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/

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.**
- **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?**

The 2013 RexxLA Symposium

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

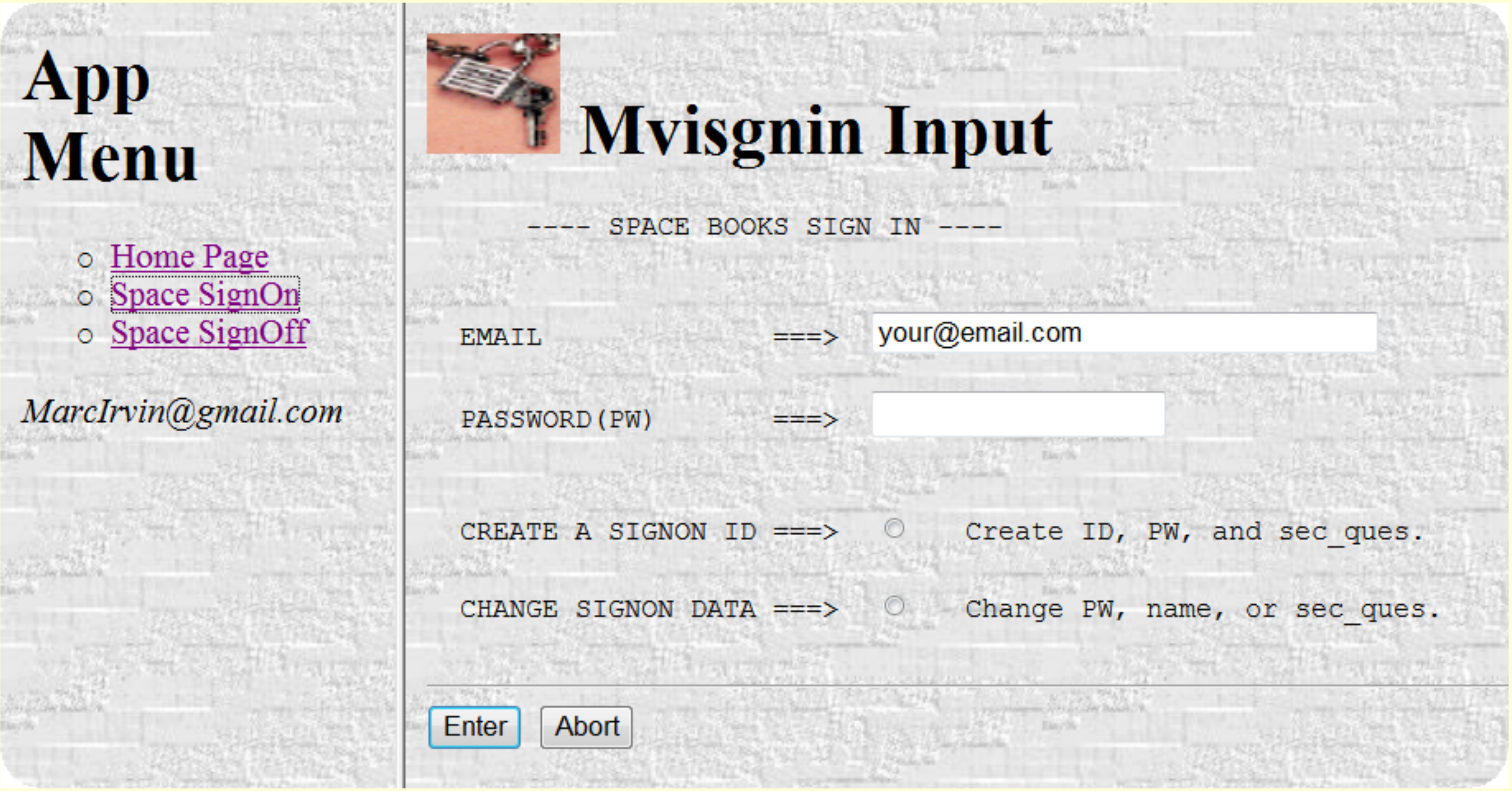
Screen layouts that are static as obsolete 3270 dumb terminal.

```
/*
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.



App Menu

- Home Page
- Space SignOn
- Space SignOff

MarcIrvin@gmail.com

Mvisgnin Input

```
---- SPACE BOOKS SIGN IN ----

EMAIL                ===>  your@email.com

PASSWORD(PW)         ===>

CREATE A SIGNON ID ===>  ○  Create ID, PW, and sec_ques.

