

- Nutzen
- Technik
- Beispiele

# **AUTOMATISIERTES METADATEN MANAGEMENT**



- Installation
- SAS Clients
  - 1. Automatisiert durch Eigenentwicklungen
  - 2. Automatisiert durch SAS Systemkomponenten



# 1. Metadaten durch Eigenentwicklungen

Warum???

- Menge der Objekte
- Zeit
- Qualität
- Nachvollziehbarkeit
- Reproduzierbarkeit
- Revisionsanforderung

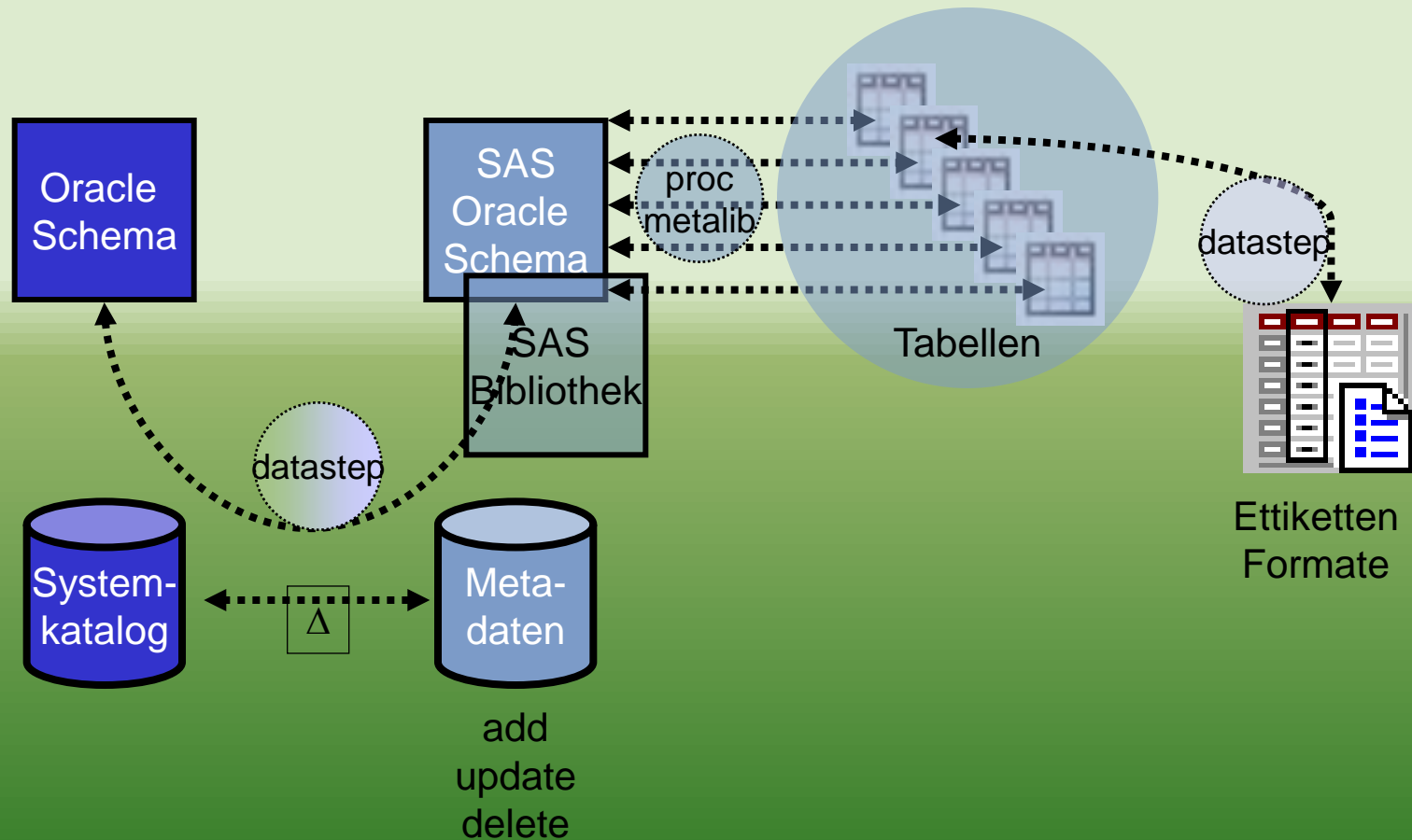
Pro Umgebung:

