

stonebranch

Automation Best Practice The Story of Stonebranch at AXA

Thomas Palm, AXA Tech



Vorstellung

- Name : Thomas Palm
- Alter : 48
- Kinder : Ja, 2 alles Mädchen 😊
- Hobbies : Jogging, Squash und Motorrad fahren
- Work : AXA seit Juli 2000.



Verantwortlich für...

Scheduling (OPC, TWS, IWS...), Stonebranch-Produkte (UEM, UDM, UCMD, USAP) national und international, Cleo's B2B; File Transfer generell

- Versicherung, Vermögensanlagen und Finanzierungen
- Zentrale in Paris beheimatet
- 5 Rechenzentren verwaltet von AXA Technology
- 166,000 Mitarbeiter weltweit (2016), Einnahmen €100,2 Milliarden (2016)

- AXA Technology Services Germany GmbH
- NESD-Region (Deutschland und Belgien)
- Pro Monat :
 - ~1,5 - 2 M Jobs dezentral (Linux, Windows) + ~ 400k SAP Jobs
 - ~ 3500 – 4000 servers, 1 Mainframe (steht in Paris), 8 Umgebungen



- 01 Grundsätzliches setup / standards
- 02 Universal Command (UCMD) – Script control für Linux / Windows
Universal Data Mover (UDM) – File Transfer
- 03 Universal Event Monitor (UEM) – event driven & demand driven
Universal Connector for SAP (USAP) – ABAP control
- 04 Access Control List (ACL)
- 05 Vorteile durch dieses Setup mit Hilfe von Stonebranch Produkten
- 06 Typische Benutzung im täglichen Geschäft
- 07 Zusammenarbeit mit Stonebranch



Allgemeines

- Welche Szenarien muss ich betrachten ?
- Wo möchte ich skripte speichern ?
- Security ? (Verschlüsselung, logon, user....)
- Wie viele Jobs erwarte ich ?
- Verzeichnis-Struktur auf dezentralen Maschinen
- Job log's



Wie kann ich den unterschiedlichen Szenarien begegnen ?

- Menge: WLM und spezielle job class
- Workstations : Eine workstation pro logische Einheit
- Special Resources : Eine Special Resource pro Maschine
- Troubleshooting :
Jobname = Skript name/ABAP
1 logischer Step pro job

Workstation und Special Resources

■ Workstation:

CDM1	CPU File transfer via UDM
CM01	CPU Control Windows via UCMD
CPU1	CPU Control LINUX via UCMD
CS00	CPU Control SAP via USAP
CUEM	CPU Event control via UEM
CMAI	Maintenance dezentral

■ Special Resource:

UDM :	Jede teilnehmende Maschine
UCMD :	-i Parameter → Special Ressource name
UEM :	-i Parameter → Special Ressource name
USAP :	Instanzname und Mandant



Prozeduren von Stonebranch

- Wir mussten die Prozeduren von Stonebranch anpassen damit sie zu unseren Standards passen
- Nachfolgend ein Beispiel aus der UDM-Prozedur B0CVUDM:
- `//UDMGEUX0 DD DISP=SHR,DSN=&UNVPRE1..PWD.PROD(UDMGEUX0)`

Ein typischer Linux job

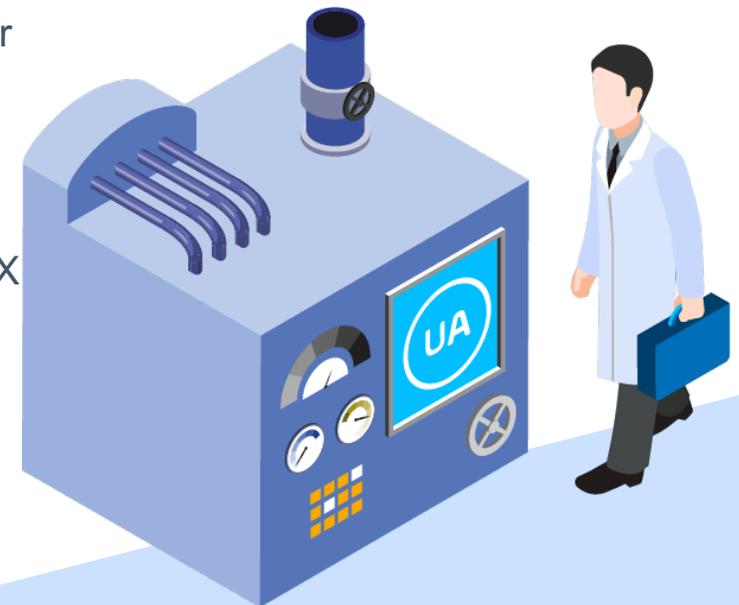
```
//$AIZAS10 JOB '860000-PROD','UCMD-HOLLIP03',  
// MSGCLASS=4,CLASS=B,REGION=20M,  
....
```

```
//ST010 EXEC B0CVUCMD  
//SCRIPT DD *  
/customer/sbin/pexec PROD Produser AIZAS10  
//SYSIN DD *  
-i HOLLIP03  
-x FILELX  
-s SCRIPT  
-cmdid %cmdid  
-restart AUTO
```

