\User\Tools: Unix Tips with Scripts

Keven Miller
Valtek International

Introduction

We are using Oracle Applications version 10.4.2 on an NCR 3525.

Various interface problems with Oracle can be resolved with unix scripts. We have sales representatives that require their own copies of reports and output, but their current access is with a modem. We have provided kermit and zmodem transfer scripts registered as Oracle Tools.

At times our DBA wishes to know of problems as they start, not after they have been festering for 30-60 minutes. We use numeric pagers to inform us if the concurrent manager is down or backlogged. A pager could also tell our operators of a failed backup soon after failure. Numeric pagers can be accessed with the standard unix cu command.

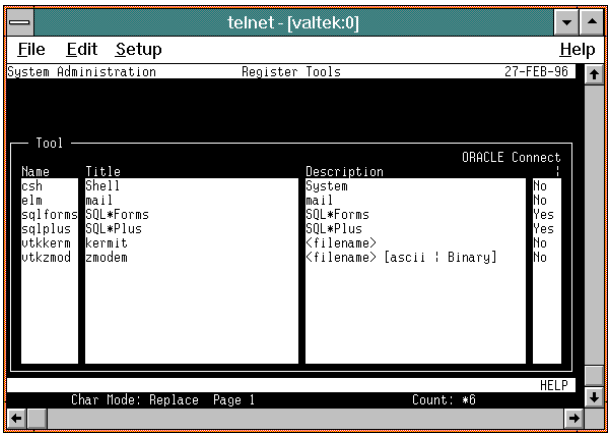
Oracle Forms 2.3 has a PRINT function that prints the current screen to the default system printer. With a script, users can now select the printer they desire.

Kermit and Zmodem

User tools must be registered or defined in Oracle Applications. From System Administrator, this is done in the Register Tools screen found at \ Navigate Install Tools. The following screen has these columns:

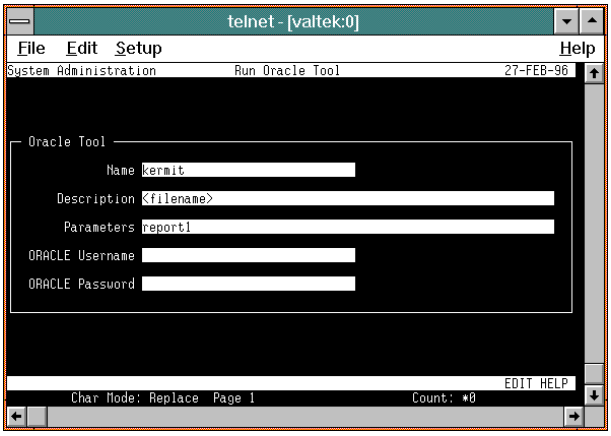
Name	unix command or script name (it must be in the execution path of the user).
Title	tool name for the user.
Description	we use to explain parameter usage.
Oracle Connect	Oracle passes password info to applications as in Sql*Plus.

In the following screen, I have defined kermit to have 1 parameter; the file to send. Zmodem has 2 parameters. The first is the file to send, and the second, which is optional, defines a transfer mode.



\ Navigate Install Tools

Users can copy report output from the request screen to a known filename in their directory. Then they can transfer the file with either kermit or zmodem from the Oracle Tool screen found at \ User Tools.



\ User Tools

The following scripts show how we use kermit and zmodem.

```
#---Kermit-------------------------------------------------------
#
#  Program: Kermit send shell
#  Created: 04 Dec 95 Keven Miller
#    Parms: $1 =file to send
#  Updated:
#
#----------------------------------------------------------------
```

```
  file=$1
  user=`logname`
  tty=`tty`
#-----------------------------------------------------------------
# send kermit
#
  kermit -s $file
#-----------------------------------------------------------------
# Notify caretaker
#
  wall -g kevenm << EOF
Kermit $file by $user $tty
EOF
#-----------------------------------------------------------------
  exit 0
#-----------------------------------------------------------------
#---Kermit--------------------------------------------------------


#---Zmodem--------------------------------------------------------
#
#  Program: Zmodem send shell
#  Created: 04 Dec 95 Keven Miller
#    Parms: $1 =file to send
#           $2 =A, a, ascii, B, b, binary
#              1st letter used; if not a, it defaults to b
#  Updated:
#
#-----------------------------------------------------------------
  file=$1
  user=`logname`
  tty=`tty`
#-----------------------------------------------------------------
# get mode
#
  m=`echo $2 | cut -c1-1`

  if [ "$m" = "A" -o "$m" = "a" ]; then
    m="a"
    mode="Ascii "
  else
    m="b"
    mode="Binary"
  fi
#-----------------------------------------------------------------
# send zmodem -y=delete existing remote file
#                    -a/b=ascii/binary transfer
#
  sz -y$m $file
#-----------------------------------------------------------------
# Notify caretaker
#
  wall -g kevenm << EOF
Zmodem $file $mode by $user $tty
EOF
#-----------------------------------------------------------------
  exit 0
#-----------------------------------------------------------------
#---Zmodem--------------------------------------------------------
```

