How do I launch a Frog? Let me count the ways……

Anna Dawson
IBM (UK) Ltd

November 2018
Session NG

Abstract

- Over the years SOE/WAPL has been associated with the concept of juggling frogs, a reference to the developers analogy that working with the IBM Workload Scheduler for z/OS (IWSz) Programming Interface (PIF) is somewhat akin to juggling frogs, its slippery and bits of it, like frogs' legs, get in the way - it's difficult, but with time and perseverance it can be done.
- However, not everyone has the desire to juggle frogs, they just like an easy way to get IWSz PIF to do something.
- This session is a basic overview of the various ways you can **launch** WAPL, as a job or a started task, as an IWSz operation, from IWSz exits, or even customized in the panels.
- With luck there may even be a demo of some simple WAPL tasks.

Running WAPL from a batch job

By far the simplest way to run WAPL is in batch

```
//RUNWAPL   EXEC EQQYXJPX,
//          SUBSYS=IWSX
//OUTBL     DD SYSOUT=*
//OUTDATA   DD SYSOUT=*
//SYSIN     DD *
SHOW OPTIONS
```

Proc to use from APAR PI79321

Controller name

Optional Batch Loader output

Optional ILSON output

It’s a Job

This job is to list the successors of a specific job

```
//DAWSOA1W JOB ,'WAPL TEST',
//          CLASS=A,MSGCLASS=H
//          JCLLIB ORDER=OPCUSR.WAPL.EQQJOBS      ← My procedure library
//RUNWAPL   EXEC EQQYXJPX   (EQQYXJCL superseded) ← My procedure
//SYSIN DD *
 LISTJOB OPJCD10A DETAIL(SUCC) ADID(WAPLTESTCYCLE)
```

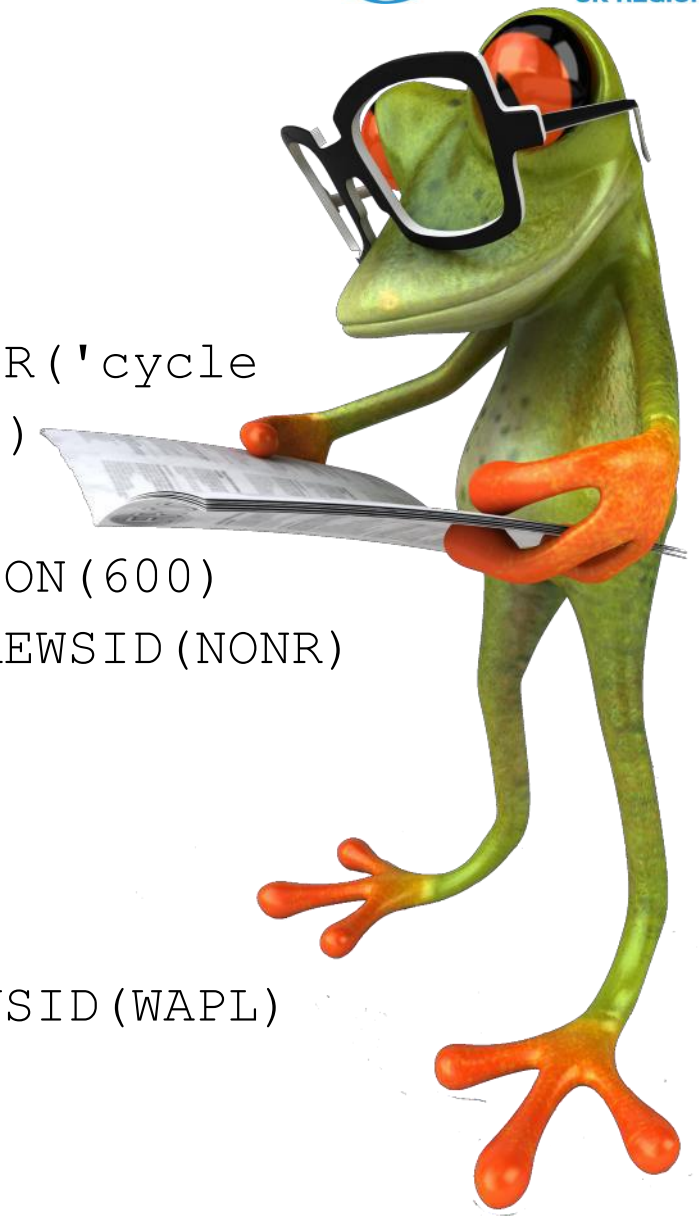
This JCL may be scheduled, or the job can just be submitted, assuming you have the correct authority ☺