- Oracle – Server ( $\sim 8$ ), - Bibliotheken ( $>150$ ), - Datenbankschemen ( $>100$ )
- Benutzergruppen ( $>600$ )
- Berechtigungen



# 1. Metadaten durch Eigenentwicklungen

## Anwendung 1: Abbildung von Oracle Datenstrukturen





# 1. Metadaten durch Eigenentwicklungen

## Anwendung 1: Abbildung von Oracle Datenstrukturen

- Vorteile des automatisierten Prozesses
  - Prozess ist reproduzierbar, automatisierbar, konsistent
  - Massendatenfähig
  - Automatische Übernahme von Veränderungen der Oracle Welt in die SAS Metadaten
- Vorteile der Verwendung der Metadatenengine:
  - Keine Zugangsdaten (Userid, Password) im Quellcode
  - Keine Zugriffsoptionen (z. B. commit, direct\_sql etc.)
  - Keine umgebungsspezifischen Angaben
  - Anwendung von Etiketten und Formaten auf Daten einer Access-Schnittstelle
  - Metadaten – Zugriffsberechtigungen kommen zum Einsatz



# 1. Metadaten durch Eigenentwicklungen

## Anwendung 1: Abbildung von Oracle Datenstrukturen - Vorteile

Metaengine libname statement:

```
LIBNAME DISCO META LIBRARY = ,A_ALLM_DISCO';
```

Dadurch generiertes libname statement:

```
LIBNAME DISCO ORACLE  
CONNECTION=GLOBAL DBCOMMIT=0  
DIRECT_EXE=DELETE INSERTBUFF=1000  
PRESERVE_COL_NAMES=YES  
PRESERVE_TAB_NAMES=YES  
DIRECT_SQL=YES READBUFF=2000  
SHOW_SYNONYMS=YES  
UPDATEBUFF=1000 PATH=DME01R1  
SCHEMA=A_ALLM_DISCO  
USER=LESEBERECHTIGTER  
PASSWORD="{SAS002}KF0849DXWF901" ;
```



# 1. Metadaten durch Eigenentwicklungen

## Anwendung 1: Abbildung von Oracle Datenstrukturen

### Vorteile

	VSTAG_NR	VSTAG_DAT	LIMIT_ID	MANDANT_SL	HOTSPOT_KL_ID	RISIKOART_SL	LIM_ART_SL	PORTF_KTX_SL	LIM_GROE_SL	LIM_T
1	6	03JUN2008	356	1	1	1	10	10001	4	1
2	6	03JUN2008	365	1	1	1	10	10001	4	1
3	6	03JUN2008	369	1	1	1	10	10001	4	1
4	6	03JUN2008	5921	1	1	1	10	10001	4	1
5	6	03JUN2008	5922	1	1	1	10	10001	4	1
6	6	03JUN2008	5923	1	1	1	10	10001	4	1
7	6	03JUN2008	5924	1	1	1	10	10001	4	1
8	6	03JUN2008	5925	1	1	1	10	10001	4	1
9	6	03JUN2008	5926	1	1	1	10	10001	4	1
10	6	03JUN2008	5927	1	1	1	10	10001	4	1