Numeric Pager

To use numeric pagers, dial the pager number from a touch tone phone, wait for the request tone, then dial the number to be displayed on the pager and end with the '#' key. All this can be done with a modem dial-out. The following script uses the standard unix command cu to dial the pager. Before it can be used, two system files need to have lines appended to them for cu to call out, not have a modem connect, and yet think it was a successful call.

```
/etc/uucp/Devices
  #Valtek 7-FEB-96 Keven Miller
  pager term/00,M - 9600 pager \D
```

This line defines the name 'pager' to use device term/00 found in /dev. The ',M' means to ignore carrier detect on device open. Use 9600 baud, and use the entry in the Dialers file named 'pager'.

```
/etc/uucp/Dialers
  #Valtek 7-FEB-96 Keven Miller
  pager =,-, "" \MATS7=10DT\T\r\c  CARRIER \m\c
```

This line defines the dialer name 'pager' and describes how to communicate with the modem. The next four characters are user-to-modem mapping characters.

```
=       user;     wait for 2nd dialtone
,       modem; pause for a fixed time
-       user;     pause for a fixed time
,       modem; pause for a fixed time
```

What follows next is called a chat script. These are pairs of receive and send sequences to communicate with the modem.

```
    ""                      receive nothing
    \MATS7=10DT\T\r\c ignore carrier detect,
                            send char sequence with
                            phone (\T), and a return with
                            no newline (\r\c)
    CARRIER                 receive sequence like
                            'NO CARRIER'
    \m\c                    require carrier detect, no
                            newline
```

Most likely you will have to have your unix system manager edit these files.

After you have completed the setup, you need to place your numeric pager phone number into this script. The script has two parameters.

$1	the number to display (note that pagers display numbers as phone numbers)
$2	unit name

A unit is used to select a pager phone number. Here I have defined two units named barb and dave.

The phone number passed to the cu command is this:

```
9=$phone-----$display#
```

9	get outside line
=	wait for 2nd dialtone
$phone	pager number
-----	5 pauses to wait for answer and request tone
$display	number to display on pager
#	end pager number and call

```
#---Pager-----------------------------------------------------------
# Numeric Pager
# Created: 25-JAN-96 Keven Miller
#   Parms: $1 = display number for pager.
#             If <=0, then no page. Range is 1-(30 digit)
#             $2 = pager unit id
# Returns: 0 paged ok
#          1 display number <= 0
#          2 undefined unit name
#
# System files to edit
#    /etc/uucp/Devices
#    #Valtek 7-FEB-96 Keven Miller
#    pager term/00,M - 9600 pager \D
#
#    /etc/uucp/Dialers
#    #Valtek 7-FEB-96 Keven Miller
#    pager =,-, "" \MATS7=10DT\T\r\c  CARRIER \m\c
#
# Updated:
#--------------------------------------------------------------------
rtv=0
display=$1
if [ $display -le 0 ]; then
  rtv=1
fi
#--------------------------------------------------------------------
unit=`echo "$2" | tr "[A-Z]" "[a-z]"`
if [ "$unit" = "barb" ]; then
  phone=5551001       #Barb's pager
elif [ "$unit" = "dave" ]; then
  phone=5551002       #Dave's pager
else
  rtv=2
  phone="Unknown"
fi
#--------------------------------------------------------------------
echo ""
case "$rtv" in

  0 )
    echo "Pager: $unit $phone $display"
    echo "--- Start: `date +%T` ---"
# start cu using device pager, and phone number
    cu -c pager 9=$phone-----$display#
    echo "---  Stop: `date +%T` ---"
    ;;

  1 )
    echo "Pager <display-number> <person>"
    echo "--- Display number must be > 0 ---"
    ;;

  2 )
    echo "Pager <display-number> <person>"
    echo "--- Unknown Paging unit ---"
    ;;
esac

exit $rtv
#--------------------------------------------------------------------
#---Pager-----------------------------------------------------------
```

Oracle Forms 2.3 PRINT

When you press PRINT from your form, Oracle needs a method to send the screen to your system printer. I found in our Oracle7 Server for Unix Administrator's Reference Guide that the default method uses lpr. This can be changed with 2 environment settings

ORACLE_LPPROG	defines the command or script to use to print.
ORACLE_LPARGS	defines constant parameters for the command or script.