Jobname = Script name (ohne das erste Zeichen)

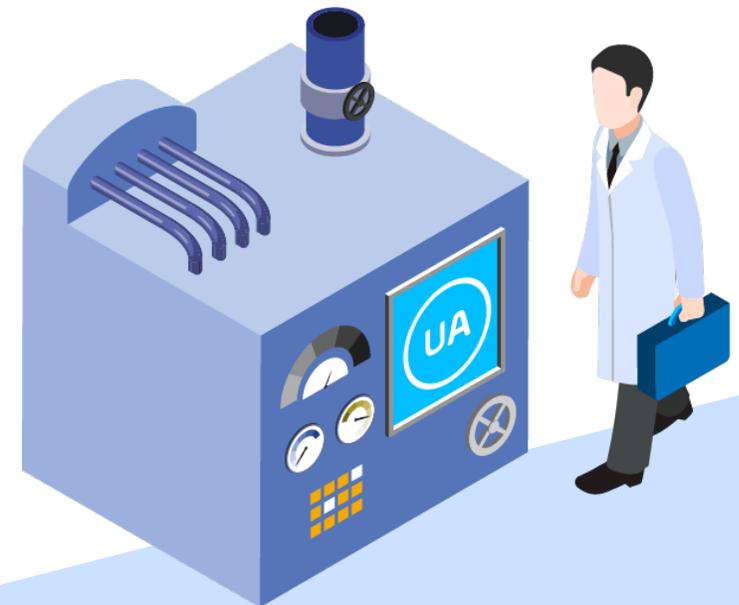
hostname (-i Parameter) → Special resource
verschlüsselter technischer User

%OADID.%OJOBNAME.%OOPNO.%CYYMMDD.%CHHMMSSX



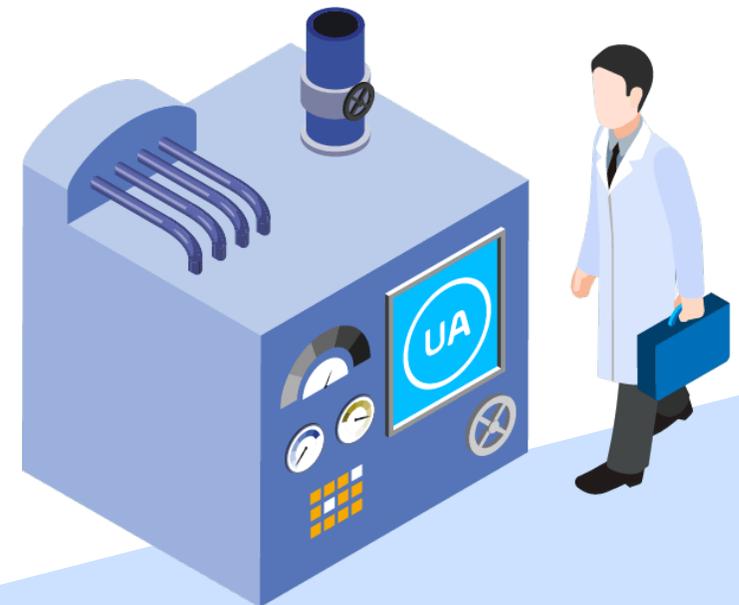
pexec für Linux

- /admin/sbin/pexec oder /customer/sbin/pexec
- Selbst geschrieben
- Erlaubt den Zugriff und die Ausführung von Skripten mit sicheren technischen Usern / Umgebung
- Ändert jeden Returncode zwischen 1-7 zu 8
- Nur für Linux verfügbar



