```
*****************************************************************************
install.doc - Vancouver Utilities Install Guide for Unix/Linux [libinstall]
*****************************************************************************
```

** Installation Guide for Unix/Linux - CONTENTS **

** Installation Guide - CONTENTS (continued) **

** A0. Vancouver Utilities - Installation Guide **

The Vancouver Utility programs are encoded with a unique registration#
to aid in the prevention and detection of unauthorized copying.

A generous amount of free telephone support is included in the purchase
price (see more details in 'uvprices.doc'). User comments are encouraged
and any suggestions may be incorporated into future updates.

Owen Townsend, UV Software, 4667 Hoskins Rd., North Vancouver BC, V7K2R3
Tel: 604-980-5434 Fax: 604-980-5404
Email: owen@uvsoftware.ca Web: https://www.uvsoftware.ca

** Documentation to Install Vancouver Utilities **

The Documentation to Install the Vancouver Utilities is available in 3 formats:

1. HTML - 'www.uvsoftware.ca/install.htm'

2. PDF - 'www.uvsoftware.ca/install.pdf'

3. TEXT - after downloading & expanding the Vancouver Utilities zip/tar file
- you can access the text version as follows

#3. vi /home/uvadm/doc/install.doc

All documentation is created as text files with the vi editor & converted
to HTML using the uvcopy utility. The text version is also converted to .pdf
which you could download & print with windows or Linux open office.

After installing the Vancouver Utilities, you can print any documentation file
in /home/uvadm/doc/... with the 'uvlist' utility which converts text to PCL5.
You need a PCL5 compatible printer & you need to configure the printer in Linux
& export the printer name in the common_profile or bash_profile as follows

export UVLPDEST=-dPRINTERNAME <-- about line 95 in common_profile_uv

Then you can run the 'uvlp13D' script (calls uvlist to print 13 cpi Duplex)

#4. uvlp13D /home/uvadm/doc/install.doc

```
*****************************************************************************
A1.          install.doc - Installation Overview & Recommendations
*****************************************************************************

                  ** 3 versions of Vancouver Utilities **

Class A - complete package including mainframe conversion tools
        - includes source code for compile on unix/linux/windows
        - distribution has precompiled binaries for Linux & Windows

Class B - Vancouver Utilities without mainframe conversion tools
        - includes source code for compile on unix/linux/windows
        - distribution has precompiled binaries for Linux & Windows

Class C - Vancouver Utilities without mainframe conversion tools
          and without source code
        - executables only, intended for windows

                      ** 2 Distribution Files **

uvadm.tar.gz - tar archive for Unix/Linux (relevant to AIX w/o zip utility)
uvadm.zip    - zip archive for Linux &/or Windows (same contents as uvadm.tar)

These are alternative formats, but the contents are the same. You use whichever
1 is appropriate for your system. Both may hold 3 sets of binaries as follows:

                   ** subdirs of pre-compiled binaries **

bin       - binaries compiled on Red Hat or ubuntu linux
binDOS    - '.exe's  compiled by minGW on Windows

                      ** Installation Guides **

'install.doc'    - Installation on Unix/Linux systems
                 - source code is provided for compiling on unix/linux
                 - But binaries are included for the Linux standard

'WindowsDOS.doc' - Installation on native Windows without SFU
                 - intended to run in a command window
                 - 'executables only', .exe binaries supplied in 'binDOS'
```

```
*****************************************************************************
A2.              Vancouver Utilities Installation Guide
*****************************************************************************

The media includes directories for source, documentation, installation
scripts, test files, library archives, etc. After the tar extract or unzip,
the home directory should contain the following sub-directories:

                ** sub-directories in uvadm after restore **

/home/uvadm          <-- usual location of uvadm
/opt/uvsw/uvadm      <-- alternate location of uvadm
:-----batDOS           <-- BAT files for VU on native Windows
:-----binDOS           <-- binaries for VU on native Windows (cc by MinGW)
:-----bin              <-- binaries (uvcopy,uvsort,etc) distros are RedHat Linux
:-----ctl                - control files for various purposes
:-----dat1               - test data files
:-----demo             <-- demo files for UVdemos tutorials
:     :------...         - dat1/2,jcl2/3,ftps,pf,sf,sqls,tmp
:-----doc                - Vancouver Utilities documentation (text)
:-----dochtml          <-- documentation in HTML (uploaded to www.uvsoftware.ca)
:-----env                - profiles for Unix/Linux, SFU, Cygwin,& Uwin
:-----hdr                - hdr files for C compiles
:-----htmlcode           - HTML Index files for doc/dochtml
:-----lib                - libraries for C compiles (subfunctions,DISAM,etc)
:-----mf               <-- MainFrame derived test/demo files
:     :------...         - fewer/shorter/simpler testfiles than in mvstest/...
:-----mvstest          <-- test/demos for MVS JCL/COBOL mainframe conversions
:     :------...         - many subdirs omitted, see 'JCLcnv1demo.doc#Part_3'
:-----pf               <-- Parameter Files for uvcopy
:     :-----adm          - administrative jobs
:     :-----demo         - demo jobs
:     :-----IBM          - IBM mainframe conversion jobs
:     :-----util         - utility jobs
:-----sf               <-- Script Files
:     :-----adm          - administrative scripts
:     :-----demo         - demo scripts
:     :-----IBM          - IBM mainframe conversion scripts
:     :-----util         - utility scripts
:-----sfun             <-- ksh functions (used in converted JCL/scripts)
:-----src              <-- Vancouver Utilities C source code
:-----srcf               - C source for various sub-functions
:-----tbls               - miscellaneous tables
:-----tf                 - test files for various examples in doc
:-----tmp                - tmp subdir (test/demo outputs)
:-----vsetest          <-- test/demos for VSE JCL/COBOL mainframe conversions
:     :------...         - many subdirs omitted, see 'VSEJCL.doc'
```

```
*****************************************************************************
A3.              Vancouver Utilities Installation Guide
*****************************************************************************

Here is a little more info about some of the more important sub-directories
included in the Vancouver utility distribution.

bin       - directory to receive the executables from compiles & links
          - uvcopy, uvsort, uvqrpg, uvlist, uvcp, uvhd, uvhdcob, jclunix5
          - distributed with binaries

batDOS    - scripts (batch files) for the Windows/DOS Vancouver Utilties

binDOS    - executables compiled with 'MinGW' (Minimalist GNU for Windows)
          - to run in the DOS window under Windows
          - unix/linux users may may transfer to Windows PCs
            (uvhd,uvlist,uvsort,uvcp,uvcopy,uvqrpg)

ctl       - conversion tables, etc (for data file & program conversions)

dat1      - test data files for various pre-programmed conversion jobs
          - fixed length mainframe type files (vs text files in subdir tf)

doc       - documentation for Vancouver Utilities (in text format)

dochtml   - documentation in HTML, for upload to 'https://www.uvsoftware.ca'
          - converted to HTML by uvcopy from doc/... text documentation

env       - environment scripts & profiles for Unix/Linux, SFU, Cygwin, Uwin

hdr       - header library for the C programs
lib       - object code libraries (archives) for the uv subfunctions
            (srcf) & for the D-ISAM Indexed file support.

mvstest   - files to demo mainframe conversions 'jclcnv1demo.htm'

pf        - pre-programmed jobs (parameter files) for uvcopy applications
          - over 500 useful jobs (many suggested by end users)

sf        - scripts to compile programs, subfunctions, etc
          - also many end user scripts (see scripts1.doc in vol 1)

sfun      - Korn shell 'functions' (exportgen0/1, jobset, logmsg, etc)
          - mostly for running scripts converted from Mainframe JCL

src       - C source programs (uvcopy,uvsort,uvcp,uvlist,uvhd,etc)
          - script ccuvall compiles src/... into bin/...
srcf      - C source for subfunctions compiled separately
            & archived to lib/uvlib64.a (see scripts ccuvf & ccuvfa)

tf        - test data files for uvcopy & the pre-programmed jobs

vsetest   - test files to demo mainframe conversions (see 'VSEJCL.doc')
```

****
A4. Vancouver Utilities Installation Guide
****

** Installing Vancouver Utilities - Summary **

1. Setup user account 'uvadm' to serve as Vancouver Utilities administrator.
   Usual home directory is /home/uvadm, but could be elsewhere if you change the
   environmental variable 'export UV=/home/uvadm' in the common_profile.

2. We have suggested setting up group 'apps' for uvadm & other users who
   will be working with Vancouver Utilities.

3. Download the Vancouver Utilities from the UV Software website into the uvadm
   home directory,& unzip or untar (from archives uvadm.zip or uvadm.tar.gz).

4. You may need to modify the profiles as required for your site.
   We recommend setting up user 'appsadm' to store your modified versions
   of the profiles, so you do not lose them if you install a new version
   of Vancouver Utilities.

5. Setup user 'appsadm' & copy /home/uvadm/env/* to /home/appsadm/env
   Then modify .profiles or .bash_profiles to call the common_profile
   from /home/appsadm/env/common_profile. See short versions of the
   bash_profile & common_profile listed on the following pages.

6. The common_profile gives you (the system administrator) 1 place to control
   program & data-file paths for all users on your system.

7. You should rename the supplied common_profile_uv to common_profile_xxx
   where 'xxx' identifies your organization (company ID).

8. Setting up 'appsadm' (separate from uvadm) saves your customized version
   of the common_profile from being lost or over-written when you install
   new versions of 'uvadm'. All bash_profiles (.bash_profile or .profile)
   should call the comon_profile from /home/appsadm/env/common_profile
   (not from /home/uvadm/env/common_profile).

9. The C source utility programs are stored in $UV/src & are compiled
   into $UV/bin. You may need to recompile the C source programs for
   some versions of Linux (see pages 'C1' - C7).

** setup 'appsadm' - Applications Administrator **

I strongly recommend setting up a site administrator account 'appsadm'
(separate from uvadm) to hold various profiles & scripts common to all
users of the Vancouver Utilities.

This will save duplication of effort in setting up other users of Vancouver
Utilities because these profiles & scripts will be common to all users.
This makes it easier to make changes, add new users, etc.

You will also find APPSADM useful for establishing profiles, scripts,&
environmental variables for your other application packages as well as
Vancouver Utilities.

** 'profiles' provided in /home/uvadm/env **

```
/home/uvadm/env          <-- profiles provided here
:-----bash_profile_uv      - copy/rename to .profile (ksh) or .bash_profile (bash)
:                          - defines RUNLIBS/RUNDATA for programmers & operators
:-----common_profile_uv    - common profile (called by bash_profile)
:                            defines PATH's etc using $RUNLIBS/$RUNDATA
:
/home/appsadm/env        <-- setup user 'appsadm' & copy from /home/uvadm/env/*
:-----bash_profile_xxx     - customize & copy to homedirs .profile or .bash_profile
:-----common_profile_xxx   - common profile (called by bash_profile)
```

You should setup an application administrator userid 'appsadm',
copy /home/uvadm/env/* to /home/appsadm/env,& customize profiles
there depending on the locations of their libraries & data.
Do NOT customize profiles in /home/uvadm/env/... because they would be
overwritten when a new version of Vancouver Utilities is installed.

We recommend the concept of 'stub' & 'common' profiles. The shell profile in
each user's homedir is a 'stub' that calls the 'common_profile'
which is stored in /home/appsadm/env/...

1. The supplied 'bash_profile_uv' is copied to $APPSADM/env/ & renamed
   'bash_profile_xxx' (your master version vs UV Software supplied version)
   Then copy to user homedirs, renaming as '.bash_profile' for bash shell
   - or '.profile' for Korn shell.
   - defines RUNLIBS as testlibs/prodlibs for programmers/operators
   - defines RUNDATA as testdata/proddata for programmers/operators

2. 'common_profile' defines the 'PATH's using $RUNLIBS,$COBDIR,$UV,etc
   For example: export PATH=$PATH:$RUNLIBS/jcls (converted JCL/scripts).
   Defines software superdirs (uvadm, COBDIR, ORACLE_BASE, ORACLE_HOME, etc)

3. '$RUNDATA' determines data-file locations indirectly as follows:
   $RUNDATA defines the superdir housing all data-files. All JCL/scripts call a
   common function 'jobset51' which changes directory to $RUNDATA (cd $RUNDATA).
   The JCL converter inserts jobset51 at the begining of all converted
   JCL/scripts and then addresses all data files relative to $RUNDATA.

Note that stub profiles must call 'common_profile' using '.'
(dot execution), which means the 'export's made in the common_profile
will still be effective on return to the users profile.

This system is a big advantage for any site with multiple users, it means the
sysadmin can update common_profile once in 1 place & those changes are
effective for all users.

See stripped-down versions of the stub & common profiles listed below:

** bash_profile_uv or bash_profile_uv (same contents) **

```
# bash_profile_uv - bash_profile for Vancouver Utilities
# .bash_profile <-- must be copied/renamed to '.bash_profile' in your homedir
#                 - a 'stub_profile' that calls a 'common_profile'
#                 - by Owen Townsend, update Dec 2020
#
# bash_profile_ABC - users should copy/rename their version of bash_profile
#                    store their master copy in /home/appsadm/env/...
#                    copy to user homedirs, renaming .profile or .bash_profile
# common_profile_ABC - copy/rename their common_profile to /home/appsadm/env/...
#                      called by their bash_profiles from $APPSADM/env/...
#
# bash_profile & common_profile - distributed in $UV/env/...
# - copy to $APPSADM/env/... (/home/appsadm/env/...) & modify for your site
# - do not modify profiles in $UV because new versions of uvadm would overwrite
# - see bash/common profile listings at 'uvsoftware.ca/install.htm#A6' & A7
#
#           ** define RUNLIBS/RUNDATA/CNVDATA before call to common_profile **
#
# bash_profile defines RUNLIBS/RUNDATA/CNVDATA for common_profile to define PATHs to LIBS & DA
#  - Modify definitions below depending on self-training or Conversion/Production
#
# Mainframe JCL/COBOL/DATA Migrations - libs/data in homedir for testing/training
export RUNLIBS=$HOME/testlibs1 RUNDATA=$HOME/testdata1 CNVDATA=$HOME/cnvdata1
#=============================================================================
#
# Alternante Mainframe JCL/COBOL/DATA conversion/production in separate file systems
#   export RUNLIBS=/p1/apps/ABClibs1 RUNDATA=/p2/apps/ABCdata1 CNVDATA=/p3/apps/ABCcnvdata1
#   ========================================================================================
#   - better to use larger file systems for production (p1,p2,p3/apps/ABC... vs $HOME)
#   - appended digit '1' for future possible alternates ...libs2/...data2/...cnvdata2,etc
#   - could have multi-companies/systems on same machine, example for ABC company
#   - RUNDATA could be defined differently for different programmers for testing
#   - see 'uvsoftware.ca/jclcnv1demo.htm#1B3' RUNLIBS/RUNDATA defines for migrations
#
# Vancouver Utilities Demos/Tutorials & QuizGame for self training Linux Command Line Tools
export UVDEMOS=$HOME/demo QUIZFILES=$HOME/demo/quiz2f QF=$QUIZFILES
#=================================================================
#  - the $UV/demo/... directory can be copied to user's homedir for self-training
#    Linux Command Line Tools, vi/vim, C programming, uvcopy, & play quizgame (quiz2c)
#  - $HOME/demo/... already setup for guest login users (no need to download/install VU)
#    if you have downloaded/installed VU --> cp -r $UV/demo to $HOME/demo <--
#  - see demo files at 'uvsoftware.ca/uvdemos.htm#1B5'
#
#                     ** aliases for easy navigation **
#
# The following aliases are defined in the common_profile & in .bashrc for easy navigation
#  - quick changes between the various directories which could have long path names
#
#  alias cdc='cd $CNVDATA'   <-- cd to data conversion super directory
#  alias cdd='cd $RUNDATA'   <-- cd to data testing/production super directory
#  alias cdl='cd $RUNLIBS'   <-- cd to JCL/COBOL/script conversion/testing/production librarie
#  alias cdm='cd $HOME/demo' <-- cd to self-training Linux Tools, C programming, games, uvcopy
#
#
```

```
#             ** 'profile_uv' (GUI logins) vs 'bash_profile_uv' (text logins) **
#
# .bash_profile executed by text based command line logins (ctl-alt-functions F3-6 on Ubuntu)
# .profile executed by GUI logins on Ubuntu (default .profile would not call our common_profil
# If you want GUI logins to call our common_profile & setup PATHs to utilities (as above)
# - could copy $APPSADM/env/bash_profile_uv to .profile in home-dirs of GUI logins
# - we also provide *THIS* $APPSADM/env/profile_uv for copy to .profile of GUI logins
# - also see $APPSADM/env/Xresources_uv to be copied to homedirs of GUI logins
#
export APPSADM=/home/appsadm #Jan2020 APPSADM def in .profile_bash (for WSL Windows Subsystem
# CALLER=$(cat /proc/$PPID/comm)
echo "Executing--> .bash_profile (copied/renamed from \$APPSADM/env/bash_profile_uv)"
echo " - Vancouver Utilities bash_profile in login homedir, will call common_profile"
echo " - LOGNAME=$LOGNAME HOME=$HOME PWD=$PWD APPSADM=$APPSADM"
echo "Calling--> . $APPSADM/env/common_profile_uv"
#
. /home/appsadm/env/common_profile_uv   # common_profile called from /home/appsadm/env/...
#=====================================   # - NOT from /home/uvadm/env/...
# - must setup appsadm to store common_profile, so not lost when uvadm updated
# - see more at www.uvsoftware.ca/install.htm#A4
echo "HOSTNAME=$HOSTNAME LOGNAME=$LOGNAME APPSADM=$APPSADM UV=$UV"
echo "RUNLIBS=$RUNLIBS RUNDATA=$RUNDATA"
#
#       ** misc items that user may need to override common_profile defs **
# export TERM=linux       # TERM - modify depending on your terminal
# stty erase '^?'         # erase char - modify depending on your terminal
# stty intr '^C'          # interrupt ^C, (probably already default ?)
# export UVLPDEST="-dlp0" # default destination for uvlp(uvlist) scripts
#                         # change to a printer near you & un-comment
#
#                       ** user aliases, etc **
# alias l='ls -l'         # save keystrokes on very often used commands
# - see common_profile for several more aliases
# - add more here depending on user preferences
#
#                     ** TEST or PRODuction **
# export TESTPROD=P000   # P___ for PRODuction
export TESTPROD=T000     # T___ for TEST
# - PRODuction profiles TESTPROD=P*, developer TEST profiles TESTPROD=T*
# - JCL/scripts can test $TESTPROD to control various differences desired
# - used to determine if programmer 'T'esting or 'P'roduction
# - bytes 2,3,4 of P/T___ reserved for future use as required
#   if [[ "$TESTPROD" == P* ]]  <-- test only 1st byte for Test/Prod
#   if [[ "$TESTPROD" != T* ]]  <-- assume Production if not Test
#Note - Test/Prod code relevant only to mainframe migration JCL/scripts sites
#     - migration sites would move this code up prior to calling common_profile
#       so common_profile could modify PATH,etc depending on Test/Production
#
#
```

```
#                   ** Console Logging - optional **
# - uncomment 9 '##' lines below to activate console logging
# - must setup subdirs matching $LOGNAME in $LOGDIR/log1/...,log2/...,log3/...
#   (usually LOGDIR=$APPSADM in common_profile)
# - subdirs log1,log2,log3 hold logfiles for: current file, month, lastmonth
# - see details at www.uvsoftware.ca/admjobs.htm#Part_6
# - console logging for production operators to capture entire logon session
# - programmers can use the 'joblog1' script to capture log for 1 job at a time
 ## login1 || exit 2          # exit here if 2nd login
 ## if [[ ! -f .bashrc ]]; then
       ## echo "logging requires .bashrc (copy/rename from $AA/env/bashrc)"; read reply; exit
 ## logfixA $LOGNAME          # process log1 file to log2 (to allow read/print)
 ## echo "--> logview   <-- execute logview script to see prior console logs"
 ## echo "logging requires .bashrc/.kshrc with PS1='<@$HOST1:$LOGNAME:$PWD >'"
 ## echo "logging requires $LOGNAME subdirs in \$LOGDIR/log1 & log2"
 ## if [[ -d $LOGDIR/log1/$LOGNAME && ( -f .kshrc || -f .bashrc) ]]; then
 ##    echo "script $LOGDIR/log1/$LOGNAME/$(date +%y%m%d_%H%M%S)"
 ##    exec script $LOGDIR/log1/$LOGNAME/$(date +%y%m%d_%H%M%S)
 ## fi
# 'exec script' must be the last non-comment line in the profile
# 'script' disables aliases & umask 002 - put in .bashrc/.kshrc to be effective
# ==============================
# cp $APPSADM/env/bashrc .bashrc  # copy to your homedir restoring correct name
# ==============================
# After uvadm installed at $UV (/home/uvadm or /opt/uvsw/uvadm)
# - setup appsadm at $APPSADM (/home/appsadm or /opt/uvsw/appsadm)
# - copy $UV/env/* $APPSADM/env
# - modify $APPSADM/env/common_profile & bash_profile for your site
# - copy $APPSADM.env/bash_profile to user .profiles
# Then all user profiles call common_profile from $APPSADM/env/...
# to prevent loss of customized common_profile when new version uvadm installed
#
#             ** seldom used definitions stored at bopttom of profile **
# 5. Programmer testing/development projects, Example track US Election candidates
#    export RUNLIBS=$HOME/demo/election1 RUNDATA=$HOME/demo/election1 CNVDATA=$HOME/demo/elect
#    ================================================================================
#    - 1st demo election1/, then election2/, then election3, See uvsoftware.ca/uvdemos.htm#Par
#
```

```
                    ** common_profile_uv **

#*common_profile_uv      - common_profile for UV Software's offline machines
#                        - complete Vancouver Utilities
#                        - by Owen Townsend, Oct 2020
# common_profile_uv_web1 - for uvsoft webfaction sublogins uvsoft01-uvsoft19
#                        - complete Vancouver Utilities including bin/... & src/...
# common_profile_uv_web2 - for uvsoft webfaction sublogins uvsoft20-uvsoft29
#                        - empty bin/... & src/... for uvsoft20-29
#                        - bin PATH /home/uvsoft00/uvadm/bin/...
#Note - do not confuse 'common_profile_uv_web1/2' with 'common_profile_uv'
#     - clients will use 'common_profile_uv" when they install on their machines
# common_profile_ABC - users should copy/rename their version of common_profile
#                      & store their version in /home/appsadm/env/...
# bash_profile_uv  - also copy BASH/STUB profile to $APPSADM/env (/home/appsadm/env)
#                    for site customizations
# bash_profile_ABC - might append suffix to identify your customized versions
#                  - then copy to user homedirs & renamed as .profile or .bash_profile
#  - bash stub profile defines RUNLIBS/RUNDATA/CNVDATA for common_profile to modify PATHs
#  - users may define depending on their current project (migrations,testing,development)
#    export RUNLIBS=$HOME/testlibs RUNDATA=$HOME/testdata CNVDATA=$HOME/cnvdata
#    export RUNLIBS=$HOME/demo RUNDATA=$HOME/demo CNVDATA=$HOME/demo
#
# common_profile & bash_profile - distributed in $UV/env/... (usually /home/uvadm/env)
# - copy to $APPSADM/env/... (/home/appsadm/env/...) & modify for your site
# - do not modify profiles in $UV because new versions of uvadm would overwrite
# - see stub/common profile listings at 'uvsoftware.ca/install.htm#A6' & A7
#
#                  ** begin code for common_profile **
#
export UV=/home/uvadm           # UV homedir symbol used below
# export APPSADM=/home/appsadm  # APPSADM defined in .bash_profile
export LOGDIR=$APPSADM          # console logging subdirs log1,log2,log3
export FPATH=$APPSADM/sfun      # Function FPATH here for echo displaays
# CALLER=$(cat /proc/$PPID/comm)
echo " "
echo "Executing--> \$APPSADM/env/common_profile_uv (APPSADM=$APPSADM)"
echo "LOGNAME=$LOGNAME BASHPID=$BASHPID process=$$"
echo "HOME=$HOME APPSADM=$APPSADM UV=$UV FPATH=$FPATH"
#
# setup PATH for Vancouver Utilities programs & scripts (uvadm & appsadm)
# - append onto system PATH, using symbols defined above ($UV, $APPSADM, etc)
export PATH=$PATH:./sf:$HOME/sf:$HOME/bin:$APPSADM/sf:$APPSADM/bin
export PATH=$PATH:$UV/sf/adm:$UV/sf/demo:$UV/sf/util:$UV/sf/IBM
export PATH=$PATH:$UV/bin:$UV/help:./bin
#Note - Search Priority for scripts & programs in subdirs sf/... & bin/...
#     - #1=current directory, #2=$HOME, #3=$APPSADM, $4=$UV
#     - #5=$UV/sf subdirectoried to adm,demo,util,IBM
# setup PATH for JCL/scripts converted from mainframe Vancouver Utils
# - see www.uvsoftware.ca/jclcnv1demo.htm or www.uvsoftware.ca/vsejcl.htm
export PATH=$PATH:$RUNLIBS/sf:$RUNLIBS/jcls:$RUNLIBS/jts:$RUNLIBS/db2s/
#
#Note Re 'FPATH' - defined prior to 'echo's above for console display
# FPATH - defines directory of Korn shell functions (called by VU JCL/scripts)
#       - jobset51,exportgen0,exportgen1,jobend51,jobabend51,testcc,logmsg,etc
# export FPATH=$UV/sfun       # functions distributed in /home/uvadm/sfun/...
```

```
# export FPATH=$APPSADM/sfun  # copied to /home/appsadm/sfun/... for customization
# export FPATH=$RUNLIBS/sfun  # OR to $RUNLIBS for more flexibility if required
#
# setup 'PFPATH' for uvcopy & uvqrpg interpreter to find Parameter Files (jobs)
export PFPATH=./pf,$RUNLIBS/pf,$RUNLIBS/pfs
export PFPATH=$PFPATH,$HOME/pf,$UV/pf/adm,$UV/pf/demo,$UV/pf/util,$UV/pf/IBM
# - UV/pf/... follows RUNLIBS,APPSADM,HOME to allow user duplicate names
# - uvcopy accepts ',' delimiters as well as ':' in case of SFU on Windows
#
# define 'GDGCTL' location of gdgctl51I.dat/.idx
# - see doc at www.uvsoftware.ca/jclcnv4gdg.htm#5G1
if [[ -z "$GDGCTL" ]]; then export GDGCTL=$RUNDATA/ctl; fi #<-- set default
# - see GDG control file discussed at www.uvsoftware.ca/jclcnv4gdg.htm#5A2
#
# Define CTLMAPDIR for uvhdcob (display COBOL copybook fieldnames beside data fields)
export CTLMAPDIR=$HOME/mf/maps  #<-- uvhdc1 demos /home/uvadm/dat1/... & /home/uvadm/maps/...
# export CTLMAPDIR=$RUNLIBS/maps #<-- comment out above defaults this for uvhdc2
export COBMAPDIR=$RUNLIBS/maps  # for uvhdcob (display data with fieldnames)
export UVHDCOBROP=m45           # uvhdcob display 45 lines
# export UVHDROP=l64            # uvhd display 64 chars/line, Changed to 100 Dec2020
# export UVHDROP=l100           # uvhd display 100 chars/line, default in uvhd.c
#
# Indexed file extension controls for Vancouver Utilities
export ISDATEXT=".dat" # .dat/.idx Indexed files for uvsort,uvcopy,uvcp,etc
#                      # uvsort,etc expects .dat on data partition of ISAM files
#                      # COBOL equivalent is 'IDXNAMETYPE=2' in $EXTFH/extfh.cfg
#                      # ISDATEXT new way to control DISAM .dat extension Apr2010
export DISAMEXT="dat"  # DISAMEXT old way prior to Apr2010
#                      # - omit both or set null if you want NO .dat extension
#
# printer destinations for VU laser printing scripts
# - modify UVLPDEST to the network printer closest to you
export UVLPDEST="-dMS610USB"    # default dest for uvlp(uvlist) scripts
export UVLPOPTN="-onobanner"    # for unix/linux (SFU does not allow)
export UVHDPRINT=uvlp16         # script for uvhd 'i' immediate print command
export UVHDPWIDE=uvlp14L        # script for uvhd 'iprint' Landscape 100 chs/line
#-------------------------------------------------------------------------------
#
#                  ** TERM, erase, interupt, etc **
# stty erase '^?'           # erase char - modify depending on your terminal
#                         #  '^?' for linux/at386, '^H' for vt100,ansi,xterm
# stty -icrnl             # ensure CR x'0D' omitted & only LF x'0A' inserted
# stty intr '^C'          # interrupt ^C, (probably already default ?)
#
#                 ** UV Recommended items **
umask 002                     # permissions 775 dirs, 664 files
set -o ignoreeof              # disallow logoff via control D (use exit)
export HOSTNAME               # should already be set
export HOST=${HOSTNAME%%.*}   # extract 1st segment of $HOSTNAME
devtty=$(tty)                 # capture terminal (might add to PS1 prompt)
export tty=${devtty#/dev/}    # remove prefix /dev/
export PS1='<@$HOST:$LOGNAME:$tty:\D{%H%M}:$PWD> '
export EDITOR=vi              # for Korn shell history
export VISUAL=vi              # for Korn shell history
export HISTSIZE=3000;         # Korn shell history file size
export TD8=$(date +%Y%m%d)
export TD6=$(date +%y%m%d)
export EM=$HOME/em            # convenience for Owen (EMail directory)
export EMTD6=$HOME/em/$TD6    # convenience for Owen (EMail directory)
export AA=$APPSADM            # convenience since $APPSADM often used
```

```
export RL=$RUNLIBS            # convenience since $RUNLIBS often used
export RD=$RUNDATA            # convenience since $RUNDATA often used
export UVBAK=/home2/uvbak     # convenience for backup scripts & alias
export QJS=$RUNLIBS/qjs       # uvcopy jobs to replace QIKJOB
export EZTS=$RUNLIBS/ezts     # uvcopy jobs to replace EASYTRIEVE
export LOGMSGACK=n            # disable ACK option in logmsg2 in JCL/scripts
export LC_ALL=C               # disable UTF8 in errmsgs for C compiler
#                             # sort alpha ABC-Zabc-z vs AaBbCc-Zz
export SPELLDIR=$APPSADM/admin/spell  # vim spellfiles for all users
# - see script $UV/sf/adm/updatespellfile
# - sorts spellfile.adds into $SPELLDIR/en-utf-8.add & overwrites existing
export TBFR=/home7/owen/.thunderbird/3io8dews.default/ImapMail/mail.webfaction.com
# vi $TBFR/msgFilterRules.dat # easy access to Thunderbird filters (as root)
#
#                      ** aliases **
# alias commands to prompt for overwrite (highly recommended)
# - use option '-f' when you have many files  (rm -f tmp/*, etc)
alias rm='rm -i'              # confirm removes
alias mv='mv -i'              # confirm renames
alias cp='cp -i'              # confirm copy overwrites
alias l='ls -l'               # save keystrokes
alias lsd='ls -l $1 | grep ^d'   # list directories only
alias vi='vim'                # use vim for Linux
alias more='less'             # alias to 'less' if preferred
alias df='df -BM'             # df report sizes in MegaBytes
alias du='du -sBM'            # du Block size MegaBytes
alias grep='grep -nHd skip'   # ensure filename & line# on matching lines
alias uname='uname -a'        # ensure -a on uname (All info)
alias cdb='cd $UVBAK'         # UV Software backup superdir
alias cdl='cd $RUNLIBS'       # quick access to LIBS superdir
alias cdlc='cd $RUNLIBS/ctl'  # quick access to LIBS/control-files
alias cdd='cd $RUNDATA'       # DATA superdir
alias cdc='cd $CNVDATA'       # data CONVERSION superdir
alias cdk='cd $CMPDATA'       # data COMPARISON superdir
alias cdm='cd $HOME/demo'     # quick access to $HOME/demo/... self-training
alias cdqd='cd $HOME/demo/quiz2d' # see www.uvsoftware.ca/uvdemos1.htm#3H1
alias rmtmp='rm -f tmp*/*'    # often used to clear tmp/* tmp1/* tmp2/*
alias tree='tree -F'          # append '/' on dirs & '*' on executables
alias h='history'             # see all history commands
alias h40='history 40'        # see last 40 comamnds
alias ping='ping -c3'         # stop after 3 pings
alias dig='dig +short'        # find IP# for domain, dig -x find domain for IP#
alias cde='cd $EM'            # EMail directory
alias cdem='cd $EM/$TD6'      # EMail directory for today
alias mkem='mkdir $EM/$TD6; cd $EM/$TD6; touch ${TD6}a; vi ${TD6}a;'
# aliases - ineffective if console logging activated (in user stub profile)
#         - ifso, place aliases in .bashrc (or .kshrc, for ksh)
#-------------------------------------------------------------------------
# Verify that critical environmental variables have been defined
# (by stub_profile or this common_profile)
if [[ "$UV" = "" || "$APPSADM" = "" ]]; then
   echo "UV=$UV or APPSADM=$APPSADM not defined"
   echo "- enter to continue"; read $reply; fi
if [[ "$RUNLIBS" = "" || "$RUNDATA" = "" ]]; then
   echo "RUNLIBS=$RUNLIBS or RUNDATA=$RUNDATA not defined"
   echo "- enter to continue"; read $reply; fi
#-------------------------------------------------------------------------
# keep logins with last 50 logins (mostly for SSH logins on UVSW website)
test -d .login || mkdir .login  # make directory if not already present
CLIENT=${SSH_CLIENT%% *}        # get SSH_CLIENT (IP#)
```

```
echo "$LOGNAME $CLIENT $(date +%Y%m%d_%H%M)" >>.login/logins
tail -n 50 .login/logins >.login/loginstmp
mv -f .login/loginstmp .login/logins
#-------------------------------------------------------------------------
#        ** Micro Focus COBOL 2.2 update2 Eclipse on RHEL 7 June2015 **
# export COBDIR=/opt/microfocus/VisualCOBOL
# export JAVA_HOME=/usr/local/java32
# export PATH=$COBDIR/bin:$JAVA_HOME/bin:$PATH
# export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$COBDIR/lib:$JAVA_HOME/lib
# export COBCPY=$COBDIR/cpylib
# export CLASSPATH=$COBDIR/lib/mfcobol.jar:$COBDIR/lib/mfcobolrts.jar:$COBDIR/lib/mfsqljvm.jar
# export COBMODE=64
# export EXTFH=$UV/ctl/extfh.cfg   # file handler options IDXNAMETYPE=2 FILEMAXSIZE=8
# export CBLX=$RUNLIBS/cblx    # path for loading COBOL programs
#                       ** AIX COBOL **
# set default file type for JCL converter to AIX COBOL
# - other code at 'http://www.uvsoftware.ca/admjobs.htm#1C3' or $UV/env/archive/
# export COBRTOPT=FILESYS=QSAM
# converted JCL/scripts allow override via cft=XXX, for example:
# exportfile CUSTMAS data1/ar.custmas.master #cft=QSAM  <-- as generated
# exportfile CUSTMAS data1/ar.custmas.master cft=STL    <-- uncomment & change type
#                  ** Microsoft SQL Server **
# see www.uvsoftware.ca/sqldemo.htm#Part_6
# export PATH=$PATH:/opt/mssql-tools/bin
# export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/microsoft/msodbcsql/lib64
# export ODBCSQL="ODBC Driver 13 for SQL Server"
# export DATABASE=testdb   # $(DATABASE) used in table create & load scripts
# export ODBCINI=/etc/odbc.ini
# export ODBCSYSINI=/etc/  #<-- Directory with ODBC config (not File odbcinst.ini)
#            ** GNU COBOL testing Oct 2019 **
# export GCHOME=/home/gcobol
# export GCDIR=/home/gcobol/cob
#         ** optional for WSL (Windows Subsystem for Linux) **
#  Example#1 - export WINUSER using wslpath to get windows %USERPROFILE% (in C:\USERS\...)
## WINUSER=$(wslpath $(cmd.exe /C "echo %USERPROFILE%"))
## export WINUSER=$(echo $WINUSER | tr -d '\r')
## echo "WINUSER=$WINUSER"
# Example#2 - setup variables for Both Windows & Linux using 'WSLENV' (translates path differe
## C:\uvadm> set UVADM=C:\uvadm    <-- set variable in windows
##           set WSLENV=UVADM/p    <-- WSLENV sets up variable UVADM for Both Windows & Linux
#  C:\uvadm> echo %UVADM%          --> shows value "C:\uvadm" as expected
#  C:\uvadm> wsl                   <-- run WSL (or bash)
#  /mnt/c/uvadm>                   <-- now running Linux, prompt changed as per common_profile
#  /mnt/c/uvadm> echo $UVADM       <-- test, see if same variable now shows Linux path
#  /mnt/c/uvadm> /mnt/c/uvadm      --> proves $UVADM on Linux equivalent of %UVADM% on Windows
#            ** define directories for uvcopy mailx1 or mutt1 **
# export MAILDATA=maildata        #<-- input data files
# export MAILMSGS=mailmsgs        #<-- MSG files created for input to mailx utility
# export MAILSCRIPTS=mailscripts  #<-- scripts created to execute mailx utility
#            ** optional software **
# - Micro Focus COBOL, AIX COBOL, GNU COBOL, COBOL-IT
# - Microsoft SQL Server Oracle mySQL Morada RPG
# - WSL (Windows Subsystem for Linux)
#Note - code removed for seldom used items, Feb 2016
# - but prior version saved in $UV/env/archive/common_profile_uv_20160215
# - contains items that may need to be recovered, such as:
# - Micro Focus COBOL, COBOL-IT, Oracle, MySQL, Morada RPG, SQL Server
# - see listings at www.uvsoftware.ca/admjobs.htm#1C1 & 1C2,1C3,etc
#------------------------ end of common_profile --------------------------
```

