WAGen5
and
learning to walk with a riZing frog

Lukas Essig, Michael Witzgall, 29.05.2019

FIDUCIA GAD
ZUKUNFTSERFAHREN

Agenda

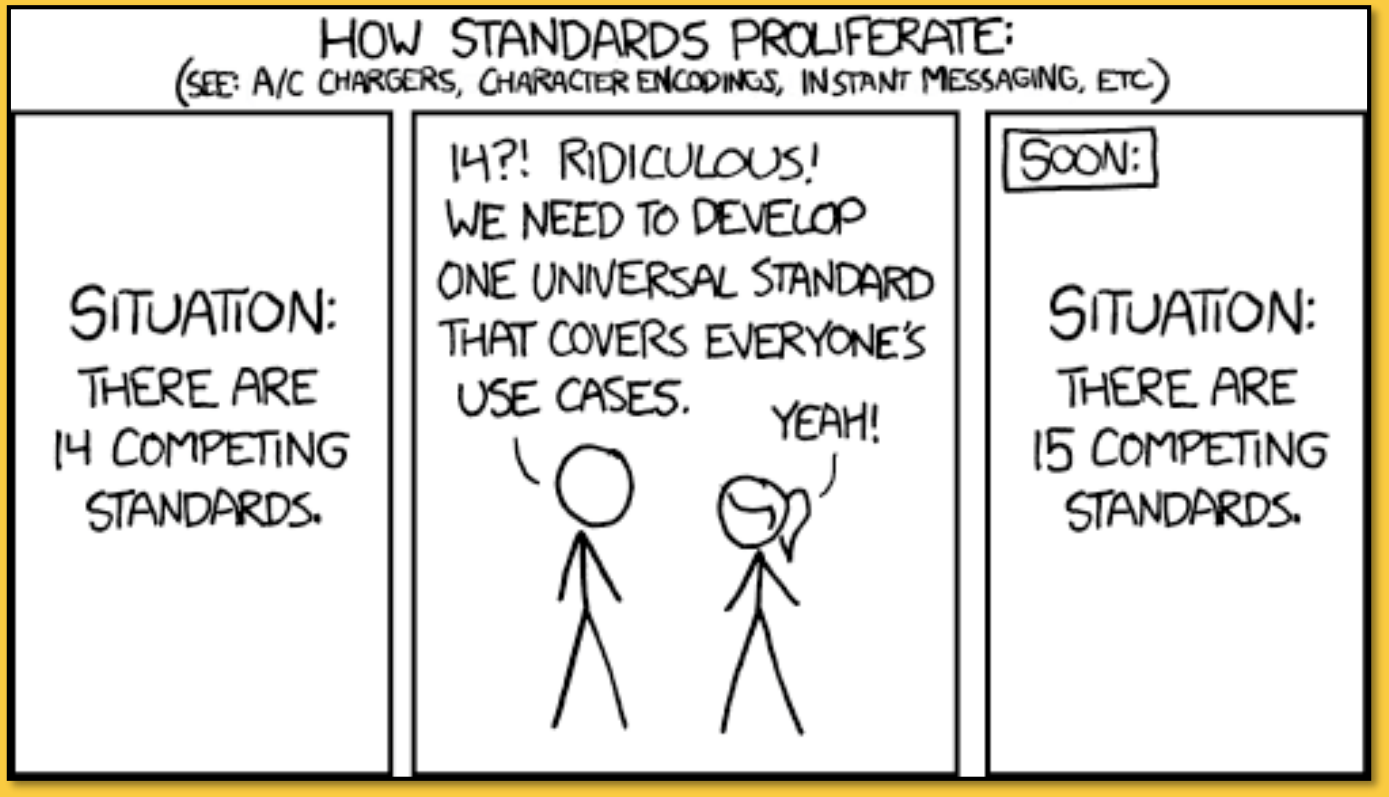
1 **WAGen5**

2 learning to walk with a riZing frog

© Fiducia & GAD IT AG

Bevor wir anfangen…