Ein typischer Windows job

```
//$AI99A05 JOB '860000-PROD','UCMD-HOLMFS04',  
// MSGCLASS=4,CLASS=B,REGION=20M,  
// COND=(4,LT)  
....  
//ST010 EXEC BOCVUCMD  
//SCRIPT DD *  
c:\Batch\PROD\AI99A05.cmd  
//SYSIN DD *  
-i holmfs04  
-x FILEMSDE  
-s SCRIPT  
-cmdid %cmdid  
-restart AUTO
```



Standards und Prozeduren

- Prozeduren wurden erstellt um die syntax im job zu minimieren
- 6 Prozeduren...

- UDMS2MFT / UDMS2MFB
- UDMMF2ST / UDMMF2SB
- UMDMF3PT / UDMMF3PB



Aufruf Prozedur UDMMF3PB als Beispiel

```
call UDMMF3PB +
sserver=l13app01.applications.services.axa-tech.intraxa +
udmusrs=UDMGEX0 +
tserver=ftst21.applications.services.axa-tech.intraxa +
udmusrt=UDMGEX0 +
sfile=/transfer/prod/trans/*.req +
tfile=/transfer/ge/prod/out/AVBBC01/ +
▪ call UDMMF3PB copy=move +
```

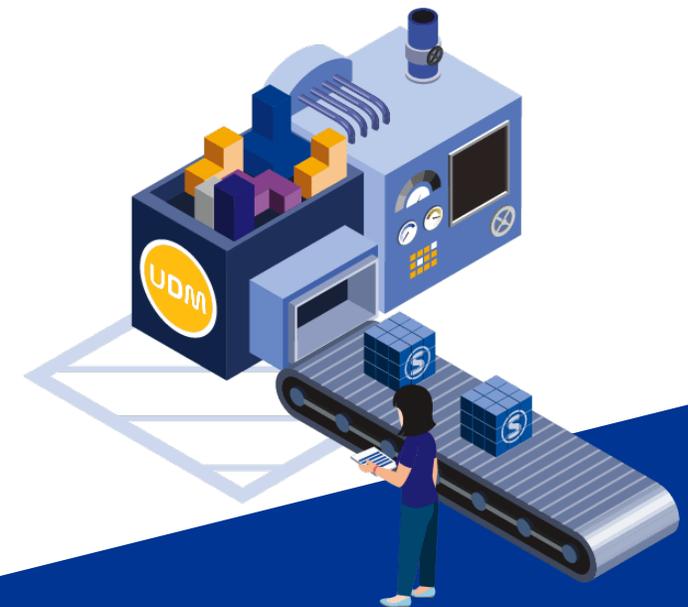


Prozedur UDMMF3PB als Beispiel

```
set copy=copy
if $(debug.exists) EQ yes
  set echo=yes
end

report progress=yes
print msg="### Sourceserver=$(sserver)           "
print msg="### Targetserver=$(tserver)          "
print msg="### copy=$(copy)                     "
```

....



Prozedure UDMMF3PB als Beispiel

```
if $(encrpt.exists) EQ no  
  set encrpt=AES256-SHA  
if $(comp.exists) EQ no  
  set comp=hasp  
end
```

```
open source=$(sserver) xfile=$(udmusrs) +  
  target=$(tserver) xfile=$(udmusrt) +  
  encrypt=$(encrpt) compress=$(comp) +  
  nft=$(nft)
```

....