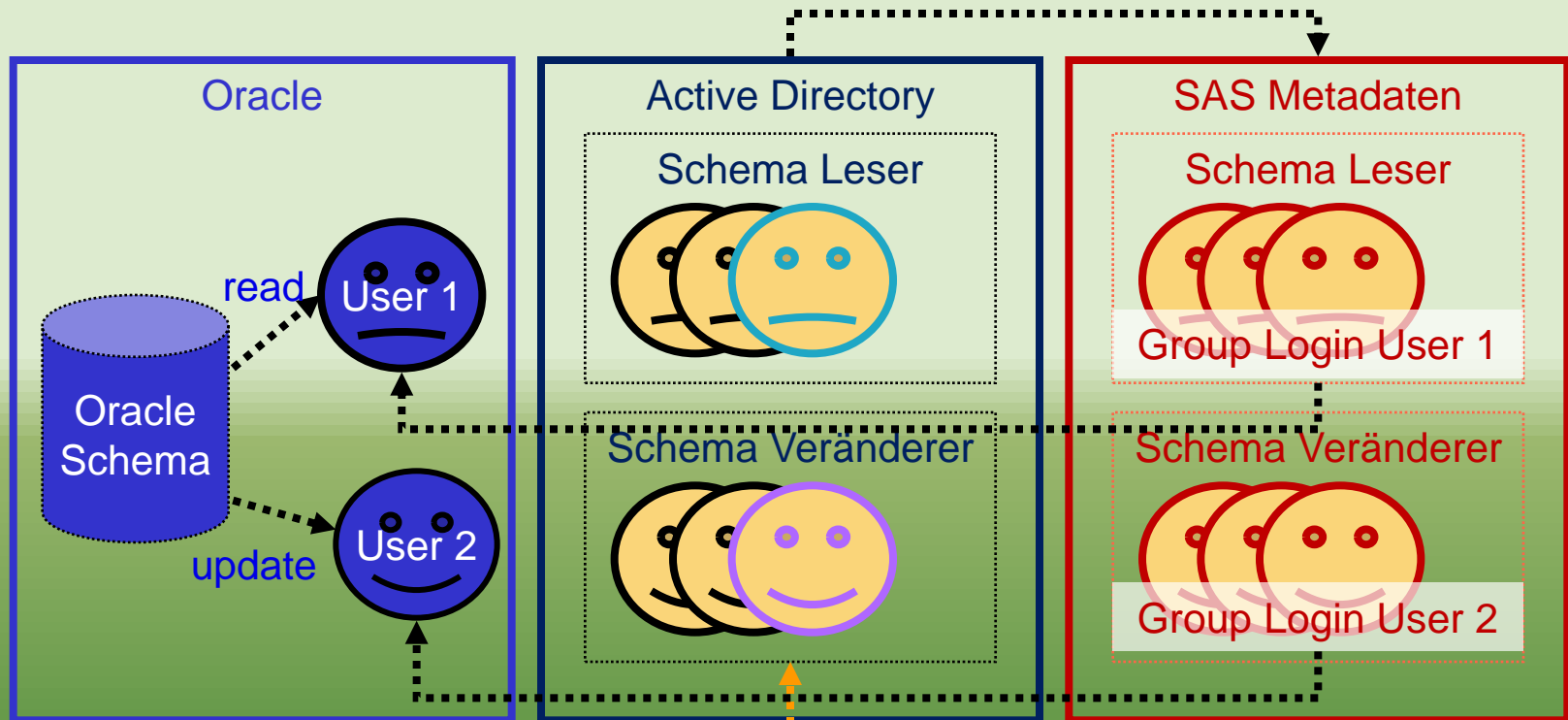
	Versionsnummer	Datum des Stichtages der Version	Limit-ID	Mandant - Schlüssel 282	Hotspotklasse	LMS-Risikoart - Schlüssel 3500	Limitart - Schlüssel 3069	Portfoliokontext - Schlüssel 2314 - LIM_ART_SL=3 -> 2 (IPEX), ...	Limitierte Größe - Schlüssel 3118
1	6	03.06.2008	356	KfW	1	keine	AL-LMS	TK01	Z0
2	6	03.06.2008	365	KfW	1	keine	AL-LMS	TK01	Z0
3	6	03.06.2008	369	KfW	1	keine	AL-LMS	TK01	Z0
4	6	03.06.2008	5921	KfW	1	keine	AL-LMS	TK01	Z0
5	6	03.06.2008	5922	KfW	1	keine	AL-LMS	TK01	Z0
6	6	03.06.2008	5923	KfW	1	keine	AL-LMS	TK01	Z0
7	6	03.06.2008	5924	KfW	1	keine	AL-LMS	TK01	Z0
8	6	03.06.2008	5925	KfW	1	keine	AL-LMS	TK01	Z0