** .vimrc - set options for vim **

```
" vimrc - setups for vi/vim, stored in /home/uvadm/env/vimrc
"       - copy to your homedir & rename as .vimrc
"
" set HighLighting for search matches & current cursor line
set hlsearch     "<-- HighLights search words (yellow)
set cursorline   "<-- Highlights cursor line (light blue)
"
" restore cusrsor to previous position when reopening a file
if has("autocmd")
  au BufReadPost * if line("'\"") > 0 && line("'\"") <= line("$")
    \| exe "normal! g'\"" | endif
endif
"
" increase yank max lines per buffer a-z, from default (50 lines) to 1000 lines
set viminfo='20,<1000   "note max 20 files & max 1000 lines
" -------------------- end of .vimrc ----------------------
```

** copy .vimrc to your $HOME **

It is highly recommeded to copy above .vimrc to your homedir. For visibility, '.vimrc' is named as 'vimrc' in both $UV/env/vimrc & $APPSADM/env/vimrc.

```
cp $APPSADM/env/vimrc $HOME/.vimrc
==========================================
```

- copy to your homedir, renaming to '.vimrc'

You will love using vim with $HOME/.vimrc - highlighting the cursor line & search words make vim so much nicer & easier to use.

```
************************************************************************
B1.                    Installing the Vancouver Utilities
************************************************************************

                     ** uvadm - Vancouver Utility homedir **

Here are the most relevant subdirs in /home/uvadm, emphasizing env/profiles
that must be copied to /home/uvadm/appsadm/env.

/home/uvadm          <-- usual location of uvadm
/opt/uvsw/uvadm      <-- alternate location of uvadm
:--bin                 <-- binaries (uvcopy,uvsort,etc) distros are RedHat Linux
:--ctl                   - control files for various purposes
:--dat1                  - test data files
:--doc                   - Vancouver Utilities documentation (text)
:--env                 <-- profiles for Unix/Linux users & administrators
:  :--bash_profile_uv    - defines RUNLIBS/RUNDATA for the common_profile
:                          copied to homedirs, renamed .profile or .bash_profile
:  :--common_profile_uv - common profile (called by bash_profile)
:                          defines PATH's etc using $RUNLIBS/$RUNDATA
:                        - env/common_profile,etc must be copied to /home/appsadm/
:--mvstest             <-- test/demos for MVS JCL/COBOL mainframe conversions
:  :--...                - many subdirs omitted, see 'jclcnv1demo.htm#Part_3'
:--pf                  <-- Parameter Files for uvcopy & uvqrpg
:--sf                  <-- Script Files
:--sfun                  - ksh functions used in converted JCL/scripts
:                        - jobset51,jobend51,exportgen0,exportgen1,logmsg1,etc
:                        - must be copied to /home/appsadm/
:--src                 <-- Vancouver Utilities C source code

                     ** appsadm - Application Administrator **

We recommend you setup a login/userid 'appsadm' to serve as the applications
administrator for the unix/linux site. The appsadm directory would hold
profiles, functions scripts, crontabs, log files, etc used in application admin.
Here are some suggested sub-directories:

/home/appsadm        <-- usual location of appsadm
/opt/uvsw/appsadm    <-- alternate location of appsadm
:----bin                    - binaries for site developed/modified programs
:UV--ctl                    - control files for converting JCL & COBOL & GDG files
:UV--env                  <-- profiles copied from /home/uvadm/env/...
:   :---bash_profile_xxx    - defines RUNLIBS/RUNDATA for the common_profile
:                           - copied to homedirs, renamed .profile or .bash_profile
:   :---common_profile_xxx - common profile (called by bash_profile)
:                             defines PATH's etc using $RUNLIBS/$RUNDATA
:----logs                   - console logs from nightly 'cron' scripts
:----pf                   <-- uvcopy jobs developed/modified by site admin
:----sf                   <-- shell scripts developed/modified by appsadm
:UV--sfun                 <-- functions for JCL/scripts (jobset51,exportgen0,etc)
:----src                    - source for any programs developed/modified by appsadm

Note - 'UV' marks directories that must be copied from uvadm to appsadm
     - other subdirs are optional & may remain empty initially
     - must copy profiles from uvadm/env to appsadm/env, because new versions
       of uvadm would overwrite any customized profiles in uvadm/env
     - customize common_profile in /home/appsadm/env/... depending on your site
```

B2. First Time Setup/Install Vancouver Utilities

If you already have an old version of uvadm, please skip to page 'B11' --->

** setup uvadm Vancouver Utilities admin user **

#1. login as 'root'

#2. groupadd apps <-- setup group 'apps', if not already setup
- or could be uvgrp, ITgrp, etc

Note - 'apps' is the group suggested for the group of programmers who will be working on a common set of files (JCL, COBOL, DATA)

#3. useradd -m -g apps -s /bin/bash uvadm <-- setup user 'uvadm'
- option -g specifies group 'apps'
- use option '-s' to specify login shell as 'bash'
- could specify '-s /bin/ksh' if preferred
- JCL/scripts code 1st line '#!/bin/ksh' because only the Korn shell has '$FPATH' to functions $APPSADM/sfun/... (jobset51,exportgen0,etc) & allows 'autoload' at beginning of JCL/scripts to declare functions that may be called within that script

#3a. useradd -m -g apps -s /bin/bash -d /opt/uvsoftware/uvadm uvadm
- could use '-d' to specify home dir (if other than default /home/uvadm)
- may install uvadm elsewhere, such as /opt/uvsoftware/uvadm, if you change variable 'UV' in common_profile to point to it

#4. passwd uvadm <-- setup password desired

#5. chmod 755 /home/uvadm <-- allow other users to copy files from uvadm/...
- required for many Vancouver Utility procedures

#6. exit (logout from root)

1. group 'apps' is suggested for the group of programmers who will be working on a common set of files (JCL, COBOL, DATA). These users must be in same group & have umask=002 in the profile so files will be created as 664 & directories as 775 to allow common updates

2. user 'appsadm' is suggested as an applications administrator. Use appsadm to hold customized files, such as env/common_profile_xxx, called by users bash_profiles (.profile or .bash_profile in their homedirs).

3. The common set of directories (updated by various programmers) must have permissions 775 (files 664) achieved by umask=002 in the common_profile.

4. You can protect non-common directories by changing permissions to 755. Programmers can do this for their $HOME directory & you the appsadm can do it for uvadm & appsadm.