https://xkcd.com/927/



Command ===> CH ALL STANDARDS INTERFACES

© Fiducia & GAD IT AG

WAGen5

© Fiducia & GAD IT AG

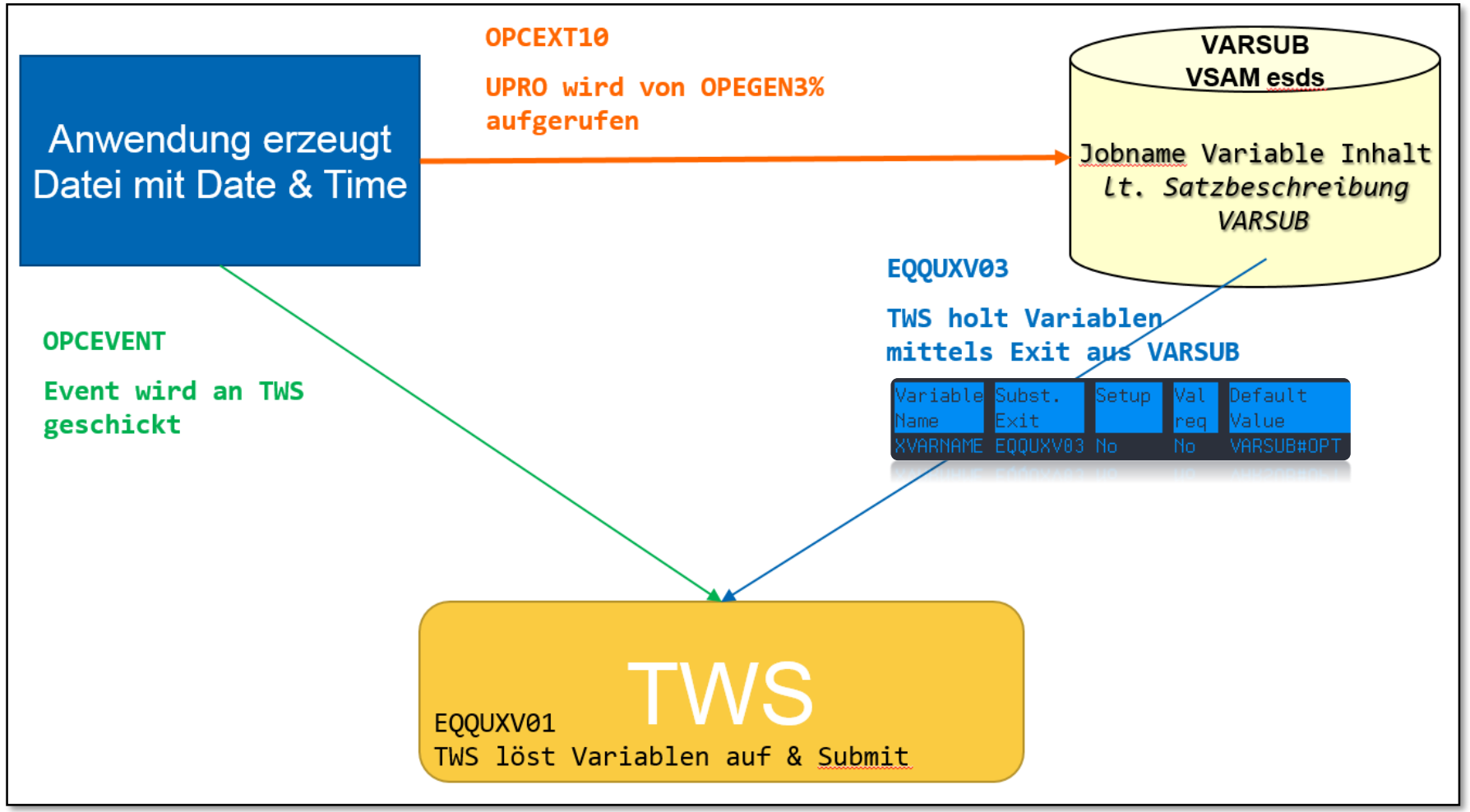
WAGen5

Workload Automation Generation 5

OPEGEN50 – Variablenübergabe und Add Application(s)