If you need to ‘debug’ or like to understand what’s happening under the covers you can add a EQQYLOGx DD

```
//EQQYLOG5 DD SYSOUT=*
```

The output from my job

```
Application: ADID(WAPLTESTCYCLE) ADVALFROM(150117) DESCR('cycle
             solution') OWNER(ANNA) DSMOOTHING(00000000)
             DLIMFDBK(00000100)
  Operation: WSID(OPAC) OPNO(010) JOBN(OPJCD10A) DURATION(600)
    Predecessor-INT: DUMMYJOB PREADID(WAPLTESTCYCLE) PREWSID(NONR)
                     PREOPNO(001)
    Resource: RESOURCE(MY.RESOURCE)
    User field: UFNAME(JCL) UFVALUE(JB#)
    User field: UFNAME(MEMBER) UFVALUE(OPACD10)
    Successor-INT: OPJCCMD1 SUCADID(WAPLTESTCYCLE) SUCWSID(WAPL)
                   SUCOPNO(255)
```



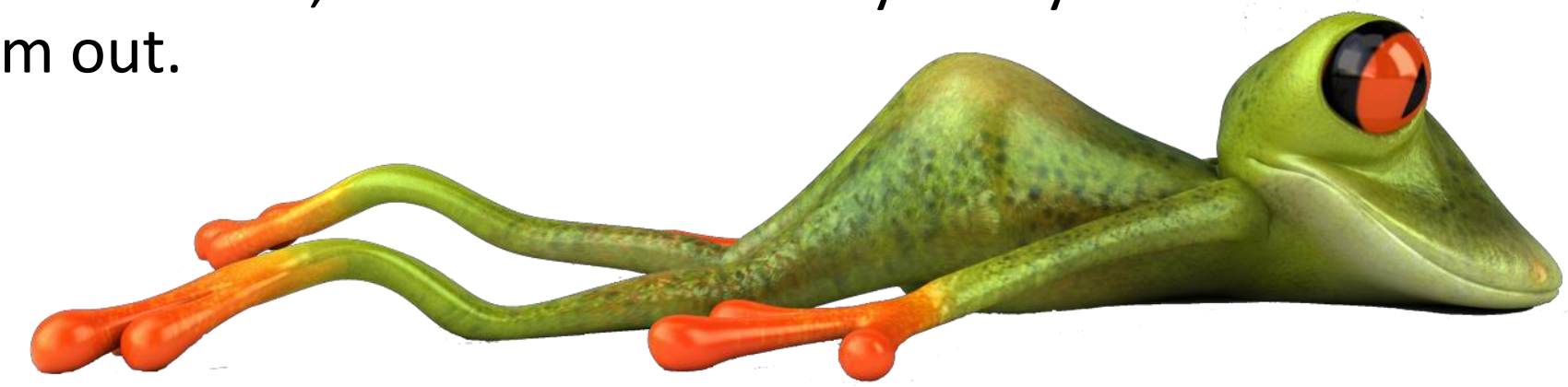
It’s an STC (1)

Started by a z/OS start command when a WAPL procedure resides in the JES Procedure concatenation,

or, scheduled by IWSz from an STC workstation.

The setting up of started task support in IWSz has a lot of ifs and buts associated with it. It depends on several things, like whether you are using Restart & Clean-up, as to how it’s actually set-up.

It is documented in the manuals, but it’s best left to your System Programmer to fathom out.