CHANGE SIGNON DATA ===>  ○  Change PW, name, or sec_ques.
```

The 2013 RexxLA Symposium

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

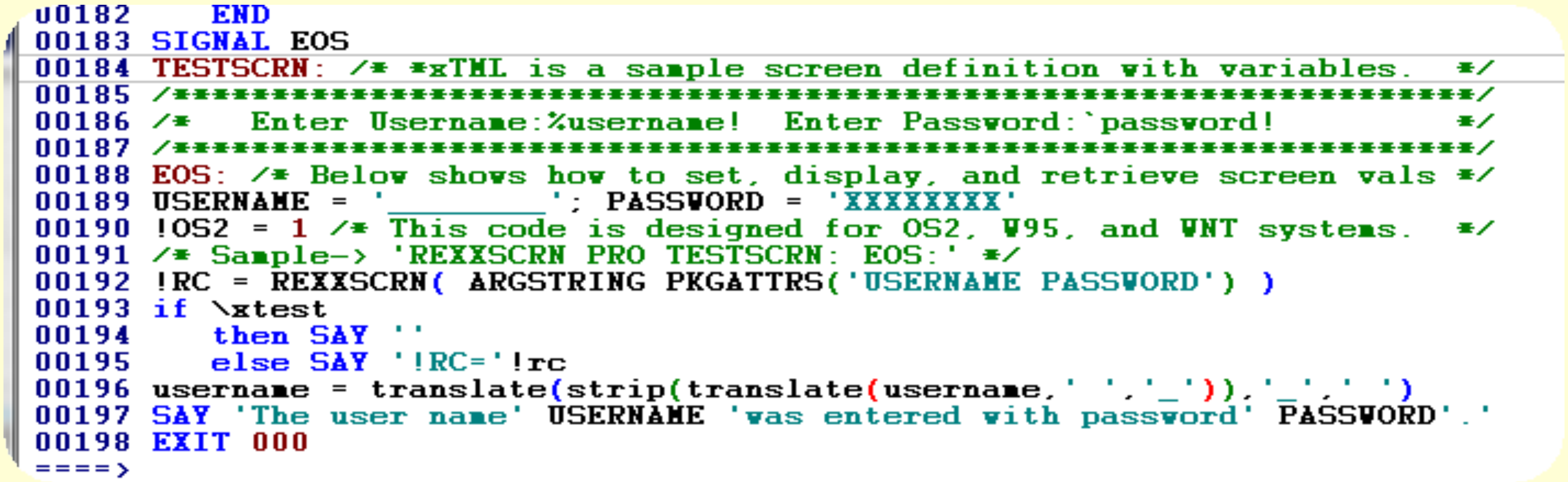
Raw and fast Rexx code that talks directly to and from browser.

```
SIGNINIT:  /* display the signin screen */
xnam = 'mvisgnin.'!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
```

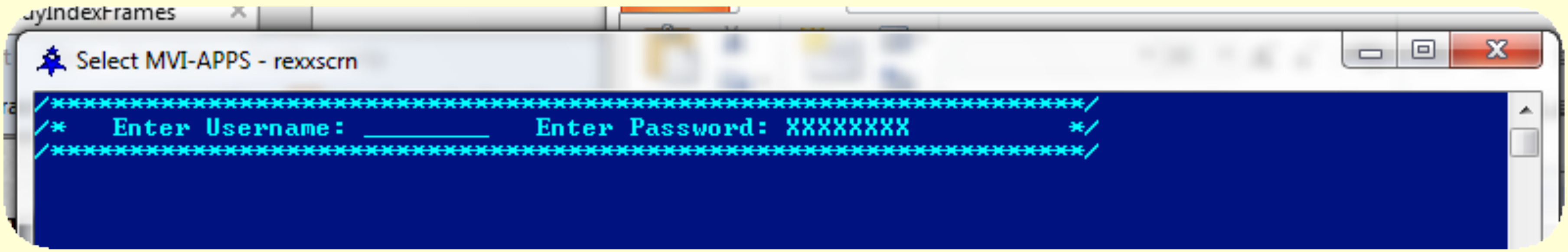
The 2013 RexxLA Symposium

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.