© Fiducia & GAD IT AG

the ‚solid core' way

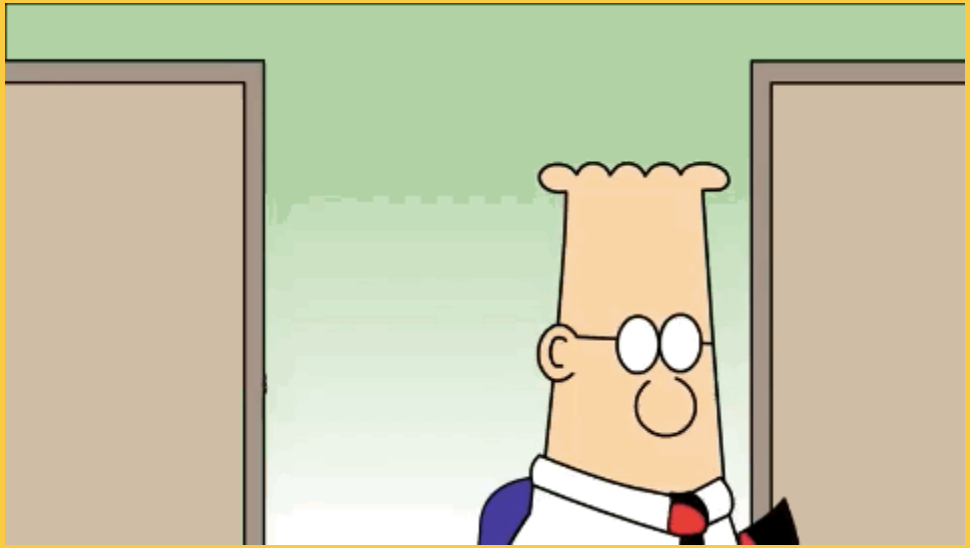


© Fiducia & GAD IT AG

the ‚problem‘

Massengeschäft bei VARSUB
→ „Abholer“ überholen sich

Zwischenlösung mit Prompt-Variablen nicht restart-fähig

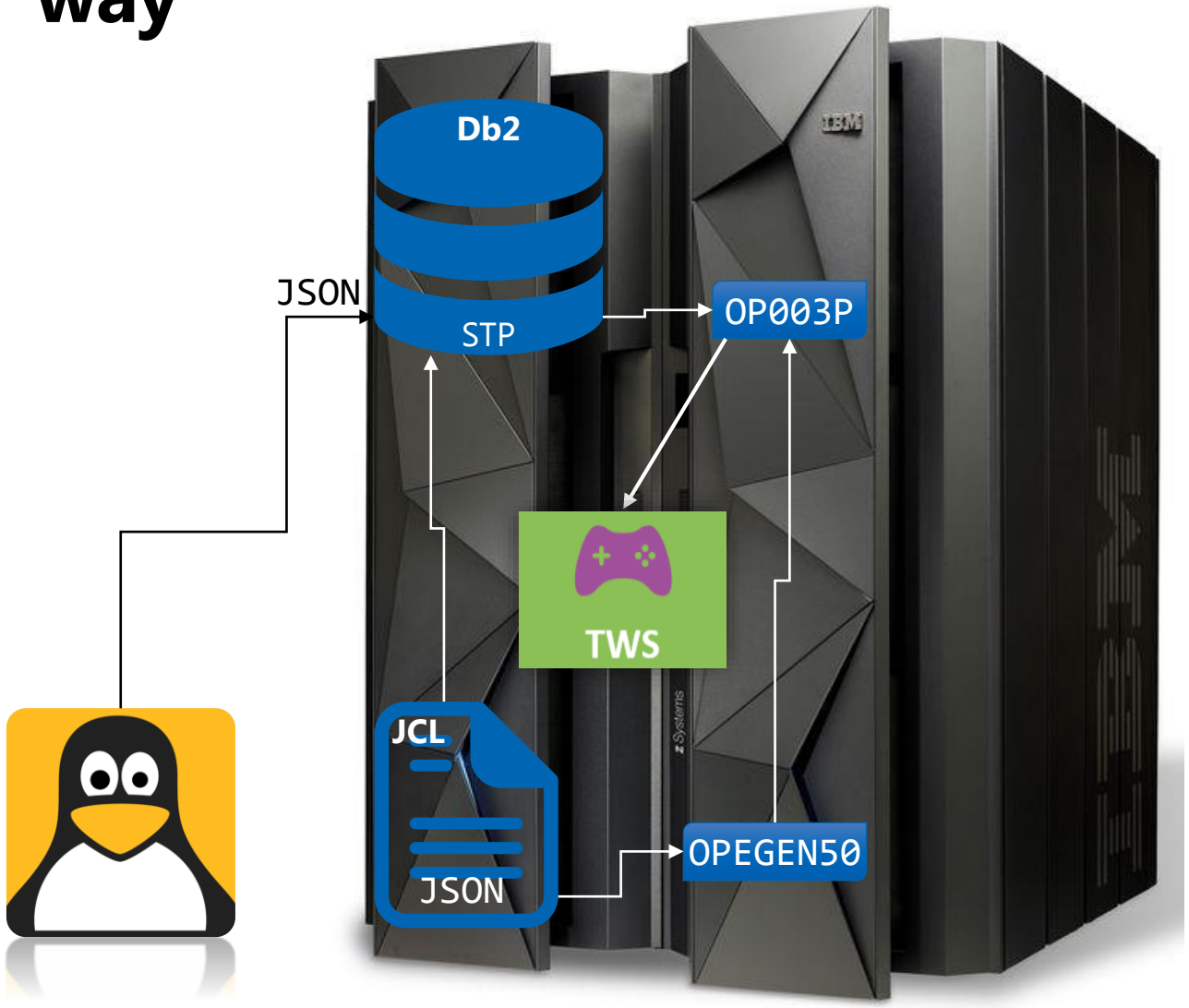


Aufrufer <u>müssen</u> auf dem Host sein

ETT liefert keinen „richtigen“ RC

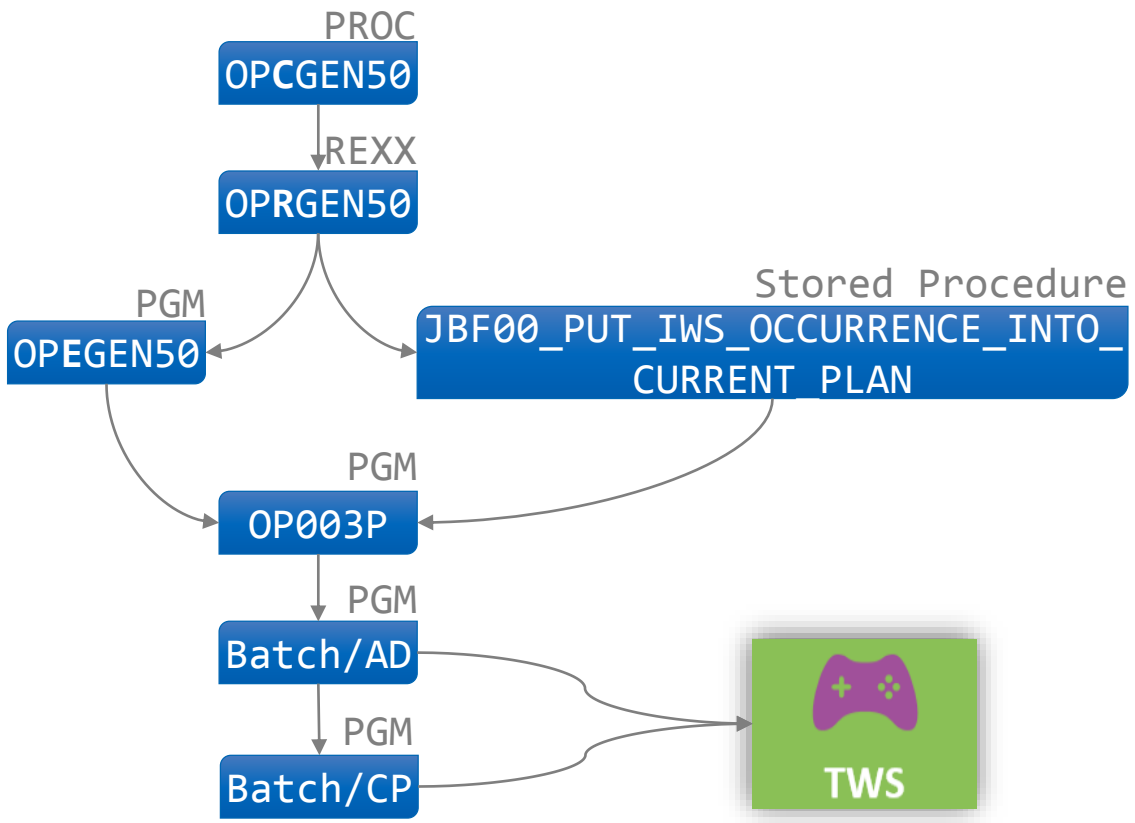
© Fiducia & GAD IT AG

the ‚cloud' way



© Fiducia & GAD IT AG

the ‚magic‘



© Fiducia & GAD IT AG

the ‚proc‘

- Prozedur OPCGEN50
  - ermöglicht beide Aufrufarten
  - kapselt REXX OPRGEN50 und Steuerkarten für Batch/AD & CP

- REXX OPRGEN50
  - überprüft JSON (HWTJSON – Web Enablement Toolkik)
    - Syntax
    - Semantik (korrektes TWS-System)

```
//OPCGEN50 EXEC OPCGEN50,DB2SYS='DB2U',SUBSYS='OPK',MODE='DB2'
//*OPCGEN50 EXEC OPCGEN50,SUBSYS=OPK
//OPRGEN50.JSONIN DD  *,SYMBOLS=JCLONLY
{
"SUBSYSTEM"    : "OPK",
"TABLEOWNER"   : "TWSADM",
"APPLICATIONS" :
 { "UK00TEST#LUK" : { "VarnameA" : "GERMAN:",
                      "VarnameB" : "Mitochondrien sind Kraftwerke der Zelle"
                    }
 , "UK00TEST#LUK" : { "VarnameA" : "ENGLISH:",
                      "VarnameB" : "Mitochondria are power plants of the cell"
                    }
  }
}
```