5. When setting up directories for a group of programmers (assuming off line machine), it is better to ensure umask=002 for directories 775 & files 664, amd that they are all in the same group. Otherwise it is very frustrating to the programmers to have permissions problems at every turn.

B3. First Time Setup/Install Vancouver Utilities

** Download/Install Vancouver Utilities **

This assumes UV Software has supplied you with a userid/password to download 'uvadm.zip' from the UV Software web site.

```
#1. Login as 'uvadm' --> /home/uvadm      <-- usual location of uvadm
#1a. cd $UV  ----------> /opt/uvsw/uvadm  <-- alternate location uvadm

#2. ftp uvsoftware.ca    <-- FTP
    =================
#2a. user ----> uvsoft99  <-- userid could be uvsoft2-uvsoft99
#2b. passwd --> xxxxxxx   <-- password supplied by UV Software
#2c. binary
#2d. get uvadm.zip
#2e. bye

#3. unzip uvadm.zip       <-- unzip Vancouver Utilities zip archive
    ===============         - ensure logged in user 'uvadm' (not root)

#3a. gunzip uvadm.tar.gz  <-- OR - decompress gzip archive (AIX users)
     ===================         - ensure user 'uvadm' (not root)

#3b. tar xvf uvadm.tar    <-- extract files from tar archive (AIX users)
     =================      - AIX does not include 'zip' utility
```

** setup/modify uvadm profile for your site **

```
#1. Login 'uvadm' --> /home/uvadm      <-- login uvadm (if not already)
#1a. cd $UV  -------> /opt/uvsw/uvadm  <-- alternate location uvadm

#2. cp env/bash_profile_uv .bash_profile <-- copy rename for bash shell
    ====================================

#2a. cp env/bash_profile_uv .profile     <-- OR - rename for Korn shell
     ===============================

#3. vi .bash_profile    <-- could modify profile now, BUT recommend
    ================        after copy to appsadm, see next page -->

#4. vi env/common_profile_uv  <-- could modify profile now, BUT recommend
    ========================      after copy/rename to appsadm, next page -->

#5. exit               <-- logoff
    ====

#6. Login uvadm        <-- log back in to make profiles effective
    ===========
```

```
B4.                First Time Setup/Install Vancouver Utilities
-------------------------------------------------------------------------------

                          ** Create 'appsadm' **

#1. login as 'root'

#2. groupadd apps     <-- setup group 'apps', if not already setup
    =============       - already done when uvadm setup on page 'B2'

#3.  useradd -m -g apps -s /bin/bash appsadm   <-- setup user 'appsadm'
     ======================================

#4. passwd appsadm     <-- setup password desired
    ==============

#5. chmod 755 /home/appsadm <-- allow other users to copy files from appsadm/...
    ========================   - required for many Vancouver Utility procedures

#6. exit                <-- exit from root

                  ** setup/modify appsadm profiles, etc **

#1. login as 'appsadm' --> /home/appsadm      <-- usual location appsadm
#1a. cd $APPSADM  -------> /opt/uvsw/appsadm  <-- alternate location appsadm

#2. mkdir ctl env logs sfun tmp <-- setup subdirs required
    ============================

#3a. cp /home/uvadm/ctl/* ctl     <-- copy control files from uvadm to appsadm
     ========================
#3b. cp /home/uvadm/sfun/* sfun   <-- copy functions from uvadm to appsadm
     ==========================     - for JCL/scripts (jobset51,exportgen0,etc)
#3c. cp /home/uvadm/env/* env     <-- copy profiles from uvadm to appsadm
     ========================

#4a. mv env/bash_profile_uv env/bash_profile_xxx
     ========================================
     - rename UV Software's version to identify as your company's version

#4b. vi env/bash_profile_xxx     <-- modify bash_profile as required
     ========================       - see details on the next page -->

#5a. mv env/common_profile_uv env/common_profile_xxx
     ================================================
     - rename UV Software's version to identify as your company's version
     - chnage 'xxx' to whatever you wish to identify your organization

#5b. vi env/common_profile_xxx   <-- modify common_profile as required
     ==========================     - see details on the next page -->

#6. cp env/bash_profile_xxx .bash_profile  <-- copy/rename for bash
    ====================================      (.profile for ksh)

#6a. cp env/bash_profile_xxx /home/userxx/.bash_profile
     ================================================
     - copy your modified bash_profile to other user's homedirs now or later

#7. exit
```

** Modify /home/appsadm/env/common_profile **

On the previous page, we setup userid appsadm & copied the profiles from /home/uvadm/env/* to /home/appsadm/env/... We will now modify the profiles if & as required for your site. Please see listings of profiles starting at 'ADMjobs.doc#1C0' & discussions starting at 'ADMjobs.htm#1D1'.

Note that the common_profile_uv is copied, renamed, modified in,& called from /home/APPSADM/env/common_profile_xxx (not /home/UVADM/env/common_profile_uv). This avoids losing your site specific modifications when new versions of Vancouver Utilities are installed. We must logout & login again to ensure any profile modifications (performed on prior page) are activated.

```
#1. login 'appsadm' --> /home/appsadm      <-- usual location appsadm
#1a. cd $APPSADM  ----> /opt/uvsw/appsadm  <-- alternate location appsadm

#2. vi env/common_profile_xxx    <-- edit your common profile
    ==========================
```

#2a. Modify TERM & 'stty erase' character depending on user's terminal (distribution has TERM=linux & stty erase '^?')

```
export TERM=linux       # TERM - modify depending on your terminal
#================       # (vt100,xterm,at386,ansi,etc)
stty erase '^?'         # erase char - modify depending on your terminal
#==============         #  '^?' for linux/at386, '^H' for vt100,ansi,xterm
```

#2b. Change laser printer name defined by 'UVLPDEST' environmental variable used by the 'uvlist' utility & scripts 'uvlp12', etc.

```
export UVLPDEST="-dlp0" # change 'lp0' to site laser printer
========================
```

#2c. Change 'COBDIR', environmental variable defining where Micro Focus COBOL COBOL is installed on your system

```
export COBDIR=/opt/microfocus/cobol  <-- default install location
==================================    - change if your site different
```

#2d. other changes as dsired ???

** common_profile called by bash_profile's **

The common_profile should be called by all stub profiles (.bash_profile or .profile in homedirs of all users who wish to use Vancouver Utilities).

The bash_profile_uv supplied in /home/UVADM/env/bash_profile_uv calls common_profile_uv from /home/UVADM/env/common_profile_uv, but ONLY for uvadm. When bash_profile_uv is copied to /home/APPSADM/env/bash_profile_xxx, it should be changed to call from /home/APPSADM/env/comon_profile_xxx, BEFORE copying to user homedirs (renaming as .bash_profile or .profile depending shell bash/ksh). See these instructions on the next page.

B6. Installing the Vancouver Utilities

** modify /home/appsadm/env/bash_profile_xxx **

The bash_profile_xxx (copied/renamed as .bash_profile or .profile) in user homedirs calls the common_profile_xxx from /home/appsadm/env/common_profile, which has most of the profile code and is much more convenient to maintain in 1 place than having to update profiles in the various user homedirs. Please see listings of profiles starting at 'ADMjobs.doc#1C0' & changes starting at 'ADMjobs.htm#1D1'.

After copying bash_profile_uv to /home/appsadm/env/bash_profile_xxx, the most important change is to call common_profile_xxx from /home/apspadm/env/... rather than from /home/uvadm/env/... which is OK only for uvadm.

If you are performing JCL conversion, you would modify the RUNLIBS & RUNDATA definitions in the stub profile depending on your directory locations of JCL,COBOL,& DATA. Initial definitions are $HOME/testlibs & $HOME/testdata, good for self-training, allows users to copy demo JCL/COBOL/DATA to their homedirs & perform the test/demo conversions documented in 'jclcnv1demo.doc'.

#1. login 'appsadm' --> /home/appsadm <-- usual location appsadm
#1a. cd $APPSADM ----> /opt/uvsw/appsadm <-- alternate location appsadm

#2. vi env/bash_profile_xxx <-- modify 'stub' profile for your site
    ======================
    - Ensure the call to the common_profile is from $APPSADM

```
 . /home/appsadm/env/common_profile_xxx
#====================================
 . /opt/uvsw/appsadm/env/common_profile_xxx #<-- alternate location
#========================================
```

- Minimize other changes, since majority of profile code should be maintained in the common_profile. Any changes after the call to the common_profile would override the definitions in the common_profile. Various users may wish to add their favorite aliases, etc
- Some users might need to modify TERM & 'stty erase' character depending on their terminal (distribution has TERM=linux & stty erase '^?')
- Users can change UVLPDEST to define a laser printer near them. UVLPDEST is an environmental variable used by uvlist & scripts uvlp12,etc
- Modify RUNLIBS & RUNDATA as required. Initial $HOME/testlibs & $HOME/testdata for self-training. Change later to common locations for the real conversions.

#3. cp env/bash_profile_xxx .bash_profile <-- OR .profile for ksh vs bash
    ====================================
    - copy modified bash_profile to .bash_profile or .profile of user appsadm

#4. Append the bash_profile to homedirs of users who are going to use the
    Vancouver Utilities. Could just copy if user has not customized profiles.

#4a. cat env/bash_profile_xxx >>.bash_profile <-- copy/rename for bash shell
     ======================================
#4b. cat env/bash_profile_xxx >>.profile <-- copy/rename for Korn shell
     ================================

B7. Installing the Vancouver Utilities

** Compile Vancouver Utility C programs **

Vancouver Utility C program binaries supplied in $UV/bin/uvcopy,uvsort,etc were compiled on Ubuntu Linux & you may need to recompile on your machine/OS.
You can test by attempting to execute the binaries in $UV/bin/..., for example:

```
#1. Login uvadm --> /home/uvadm      <-- login uvadm (if not already)
    ============
#2. uvcopy    <-- attempt to execute supplied version (compiled on Ubuntu)
    ======
```

If OK, uvcopy will display version dates & quit (since no arguments given)
IF NOT OK, you get "cannot execute binary file Exec format error", And you need to recompile as shown on pages 'C1' - C3. For example on RHEL7:

```
ccuvall LNX H64 uvlib64.a disamRHEL7.a - compile on RedHat Enterprise Linux 7.xx
=====================================
```

** Recommended program additions **

You may need some additional programs depending on your intended uses of VU.

** ksh - Korn Shell **

Many of the supplied scripts in $UV/sf/adm,demo,util,IBM specify #!/bin/ksh (vs #!/bin/bash) especially relevant for JCL conversions, but also required for scripts using supplied functions such as exportgen0/exportgen1 (GDG files).
The Korn shell provides 'autoload' a powerful feature that allows you to store your functions in a separate directory specified by 'export FPATH=...'
The default in the common_profile is 'export FPATH=$APPSADM/sfun' which assumes you have copied $UV/sfun/* to $APPSADM/sfun/.

```
yum install ksh           <-- install ksh on RedHat Enterprise
===============

sudo apt-get install ksh   <-- install ksh on ubuntu
========================

cp $UV/lib/ksh93 /bin/ksh  <-- for Linux, you can copy/rename supplied version
=========================
```

B8. Installing Additional Utilities

** dcheck **

'dcheck' checks the validity of Indexed files compatible with Mucro Focus COBOL & the D-ISAM file handler used by the VU programs (uvcopy,uvsort,etc). Several versions are supplied in $UV/lib/dcheck/dcheckAIX,dcheckRHEL6,decheckRHEL7,etc Copy/rename the relevant version to the system bin/ or VU bin, for example:

```
cp $UV/lib/dcheck/dcheckRH7 $UV/bin/dcheck
```

** tree **

'tree' is a great tool to draw directory/file structures for documentation. For example see 'uvsoftware.ca/uvdemos2.htm#1A1'. You could also use for your docuementation.

```
yum install tree           <-- install ksh on RedHat Enterprise
```

```
sudo apt-get install tree  <-- install ksh on ubuntu
```

```
tree --version             <-- see version & authors
```

```
tree v1.7.0 (c) 1996 - 2014 by Steve Baker (ice@mama.indstate.edu)
Thomas Moore, Francesc Rocher, Florian Sesser, Kyosuke Tokoro
```

```
*************************************************************************
B9.               Installing the Vancouver Utilities
*************************************************************************

                     ** setup your user account **

Setup your user login account (& other users you know will be needed). We will
show as 'userxx' but you would proably use first names.

#1. login as 'root'

#2a. useradd -m -g apps -s /bin/bash userxx
     ====================================
     - use option '-s' to specify login shell as 'bash'
       OR use #2b. below for 'ksh'

#2b. useradd -m -g apps -s /bin/ksh userxx
     ====================================
     - use option '-s' to specify login shell as 'ksh' (Korn shell)

#3. passwd userxx           <-- setup password desired
    ==============

#4. chmod 755 /home/userxx  <-- allow copying between your user accounts
    ======================    - optional

#5. cp $UV/env/bash_profile_uv /home/userxx/.bash_profile
    ======================================================
    - copy supplied profile to homedir renaming to '.bash_profile'
    - calls the common profile from $APPSADM/env/common_profile_uv
    - see profiles listed above on pages '1B2' & '1B3'

           ** OR, Alternative #5a. below (vs #5. above) **

#5a. cp $APPSADM/env/bash_profile_xxx /home/userxx/.bash_profile
     ============================================================

Use #5a. if you have a customized version of the bash_profile saved in
$APPSADM/env/...  Customized versions of the bash_profile probably not required
for most sites, but all sites should use a customized version of the common_profile.
Add suffix to identify your company - commmon_profile_XXX (vs common_profile_uv).
```

```
*****************************************************************************
B10.         Install VU on Windows WSL (Windows Subsystem for Linux)
*****************************************************************************
```

As of 2019, you can install Vancouver Utiltities on Windows systems using WSL (Windows Subsystem for Linux), compile the VUs under Linux & execute the many Korn/Bash scripts supplied with Vancouver Utilities.

Prior installs of Vancouver Utilities on a windows systems used the native Windows .exe versions of Vancouver Utilities, distributed in binDOS/uvcopy.exe,uvsort.exe,etc and could not execute the many Korn & bash scripts supplied with Vaancouver Utilities.

Please see 'uvsoftware.ca/windowsdos.htm#D1' - D5.