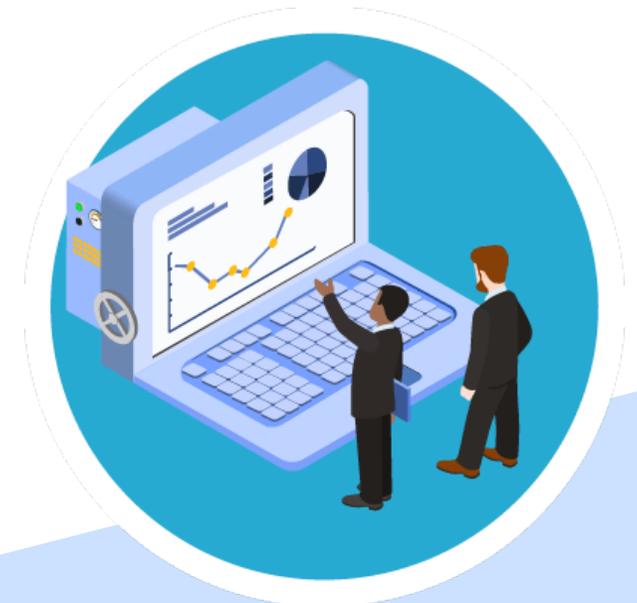
Event oder demand driven ?

Wann sollte UEM in **event driven** mode genutzt werden ?

- 24 Stunden Überwachung
- Keine Ahnung ob oder wann eine Datei erscheint
- Keine Ahnung wie viele Dateien

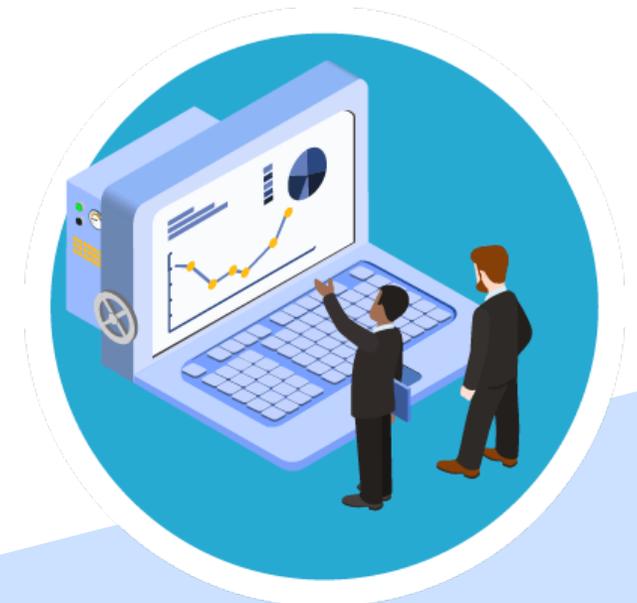
Wann sollte UEM in **demand driven** mode genutzt werden ?

- Eine definierte End-Zeit oder eine bestimmte Überwachungsdauer wird benötigt
- Für den Fall das eine Datei nicht in der vorgegebenen Zeit erscheint, soll eine Reaktion erfolgen



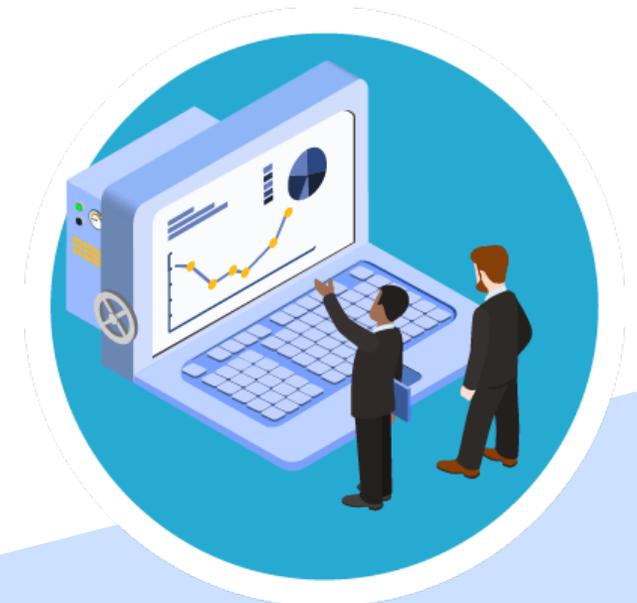
Setup für UEM, event definition

```
begin_event
event_id FS01A00
event_type FILE
comp_name uems
state enable
handler_opts "FS01A00"
triggered_id start_ETT
filespec "/transfer/ge/prod/in/ship2/SHP_forex_*"
rename_file yes
rename_filespec "/transfer/ge/prod/in/ship2/tmp/${origname}.${origext}"
end_event
```