© Fiducia & GAD IT AG

the ‚rexx‘

```
/*-------------------------------------------------------------------*/
/* Call the STP or module direct in the given Subsys and Schema      */
/* RC of STP is in prmOut                                            */
/*-------------------------------------------------------------------*/
execute:
 if MODE = 'PGM' then do
   call log 'Calling OPEGEN50 native', inf
   address tso
   'CALL *(OPEGEN50)'
   if rc <> 0 then do
     call log 'Error in calling module OPEGEN50',err
     call myexit rc
   end
  end
  /* call db2 */
 else if MODE = 'DB2' then do
  call log 'Calling OPEGEN50 via DB2-STP', inf
  "SUBCOM DSNREXX"                          /* HOST CMD ENV AVAILABLE? */
  IF RC THEN                                /* NO--MAKE ONE            */
     S_RC = RXSUBCOM('ADD','DSNREXX','DSNREXX')
  ADDRESS DSNREXX
  "CONNECT" DB2SYS
   prmIn = json
   prmOut = 0
   call log "Parms for STP  prmIn: " LENGTH(prmIn) " prmOut: " prmOut,inf
   "EXECSQL CALL",
   DB2SCHEMA!!"."!!stpName!!" (:prmIn,:prmOut)"
   _rc = SQLCODE
   If _rc < 0 then CALL SQLCA
   rc_stp = prmOut
   ADDRESS DSNREXX "DISCONNECT"
   _rc = SQLCODE
   IF _rc ^= 0 THEN CALL SQLCA
   S_RC = RXSUBCOM('DELETE','DSNREXX','DSNREXX') /* REMOVE CMD ENV  */
   if rc_stp <> 0 then do
     call log 'Error in calling STP',err
     call myexit rc_stp
   end
  end
  call log 'Call of OPEGEN50 successfully',inf
return
```


© Fiducia & GAD IT AG

the ‚program‘

© Fiducia & GAD IT AG

Demo - I

```
Sitzung A | [62 x 160] | A10CLSSE
Datei  Bearbeiten  Anzeige  Kommunikation  Aktionen  Fenster  Hilfe

 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Iest  Help
MVSE       XCE35D1.J.CNTL(OPGEN50) - 01.32                    Columns 00001 00080
Command ===>                                                   Scroll ===> CSR
****** ******************************** Top of Data ********************************
000001 //OPGEN50   JOB 87409C09,'ESSIG',REGION=200M,
000002 //              NOTIFY=&SYSUID,CLASS=H,
000003 //              MSGLEVEL=(1,1),SCHENV=MVSR
000004 //*
000005 //*OPCGEN50 EXEC OPCGEN50,DB2SYS='DB2U',SUBSYS='OPK',MODE='DB2'
000006 //OPCGEN50 EXEC OPCGEN50,SUBSYS=OPK
000007 //OPRGEN50.JSONIN DD  *,SYMBOLS=JCLONLY
000008 {
000009 "SUBSYSTEM"    : "OPK",
000010 "TABLEOWNER"   : "TWSADM",
000011 "APPLICATIONS" :
000012  { "UK00TEST#LUK" : { "VarnameA" : "GERMAN:",
000013                      "VarnameB" : "Mitochondrien sind Kraftwerke der Zelle"
000014                    }
000015  , "UK00TEST#LUK" : { "VarnameA" : "ENGLISH:",
000016                      "VarnameB" : "Mitochondria are power plants of the cell"
000017                    }
000018   }
000019 }
****** ******************************** Bottom of Data *****************************

 OPK       *OPGEN50 SDSF      -EXEC
```