Running WAPL from an Application

If your Systems Programmer sets up STC workstations

```
---------------------------- OPERATION USER FIELDS ----------- Row 1 to 2 of 2
Command ===>                                                 Scroll ===> CSR

Enter/Change data in the rows, and/or enter any of the following
row commands:
I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete

Application      : DH#WAPL                        WAPL Demo
Operation        : CMD1 001     Run WAPL commands
Jobname          : TWSXCMD1

Row  User Field Name  User Field Value
cmd                   ----+----1----+----2--
'''' EQQ-SYSIN-01     SHOW OPTIONS
'''' EQQ-SYSIN-02     SHOW VARIABLES
******************************** Bottom of
```

STC workstation

Standard Procedure/Job

Based on EQQWCMD1/2

User Fields and WAPL work for all CPU type workstations

User Fields with EQQ-SYSIN- prefix

My Proc…….

```
//EQQYXJPX PROC @=,
//    ARGS='',
//    CMD=EQQYXTOP,
//    REG=4M,
//    SUBSYS=OPJC,
//    VER=V930,
//    #=
..
```

```
//EQQYXTOP EXEC PGM=IKJEFT01,
//          REGION=&REG,
//          PARM=&@'&CMD &SUBSYS-&VER &ARGS'&#
//STEPLIB   DD DISP=SHR,DSN=OPC.&VER..SEQQLMD0
//SYSPROC   DD DISP=SHR,DSN=OPC.V930.SEQQMISC
//EQQMLIB   DD DISP=SHR,DSN=OPC.&VER..SEQQMSG0
//EQQMLOG   DD SYSOUT=*
//EQQDUMP   DD SYSOUT=*
//EQQSMTP   DD SYSOUT=(B,SMTP)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN   DD DUMMY
```

It’s an STC (2)

Started by EQQUX007 (issues the start command)

```
------------------------------ OPERATION USER FIELDS ----------- Row 1 to 7 of 7
Command ===>                                                    Scroll ===> CSR

Enter/Change data in the rows, and/or enter any of the following
row commands:
I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete

Application      : @TEST#X7#EMAIL                send email (WAPL STC)
Operation        : OPJC 010
Jobname          : OPJCMAIL

Row
cmd  User Field Name     User Field Value
                         ----+----1----+----2----+----3----+----4----+----5----
'''' SYSCMD-C            S OPJCX7M,JOBNAME=&JN,ARGS='OPID(&AD &IA &OP)',S=C
'''' JCL                 JB#
'''' MEMBER              OP@CD30
'''' EQQ-X7C-01          VARSET TXT VALUE(This is a demo of using WAPL STC to
'''' EQQ-X7C-02          run SENDMAIL. Started by EQQUX007 when a job
'''' EQQ-X7C-03          completes. Hope it works okay)
'''' EQQ-X7C-04          INCLUDE VARDATE(TESTMAIL)
********************************* Bottom of data ********************************
```

This process requires that EQQUX007 has been set-up to read user fields and act on the data found.