```
   END
SIGNAL EOS
TESTSCRN: /* *xTML is a sample screen definition with variables.   */
/*******************************************************************/
/*   Enter Username:%username!  Enter Password:`password!          */
/*******************************************************************/
EOS: /* Below shows how to set, display, and retrieve screen vals */
USERNAME = '________'; PASSWORD = 'XXXXXXXX'
!OS2 = 1 /* This code is designed for OS2, W95, and WNT systems.   */
/* Sample-> 'REXXSCRN PRO TESTSCRN: EOS:' */
!RC = REXXSCRN( ARGSTRING PKGATTRS('USERNAME PASSWORD') )
if \xtest
   then SAY ''
   else SAY '!RC='!rc
username = translate(strip(translate(username,' ','_')),'_',' ')
SAY 'The user name' USERNAME 'was entered with password' PASSWORD'.'
EXIT 000
====>
```



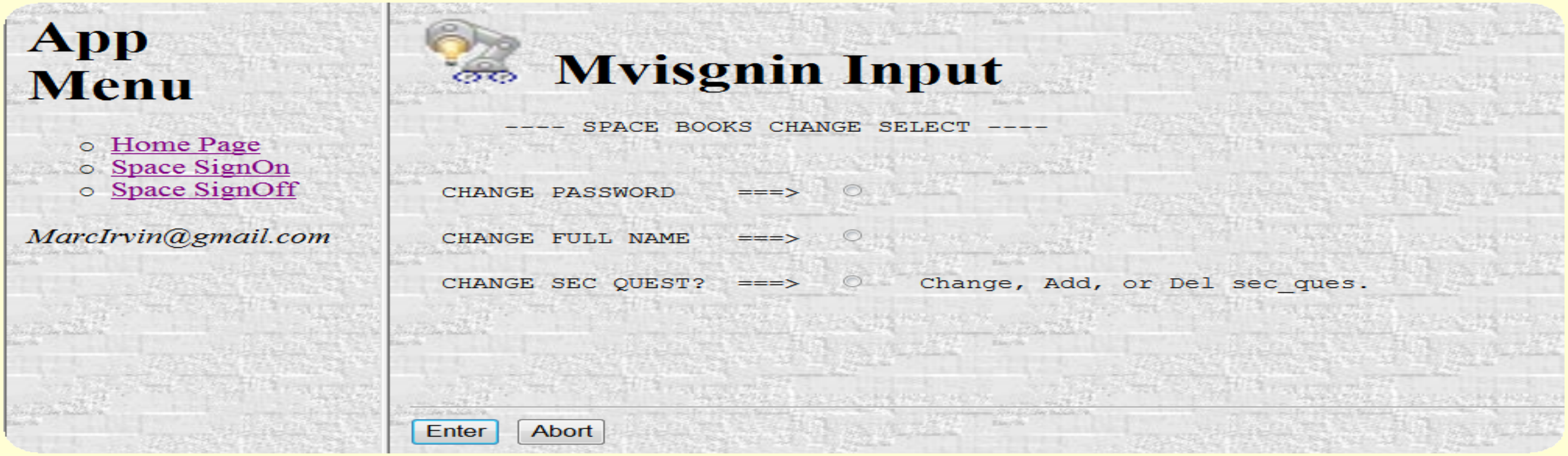
Select MVI-APPS - rexxscrn