© Fiducia & GAD IT AG

Demo - II

Java - dev/src/Opcgen50.java - Eclipse SDK - C:\Users\xce35d1\workspace

File Edit Source Refactor Navigate Search Project Run Window Help

Opcgen50.java | input.json | password

dev ▸ src ▸ (default package) ▸ Opcgen50 ▸

```
1⊕import java.nio.file.Files;
5
6
7 public class Opcgen50 {
8     private static Logger log = null;
9
10    /**
11     * @param args
12     */
13    public static void main(String[] args) {
14        log.info("Program started");
15        String urlPrefix = "jdbc:db2:";
16        String url = "jdbc:db2://edb2u.mvsbrz.fiducia.de:4000/DB2U";
17        String user = "xce35d1";
18        String password = "";
19        String stpName = "TTA.JBF00_PUT_IWS_OCCURRENCE_INTO_CURRENT_PLAN";
20        String json = "";
21        int rc;
22
23        Connection con;
24        CallableStatement cstmt;
25
26          try {
27              Class.forName("com.ibm.db2.jcc.DB2Driver");
28              log.info("**** Loaded the JDBC driver");
29              // Create the connection using the IBM Data Server Driver for JDBC and SQLJ
30              password = new String(Files.readAllBytes(Paths.get("data/password")));
31              con = DriverManager.getConnection (url, user, password);
32              cstmt = con.prepareCall("CALL "+stpName+"(?,?)");
33              json = new String(Files.readAllBytes(Paths.get("data/input.json")));
34              cstmt.setString (1, json);
35              // Set input parameter (DB2 command)
36              cstmt.registerOutParameter(2, Types.INTEGER);
37              log.info("executing stp");
38              cstmt.executeUpdate();            // Call the stored procedure
39              rc = cstmt.getInt(2);        // Get the output parameter values
40              log.info("RC was:"+rc);
41              // Register output parameters
```

Problems | Javadoc | Declaration | Console | Call Hierarchy | Search | Progress

No consoles to display at this time.

Writable | Smart Insert | 7 : 24

© Fiducia & GAD IT AG

the benefits

Schnittstelle aufrufbar von zOS & dist.

Aufrufer erhält „echten“ RC

verbesserte Security durch PIF-Nutzung

erweiterbar als REST-Service

© Fiducia & GAD IT AG

Fragen? Fragen!

FIDUCIA GAD
ZUKUNFTSERFAHREN

learning to walk with a riZing frog

FIDUCIA GAD
ZUKUNFTSERFAHREN

Agenda

1 **How does the frog looks like?**

2 How to grow my own frog?

3 How to play with the frog?

© Fiducia & GAD IT AG

How does the frog looks like?

Data Commands (PIF)

LIST, SELECT, DELETE, MODIFY, INSERT, DELETE

Batch Loader

ADSTART, CLSTART, ETTSTART, WSSTART, etc.

Current Plan Commands

HOLD, RELEASE, NOP, UNNOP, ALTER, etc.

Function Commands

ADD, LISTJOB, LISTSTAT, GETDATES, etc.

TSO Commands

OPSTAT, SRSTAT, WSSTAT, OPINFO, etc.

Occurrence Commands

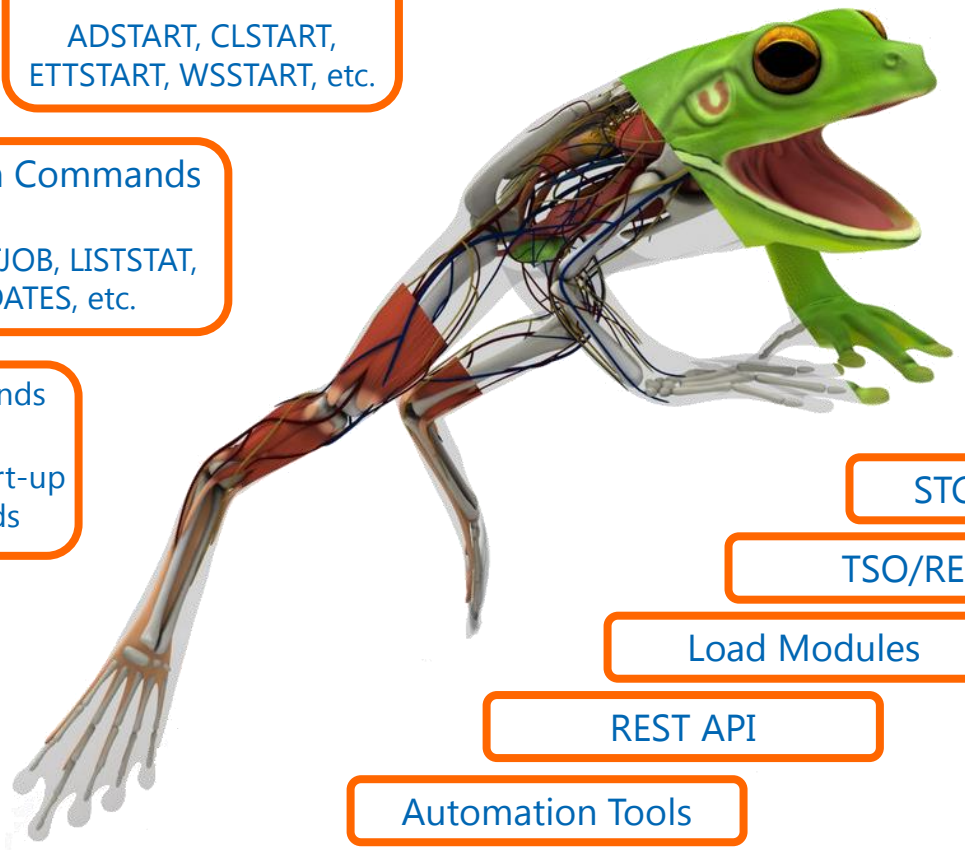
RUNIF & ALTIF as start-up actions in userfields

Communication

SENDMAIL, SENDMSG, CONSOLE, DISPLAY, etc.