Next is the tricky part. Running a script for printing is the same as calling

```
#HOST 'script parms' NOSCREEN
```

in a form trigger.

Oracle runs its applications with the tty in raw, noecho mode. Scripts can easily echo text to the screen. To get input from the user, however, you will need to set the tty back to a usable state, ie: cooked mode and echo.

The following script saves the current stty setting and resets it to a friendly state. Then using VT100 escape sequences, it asks the user for a printer on the standard Oracle message line. After that it prints the file that Oracle placed the screen image into, or tells the user what error the print spooler returned.

```
#---OraPrint-----------------------------------------------------
#!/bin/sh
# Created: 01-Feb-96 Keven Miller
#   Parms: $1 = file to print
# Updated:
#----------------------------------------------------------------
file=$1   #Oracle passes 1 parm as file,
          #(default is form.lis)

#VT/ansi terminal escape codes for
up="⇦[A          # cursor up (esc)[A
inverse="⇦[7m" # inverse text (esc)[7m
ptron="⇦[5i"     # vt100 printer on (esc)[5i
ptroff="⇦[4i"    # vt100 printer off (esc)[4i
emptyline="
" #70 blanks
#----------------------------------------------------------------
rtv=0
if [ ! -r $file ]; then
  msg="Cannot read file: $file"
else

#$PRINTER is user default for lpr
#$LPDEST  is user default for lp
#lpstat -d returns system default printer

  dev=$PRINTER
  if [ -z "$dev" ]; then
    dev=$LPDEST
  fi
  if [ -z "$dev" ]; then
    dev=`lpstat -d | cut -f5 -d: | tr -d " "`
  fi

#place cursor on msgline, blank it out,
# then ask for printer
  echo "$up\r$inverse$emptyline\rOutput printer
[$dev]? \c"
#----------------------------------------------------------------
#Oracle APPS puts tty in raw mode (and other settings)
#need to save current settings, reset to sane setting
#and get user input. Then reset back to oracle settings
#get current oracle stty settings
  oldstty=`stty -g`
#enable std input settings
  stty icanon echo icrnl opost onlcr >/dev/null

  read prn

  if [ -z "$prn" ]; then
    prn=$dev
  else
    prn=`echo "$prn" | tr "[A-Z]" "[a-z]"`
    if [ "$prn" = "y" ]; then
      prn=$dev
    fi
  fi

  if [ "$prn" = "0" -o "$prn" = "n" ]; then
    msg="NO Print"          #allow user to cancel
  elif [ "$prn" = "1" -o "$prn" = "pc" ]; then
    echo "$ptron\c"         #send to PC local printer
    cat $file
    echo "$ptroff\c"
    msg="Printing $file to PC Printer"
  else
#use lp, trap err msg
    msg=`lp -d $prn $file 2>&1 >/dev/null`
    rtv=$?

    if [ $rtv = 0 ]; then
      msg="Printing $file to $prn"
    else
#chg multi-line msg to 1 line
      msg=`echo "$msg" | tr -ds "\012" " "`
#cut out leadin 'UX:'
      msg=`echo "$msg" | cut -d: -f2-4`
    fi
  fi

  stty $oldstty >/dev/null   #reset to oracle stty settings
fi
#----------------------------------------------------------------
# cursor to msgline, blank it, display error/print msg
echo "$up\r$inverse$emptyline\r$msg\c"
sleep 2
# Blank msgline, because Oracle does not know that we
# wrote on it
echo "\r$inverse$emptyline\c"
#----------------------------------------------------------------
# notify caretaker
# if error or if user's default is not the selected printer
#
if [ $rtv != 0 -o "$dev" != "$prn" ]; then
  wall -g kevenm << EOF
```

```
OraScrn $LOGNAME($dev): $prn=$rtv $file $msg
EOF
fi
#-------------------------------------------------------------------
#---OraPrint--------------------------------------------------------
```

Conclusions

These may seem to be trivial problems to some, but users are always ready to compliment development teams for anything that resolves a current problem. These were simple ideas that can be programmed quickly for fast results!

For an example of using the pager script, see the paper titled “Managing the Concurrent Manager (or How to Herd Cats)” by Barbara Matthews.

About the Author

Keven Miller
Programmer/Application Development
Valtek International
INET: kevenm@valtek.com
CIS: 70142,3647