# 1. Metadaten durch Eigenentwicklungen

## Anwendung 2: Zugriffsteuerung

### Beispiele



Antragsworkflow:

Anwender 1 beantragt Mitgliedschaft in Gruppe Schema Leser

Anwender 2 beantragt Mitgliedschaft in Gruppe Schema Veränderer



## 2. Metadaten durch SAS Systemkomponenten

- Import von Benutzer und Benutzergruppen  
z. B. Active Directory Import
- Import von Datenstrukturen mit proc metalib

### SASPlatformObjectFramework (9.2):

- Setzen von Passwörtern (SetPassword.exe)
- Erzeugung von Ordnerstrukturen (MakeFolder.exe)
- Import und Export von Objekten (ImportPackage.exe, ExportPackage.exe)
- Bereinigung von verwaisten Objekten (AnalyseMetadata.exe)



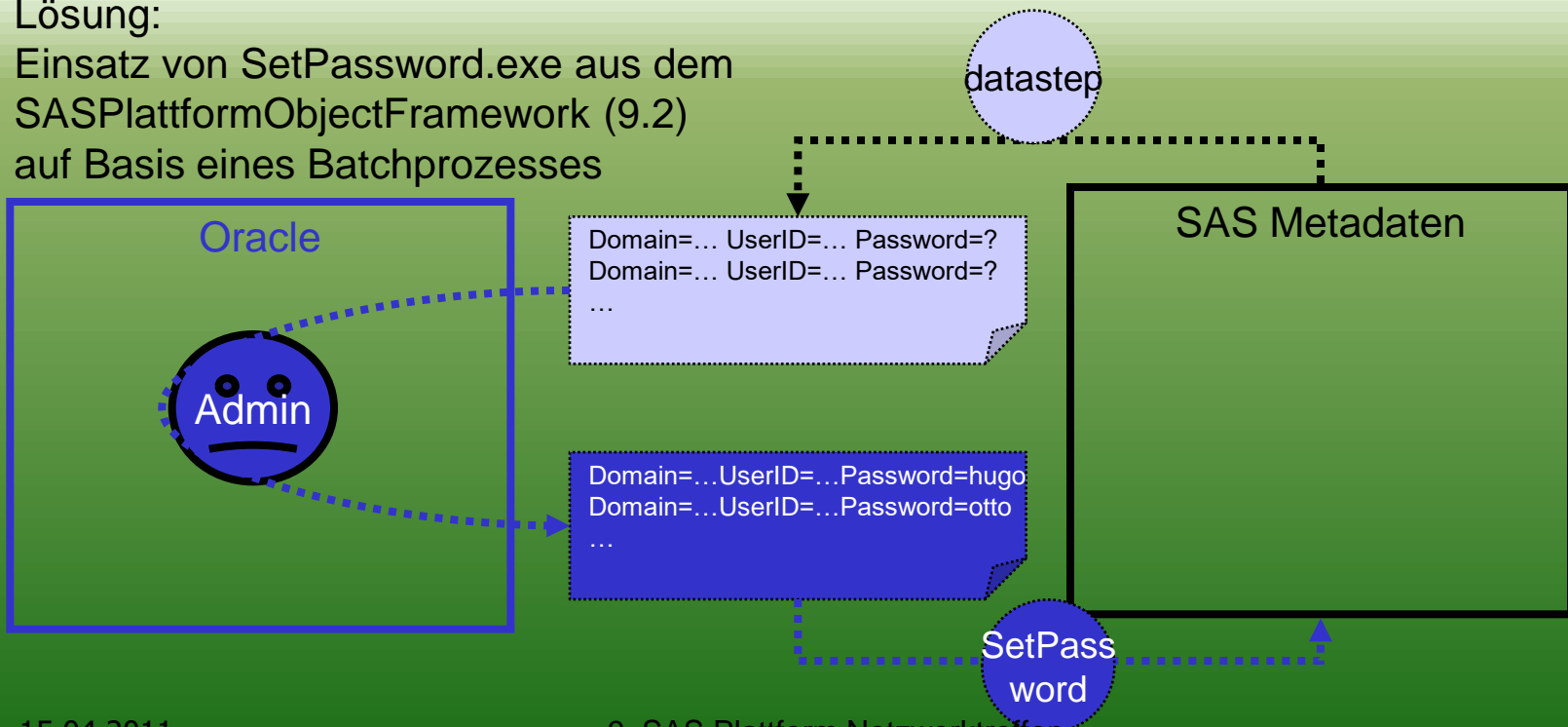
## 2. Metadaten durch SAS Systemkomponenten