D1 - Install WSL (Windows Subsystem for Linux) to run the linux version of
     Vancouver Utilities on a Windows system, 1st doc Feb 2020 Windows 1903
D2 - Installing Vancouver Utilities for WSL - Overview/Notes
D3 - Windows & Linux Directories relevant to VU profiles,programs,& scripts
D4 - Setup users for Linux versions of Vancouver Utilities on Windows WSL
D5- Setup profiles for Linux users of Vancouver Utilities on Windows WSL

```
*************************************************************************
B11.         UPGRADING to New Versions of the Vancouver Utilities
*************************************************************************

If this is your 1st version of the Vancouver Utilities package, you have just
performed initial install using procedures on pages B1-B6, and you can now
skip to page 'C1' (compiling the C programs) --->

             ** Preparations to Install Update Version **

If there is any doubt about the usefulness of the old uv version, the
following procedures will allow for retrieving user modified files from
the old uvadm directories into the new. It will be easier to upgrade to the
next version of this package if you do not make changes in the 'uvadm'
directory or add your own stuff into the uvadm directories.

Note - Method#1 (this page) works for all machines
     - #1 is mandatory for SUN Solaris because you cannot move homedirs
     - logins would still go to the original subdir (uvadm.old)
     - Method#2 (next page) works for most machines & is a little shorter

             ** method#1 - mandatory for Sun (OK on all machines) **

#1. login: root    - you need to be root, but ONLY for the 1st 7 steps below

#2.  cd /home             <-- change above uvadm (usual location /home/uvadm)
#2a. cd /opt/uvsw         <-- alternate location /opt/uvsw/uvadm

#3. rm -rf uvadm.old        - remove any existing uvadm.old (grandfather)
    ================

#4. cp -rp uvadm uvadm.old  - make backup copy of the uvadm directory
    ======================    'r' option copies all subdirs
                              'p' option preserves uvadm owner & permissions

#5. rm -rf uvadm/*          - clear all contents of uvadm directory
    ==============          - before reading new uvadm.tar or uvadm.zip
                              (on the next page under uvadm login)

#6a. cp -p uvadm.old/.bash_profile uvadm   (for Linux/bash)
     ==================================
              --- OR ---
#6a. cp -p uvadm.old/.profile uvadm        (for Unix/ksh)
     ==============================

#6b. cp -rp uvadm.old/env uvadm     <-- copy the common profiles (subdir)
     ==========================
     - copy existing uvadm profiles over to the now empty uvadm subdir
     - preserving any customizations made on prior version

#7. exit (from root)

Install the new utilities as directed on the page 'B13' --->
```

```
*************************************************************************
B12.         UPGRADING to New Versions of the Vancouver Utilities
*************************************************************************

               ** method#2 - most machines (except SUN) **

Note - Method#1 (prior page) is mandatory for SUN Solaris
     - because you cannot move home directories on SUN
     - logins would still go to the original subdir (uvadm.old)
     - Method#2 (this page) works for most machines & is a little shorter

              ** Preparations to Install Update Version **

#1. login: root    - you need to be root, but ONLY for the 1st 8 steps below

#2.  cd /home             <-- change above uvadm (usual location /home/uvadm)
#2a. cd /opt/uvsw         <-- alternate location /opt/uvsw/uvadm

#3. rm -rf uvadm.old        - remove any existing uvadm.old (grandfather)
    ================

#4. mv uvadm uvadm.old      - rename existing 'uvadm' as 'uvadm.old'
    ==================

#5. mkdir uvadm             - make new dir for new version of VU
    ===========
#5a. chmod 775 uvadm        - set desired permissions
     ===============
#5b. chown uvadm uvadm      - ensure owned by user 'uvadm'
     =================
#5c. chown ???? uvadm:apps  - ensure group matches desired group ????
     ====================

Note - group ID should match the programmers who will be working with VU
     - now copy existing uvadm profiles over to the now empty uvadm subdir
     - preserving any customizations made on prior version

#6. cp -p uvadm.old/.bash_profile uvadm  <-- copy existing profile to new
    ===================================
              --- OR ---
#6a. cp -p uvadm.old/.profile uvadm      <-- OR - copy .profile if Korn shell
     ==============================

#7. ensure .bash_profile calls common_profile from $APPSADM/env/... (not $UV/env/...)

#8. exit (from root)

Install the new utilities as directed on the next page --->

Step #4 above allows you to retrieve any modified/customized pf/sf files at any
time from uvadm.old into the new uvadm directory. This also allows you to
quickly revert to your former versions if you encounter a problem.
```

B13. UPGRADING to New Versions of the Vancouver Utilities

** Download/Install new version of Vancouver Utilities **

This assumes UV Software has supplied you with a userid/password to download 'uvadm.zip' from the UV Software web site.

```
#1. Login 'uvadm' --> /home/uvadm      <-- login uvadm (if not already)
#1a. cd $UV  -------> /opt/uvsw/uvadm  <-- alternate location uvadm

#2. ftp uvsoftware.ca    <-- FTP
    =================
#2a. user ----> uvsoft99  <-- userid could be uvsoft2-uvsoft99
#2b. passwd --> xxxxxxx   <-- password supplied by UV Software
#2c. binary
#2d. get uvadm.zip
#2e. bye
```

Note - Before unzipping the uvadm.zip archive, ensure:
- logged in as uvadm & in the uvadm homedir $UV (usually /home/uvadm)
- $UV homedir is empty (old contents previously saved in /home/uvadm.old)
- empty except for new uvadm.zip just downloaded

```
#3. unzip uvadm.zip       <-- unzip Vancouver Utilities zip archive
    ===============         - ensure logged in user 'uvadm' (not root)

       --- OR ---

#3a. gunzip uvadm.tar.gz  <-- OR - decompress gzip archive (AIX users)
     ====================        - ensure user 'uvadm' (not root)
#3b. tar xvf uvadm.tar    <-- extract files from tar archive (AIX users)
     =================      - AIX does not include 'zip' utility
```

** Updating profiles in $APPSADM/env/ **

1. The initial install instructions on pages B1-B6 copied the profiles from /home/uvadm/env/... to /home/appsadm/env/... and any site-specific changes were made in /home/appsadm/env/...

2. All user stub profiles (.bash_profile or .profile) in user homedirs call the commmon_profile from /home/appsadm/env/... (except for uvadm).

3. So, when new versions of uvadm are installed, you do NOT have to worry about overwriting any of your profile customizations (in /home/appsadm/...).

** Updating functions in $APPSADM/sfun/ **

This applies only to users who are migrating mainframe JCL to unix/linux.

The 1st time UV install instructions setup user appsadm and copied the JCL/script functions from $UV/sfun/* to $APPSADM/sfun/ (see #3B on page 'B4'). This is done because users may need to modify the functions (jobset51, etc) and they would not want to overwrite their modifications when a new version of Vanouver Utilities is installed in /home/uvadm (/home/appsadm/ unchanged).

BUT, if you have not yet made any customizations in $APPSADM/sfun & know that the new version of $UV does contain desired uppdates (in /home/uvadm/sfun/.) then you should copy $UV/sfun/* to $APPSADM/sfun/. (#3B on page 'B4').

#1. Login appsadm --> /home/appsadm
==========

#3b. cp /home/uvadm/sfun/* sfun/
==========

** Updating common_profiles in $APPSADM/env/ **

The 1st time UV install instructions setup user appsadm and copied the common_profiles from $UV/env/* to $APPSADM/env/ (see #3C on page 'B4').

It is less likely that you want to copy new common_profiles from new $UV/env to $APPSADM/env/ since you must have had this working on 1st version, but if desired you could.

#1. Login appsadm --> /home/appsadm
==========

#3c. cp /home/uvadm/env/* env/
==========

You will then need to repeat the customization of your common_profile as per #4a - #6a on page 'B4'.

B15. INSTALLATION - Creating uvadm Directory & Copying the Media

** Notes re owner, group,& permissions **

You must be uvadm when you copy the media (on the previous page) in order that the new files will be owned by uvadm & have desired permissions & group. If you forgot to exit root & login uvadm, you could correct as follows:

```
#1a.  Login as root              - login as root
#1b.  cd /home/uvadm             - change to uvadm home dir

#2.  chown -R uvadm uvadm        - change owner of all dirs & files in uvadm

#3.  chgrp -R group? uvadm       - change group of all dirs & files in uvadm

#4.  chmod -R 775 uvadm        <-- DON'T do this, not all same

#4a. chmod    775 uvadm/*         - all directories should be 775 or 755
#4b. chmod    775 uvadm/sf/*/*    - all scripts should be 775
#4c. chmod    775 uvadm/bin/*     - all executable programs should be 775
                                    (not yet compiled see page C1 ahead)
#4d. chmod    664 uvadm/dat1/*    - files in other subdirs should be 664
#4e. chmod    664 uvadm/.../*     ------------- etc ------------


#5a. chown uvadm .bash_profile    - ensure .bash_profile owner is uvadm
#5b. chmod 664 .bash_profile      - ensure read/write .bash_profile
      ---- OR ----
#5a. chown uvadm .profile       - ensure .profile owner is uvadm
#5b. chmod 664 .profile         - ensure you can read/write uvadm .profile

#6. exit                        - exit root
```

** chmod1 - change perms on all files & subdirs **

The 'chmod1' script will change permissions on entire directory trees, using 'find' to process all levels of sub-directories. This script can save hours of manual investigation & correction.

For directories that must be shared among several users, we recommend 775 for directories & 644 for files. Any executable files must be manually corrected back to 775 after running chmod1.

You must login or su to 'root' to run this script since it changes permissions. In the instructions below, I have included an 'export' to add the uvadm script subdirs to root's PATH (could add permanently in root's profile for future use).

```
#1. su root                     <-- switch to root

#2. export PATH=$PATH:/home/uvadm/sf/adm
    ====================================
    - add to PATH, so root can find chmod1

#3. chmod1 directory dir-perms file-perms  <-- command format
    ====================================

#3a. chmod1 directory 775 664              <-- recommended permissions
     ========================

#4. Correct any executable files back to 775 (restore execute bits)

#4a. chmod 775 .../.../bin/*      <-- restore execute bits on programs
#4b. chmod 775 .../.../scripts/*  <-- restore execute bits on scritps

#5. exit         <-- exit from root
    ====
```

** Reverting to old version if problems occur **

Since we have saved the old '/home/uvadm' (as /home/uvadm.old), we can quickly revert to the previous version if you encounter a problem. UV Software will help you resolve the problem as soon as possible, but you may need to run your production jobs with no waiting.

#1. login as root

#2. mv /home/uvadm /home/uvadm.new <-- rename just installed version

#3. mv /home/uvadm.old /home/uvadm <-- rename old version as current

Note - on some systems (such as SUN) 'mv' (rename) does not work
- logins go the originally setup login directory
- use the following procedure:

#2. rm -rf /home/uvadm/* <-- remove new version

#3. cp -rf /home/uvadm.old/* /home/uvadm <-- copy old version back

```
********************************************************************************
C1.                 Compiling Vancouver Utility C Programs
********************************************************************************
```

** Compile 'C' programs **

The Vancouver Utilities are distributed with the 'C' utility programs already compiled (into /home/uvadm/bin) using Red Hat Enterprise 5.1 (64 bit AMD Opteron). You must recompile for other systems.

The following script will compile all 'uv' programs assuming you have restored the distribution media to your 'uvadm' directory & set your path to include the 'sf' directory of scripts. Please see page 'C5' ahead to compile the 'ux' programs that support Indexed Sequential Variable length records compatible with Micro Focus COBOL IDXFORMAT3 file types.

You can modify the 'C' compiler called by the compile scripts by exporting an alternate value for 'UVCC' as shown below. Might be used on machines that do not have the ANSI C 'cc' compiler, but do have 'gcc'.
(see more info at the bottom of the next page)

```
Login 'uvadm' --> /home/uvadm      <-- login uvadm (if not already)
cd $UV --------->  /opt/uvsw/uvadm  <-- alternate location uvadm

export UVCC=gcc        - causes compile scripts to use 'gcc' vs 'cc'

ccuvall OStype H32/H64 uvlib32/64.a disamTYP32/64.a [options]  <-- format
=====================================================
```

H32/H64 - bits in a 'long' (C programming long integer)
- On 32 bit hardware, we still have 64 bit integer accumulators using 'long long' (software implementation for 64 bit integers)

H32/H64 - bits in a 'pointer' (C program address pointer)
- 64 bit hardware would always have 64 bit pointers
- DEC alpha, Itanium, 64 bit x86 chips, 64 bit AMD opteron, etc

Most computers now have 64 bit hardware & operating systems, and the compilers have options to compile C programs in 32 bit mode or 64 bit mode. The 64 bit mode causes 'long' integers & 'pointers' to be defined as 64 bits vs 32 bits.

The Vancouver Utilities do not necessarily need to be compiled in 64 bit mode. 64 bit accumulators are available via software on 32 bit hardware by using 'long long'.

Note - compile arguments changed in January 2009 to H32/H64
- replacing prior arguments L32/L64 + P32/P64 (Longs + Pointers)
- LF64 Large Files (> 2 gig) option no longer required (always provided)

C2. Vancouver Utilities Compile script Options

** LNX (linux) **

```
ccuvall LNX H64 uvlib64.a disamRHEL6.a - compile on RedHat Enterprise Linux 6.xx
=======================================   64 bit Hardware integers & pointers
                                        - DISAM COBOL compatible Indexed files

ccuvall LNX H64 uvlib64.a disamRHEL7.a - compile on RedHat Enterprise Linux 7.xx
=======================================   64 bit

ccuvall LNX H64 uvlib64.a disamCentOS.a  - compile on CeentOS 64 bit machine
=======================================

ccuvall LNX H64 uvlib64.a disamUB1604.a  - compile on Ubuntu 16.04
=======================================

ccuvall LNX H32 uvlib32.a disamLNX32.a - compile on Linux Intel 32 bit machine
=======================================   with 64 bit int acums via SoftWare

Note - 32 bit compiles do allow for 64 bit integers & Large Files > 2 gig
     - since 64 bit accumulators are provided by software (long long)
     - since Large Files (> 2 gig) are provided by software (open64 & lseek64)
```

** AIX compile **

```
export UVCC=xlc                       <-- specify AIX C compiler (vs default cc)
===============
ccuvall AIX H32 uvlib32.a disamAIX32.a -q32   <-- for IBM AIX 32 bit & DISAM
=========================================
ccuvall AIX H64 uvlib64.a disamAIX64.a -q64   <-- AIX 64 bit compile
=========================================
ccuvall AIX H64 uvlib64.a disamAIXnull.a -q64 <-- if disamAIX64.a missing
=============================================   - see 'D1' create disamAIXnull.a
```

** SUN **

```
ccuvall SUN H32 uvlib32.a disamSUN32.a       <-- SUN Solaris defaults to 32 bit
===================================
ccuvall SUN H64 uvlib64.a disamSUN64.a -m64  <-- 64 bit option on newer SUNs
=========================================
ccuvall SUN H64 uvlib64.a disamSUN64.a -xarch=v9 <-- 64 bit on older SUNs
=============================================

Note - '-m64' or '-xarch=v9' is the SUN option for 64 bit longs & pointers
     - 32 bit versions probably work just as well
     - since software 'long long's are used to get 64 bit accumulators
```

C3. Vancouver Utilities Compile script Options

** HPUX 'cc' compiler **

Note - also see below HPUX 'gcc' (Gnu open source, free download)

```
ccuvall HP H32 uvlib32.a disamHP32.a -Ae <-- 32 bit hardware
=======================================   - 64 bit integer via long long
-Ae ANSI C compiler option no longer required as of HPUX 10.20

ccuvall HP H64 uvlib64.a disamHP64.a "-Ae +DD64"  <-- 64 bit on HPUX cc
================================================
ccuvall HP H32 uvlib32.a disamHP32.a "-Ae +DD32"   <-- 32 bit on HPUX cc
================================================
```

- "-Ae" ANSI C compiler option no longer required as of HPUX 10.20
- '+DD64' is HP's option for 64 bit mode (longs & pointers 64 bits)
- VU option 'H32' achieves 64 bit integer acums via 'long long' (vs long)
- change +DD64/+DD32 to +DA2.0W/+DA2.0N for HP-UX on PA-RISC
- You may specify compile options as the last argument to ccuvall.
- must specify 'multiple' compile options in double quotes
- could specify '-g' for the debug option

** HPUX IA64 'gcc' compiler **

```
export UVCC=gcc    <-- causes ccuvall to execute 'gcc' vs 'cc
===============
ccuvall HP H32 uvlib32.a disamHPIA32.a -milp32
============================================
ccuvall HP H64 uvlib64.a disamHPIA64.a -mlp64   <-- 'mlp64' vs 'milp32'
============================================     - 64 bit recommended
```

** test your hardware/compiler 32bit/64bit **

See page 'E1' test program 'test64c' to determine if your hardware/compiler is 64 bit capable & if BIG-end or little-end. May compile & execute as follows but see sample output on page 'E1'.

```
cc src/test64c.c -obin/test64c   <-- compile program (cc or gcc)
==============================
bin/test64c                      <-- execute
===========                        - expected results on page 'E1'
```

C4. Vancouver Utilities Compile script Options

** compiling uvhd & uvhdcob **

The 'ccuvall' script does compile uvhd & uvhdcob, but if you need to run ccuvall in 32 bit mode, you should compile these in 64 bit mode as follows.