Setup für UEM, handler definition

```
begin_handler
  handler_id start_ETT
  handler_type SCRIPT
  begin_script
    stmt "#!/bin/bash"
    stmt "exec 1>/tmp/udm_$(1).log"
    stmt "exec 2>>/tmp/udm_$(1).log"
    stmt "mv $UEMRENAMEDFILE $UEMRENAMEDPATH/$UEMORIGFNAME"
    stmt "/opt/universal/bin/udm -s /opt/universal/scripts/S2MF -
-o ett=$(1)"
    stmt "exit"
  end_script
  userid AVUSERP
end_handler
```



Setup für UEM, handler definition

```
open target=cpub xfile=/etc/universal/pwdfiles/udmgeux0
```

```
...
```

```
filesystem target=dsn
```

```
attrib target createop=replace lrecl=80 dataclas=DVFB dsorg=PS unit=PROD rlse=RLSE
```

```
copy local=/opt/universal/scripts/ett_file.txt target='VPAX.JOB.$(ett).ETT'
```

```
if $_rc NE 0
```

```
    set _rc=8
```

```
    exit
```

```
end
```

```
close
```



Setup für UEM, TWS

- Using ETT (Event Trigger Tracking)
- VPAX.JOB.FS01A00.ETT FS01A#PCORE1#FRX

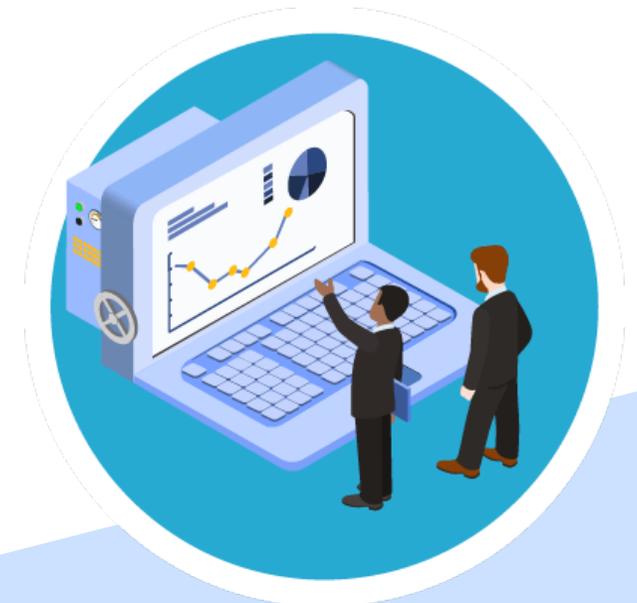
Vorteile aus der Kombination von UEM und TWS :

- Single point of control in TWS, nur der Anstoß ist nicht in realtime zu sehen
- Job logs werden z.B.: in BETA92 zentral auf dem Mainframe gespeichert (Revision, ein Ort für alle logs)
- Dasselbe Verhalten für Linux und Windows



Setup für UEM, TWS

```
//ST010 EXEC BOCVUEM
//SYSIN DD *
-i ftst21.applications.services.axa-tech.intraxa
-x UDMGEUX0
-event_type file
-filespec /transfer/ge/prod/in/copernic/0004/acs/COP_FR_004_*. *
-rename_file no
-inact_date_time +15
-l audit -max_count 1
//ST030 EXEC PGM=COLBR14,PARM='8',
// COND=(0,EQ,ST010.UEM)
```



Wie steuern wir ABAPs ?

- Auch hier,

ABAP = Jobname

- // SAPDEST=EEP100, → Dieser Eintrag findet sich in der rfc.ini wieder. Dort ist ein SAP Router String hinterlegt
- // SAPPWD=EEP → Verschlüsselte user-credentials

Ein Sap-Job besteht aus 2 Schritten :

- 1.) SAP-Job id auslesen
- 2.) Start des aktuellen und gültigen ABAP als Kopie der Definition im SAP



Standardisierte ACL für alle Server

ucmd_access ALL,Production,Production,allow,auth

ucmd_access ALL,* ,Production,deny,auth

ucmd_access ALL,XXXI,XXXI,allow,auth

ucmd_access ALL,* ,XXXI,deny,auth

ucmd_access ALL,XXXZ,XXXZ,allow,auth

ucmd_access ALL,* ,XXXZ,deny,auth



Da ist ein (e)

- ...Patch Tag für Windows ? → Schließe die Workstation CM01
- ...Server mit Problemen ? → Schließe die Special Resource (hostname)
- ...zu hoher workload auf der Maschine ? → limitiere die Anzahl paralleler Jobs für diese Special Ressource
- ...IPL auf einer CPU ? → Nimm die CPU aus der Workload raus
- ...SAP-Instanz mit Problemen ? → Schließe die Special Resource (SAP-Instance name)
- ...Disaster Recovery Test ? → Schließe alle Workstations außer CMAI

Du erreichst....