For an exit that already does this contact myself or Dean

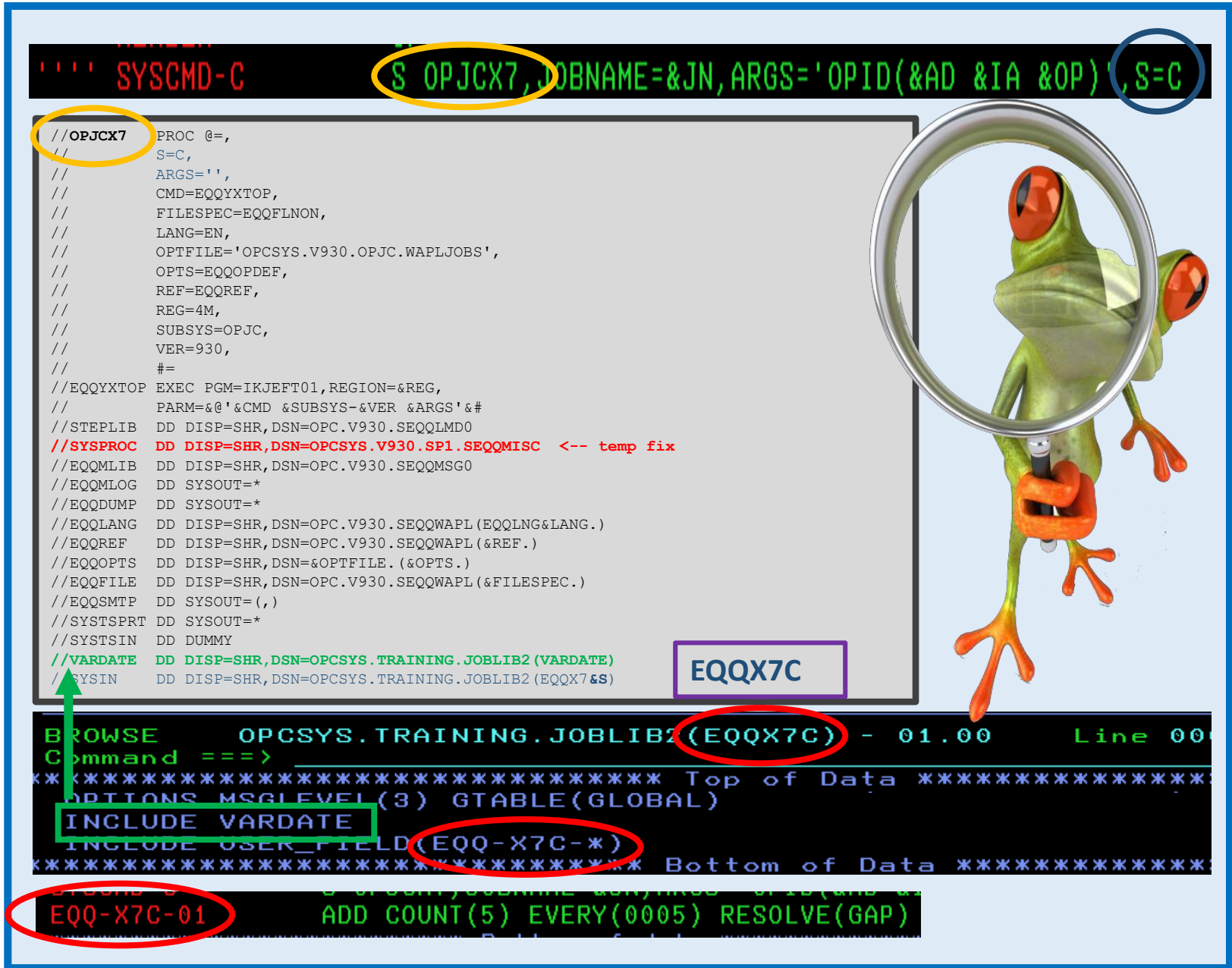
EQQUX007 & WAPL STC

Our EQQUX007 exit reads the user field names looking for SYSCMD-C (where C is the new status of the operation, could be E, S etc..)

It issues the command detailed in the user field value, in this case it a start command for procedure OPJCX7, which must exist in the JES2 PROCLIB concatenation

This is a WAPL STC procedure which uses the OPID arguments to know which operation caused the task to be started

The STC then finds the WAPL related user fields to know what action to take.



It’s a REXX / CLIST

```
/* REXX */
/*--------------------------------------------------------------------+
 | Initialise parameter stem                                          |
 +--------------------------------------------------------------------*/
PARSE ARG APPL JOBN WSID OPNO
x.          = ""
x.WAPLPFX   = "OPCUSR.WAPL"
x.TWSPFX    = "OPC.V930"
Set-up rest of WAPL environment
/*---- Queue some commands to WAPL  -----*/
QUEUE "OPTIONS FIELDSEP(,)"
QUEUE "LISTJOB "jobn" DETAIL(SUCC) ADID("appl") DISPLAY(N) OUTPUT(*)"
/*---- Run WAPL -------------------------*/
waplRC = EQQYXTOP(x.SUBSYS x.FORE x.ARGS)
if waplRC > 0 then SAY "RC="||waplRC
/*----- Pocess the WAPL output  ---------*/
DEP. = 0
DEP# = 0
DO WHILE QUEUED() > 0
   PARSE PULL waplOut
DEP.0 = DEP.0 + 1
   DEP# = DEP# + 1
   DEP.DEP# = STRIP(waplOut)
END
CAll process_waplOut
IF XTAB=YES THEN DO                  /* THERE WAS A JOB SO DISPLAY    */
 ADDRESS ISPEXEC "TBTOP TEMPTAB"
 ADDRESS ISPEXEC "TBDISPL TEMPTAB PANEL(OPCABDEP)"
 ADDRESS ISPEXEC "TBEND TEMPTAB"
END                                  /* END THE IF/THEN/DO            */
ELSE DO
 ADDRESS ISPEXEC "SETMSG MSG(OPCU003I)"
END
```