### Anwendung 3: Setzen von Passwörtern

- Anforderung 1: Zugangsdaten für Verfahrenskonten sollen nur dem Systemverantwortlichen bekannt sein.  
Beispiel: Nur Oracle Administratoren sollen Passwörter der Oracle Verfahrenskonten kennen
- Anforderung 2: Passwörter sollen regelmäßig geändert werden

Lösung:

Einsatz von SetPassword.exe aus dem  
SASPlattformObjectFramework (9.2)  
auf Basis eines Batchprozesses





## 2. Metadaten durch SAS Systemkomponenten

### Anwendung 3: Setzen von Passwörtern

SetPassword.exe Log:

```
Tue Dec 14 17:28:00 CET 2010: Verbindungsaufbau zum
Metadatenserver. Bitte warten ...
Tue Dec 14 17:28:02 CET 2010: Die Verbindung zu Metadatenserver
"secsasmeta03", Port "8561" wurde aufgebaut.
Tue Dec 14 17:28:02 CET 2010: Passwortdatei "P:\setENTW_DM.txt"
wird geöffnet ...
Tue Dec 14 17:28:02 CET 2010: Anmeldekonto werden vom
Metadatenserver abgerufen ...
Tue Dec 14 17:28:03 CET 2010: Es wurden insgesamt 122
Anmeldekonto vom Metadatenserver abgerufen.
Tue Dec 14 17:28:03 CET 2010: Anmeldekonto werden
aktualisiert...
Tue Dec 14 17:28:03 CET 2010: Folgende 122 Anmeldekonto wurden
aktualisiert:
```



## 2. Metadaten durch SAS Systemkomponenten

### Anwendung 4: Versionskontrolle von Metadatenobjekten

- Anforderung: Versionierung und Auslieferung von Metadaten

Beispiel: Stored Processes, welche im Enterprise Guide erzeugt werden, sollen mit einem vorhandenen Werkzeug der Versionskontrolle versioniert und ausgeliefert werden.