- standard jobs
- Standardisierte Prozeduren selbst bei Umgebungswechsel
- standard ACL
- restart option zu jedem Zeitpunkt



Kopieren von Jobs aus Produktion zu Integration

Welche Schritte muss ich durchführen ?

UCMD :

- Ändere Prozedurnamen von BOCVUCMD zu BICVUCMD
- Ändere “PROD Prod-user scriptname” zu “INTE Inte-user scriptname”

UDM

- Ändere Prozedurname von BOCVUDM zu BICVUDM
- Ändere Verzeichnisstruktur von prod nach inte

UEM

- Ändere Verzeichnisstruktur von prod nach inte
- Ändere ggf. hostnamen



Christian Bruns bei AXA

1. Reconnectfähigkeit bei SAP-Jobs
2. cancel und canceldump bei SAP-jobs
3. cancel und canceldump bei Skriptenausführungen (Windows/Linux)
4. Übersichtlicheres joblog bei UDM-Jobs
5. Update der Agenten mit Hilfe von Skripten in einem halb-automatischen Ablauf



Christian Bruns bei AXA, Übersichtlicheres joblog bei UDM-Jobs

Alte Joblogs

```
Processing script: UDMMF3PB
#### Default Werte für genutzte Variablen
#### halton=warn
#### createop=replace
#### trim=no
#### encrpt=AES256-SHA
#### comp=hasp
#### nft=yes
#### copy=copy
#### UDM Default Werte
UNV3992I control: Host=10.174.155.205, Port=6306
Data session established using cipher: AES256-SH
Third party session established with source (com
Transfer mode settings:
  type=binary
  trim=no
Session options:
  Keep Alive Interval: 120
  Network Fault Tolerant: yes
#### Werte für den aktuellen Transfer, im Job gesetzt
#### Sourceserver=lhuinfaa.applications.services.axa-tech.intraxa
#### Targetserver=ftst21.applications.services.axa-tech.intraxa
#### copy=move
Transfer mode settings:
  type=binary
  trim=no
```



Christian Bruns bei AXA, Übersichtlicheres joblog bei UDM-Jobs

Neue Joblogs

Processing script: dd:UNVSCR

Processing script: UDMMF3PB

#####

###

genutzte Variablen in diesem Lauf

###

Lauf vom: 26.09.2017 03:57:49
Jobname/Nummer: §FCA5A31,JOB33112
Sourceserver: lhuinfaa.applications.services.axa-tech.intraxa
Targetserver: ftst21.applications.services.axa-tech.intraxa
Source file: /u050/appdata/infa_shared/copernic/MAGN/in/COP_AC_010_*
Target file: /transfer/ge/prod/out/copernic/0010/
File Name2 kein file2 in diesem Lauf
File Name3 kein file3 in diesem Lauf
File Name4 kein file4 in diesem Lauf
createop: replace
trim: no
halton: 0
encrpt: AES256-SHA
compression: hasp
nft: yes
copy oder move: move
###

#####



Wage Dich zu fragen ... 😊



- Falls Kritik geübt werden soll, dann bitte bei mir persönlich. Am Besten an der Bar mit einem Bier 😊
 - Falls Lob geübt werden soll, dann bitte jetzt und laut
 - Für alle anderen Fälle... Klatschen...

Vielen Dank für Ihre Aufmerksamkeit und Ihre Geduld.

All dies ist nur möglich in einem guten Team.

Mein Kollege Peter Mulorz ist ein wichtiger Teil dieses Teams !

Danke auch an Dich, Peter





Danke schön !

—
Kontakt Information



Thomas Palm

Spezialist

AXA Technology Services Germany

T: +49 (0)221 148 248 19

thomas.palm@axa-tech.com