```
/*******************************************************************/
/*   Enter Username: ________   Enter Password: XXXXXXXX           */
/*******************************************************************/
```

The 2013 RexxLA Symposium

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

Global variables are handled as hidden HTML attributes.



App Menu

- Home Page
- Space SignOn
- Space SignOff

MarcIrvin@gmail.com

Mvisgnin Input

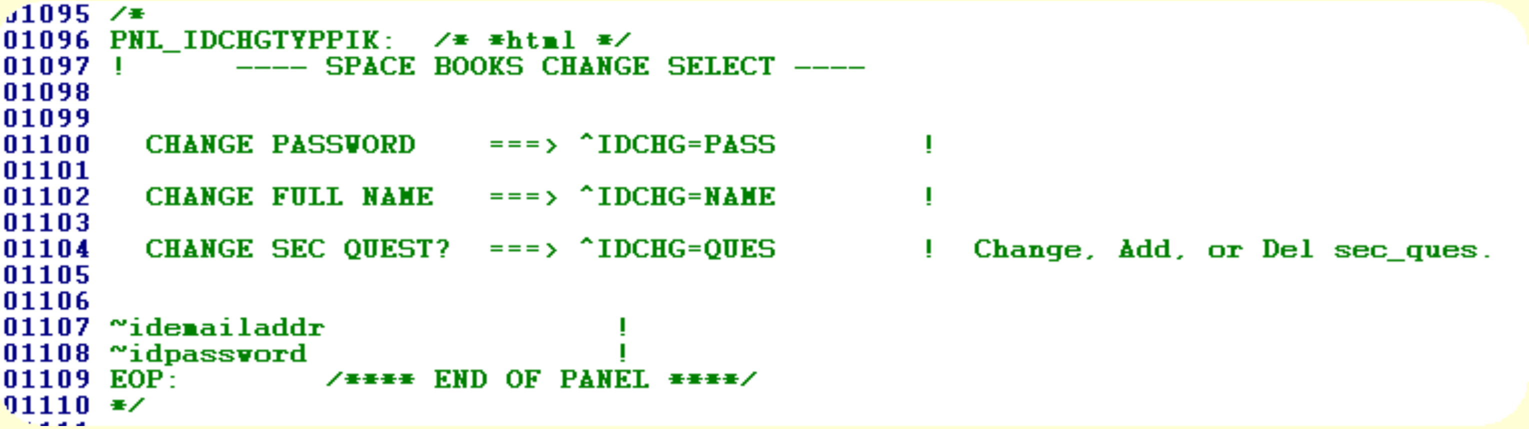
---- SPACE BOOKS CHANGE SELECT ----

CHANGE PASSWORD ===> ○

CHANGE FULL NAME ===> ○

CHANGE SEC QUEST? ===> ○ Change, Add, or Del sec_ques.

Enter Abort

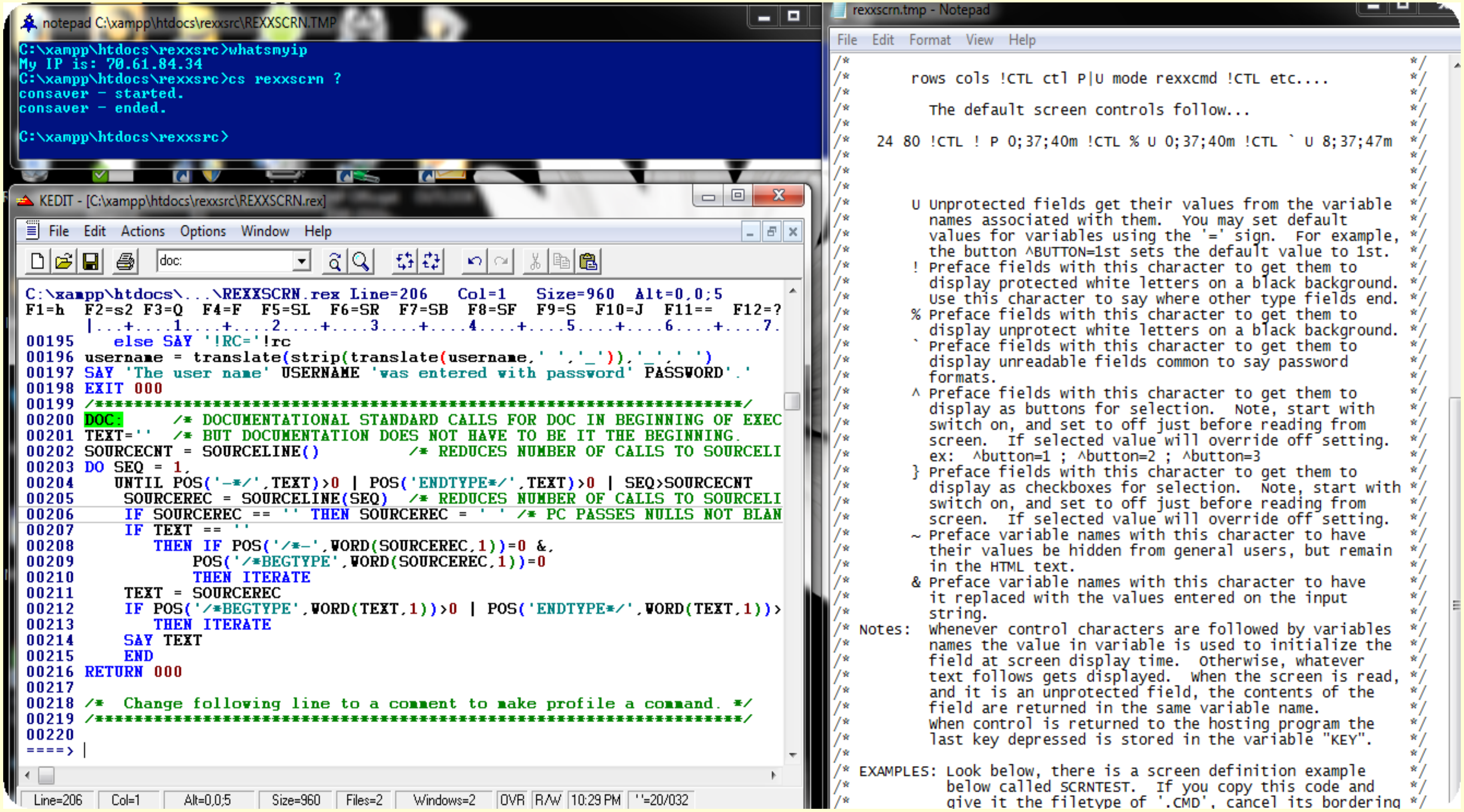