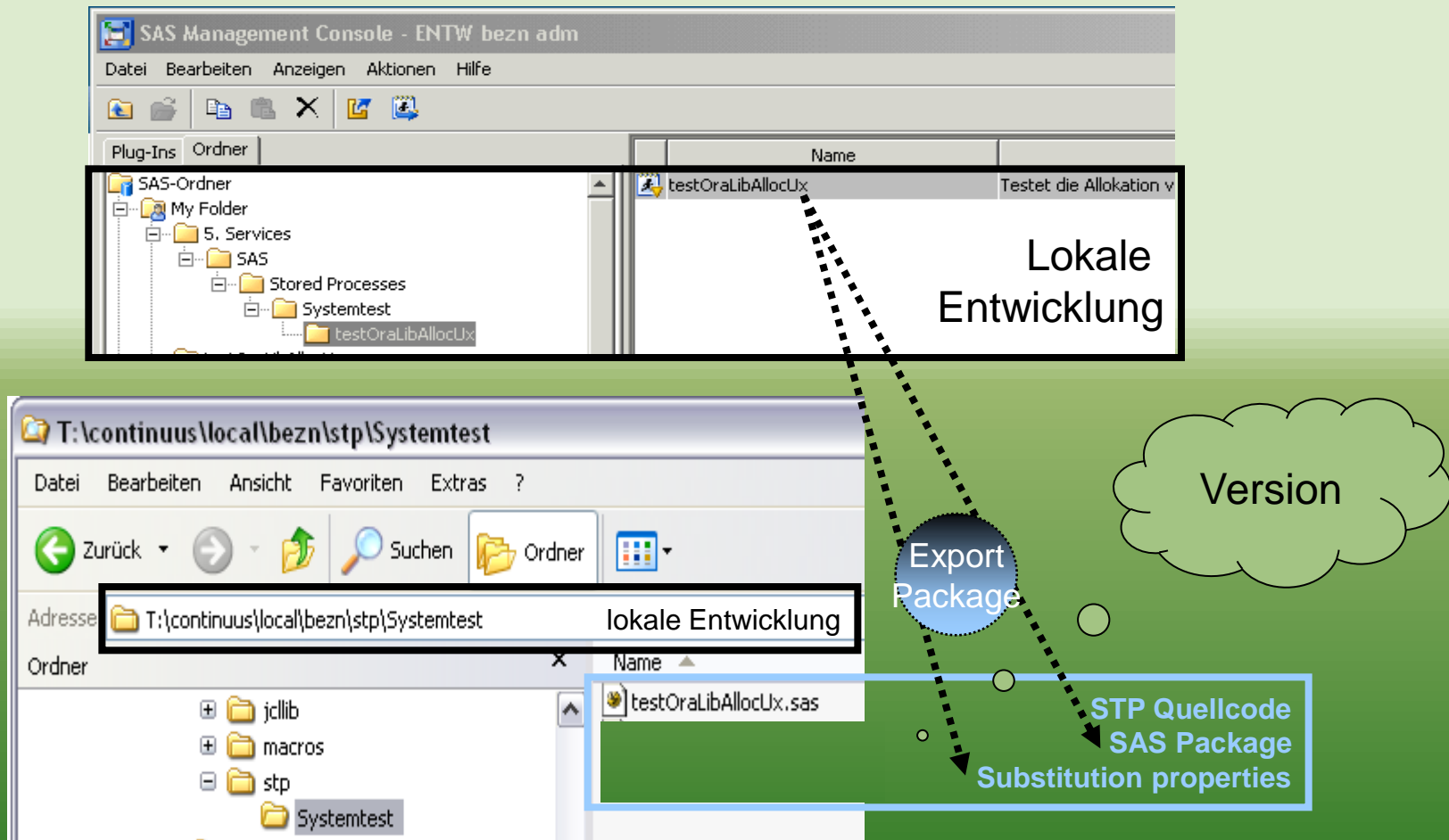
Betroffen davon sind die Vorgänge

1. CheckIn (lokale Entwicklungsumgebung → Version)  
(ExportPackage.exe)
2. Version → Entwicklungsumgebung (ImportPackage.exe)
3. Version → Testumgebung (ImportPackage.exe)
4. Version → Produktionsumgebung (ImportPackage.exe)
5. CheckOut (Version → lokale Entwicklungsumgebung)  
(ImportPackage.exe)



## 2. Metadaten durch SAS Systemkomponenten

### Anwendung 4: Versionskontrolle von Metadatenobjekten: CheckIn





## 2. Metadaten durch SAS Systemkomponenten

### Anwendung 4: Versionskontrolle von Metadatenobjekten:

#### Erzeugung des ExportPackage Aufrufs

```
E:\programme\SAS92\SASPlatformObjectFramework\9.2\ExportPackage.exe  
-objects "/users/Bezner, Gerhard/My Folder/Services/SAS/Stored  
Processes/Systemtest/testOraLibAllocUx(StoredProcess)"  
-package T:\sas\bezn\stp\Systemtest\testOraLibAllocUx.spk  
-subprop  
-profile P:\MetadataUtility\SASObjFramework\metaConfig\sasmetaENTW.swa  
-log E:\CSSAS\ENTW\SASTOOLS\expImpTool\logs\test.log
```

Vor Aufruf muss eine Kommando Datei erzeugt werden, welche die Parameter von ExportPackage.exe mit der korrekten Metadaten-Ordnerstruktur erzeugt. Dies geschieht mit Hilfe eines Datasteps, welcher aus den Parametern des Exportvorgangs die entsprechenden Metadaten bereitstellt.