Programming Language

Variables, Functions, REXX, etc.



Batch Jobs

STC workstation

TSO/REXX

Load Modules

REST API

Automation Tools

© Fiducia & GAD IT AG

Agenda

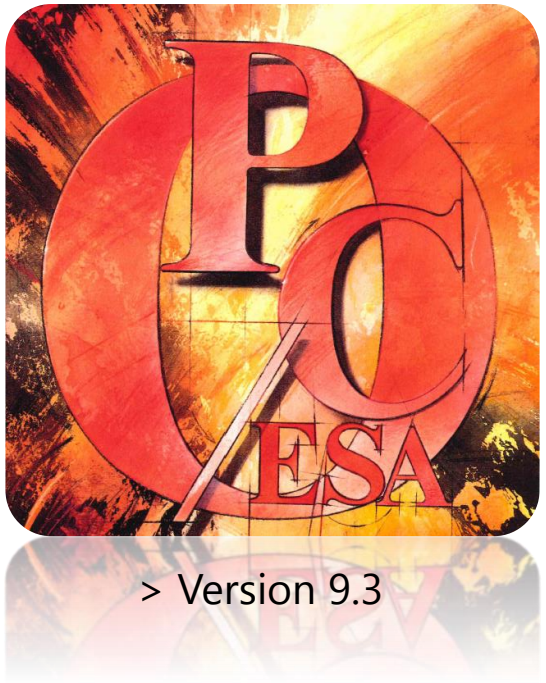
1 How does the frog looks like?

2 How to grow my own frog?

3 How to play with the frog?

© Fiducia & GAD IT AG

How to grow my own frog?



© Fiducia & GAD IT AG

How to grow my own frog?

- Beispiel Prozedur in SEQQSAMP-Library
  - Seit APAR PI79321: EQQYXJPX
    sonst: EQQYXJCL
  - Entfernt DD-Statements für EQQFILE, EQQREF, EQQLANG
  - DD-Statement EQQOPTS wird optional
  - Parameter FILESPEC durch LOADDEF ersetzt
  - Performance Optimierungen
  - Funktioniert wie im User Guide beschrieben
  - Automatische Generierung von Batchloader und ILSON-Data
- Eigenes EQQYPARM-Member notwendig
  -> DATINT(N)
- Läuft in einem TSO

© Fiducia & GAD IT AG

How to grow my own frog?

```
//OPCWAPL   PROC §=,
//          ARGS='',
//          CMD=EQQYXTOP,
//          REG=4M,
//          SUBSYS=,
//          VER=930,
//          MOD='+002',
//          #=
//*-----------------------------------------------------------
//TSO       EXEC PGM=IKJEFT01,
//          REGION=&REG,
//          PARM=&§'&CMD &SUBSYS-&VER&MOD &ARGS'&#
//SYSPROC   DD DISP=SHR,DSN=SCB.TWS0930.A.SEQQMISC
//EQQYPARM  DD DISP=SHR,DSN=SCB.TWS0000.A.PARMLIB(&SUBSYS.YPRM2)
//EQQMLIB   DD DISP=SHR,DSN=SCB.TWS0930.A.SEQQMSG0
//EQQMLOG   DD SYSOUT=*
//EQQDUMP   DD SYSOUT=*
//EQQOPTS   DD *
  OPTIONS
   GTABLE(AV40TABGLOBAL)
   MAILSERVER(FIDUCIAGAD.DE)
   POSTPROC(N)
   VARSUB SCAN SEARCH(NOGLOBAL)
//EQQYLOG1  DD SYSOUT=*
//EQQSMTP   DD SYSOUT=(B,TCPSMTP)
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD DUMMY
//OUTBL     DD SYSOUT=*
//OUTDATA   DD SYSOUT=*,LRECL=1024
//EQQCPOP   DD *
  &OADID. &OYMD1.&OHHMM. &OOPNO
```

MOD-Level abhängig von PTF-Level

DATINT(N)

Default Options

LOG mit MSGLEVEL(1)

Für SENDMAIL-CMD

Batch Loader

ILSON Data

Occurence Token

Via Loaddef können DDs überschrieben werden

© Fiducia & GAD IT AG

How to grow my own frog?