Note - These are the only 2 programs that are self-contained (do not require linking with archives of subfunctions). You can therefore compile these directly with 'cc' or 'gcc' (vs the scripts ccuvall, etc).

Note - uvhd & uvhdcob must have 64 bit integers to handle files > 2 gigs
- best achieved by 64 bit options -DH64, -m64, -q64, -mlp64 (HP gcc)
- possible (not HP) with -DH32 which uses 'long long' for 64 bit integers

```
cc -DH64 -DAIX -q64 -obin/uvhd src/uvhd.c
=========================================
cc -DH64 -DAIX  -q64 -obin/uvhdcob src/uvhdcob.c
================================================
```

Note - change 'AIX' as appropriate to 'LNX','SUN', or 'HP'
- change '-q64' to '-m64' for other machines (not HP)

** compile uvhd/uvhdcob on HP IA64 gcc **

```
gcc -DH64 -DHP -mlp64 -obin/uvhd src/uvhd.c
===========================================
gcc -DH64 -DHP -mlp64 -obin/uvhdcob src/uvhdcob.c
================================================
```

** compiling JCL converters **

'ccjclall' is provided to compile the JCL converters, a separate script (from ccuvall above), since the JCL converters are included only in the class A mainframe conversion package (not in the class B utilities package).

```
ccjclall LNX H64 uvlib64.a disamRHEL7.a      <-- RedHat Linux 64 bit
=======================================
ccjclall AIX H64 uvlib64.a disamAIX64.a -q64 <-- AIX 64 bit
===========================================
```

** disamNULL if 64 bit version missing **

If version of 'disam___' is missing (from /home/uvadm/lib/...), for your computer, see page 'D1' to compile a 'NULL' version, until UV Software can supply the required functional version.

```
*************************************************************************
C5.           Compiling 'ux' programs to access IDXFORMAT3/8 files
*************************************************************************
```

The previous pages have documented the Instructions to compile most of the Vancouver Utility programs (uvhd,uvlist,uvcp,uvsort,uvcopy,uvqrpg,etc).

This page documents the instructions required to compile the 'ux' versions (uxcp,uxsort,uxcopy.uxqrpg) that support Indexed Sequential Variable length records compatible with Micro Focus COBOL IDXFORMAT3 & IDXFORMAT8 file types.

For more info on these file types, please see the 'File Handling' documentation for Micro Focus Object or Server Express COBOL. The Micro Focus COBOL 'EXTFH' file handler is compiled into the 'ux' versions using the 'ccuvcob' script supplied with the Vancouver Utilities.

Two versions of the programs are required since only Micro Focus COBOL users (server express 2.2+) could compile 'ux' versions whereas any user can compile 'uv' versions which use D-ISAM to support Indexed Sequential FIXED record length files.

The Micro Focus IDXFORMAT3,8 files are identified to the Vancouver Utility programs as file typ=IDXf3 & typ=IDXf8. Please see uvcp.doc, uvsort.doc, or uvcopy1.doc for more info.

** compile uvlib32.a for Micro Focus on 64 bits **

On 64 bit systems (such as AMD 64), the Micro Focus compiler & script ccuvcob requires option '-m32' on any libraries linked with the ux____ programs.

Before you run ccuvcob to compile uxcopy,uxsort,uxcp,uxqrpg, you must create the 'uvlib32.a' archive as follows:

```
ccuvfa LNX H32 uvlib32.a -m32  <-- create 32 bit archive on a 64 bit machine
==============================
```

Note that the preceding 'ccuvall' script generated uvlib32.a/uvlib64.a for linking with 'uv' & 'ux' programs. You only need to generate 'uvlib32.a' to link with the 'ux' programs on 64 bit systems using script ccuvcob to include the Micro Focus variable length Indexed file handler.

Micro Focus provides cobmode=64 but I have had problems with it. I get errmsg: ld: lib/uvlib64.a(errmsg.0) relocatin R_X86_64_32 local symbol cant be used making shared object; recompile with -fPIC. Without cobmode=64 I can compile with H64, uvlib64.a,& disamLNX64.a, but execution gives signal 11.

C6. Compiling ’ux’ programs to access IDXFORMAT3/8 files

** compiling ’ux’ programs **

ccuvcob ux____ AIX/HP/LNX/SUN H32 uvlib___32/64.a disam___32/64.a <-- format

export COBMODE=32 <-- set 32 bit mode for Micro Focus ’cob’ compiler
- called by ’ccuvcob’ script to compile C programs

Example for Linux:

#1. ccuvcob uxcp LNX H32 uvlib32.a disamLNX32.a
#2. ccuvcob uxsort LNX H32 uvlib32.a disamLNX32.a
#3. ccuvcob uxcopy LNX H32 uvlib32.a disamLNX32.a
#4. ccuvcob uxqrpg LNX H32 uvlib32.a disamLNX32.a

Example for AIX:

#1. ccuvcob uxcp AIX H32 uvlib32.a disamAIX32.a -q32
#2. ccuvcob uxsort AIX H32 uvlib32.a disamAIX32.a -q32
#3. ccuvcob uxcopy AIX H32 uvlib32.a disamAIX32.a -q32
#4. ccuvcob uxqrpg AIX H32 uvlib32.a disamAIX.32a -q32

** always specify H32 for ccuvcob **

1. Script ’ccuvcob’ calls Micro Focus ’cob’ to compile the ’ux’ utilities.

2. ’cob’ defaults to 32 bit compiles, so we must specify H32

3. Archives linked with ’ux’ utilities must also be 32 bits

4. Note ’uvlib32.a compiled on prior page with option ’-m32’

5. Micro Focus does provide ’cobmode=64’ but I have had problems with it
   (see problem described on previous page).

```
*************************************************************************
C7.              Compiling Vancouver Utility C Programs
*************************************************************************

                          ** Compile Problems **

If you have 'error' messages during the compiles, please capture the
error messages and email them to me (owen@uvsoftware.ca).
Note that some systems may generate some 'warning' messages which can be
ignored if there are no 'error' messages.

ccuvall OStype H32/H64 uvlib32/64.a disam___32/64.a >ccerrs 2>&1
=================================================

- compile & direct any errmsgs into ccerrs
- email as a file attachment to owen@uvsoftware.ca

lp ccerrs           - OR print the error log & fax to UV Software Inc.

                     ** using alternate C compilers **

export UVCC=gcc    <-- causes compile scripts to use gcc vs cc

export UVCC=xlc    <-- causes compile scripts to use xlc for AIX
```

```
*************************************************************************
D1.             Compiling Vancouver Utility C Programs
*************************************************************************
```

** D-ISAM libraries available **

Here are the contents of /home/uvadm/lib/... as of January 2015. You should see a disam___32/64.a matching your machine (to link with uvcopy,uvsort,etc).

** Directory: /home/uvadm/lib **

```
     disamAIX32.a  disamAIX64.a  disamCentOS7.a  disamDOS.a
     disamRHEL6.a  disamRHEL7.a  disamSUN64.a    disamUB16.a
     uvlib32.a     uvlib64.a     uvlibDOS.a      vbisamLNX.a

     1README_disamNULL - generate dummy archive if DISAM archive
                         if not available for your machine
     archive/...  - old archives, probably obsolete
     dcheck/...   - utilities to validate Indexed files (report corruptions)
     dcheck/dcheckAIX dcheckCentOS7 dcheckRHEL6 dcheckRHEL7
     dcheck/dcheckSUN dcheckUB16 dpackLNX

     ksh93        - Korn shell for Linux machines, copy to /bin/ksh
```

** compiling D-ISAM **

D-ISAM is proprietary to Byte Designs & UV Software is licensed to distribute only the linkable archive (disamAIX64.a, disamHP64.a, disamLNX64.a, etc).

UV Software has compiled their copy of the D-ISAM source modules on the various machines & distributes only the compiled linkable archives listed above. The script to compile the source modules is listed on page 'N8'.

** dcheck - utility to verify D-ISAM files **

'dcheck' is a Byte Designs utility to validate D-ISAM files. You should copy the dcheck___ program matching your machine to /home/uvadm/bin & rename to just 'dcheck'. For example, if you are running on AIX:

```
cp lib/dcheckAIX bin/dcheck   <-- change AIX as req'd (HP,LNX,SUN,etc)
==========================
```

** if disam___ NOT available for your machine **

If you do not see the version you require, you need to compile src/disamNULL.c dummy program to lib/disamNULL.a, so you can compile uvcopy, uvsort, etc.

Please see the listing of 'disamNULL.c' on the next page which includes the instructions for compiling & archiving into lib.

D2. Compiling Vancouver Utility programs

** /home/uvadm/src/disamNULL.c **

```
/* disamNULL.c - null program used when no DISAM library required   */
/*             - to compile Vancouver Utility programs (uvcopy,etc) */
/* disamNULL allows compiling Vancouver Utilitie programs on 64 bit */
/* machines when the 64 bit version of D-ISAM is not yet available  */
/* - examples below assume that disamAIX64.a is missing for your AIX*/
/* D-ISAM file handler for Indexed files from www.bytedesigns.com  */
/* - compatible with C-ISAM and Micro Focus COBOL IDXFORMAT1       */
/* UV Software is allowed to provide only 'object code' for D-ISAM */
/* Since UVSI has only Linux at their development site, UVSI must  */
/* get time on other machines to create 64 bit versions of D-ISAM  */

/* ls -l /home/uvadm/lib   <-- see if a 64 bit version available   */
/* ======================       for your machine & operating system */
/* - disamAIX32.a,disamLNX32.a,disamLNX64.a,disamSUN64.a,etc       */
/* - disamAIX64.a missing, you could compile & archive as follows: */

/* 1. cc -c -Ihdr src/disamNULL.c       <-- compile to object (.o) */
/*    ==========================                                   */
/* 2. ar r lib/disamAIX64null.a disamNULL.o <-- create archive to  */
/*    ====================================     link with utilities*/
/* 3. rm disamNULL.o                   <-- remove object           */
/*    ==============                                               */
/*                                                                    */
/* ccuvall AIX H64 uvlib64.a disamAIX64null.a <-- compile all programs */
/* ========================================                           */
/* ccuv uvcopy AIX H64 uvlib64.a disamAIX64null.a <-- compile 1 program*/
/* ============================================                       */

/* declare structures referenced by DISAM functions */
/* - see original in srcf/iswrap.c                  */
struct keypart
  { short kp_start;
    short kp_leng;
    short kp_type;
  };
struct keydesc
  { short k_flags;
    short k_nparts;
    struct keypart k_part[5];
    short k_len;
    int   k_rootnode;
  };

/*Mar06/19 - prevent undefineds */
int iserrno;
int iserrio;
int isstat1;
int isstat2;
int isreclen;
int isrecnum;
int isrelcurr;

/
```

```
*/
/* dummy DISAM functions called by uvcopy,uvsort,uvcp,uvqrpg */
int isbuild (char *name, int len, struct keydesc *kdsc, int mode)
{ return(0); }
int isaddindex (int isfd, struct keydesc *kdsc)
{ return(0); }
int iserase (char *name)
{ return(0); }
int isindexinfo (int isfd, struct keydesc *dest, int idx)
{ return(0); }
int islock (int isfd)
{ return(0); }
int isunlock (int isfd)
{ return(0); }
int isopen (char *name, int mode)
{ return(0); }
int isclose (int isfd)
{ return(0); }
int isstart (int isfd, struct keydesc *key, int len, char *data, int mode)
{ return(0); }
int isread (int isfd, char *data, int mode)
{ return(0); }
int iswrite (int isfd, char *data)
{ return(0); }
int isdelete (int isfd, char *data)
{ return(0); }
int isrewrite (int isfd, char *data)
{ return(0); }
/*-------------------------- end disamNULL.c -------------------------*/
```

D3. Compiling Vancouver Utility programs

** Indexed file .dat for Micro Focus COBOL & Vancouver Utilities **

The Micro Focus COBOL default is NO '.dat' extension. We highly recommend
the Micro Focus COBOL option 'IDXNAMETYPE=2' to read/write Indexed files
with the '.dat' extension. You can code this in the 'extfh.cfg' file defined
by env-var $EXTFH, which is defined in the common_profile.

```
export EXTFH=$RUNLIBS/ctl/extfh.cfg  <-- define location of extfh.cfg
==================================    - EXTFH defined in common_profile

IDXNAMETYPE=2    <-- option within extfh.cfg
=============
```

Early versions of Vancouver Utilities (uvcopy,uvsort,uvcp) used only the
D-ISAM file handler for Indexed files (compatible with Micro Focus IDXFORMAT1).
The early versions of D-ISAM read either .dat or no extension, but always
created files without the '.dat' extension (compatible with Micro Focus COBOL
if not using IDXNAMETYPE=2 in EXTFH=extfh.cfg.