# Zusammenfassung: Metadaten API Java und .Net

- + Neutrale Schnittstelle zur Aktualisierung von Metadaten durch Kapselung der internen Metadatenstrukturen
  - + Sprechende Objekt- und Methodenbezeichnungen
  - + Konsistente Updates durch setzen von Commitpoints
  - + Erweiterte Möglichkeiten der Nutzung des Authorisierungsmodells durch <DOAS> Option: Durchführung von Anforderungen im Namen eines anderen Users (9.2)
- 
- Zusätzliche Programmiersprache
  - weitere Schnittstellen zu SAS (JDBC, ADO, IOM)
  - Vergleichsweise langsam (sehr langsam?)
  - Funktionen sind nicht innerhalb von SAS einsetzbar (?)



## Zusammenfassung: Datastep API (metadata\_get..., metadata\_resolve(),... etc)

- ✚ Neutrale Schnittstelle zur Aktualisierung von Metadaten durch Kapselung der internen Metadatenstrukturen
  - ✚ Halbwegs sprechende Funktionsbezeichnungen
  - ✚ Man bewegt sich innerhalb von SAS, keine weiteren Schnittstellen und Programmiersprachen
  - ✚ Verwendbar in Makros zur Programmierung von SAS-Funktionen (z. B. Verbindung zu einem Oracle Schema oder zu einer anderen „outbound“ Ressource)
  - ✚ Ideal für Funktionen mit geringen Metadatenmengen
- 
- Vergleichsweise langsam (sehr langsam)
  - Unübersichtliche Programme (viele geschachtelte Iterationen über gefundene Objekte)
  - Konsistente Updates durch setzen von Commitpoints i. A. nicht möglich (nur bei Änderungen von Berechtigungsobjekten in 9.2)



# Zusammenfassung: XML API mit PROC METALIB

- + Sehr Schnell
  - + Übersichtliche Programmierung, wenn Assoziationen relational aufbereitet werden (durch Einsatz von xml-maps)
  - + Commit Points können über einen PROC METALIB step gesetzt werden
  - + Ideal für Batch Jobs mit umfangreichen Metadaten
- 
- Verwendung nativer XML Strukturen → direkte Abhängigkeit vom Metadatenmodell
  - Kenntnis des Metadatenmodells
  - Voraussetzung ist die rudimentäre Kenntnis der XPATH Sprache
  - Einsatz von xml-maps als zusätzliche Komponente zur Erzeugung relationaler Strukturen von Metadaten



# Zusammenfassung: Metadatenserverbetrieb

Vorteile des umfassenden Einsatzes von Metadaten:

- ✚ ermöglicht eine hohe Abstraktion der Objekte von ihren physischen Attributen  
(Servernamen, Verzeichnisse, OS-Dateinamen, Optionen, Konten, Passwörtern, Datenbankschemen und Instanzen)
- ✚ Wegfall dieser Attribute
  - im Quellcode
  - beim Metadatenimport bis auf wenige Ausnahmen
- ✚ Migration von Objekten (Server, Verzeichnisse, ...) erfordern wenig Aufwand

Nachteile:

- Die Sicherung des Metadatenservers (OMABACKUP) setzt diesen in Offline Status → Kein Batchbetrieb während dieser Zeit (ca. 2 -3 Minuten) möglich.
- Data Integrations Studio ersetzt die Metadaten Objekte beim Deployment z.T. durch physische Attribute



# Zusammenfassung: Metadatenadministration als Vollzeitaufgabe

Beschränkte sich in vor-9-Versionen die „Administration“ der SAS Software auf Installation und Bereitstellung von Ressourcen, so ist sie heute zu einer Vollzeitaufgabe mit hoher Schnittstellen Präsenz und vielen integrativen Aufgaben geworden:

- Systemplattform Administratoren
- Datenbank Administratoren
- Web Administratoren
- SAP Adminstratoren
- Provisioning
- Revision