- DD-Statement EQQCPOP:
  - WAPL scannt Current Plan um Operation zum Job zu finden
  - Statement Identifiziert CPOP
- User-Fields mit Prefix EQQ-SYSIN-* werden gelesen und WAPL-Code ausgeführt (in alphabetischer Reihenfolge)
- Statement VARSUB SCAN ermöglicht Zugriff auf Occurence Variablen und Variablen Tabellen
- In Zukunft: WAPL soll den Scan nur bei Bedarf ausführen

© Fiducia & GAD IT AG

Agenda

1 How does the frog looks like?

2 How to grow my own frog?

3 How to play with the frog?

© Fiducia & GAD IT AG

How to play with the frog?

- Variablen
  - aktiviere mit VARSUB SCAN
  - VARSET-Statement
  - Prefix: „!“ oder „Ü“
  - Delimiter: „.“
  - Variablen werden vor Ausführung eines CMDs ersetzt

```
VARSUB SCAN SEARCH(NOGLOBAL)

VARSET MYVAR = 'HELLO WORLD'
VARSET MYVAR VALUE('HELLO WORLD')

DISPLAY ÜMYVAR.
```

- Objekt-Variablen
  - werden oft durch Kommandos erzeugt
  - Zugriff auf Inhalte: @V(@OBJ) bzw. §V(§OBJ)
  - Werden erst zur Ausführung aufgelöst

```
LIST [...] OBJECT(MYOBJ)

DISPLAY §V(§MYOBJ)

VARSET MYOBJVAR §V(§MYOBJ)
DISPLAY ÜMYOBJVAR.
```


© Fiducia & GAD IT AG

How to play with the frog?

- LOADDEF
  - Lädt Segment-Schema in WAPL
  - Batch Loader & Data DD-Statements überschreiben/deaktivieren/umleiten
  - Löst OUTPUT-Statement *ab*
- Data Commands:
  - List: Listet Segmente
  - Select: Erzeugt Batch Loader
  - Insert: Fügt neues Segment ein (CP/AD)
  - Modify: Modifizier Segmente (CP/AD)
  - DELETE: Löscht Segment (CP/AD)

  - Können ILSON-Daten erzeugen
  - Können Objekte erzeugen

```
LOADDEF CP* LOADER(OUTBL) DATA(-)

LOADDEF ADCOM LOADER(BL) DATA(*)
```

```
LIST CPOC ADID(MYAPPL*) IA-LE(=) STATUS(C)

LIST CPOC ADID(MYAPPL*) IA-LE(=) STATUS(C)
OBJECT(MYOBJ)

LIST ADCOM ADID(MYAPPL*) IA-LE(=) STATUS(C)
OBJECT(MYOBJ) SELECT(Y)

SELECT ADCOM ADID(MYAPPL*) VALID(=)
```


© Fiducia & GAD IT AG

How to play with the frog?

Keine ILSON-Daten

Occurences in Error

```
LOADDEF CP* DATA(-)

LIST CPOC ADID(U*) GROUP(TWSADM*) OBJECT(APPLS) STATUS(E)

DO I = 1 TO §V(§APPLS)
   VARSET ID §V(§APPLSÜI.-CPOCADI)
   VARSET IA = §V(§APPLSÜI.-CPOCIAD) !! §V(§APPLSÜI.-CPOCIAT)

   MODIFY CPOC ADID(ÜID.) IA(ÜIA.) STATUS(C)

   IF §CMD(EQ.0) THEN
      DISPLAY §LOG() "COMPLETED" §V(§APPLSÜI.-CPOCADI)
   ELSE
      DISPLAY §LOG() "COULD NOT COMPLETE" §V(§APPLSÜI.-CPOCADI)
END
```

Anzahl Elemente in Objekt

Complete

RC Handling des Modify

© Fiducia & GAD IT AG

How to play with the frog?

- Gut zu wissen:
  - FILTER-Command für CPOC geht nicht
  - Batch Loader funktioniert auch inkrementell
  - Neue Batch Loader Syntax möglich OPTIONS SYNTAX
  - MSGLEVEL(5) loggt generierte PIF-Commands
  - ILSON-Data Output <> Batch Loader
  - Max. Parameter 250 Zeichen für Rexx-Functions
  - TCPIP-TERM benötig 1 Sekunde
  - INSERT/SELECT/DELETE auf CPSRU erzeugt RC 12 anstatt 4
  - WAPL kann via REST-API aufgerufen werden

© Fiducia & GAD IT AG

Q & A!

FIDUCIA GAD
ZUKUNFTSERFAHREN