In 1998 environmental variable 'DISAMEXT' was provided for sites that wanted
to use '.dat' extensions (compatible with MF COBOL IDXNAMETYPE=2).

```
export DISAMEXT="dat"   <-- cause uvsort, etc to read/write .dat extensions
====================     - defined in common_profile
```

** ISDATEXT=".dat" D-ISAM update 2010 **

In April 2010, Byte Designs enhanced D-ISAM with env-var ISDATEXT to allow
the option of writing Indexed files without the '.dat' extension.

```
export ISDATEXT=".dat"  <-- cause uvsort, etc to read/write .dat extensions
======================    - defined in common_profile
```

You could drop the older DISAMEXT, but it will not do any harm as long as
you also define ISDATEXT=".dat".

** Indexed file capability UNIX vs mainframe **

Note that I have recoded the 'keydesc' structure to allow 9 keys, each of which
may have up to 3 parts. The OS/3 mainframe allowed only 5 single part keys.
Key duplicates & changes are not allowed on the 1st key, but are allowed
on keys 2-9. These defaults may be over-ridden if desired.

D4. Compiling Vancouver Utility C Programs

** user written subfunctions linked to uvcopy **

xxa,xxb,& xxc instructions are provided to call user subfunctions. These sub functions must have been compiled & archived to /home/uvadm/lib/uvlib32/64.a so they can be linked to uvcopy & uvqrpg.

uvsubxxa.c, uvsubxxb.c,& uvsubxxc.c are 3 dummy subfunctions provided in subdir /home/uvadm/srcf/... that you may modify with your desired C code.

```
#1. cd /home/uvadm
#2. vi srcf/uvsubxxa.c                <-- modify dummy instrn with yours
#3. ccuvf uvsubxxa LNX H32            <-- compile & archive to lib/uvlib32.a
#4. ccuv uvcopy LNX H32 uvlib32.a disamLNX32.a <-- compile&link uvcopy w subrtns
```

uvcopy instructions xxa, xxb, xxc have no operands
- areas i,j,k must be stored for subrtn arg1,2,3 (see area1,2,3 below)
- instrn ctrs 21,22,23 must be stored for arg4,5,6 (see ctr1,2,3 below)
- counters are defined here as long, but in uvcopy/uvqrpg as UVi64 which is long if H64 but long long if H32
- counters are pointers so you can store them as well as reference them

See more documentation for ’xxa’ & ’uvsubxxa.c’ in ’uvcopy3.doc’ in volume 2. Look up ’xxa’ in the instructions index on pages 3 & 4.

```
*****************************************************************************
E1.                TESTING the Vancouver Utility programs
*****************************************************************************

                  ** C program to test 64 bit integers **

A C program & a uvcopy job are supplied to test 64 bit integers.
Please see program listings, operating instructions, & expected outputs on
pages Q1 to Q4 of 'TestDemo.htm#Q1'. Here are the operating instructions &
expected output for the C program.

cc src/test64c.c -o bin/test64c   <-- compile test64c.c program
================================

bin/test64c                       <-- execute
===========                         - expected results below

 00. 0000000000004096 * 04 = 0000000000016384 = 00 40 00 00 00 00 00 00
 01. 0000000000016384 * 04 = 0000000000065536 = 00 00 01 00 00 00 00 00
 02. 0000000000065536 * 04 = 0000000000262144 = 00 00 04 00 00 00 00 00
 03. 0000000000262144 * 04 = 0000000001048576 = 00 00 10 00 00 00 00 00
 04. 0000000001048576 * 04 = 0000000004194304 = 00 00 40 00 00 00 00 00
 05. 0000000004194304 * 04 = 0000000016777216 = 00 00 00 01 00 00 00 00
 06. 0000000016777216 * 04 = 0000000067108864 = 00 00 00 04 00 00 00 00
 07. 0000000067108864 * 04 = 0000000268435456 = 00 00 00 10 00 00 00 00
 08. 0000000268435456 * 04 = 0000001073741824 = 00 00 00 40 00 00 00 00
 09. 0000001073741824 * 04 = 0000004294967296 = 00 00 00 00 01 00 00 00
 10. 0000004294967296 * 04 = 0000017179869184 = 00 00 00 00 04 00 00 00
 11. 0000017179869184 * 04 = 0000068719476736 = 00 00 00 00 10 00 00 00
 12. 0000068719476736 * 04 = 0000274877906944 = 00 00 00 00 40 00 00 00
 13. 0000274877906944 * 04 = 0001099511627776 = 00 00 00 00 00 01 00 00
 14. 0001099511627776 * 04 = 0004398046511104 = 00 00 00 00 00 04 00 00
 15. 0004398046511104 * 04 = 0017592186044416 = 00 00 00 00 00 10 00 00

Note - the 32 bit limit is at #09 above 1,073,741,824 is OK in 32 bits
     - following entries zeros or garbage if 64 bit software not working

              ** recompile with various options for 32/64 bits **

gcc src/test64c.c -o bin/test64c        <-- use 'gcc' vs 'cc'
================================
xlc -q64 src/test64c.c -o bin/test64c   <-- xlc & -q64 for AIX
====================================
cc -m64 src/test64c.c -o bin/test64c    <-- '-m64' for some machines/OSs
====================================      - SUN, Micro Focus cc
cc +DD64 src/test64c.c -o bin/test64c   <-- '+DD64' for 'cc' on HPUX IA64
====================================
gcc -mlp64 src/test64c.c -o bin/test64c <-- '-mlp64' for 'gcc' on HPUX IA64
========================================

Some OS's have a symbolic link  or any alias from 'cc' to 'gcc'.
You can look for this with following commands:

     which cc     <-- should show path location of cc
     which gcc    <-- should show path location of gcc
     ls -l /bin/*cc*      <-- list all cc programs in /bin
     ls -l /usr/bin/*cc*  <-- or might be in /usr/bin
```

E2. TESTING the Vancouver Utility programs

** uvcopy jobs to test 64 bit integers **

Page 'E1' above compiled & executed a very small 'C program' to test 64 bit integers. Now we will run similar tests using a 'uvcopy job'.

testint1 & testint2 are uvcopy jobs to test uvcopy processing 32 & 64 bit integers. You have already compiled the uvcopy interpreter with 'ccuvall' on 'C1' - 'C3'. We will give the instructions here to recompile just uvcopy for Linux on both 32 bit & 64 bit hardware/software. You should change 'LNX' to the proper code for your system (SUN,HP,AIX,SFU,etc).

#1. ccuv uvcopy LNX H32 uvlib32.a disamLNX32.a
    ==
    - compile for 32 bit hardware (long = 32 bits)
    - 64 bit integers achieved via Software (long long = 64 bits)

#2. uvcopy testint1   <-- execute uvcopy 'testint1' (32 bit limit)
    ===============   - output as shown below:

```
# Date=2019/08/27, Machine=LNX, Bits=$longbits
# testint1_data - test data for testint1 (test decimal to binary & back)
#          - uvadm/tf/testint1_data for uvcopy job uvadm/pf/testint1
#2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
#                     expected output   - - actual output in hex - - -
# decimal-input       binary hex        byte-native       byte-swapped
0000000000000000001  0000000000000001  0100000000000000  0000000000000001
0000000000000000016  0000000000000010  1000000000000000  0000000000000010
0000000000000000256  0000000000000100  0001000000000000  0000000000000100
0000000000000004096  0000000000001000  0010000000000000  0000000000001000
0000000000000065536  0000000000010000  0000010000000000  0000000000010000
0000000000001048576  0000000000100000  0000100000000000  0000000000100000
0000000000016777216  0000000001000000  0000000100000000  0000000001000000
0000000000033554432  0000000002000000  0000000200000000  0000000002000000
0000000000067108864  0000000004000000  0000000400000000  0000000004000000
0000000000134217728  0000000008000000  0000000800000000  0000000008000000
0000000000268435456  0000000010000000  0000001000000000  0000000010000000
0000000000536870912  0000000020000000  0000002000000000  0000000020000000
0000000001073741824  0000000040000000  0000004000000000  0000000040000000
0000000002147483648  0000000080000000  0000008000000000  0000000080000000
# --- 32 bit limit ---
0000000004294967296  0000000100000000  0000000001000000  0000000100000000
0000000008589934592  0000000200000000  0000000002000000  0000000200000000
0000000017179869184  0000000400000000  0000000004000000  0000000400000000
0000000034359738368  0000000800000000  0000000008000000  0000000800000000
0000000068719476736  0000001000000000  0000000010000000  0000001000000000
```

Note - entries above --- 32 bit limit --- would be zeros or garbage if 64 bit
       software emulation of 64 bit hardware not working

```
E3.               TESTING the Vancouver Utility programs
-------------------------------------------------------------------------------

                   ** uvcopy jobs to test 64 bit integers **

#3. ccuv uvcopy LNX H64 uvlib64.a disamLNX64.a
    ==========================================
    - compile on 64 bit hardware (long = 64 bits)

#4. uvcopy testint2    <-- execute uvcopy 'testint2' (64 bit limit)
    ===============

# Date=2019/08/27, Machine=LNX, Bits=$longbits
cycle      decimal-value  hex native        hex swapped
01  0000000000000000001  0100000000000000  0000000000000001
02  0000000000000000002  0200000000000000  0000000000000002
03  0000000000000000004  0400000000000000  0000000000000004
04  0000000000000000008  0800000000000000  0000000000000008
05  0000000000000000016  1000000000000000  0000000000000010
06  0000000000000000032  2000000000000000  0000000000000020
07  0000000000000000064  4000000000000000  0000000000000040
08  0000000000000000128  8000000000000000  0000000000000080
09  0000000000000000256  0001000000000000  0000000000000100
10  0000000000000000512  0002000000000000  0000000000000200
11  0000000000000001024  0004000000000000  0000000000000400
12  0000000000000002048  0008000000000000  0000000000000800
13  0000000000000004096  0010000000000000  0000000000001000
14  0000000000000008192  0020000000000000  0000000000002000
15  0000000000000016384  0040000000000000  0000000000004000
16  0000000000000032768  0080000000000000  0000000000008000
17  0000000000000065536  0000010000000000  0000000000010000
18  0000000000000131072  0000020000000000  0000000000020000
19  0000000000000262144  0000040000000000  0000000000040000
20  0000000000000524288  0000080000000000  0000000000080000
21  0000000000001048576  0000100000000000  0000000000100000
22  0000000000002097152  0000200000000000  0000000000200000
23  0000000000004194304  0000400000000000  0000000000400000
24  0000000000008388608  0000800000000000  0000000000800000
25  0000000000016777216  0000000100000000  0000000001000000
26  0000000000033554432  0000000200000000  0000000002000000
27  0000000000067108864  0000000400000000  0000000004000000
28  0000000000134217728  0000000800000000  0000000008000000
29  0000000000268435456  0000001000000000  0000000010000000
30  0000000000536870912  0000002000000000  0000000020000000
31  0000000001073741824  0000004000000000  0000000040000000
32  0000000002147483648  0000008000000000  0000000080000000 32 bit limit
33  0000000004294967296  0000000001000000  0000000100000000
34  0000000008589934592  0000000002000000  0000000200000000
                  ----- 30 lines omitted -----
63  0461168601842738790⁴  0000000000000040  4000000000000000
64  =922337203685477580x  0000000000000080  8000000000000000
65  0000000000000000000  0000000000000000  0000000000000000

See all 65 output lines at 'TestDemo.htm#Q2'
```

```
*************************************************************************
F1.              'uycopy' to access SQL DataBase Tables
*************************************************************************

In November 2008, a new version of uvcopy, 'uycopy' was created to provide
access to SQL DataBase tables. It had to be a separate version (from uvcopy)
because it requires the SQL C-API libraries for compilation at sites with an
SQL DataBase installed.

'uycopy' uses file 'typ=DBT' to define a DataBase Table & environmental
variables (DBhost1, DBuser1, DBpass1) for connection to the DataBase.
Please see documentation for uycopy in 'SQLdemo.doc#Part_2'.

              ** compiling 'uycopy' with SQL DataBase libraries **

1. You must download & install MySQL as instructed at 'SQLdemo.doc#Part_1'.

#2. ccmysqlLNX64 uycopy   <-- execute script to compile 'uycopy'
    ===================

                   ** script to compile 'uycopy' **

# ccmysqlLNX64 - script to compile Vancouver Utility C program
#              - provides access to MySQL DataBase libraries
# - by Owen Townsend, UV Software, Nov 05/2008
# - for Linux on 64 bit system, with option for debug
#
pgm="$1";
if [[ ! -f src/$pgm.c ]]; then
   echo "usage: ccmysqlLNX64 program"
   echo "       ==================="
   echo "example: ccmysqlLNX64 uycopy"
   echo "         ==================="
   echo "compile with MySQL libraries on Linux 64 bit with debug option"
   exit 99; fi;
#
sqlinclude="-I/usr/include/mysql"
sqllibs="-L/usr/lib64/mysql -lmysqlclient -lz -lcrypt -lnsl -lm -L/usr/lib64 -lssl -lcrypto"
syms="-DLNX -DH64 -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64"
#
cc $syms -Ihdr src/${pgm}.c $sqlinclude $sqllibs \
          lib/uvlib64.a lib/disamLNX64.a -o bin/${pgm}
#=====================================================
exit 0
```

```
*****************************************************************************
G1.            Vancouver Utilities for non-Unix/Linux systems
*****************************************************************************

'https://www.uvsoftware.ca/windowsdos.htm' - Windows DOS command line

'https://www.uvsoftware.ca/windowssfu.htm' - Windows Services For Unix

'https://www.uvsoftware.ca/cygwinuwin.htm' - CYGWUN/UWIN

Or see the documentation text files in the doc/ directory

/home/uvadm/doc/WindowsDOS.doc

/home/uvadm/doc/WindowsSFU.doc

/home/uvadm/doc/CygwinUwin.doc

                  ** profiles for Windows DOS, SFU, UWIN,& CYGWIN **

The profiles have been split into a 'stub profile' (bash_profile) and a
'common profile' (test or prod). This is highly recommended but has not
been done for the following profiles:

/home/uvadm/env/other_profiles/profile_DOS

/home/uvadm/env/other_profiles/profile_SFU

/home/uvadm/env/other_profiles/profile_UWIN

/home/uvadm/env/other_profiles/profile_CYGWIN

If the 'other_profiles' subdir is omitted, please see the profiles listed
in the documentation for the relevant system.
```

```
*************************************************************************
H1.              TESTING the Vancouver Utility programs
*************************************************************************
```

Please see the separate section 'TestDemo.doc'. Run these tests & ensure your outputs match the expected outputs illustrated following each test/demo. Please call/email UV Software if you find any discrepancies or need any help running these tests.

Running these 'test/demo's is also a great way to investigate the Vancouver Utilities & find out which utilities will be the most useful to you.