```
/*
PNL_IDCHGTYPPIK:  /* *html */
!      ---- SPACE BOOKS CHANGE SELECT ----


  CHANGE PASSWORD    ===> ^IDCHG=PASS        !

  CHANGE FULL NAME   ===> ^IDCHG=NAME        !

  CHANGE SEC QUEST?  ===> ^IDCHG=QUES        !  Change, Add, or Del sec_ques.


~idemailaddr                 !
~idpassword                  !
EOP:        /**** END OF PANEL ****/
*/
```

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


notepad C:\xampp\htdocs\rexxsrc\REXXSCRN.TMP
KEDIT - [C:\xampp\htdocs\rexxsrc\REXXSCRN.rex]
File
Edit
Actions
Options
Window
Help
doc:
rexxscrn.tmp - Notepad
File
Edit
Format
View
Help

```
C:\xampp\htdocs\rexxsrc>whatsmyip
My IP is: 70.61.84.34
C:\xampp\htdocs\rexxsrc>cs rexxscrn ?
consaver - started.
consaver - ended.

C:\xampp\htdocs\rexxsrc>
```

```
C:\xampp\htdocs\...\REXXSCRN.rex Line=206   Col=1   Size=960  Alt=0,0;5
F1=h  F2=s2 F3=Q  F4=F  F5=SL  F6=SR  F7=SB  F8=SF  F9=S  F10=J  F11==  F12=?
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7.
   else SAY '!RC='!rc
username = translate(strip(translate(username,' ','_')),'_',' ')
SAY 'The user name' USERNAME 'was entered with password' PASSWORD'.'
EXIT 000
/*****************************************************************/
DOC:      /* DOCUMENTATIONAL STANDARD CALLS FOR DOC IN BEGINNING OF EXEC
TEXT=''   /* BUT DOCUMENTATION DOES NOT HAVE TO BE IT THE BEGINNING.
SOURCECNT = SOURCELINE()            /* REDUCES NUMBER OF CALLS TO SOURCELI
DO SEQ = 1,
   UNTIL POS('-*/',TEXT)>0 | POS('ENDTYPE*/',TEXT)>0 | SEQ>SOURCECNT
    SOURCEREC = SOURCELINE(SEQ)  /* REDUCES NUMBER OF CALLS TO SOURCELI
    IF SOURCEREC == '' THEN SOURCEREC = ' ' /* PC PASSES NULLS NOT BLAN
    IF TEXT == ''
       THEN IF POS('/*-',WORD(SOURCEREC,1))=0 &,
           POS('/*BEGTYPE',WORD(SOURCEREC,1))=0
           THEN ITERATE
    TEXT = SOURCEREC
    IF POS('/*BEGTYPE',WORD(TEXT,1))>0 | POS('ENDTYPE*/',WORD(TEXT,1))>
       THEN ITERATE
    SAY TEXT
    END
RETURN 000

/*  Change following line to a comment to make profile a command. */
/*****************************************************************/

====> |
```

```
Line=206  Col=1  Alt=0,0;5  Size=960  Files=2  Windows=2  OVR  R/W  10:29 PM  ''=20/032
```

```
/*                                                                   */
/*      rows cols !CTL ctl P|U mode rexxcmd !CTL etc....              */
/*                                                                   */
/*        The default screen controls follow...                       */
/*                                                                   */
/*   24 80 !CTL ! P 0;37;40m !CTL % U 0;37;40m !CTL ` U 8;37;47m      */
/*                                                                   */
/*                                                                   */
/*                                                                   */
/*      U Unprotected fields get their values from the variable      */
/*        names associated with them.  You may set default           */
/*        values for variables using the '=' sign.  For example,     */
/*        the button ^BUTTON=1st sets the default value to 1st.      */
/*      ! Preface fields with this character to get them to          */
/*        display protected white letters on a black background.     */
/*        Use this character to say where other type fields end.     */
/*      % Preface fields with this character to get them to          */
/*        display unprotect white letters on a black background.     */
/*      ` Preface fields with this character to get them to          */
/*        display unreadable fields common to say password           */
/*        formats.                                                   */
/*      ^ Preface fields with this character to get them to          */
/*        display as buttons for selection.  Note, start with        */
/*        switch on, and set to off just before reading from         */
/*        screen.  If selected value will override off setting.      */
/*        ex:  ^button=1 ; ^button=2 ; ^button=3                     */
/*      } Preface fields with this character to get them to          */
/*        display as checkboxes for selection.  Note, start with     */
/*        switch on, and set to off just before reading from         */
/*        screen.  If selected value will override off setting.      */
/*      ~ Preface variable names with this character to have         */
/*        their values be hidden from general users, but remain      */
/*        in the HTML text.                                          */
/*      & Preface variable names with this character to have         */
/*        it replaced with the values entered on the input           */
/*        string.                                                    */
/* Notes: Whenever control characters are followed by variables     */
/*        names the value in variable is used to initialize the      */
/*        field at screen display time.  Otherwise, whatever         */
/*        text follows gets displayed.  When the screen is read,     */
/*        and it is an unprotected field, the contents of the        */
/*        field are returned in the same variable name.              */
/*        When control is returned to the hosting program the        */
/*        last key depressed is stored in the variable "KEY".        */
/*                                                                   */
/* EXAMPLES: Look below, there is a screen definition example       */
/*           below called SCRNTEST.  If you copy this code and       */
/*           give it the filetype of '.CMD', cancel its bordering    */
```

The 2013 RexxLA Symposium

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

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)
   if !qt = 'xidnt' /* set the hidden value now */
      then !idnt = value(!qd)
   answ.0 = !nx
   !nx = !nx + 1
   end
do !nx = 1 for answ.0
   interpret answ.!nx
   end
```

