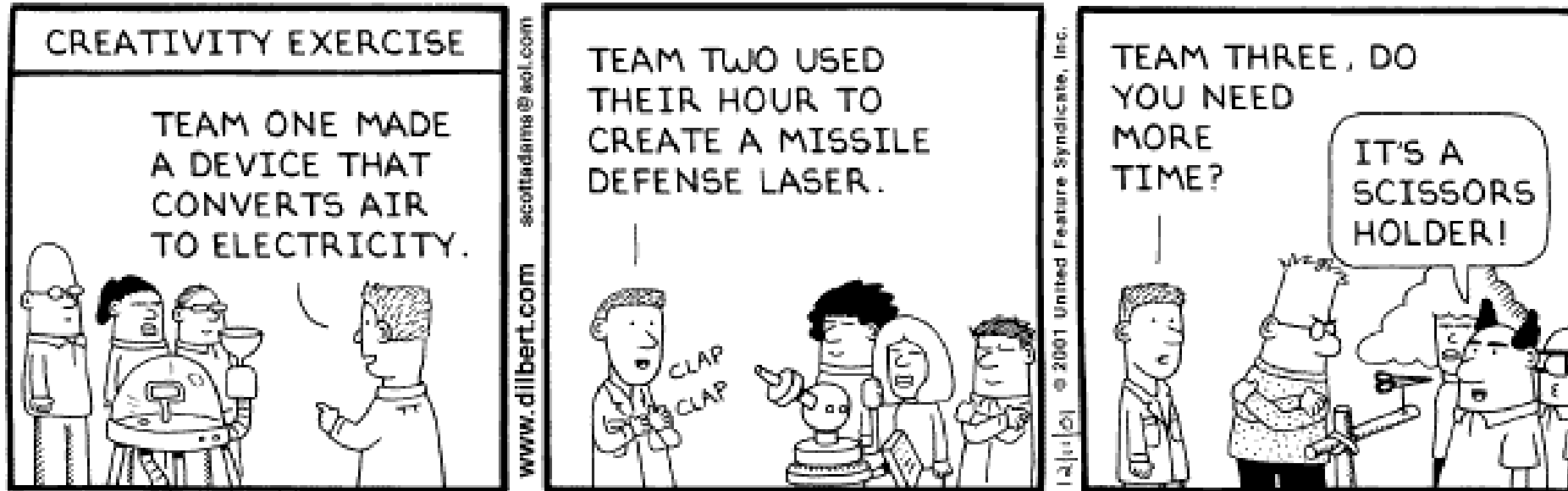
- Proc Document Name= <Lib.>Itemstore;
List / Levels = all;
Replay <object <, object>;
Make *directory*;
Setlabel *directory* „text“;
Dir *directory* | ^^;
Copy *object* To ^;
...
Run;
Quit;

→ ^ Dieses Verzeichnis (.)

→ ^^ Ein Verzeichnis hoch (..)

- OdsDocument01.sas

Noch Fragen?



Copyright © 2001 United Feature Syndicate, Inc.

Weitere Fragen oder Anregungen ?

Grischa_pfister@info-ware.de

Ab 01.04.2003

g.pfister@icasus.de

icasus