EQQJCLE

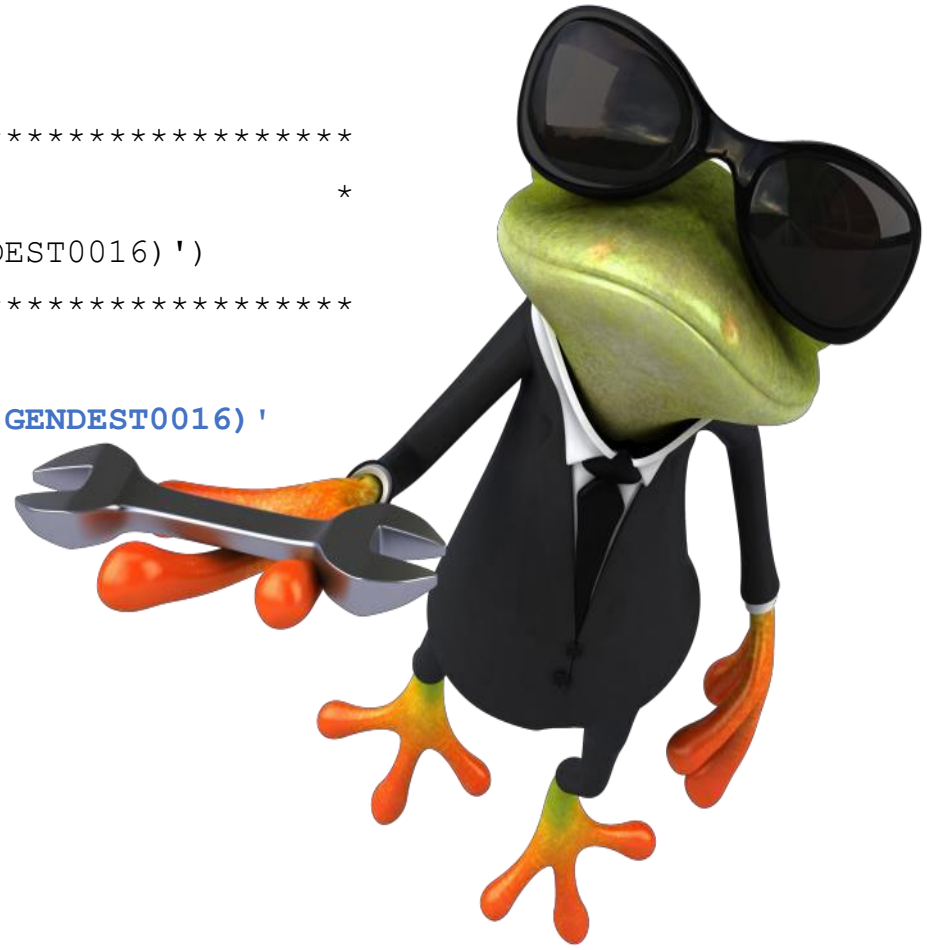
```
)PROC
 &ZSEL = TRANS(TRUNC (&ZCMD,'.')
   J,'CMD(%@TMEDIT J &opjobnm &panmod OPAC)'
   S,'CMD(%@TMSUCC &applid &opjobnm &opws &opid)'
   U,'CMD(%@TMEDIT D &opjobnm &panmod OPAC)'
   D,'CMD(%@TMDEPS &applid &opjobnm &opws &opid)'
   V,'CMD(%@SIM &opjobnm)'
   '','')
CANCEL,'EXIT'
     X,'EXIT'
     *,'?' )
)END
```

REXX Results

```
---------------------- BROWSING PREDECESSORS & SUCCESSORS ---- Row 1 t     4
Command ===> _                                                Scroll =

Predecessors and Successors of
                    Application: WAPLTESTAPPL
                    Operation  : WAPLTJB2
                    WSID Opno  : OPJC  10

 PRED              INT   Application        Jobname    WSID   OP#
 SUCC              EXT   NAme

  Predecessor      INT   WAPLTESTAPPL       WAPLTJB1   OPJC   005
  Predecessor      EXT   GSE#DEMO#AD2       LASTOP2    NONR   255
  Successor        INT   WAPLTESTAPPL       WAPLTJB3   OPJC   015
  Successor        EXT   GSE#DEMO#AD2       FIRSTAD2   NONR   001

****************************** Bottom of data ********************************
```

