The 2013 RexxLA Symposium
AN HTML WEB APP DONE ISPF STYLE

Marc Vincent Irvin
mvi@MarcIrvin.com
Representing
Move Immediate Software
at
http://MarcIrvin.com/rexxsrc/
Screen layouts that are static as obsolete 3270 dumb terminal.

Raw and fast Rexx code that talks directly to and from browser.

Imbedded panels defs logic upgraded from old ansi.sys code.

Global variables are handled as hidden HTML attributes.

Listboxes, checkboxes, radiobuttons, PWs, and more supported.

What components make it happen:

- CGIPARSE (i.e. ?var1=val1&var2=val2a+val2b&etc...)
- REXXSCRN (i.e. filename panelstart panelstop ( options... ) )
- REXXRDR (i.e. filename sequence searchkey searchmode )

Questions?
Screen layouts that are static as obsolete 3270 dumb terminal.

/*
PNL_IDSIGNIN: /* *html */
! ---- SPACE BOOKS SIGN IN ----

EMAIL ======> %IDEMAILADDR
       &emerrors

PASSWORD(PW) ======> `IDPASSWORD
       &pwerrors

CREATE A SIGNON ID ======> ^IDSIGNUP=REQ
       ! Create ID, PW, and sec_ques.

CHANGE SIGNON DATA ======> ^IDSIGNUP=CHG
       ! Change PW, name, or sec_ques.

EOP:  /**** END OF PANEL ****/
*/
The 2013 RexxLA Symposium
Dirt simple CGI web app design for people who love ISPF on VM and MVS

Screen layouts that are static as obsolete 3270 dumb terminal.
The 2013 RexxLA Symposium
Dirt simple CGI web app design for people who love ISPF on VM and MVS

Raw and fast Rexx code that talks directly to and from browser.

SIGNINIT:  /* display the signin screen */
xnam = 'mvisginin.'!idnt'.htm'
address cmd 'erase' xnam '*sysfn > nul: 2>&1'
 idemailaddr = 'your@email.com'
idpassword = ''
idsignupreq = 1 /* if selected this will be returned. */
idsignupchg = 1 /* if selected this will be returned. */
idsignup = ''
emerrors = '' ; pwerrors = ''
!x = ''
!x = !x 'idemailaddr idpassword idsignupreq idsignupchg',
 'idsignup'
attrs = space(!x,1)
Call signscrn
Signal exit
The 2013 RexxLA Symposium
Dirt simple CGI web app design for people who love ISPF on VM and MVS

Imbedded panels defs logic upgraded from old ansi.sys code.
Global variables are handled as hidden HTML attributes.
The 2013 RexxLA Symposium

Dirt simple CGI web app design for people who love ISPF on VM and MVS

Listboxes, checkboxes, radiobuttons, PWs, and more supported
What components make it happen:  

CGIPARSE

```
!nx = 1
!hs = ''
do x = 1 while !qv \= ''
   parse var !qv !qt '=' !qd ' & ' !qv
   !qd = cgideweb(!qd,'+')
   if !qd == '+' then !qd = ''
   if symbol(!qt) = 'BAD' then iterate
   ! = wordpos(translate(!qt),!hs)
   if ! \= 0
      then do
         parse value subword(!hs,!,2) with . !hx .
         parse value answ.!hx with . '=' !hv
         !hv = substr(!hv,2,length(!hv)-2)
         !qd = strip(!hv '|' !qd)
      end
   if pos('"','!qd) = 0
      then do
         answ.!nx = !qt "= '"|!qd"'
      end
   else if pos('"','!qd) = 0
      then do
         answ.!nx = !qt ' = '"|!qd"'
      end
   else iterate  /* illegal use of dbl-quote */
   !hs = strip(translate(!qt) !nx !hs)
end
```

```
What components make it happen:  REXXSCRN

SIGNSCRN:

xnam = 'mvisgnin.'!idnt'.htm'
!x = "mvisgnin.rex pnl_idsignin: eop:"
   "( *html *startoff *resp *idt("!idnt") *quiet",
   "*cginame(mvisgnin.rex)",
   "*icon(keynlock.gif)" pkgattrs(attrs)
!x = REXXSCRN(!x debug)
if !x = 0 then do
   address cmd 'TYPE' xnam '2> mvisgnin.'!idnt'.err'
   if result \= 0
      then do
         errormsg = 'Problem Typing' xnam 'into browser.'
         signal stop
      end
   end
if !x \= 0 then do
   errormsg = 'Problem displaying' xnam'.'
   signal stop
end
return 0
The 2013 RexxLA Symposium

Dirt simple CGI web app design for people who love ISPF on VM and MVS

What components make it happen: REXXRDR

RETURN POS

DOC:

/*BEGTYPE
PGMNAME: REXXRDR

FUNCTION: Randomly read named dataset using a supplied index, and
an optional search keyword.

DESCRIPTION: This EXEC when invoked will read named file using EXECIO at
the indicated location and create a stack entry for record.
If optional keyword is entered an additional stack entry
will be made defining the sequence of next record to resume
reading at.

EXEC FORMAT:

Enter command in the format shown below for CMS:

REXXRDR VFNAM VFTYP VFMOD VRECSEQ < VKEYWORD <LOCATE|FIND> > < VOPTS >

Enter command in the format shown below for TSO datasets:

REXXRDR $DSN VFDSN VFDD VRECSEQ < VKEYWORD <LOCATE|FIND> > < VOPTS >

For already allocated DDs that refer to PO datasets enter:

REXXRDR $DSN VMEM VMEMDD VRECSEQ < VKEYWORD <LOCATE|FIND> > < VOPTS >

< > - Means fields within are optional.
| - Means select for entry one or the other, not both.

VFNAM - Enter name of file to read. EX... cobolpgm
VFTYP - Enter type of file to read. EX... cobol
VFMOD - Enter mode of file to read. EX... A
VRECSEQ - Enter seq. of rec. to read. EX... 001
In CMS the file is closed after each request.

$DSN - Enter exactly as shown. EX... $DSN
VFDSN - Enter name of dataset to read. EX... 'REX.PO(TBL)'
Follow TSO's standard dataset
naming conventions. For instance,
dataset names not within quotes
will be prefixed with a TSO ID.

VMEM - If the file wanted is a member EX... (REXXTBL)
in a PO dataset(s) already allocated
to a DD name you may enter only the
the member name enclosed in perens.
The 2013 RexxLA Symposium
Dirt simple CGI web app design for people who love ISPF on VM and MVS

QUESTIONS?