The 2013 RexxLA Symposium

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

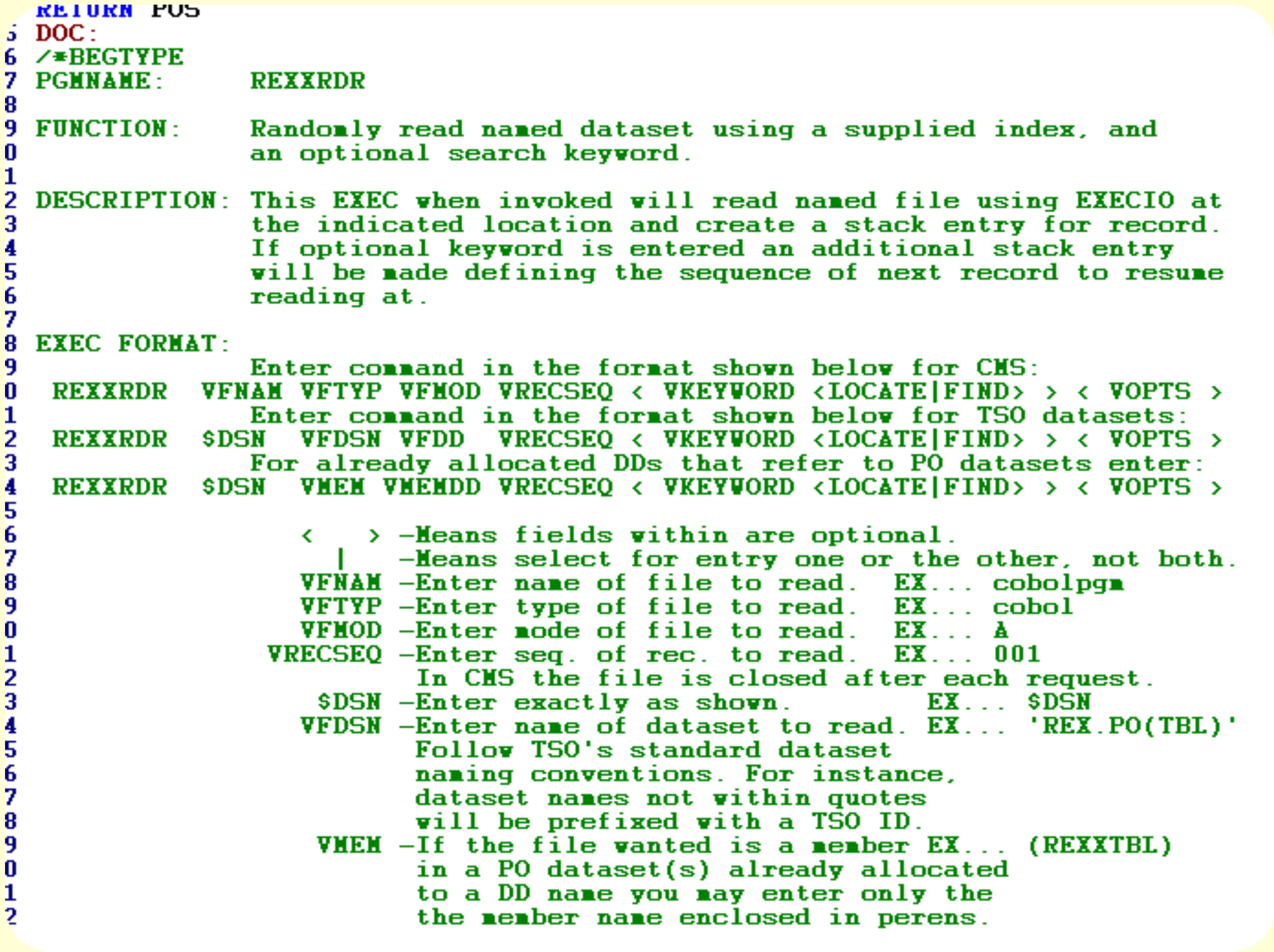
What components make it happen: REXXSCRN

```
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?