It’s a LOAD module!

```
//*******************************************************************
//* CALL WAPL LOAD MODULE                                           *
//*  'ARGUMENT(TNCRX212 CAPT   3336              GENDEST0016)')
//*******************************************************************
//WAPL     EXEC PGM=EQQWAPL,PARM='OPKC INPUT(SYSIN)
//             ARG(TNCR212  CAPT   3336              GENDEST0016)'
//STEPLIB  DD DISP=SHR,DSN=OPC.V930.SEQQLMD0
//EQQMLIB  DD DISP=SHR,DSN=OPC.V930.SEQQMSG0
.........
//SYSIN    DD *
OPTIONS MSGLEVEL(3) GTABLE(PLANTAB)
DISPLAY "CALLWAPL V01.06 - DATA(24)"
VARSUB SCAN    VARFAIL(NULL)
VARSET VARMAX  VALUE(10)
VARSET VARLEN  = !VARMAX * 27
VARSET MYDATA ENVATTR(OPTIONS,ARGUMENT) <= SETS MYDATA to the string of data defined as ARG
```

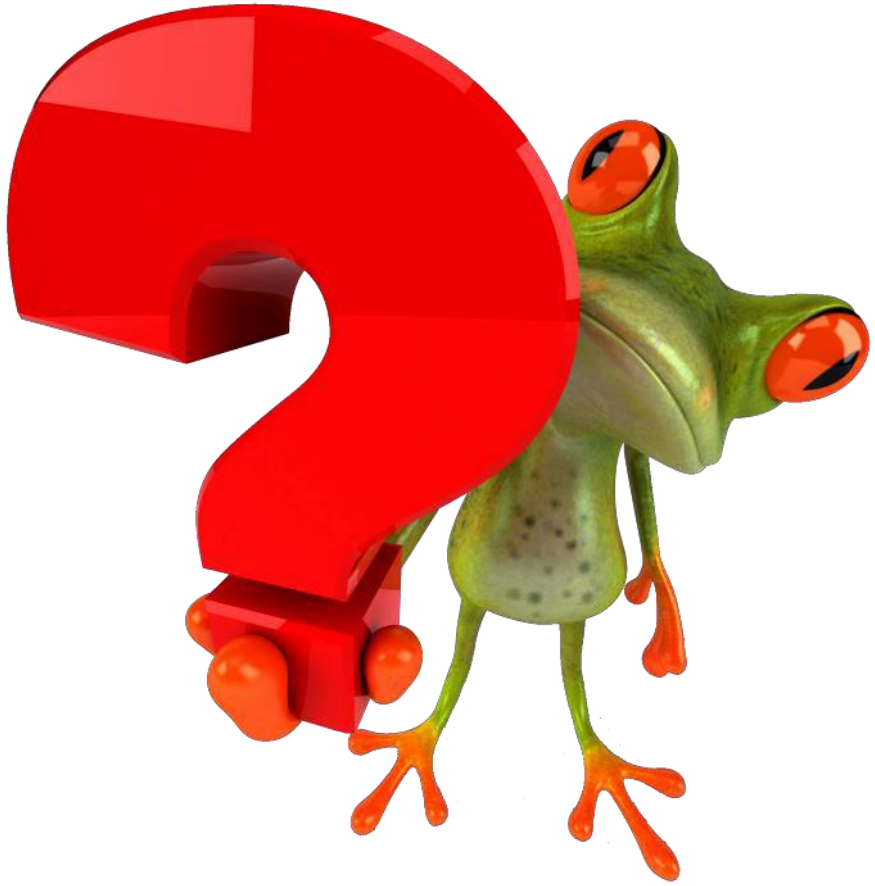


DEMO

(Hopefully)

1. Submit a WAPL job manually
2. Run a WAPL job through IWSz
3. Have EQQUX007 start a WAPL started task
4. Use WAPL to provide a customised result in EQQAJCLE

Questions



We want your feedback!

- Please submit your feedback online at ….
  - http://conferences.gse.org.uk/2018/feedback/NG

- Paper feedback forms are also available from the Chair person

- This session is NG