** Contents of 'TestDemo.doc' **

uvhd - binary file investigation & display in vertical hexadecimal
uvcp - file copy with record selection & reformatting
uvsort - file sort utility with record selection & much more
uvlist - list text files, inserting laser printer control codes
uvcopy - the most powerful Vancouver Utility (data manipulation, etc)
uvqrpg - Quick Report Generator

table1 - pre-programmed table analysis of any field by any argument
cobmap1 - pre-programmed job to create record layouts from COBOL copybooks

D-ISAM - test uvcp/uvsort processing Indexed Sequential Fixed length records
       - compatible with Micro Focus COBOL IDXFORMAT1 files

scan1d - scan all files in a directory for matches to qualified patterns
rep2 - copy files replacing patterns, qualified by other patterns
prodfix1 - uvcopy equivalent of the rep2 pre-programmed job above

testint2 - test conversion of decimal to integer & back (32/64 bit tests)
         - Be sure to run this test if you have compiled with the 64 bit option.
         - see page 'E2' or even better at 'TestDemo.htm#Q1'

testIDXL - test uxcp processing Indexed Sequential Variable length records
         - compatible with Micro Focus COBOL IDXFORMAT3 files

tabfix1 - converting tabs to blanks

More pre-programmed jobs for various useful file conversions
         - UPPER/lower case, EBCDIC/ASCII, etc

```
*************************************************************************
I1.                      Documentation Overview
*************************************************************************

The following are some of the documentation files are included on the
distribution media in subdirectory 'doc'.

install.doc    - installation guide for UNIX systems

UVdemos2.doc - Brief descriptions of uvhd,uvsort,uvcp,uvlist,& uvcopy + tutorials
   SelectJobs - select lines from 1 or all files in a directory, identified by
                multiple patterns & conditions on same line or anywhere in file
   TableJobs - create table summaries of counts & values in text files
             - tabled in memory & dumped to a file at EOF
   UVscripts - most useful shell scripts used at UV Software
             - Count Files,Lines,& KB in 1 or all subdirs of directories
             - rename files, removeCRs, alldiff2/3, uvcmpFA1, chmod1/2, dtree
             - list/move/remove files older/newer than x days
   uvcopy - data manipulation utility, power of assembler without the complexity
   uvfix1 & uvfixA - easy way to use uvcopy without writing uvcopy framework

TestDemo.doc - Testing & Demonstrating the Vancouver Utilties
             - addition to UVdemos2.doc

uvtrain.doc  - training guide for the Vancouver Utilities

uvlist.doc   - documentation listing utility
uvhd.doc     - hexdump utility for any file
uvcp.doc     - a mini-version of uvcopy that allows all parameters
               on the command line (replaces mainframe DATA utility)
uvsort.doc   - parameter driven sort for UNIX systems
               compatible with the uvcp utility (replaces SORT & SORT3)
uvqrpg.doc   - report generator

uvcopy_.doc  - documentation for the uvcopy program
               (uvcopy1.doc, uvcopy2.doc, etc thru uvcopy7.doc)
             - uvcopy is a powerful data manipulation program
               that is the basis for many applications in the pkg

JCLcnv1demo.doc - converting MVS JCL & COBOL illustrated with demo files
MVSCOBOL.doc    - converting MVS mainframe COBOL to Micro Focus on Unix/Linux
DATAcnv1.doc    - Converting EBCDIC to ASCII with or without copybooks
                - preserving packed fields & correcting zoned signs
                - convert mainframe data files to '|' delimited text files
                  for loading databases (SQL Server or Oracle)
VSEJCL.doc     - Converting VSE JCL to Korn shell scripts for Unix/Linux

UVjobs.doc     - Pre-Programmed jobs for the uvcopy interpreter
                 TABLEjobs,  HTMLjobs,
                 SCANjobs,   REPjobs,  INSERTjobs, LABELjobs,
                 LISTjobs,   COPYjobs, FIXjobs,    DROPjobs, UVjobs1, UVjobs2,
                 REFORMjobs, CMPjobs,
                 COBaids,    COBscans, ADMjobs,    AIDjobs,
                 VTOCjobs,   XREFjobs, TAPEjobs,
```

```
*****************************************************************************
I2.     Documentation available on distribution archive & on web site
*****************************************************************************

             ** Vancouver Utilities Documentation on the WEB site **

https://www.uvsoftware.ca     <-- UV doc on the web site
=========================       - OR from installed uvadm/dochtml/...

The Vancouver Utilities distribution archive now contains subdir 'dochtml'.
After install, you can point your web browser to the dochtml subdir as follows:

file:/home/uvadm/dochtml/index.htm
==================================

              ** Vancouver Utilities Doc HTML Home-Page **

   summary - Description of the Vancouver Utilties
   UVdemos2.doc - Brief descriptions of uvhd,uvsort,uvcp,uvlist,& uvcopy + tutorials
               - Selectjobs, TableJobs, UVscripts,
   uvcopy - data manipulation utility, power of assembler without the complexity
   uvfix1 & uvfixA - easy way to use uvcopy without writing uvcopy framework

   Mainframe Conversion Library
     - JCLcnv1demo,JClcnv2real,JCLcnv3aids,MVSCOBOL,VSEJCL,VSECOBOL,VSEDATA

   Vancouver Utility Program User Guides
     - uvhd,uvlist,uvsort,uvcopy,uvcp,uvqrpg,uvhdcob

   uvhd documentation - file investigation utility
   uvhd program (free download) - free sample of Vancouver Utilities
   Installation  - Vancouver Utilities Installation Guide
   Install & Demonstrate - Windows/DOS Vancouver Utilties
   Testing & Demonstrating - Unix/Linux Vancouver Utilties
   uvtrain       - Training Guide: uvhd, uvlist, uvcp, uvsort, uvcopy
   Photo Gallery - Vancouver Map and City, Owen Townsend, etc

   Pre-Programmed jobs (for the uvcopy interpreter)
     - HTMLjobs, TABLEjobs, SCANjobs, REPjobs, COBOLaids

   Unix & Linux shell scripts - 200+ scripts for conversions and everyday use
   UVprofile - UV Software, Company Profile and Products
   uvprices  - price lists and license agreements

Over 99.8% of the HTML you see here on the UV Software web site was generated
automatically. Only a few small files such as index.htm were coded manually.
The voluminous documentation (99.8%) continues to be maintained via the UNIX
'vi' editor. When website updates are made (monthly or whatever) the 99.8%
is reconverted & merged with the few small hand coded files. The result
is 'tar'd, compressed,& FTP'd to the webserver.

This may be of interest to other sites with legacy documentation.
If your legacy documentation has a clear set of rules for page headings &
section/chapter references, then you (or UV Software) could write a uvcopy
job to automatically convert your legacy documentation to HTML.
```

```
*************************************************************************
I3.              Printing any 1 section of Documentation
*************************************************************************
```

The documentation is on the distribution media & you may print as follows:

Assuming you are in /home/uvadm, using recommended profiles, have set UVLPDEST
to a PCL5 compatible Duplex laser printer with 3 hole punched paper.

```
export UVLPDEST=-dlaserxx  <-- specify your laser printer
=========================    - if not set in your profile

uvlp13D doc/install.doc    <-- use uvlp13D to print any 1 document (Duplex)
=======================      - see script listed on the next page
```

** script uvlp13D - print Duplex at 13 cpi **

```
#!/bin/ksh
# uvlp13D - Korn shell script from UVSI stored in: /home/uvadm/sf/util/
# uvlp13D - print a file at 13 cpi (100 chars on 8 1/2 x 11)
#        - DUPLEX mode (new option April 98)
#        - pg hdngs with: filename, mod-date, today-date, page#s
#        - for HP laserjet 4 printers & compatibles
#
#usage: uvlp13D filename [group1] [group2]  <-- may override default options
#       ================================
#
# - 1 of several: uvlp12,uvlp14,uvlp16,uvlp12L,uvlp14L,uvlp13D,uvlp14LD,etc
# - these scripts invokes uvlist & pipe to the spooler
# - see uvlist.doc for many group1(file) & group2(laser printer) options
# - these scripts convert group2 options into HP PCL5 escape sequences
# - scriptnames reflect commonly used uvlist options, for example:
#   group1:  p60  = 60 lines per page
#            b50e60 = Begin print at page 50 & End print at page 60
#   group2:  d1   = duplex mode
#            c13  = 13 cpi = 90 chars across on 8 1/2 " paper
#            m280 = default margin offset by 280/720 inch (for 3 hole punches)
#
# .profile should specify environmental variables for 'lp', for example:
# export UVLPDEST="-dlp0"       #<-- destination 'lp0' ("-dLPT1" for SFU)
# export UVLPDEST=""            #<-- null to use lpadmin default
# export UVLPOPTN="-onobanner"  #<-- 'nobanner' option for lp
# export UVLPOPTN=""            #<-- null disable for Windows SFU
#
if [ -f "$1" ]; then :
   else echo "ERROR - $1 is not a file"; exit 1; fi
uvlist $1 p60$2 a2d1c13n-220$3 | lp $UVLPOPTN $UVLPDEST
#=====================================================
uvln=$(basename $0)
linesbf=$(wc -l $1); linesb=${linesbf% *}; lines=${linesb##* };
echo "$uvln printing $1 on $UVLPDEST, lines=$lines"
exit 0
```

See 'uvlist.doc' for complete details on the Laser printer options and the
various scripts available.

```
*************************************************************************
K1.               Installation Guide for Vancouver Utilities
*************************************************************************

                  ** printing Vancouver Utility manuals **

If desired, you could print the more relevant documents & mount in three 3-ring
2-inch binders with clear covers to insert volume titles & identification.

Volume 1 - Installation, administration, test/demos, etc
Volume 2 - Program reference manuals
         - uvlist, uvhd, uvcp, uvsort, uvcopy, uvqrpg, etc
Volume 3 - Mainframe conversion manuals
         - JCLcnv1demo,JClcnv2real,JCLcnv3aids,MVSCOBOL,DATAcnv1
         - OR VSEJCL, VSECOBOL, VSEDATA, DATAcnv1, CNVaids,

                ** scripts to print 3 volumes of documentation **

#!/bin/ksh
# UVdocV1 - Korn shell script from UVSI stored in: /home/uvadm/sf/adm/
# UVdocV1 - print volume 1 for the Vancouver Utilities
#         - install,uvtrain,uvhelp,scripts1,ADMjobs,CNVaids,CMPjobs
#         - by Owen Townsend - UV Software Inc.
#
#usage: UVdocV1 format-options [HP compatible printer options]
#ex:    UVdocV1 p60      <-- recommended options (must enter at least 1 arg)
#       ===========
# - must enter at least 1 arg (see uvlist.doc for option details)
# - defaults shown below, over-ride if desired
#
echo "UVdocV1 - print volume #1 doc for the Vancouver Utilities"
echo "export UVLPDEST=-dlaser?  <-- change default dest to laser printer"
echo "                             (script uses UVLPDEST if defined)"
echo "UVdocV1 page-format-options [laser-printer-options]"
echo "                          <-- enter 'p60' if other defaults OK"
echo "UVdocV1 p60 d1c12m300n-240 <-- default options"
echo "        p60                   60 lines per page"
echo "            d1                DUPLEX option (append d1 to arg#2)"
echo "              c12             11 chars/inch & margins for 3 hole punch"
echo "                 m300         front margin offset right 300/720 inch"
echo "                     n-240    back margin offset right 240/720 inch"
echo "                              (Duplex back margin sign reversed)"
if [[ -z "$1" ]]
then echo "UVdocV1 requires at least 1 arg (laser options will default)"
     echo "UVdocV1 p60          <-- enter 'p60' if other defaults OK"
     exit 99; fi
uvlp13D doc/install.doc    $1 $2
uvlp13D doc/UVdemos2.doc    $1 $2
uvlp13D doc/uvtrain.doc    $1 $2
uvlp13D doc/uvhelp.doc     $1 $2
uvlp13D doc/scripts1.doc   $1 $2
uvlp13D doc/ADMjobs.doc    $1 $2
uvlp13D doc/CNVaids.doc    $1 $2
uvlp13D doc/CMPjobs.doc    $1 $2
exit 0
```

K2. printing Vancouver Utility manuals

** scripts to print 3 volumes of documentation **

```
# UVdocV2 - print volume 2 Vancouver Utilities - Program References
#   --- 25 lines omitted, similar to UVdocV1 on prior page ---
uvlp13D doc/uvlist.doc     $1 $2
uvlp13D doc/uvhd.doc       $1 $2
uvlp13D doc/uvhdcob.doc    $1 $2
uvlp13D doc/uvsort.doc     $1 $2
uvlp13D doc/uvcp.doc       $1 $2
uvlp13D doc/uvcopy1.doc    $1 $2
uvlp13D doc/uvcopy2.doc    $1 $2
uvlp13D doc/uvcopy3.doc    $1 $2
uvlp13D doc/uvcopy4.doc    $1 $2
uvlp13D doc/uvcopy5.doc    $1 $2
uvlp13D doc/uvqrpg.doc     $1 $2
exit 0

# UVdocMVS - print MVS Mainframe conversions for the Vancouver Utilities
#   --- 25 lines omitted, similar to UVdocV1 on prior page ---
uvlp13D doc/JCLcnv1demo.doc $1 $2
uvlp13D doc/MVSCOBOL.doc    $1 $2
uvlp13D doc/DATAcnv1.doc    $1 $2
exit 0

# UVdocVSE - print VSE Mainframe Conversions for Vancouver Utilities
#          - VSE may be OBSOLETE ?
#   --- 25 lines omitted, similar to UVdocV1 on prior page ---
uvlp13D doc/VSEJCL.doc     $1 $2
uvlp13D doc/VSECOBOL.doc   $1 $2
uvlp13D doc/DATAcnv1.doc   $1 $2
uvlp13D doc/VSEDATA.doc    $1 $2
exit 0
```

** summary of printed documentation **

```
Volume1 - install.doc uvtrain.doc uvhelp.doc scripts1.doc ADMjobs.doc
        - CNVaids.doc CMPjobs.doc

Volume2 - uvlist.doc uvhd.doc uvhdcob.doc uvsort.doc uvcp.doc
        - uvcopy1.doc uvcopy2.doc uvcopy3.doc uvcopy4.doc uvcopy5.doc
        - uvcopy6.doc uvcopy7.doc uvqrpg.doc

Volume3 - JCLcnv1demo.doc,JClcnv2real.doc,JCLcnv3aids.doc,MVSCOBOL.doc,DATAcnv1.doc
```

L1. Creating Vancouver Utility distribution CD

This is old documentation to create Vancouver Utilities distribution CD.

#1. Login as uvadm --> /home/uvadm

#2. clear the oldest backup subdir (using b9 for example)

#2a. rm -rf /home2/uvbak/b9/* <-- clear all old subdirs & files
#2b. rm -rf /home2/uvbak/b9/.* <-- clear old .profile etc

#3. cp -r * /home2/uvbak/b9 <-- copy all /home/uvadm/* to b9

#4. cd /home2/uvbak/b9 <-- change over to backup subdir

#5. update license & registration site & name in selected programs
(such as uvcopy & uvsort). Relevant lines are as follows:

```
sncopy(Y.version,"20090909",8,3);
sncopy(Y.license,"090909_00V_930630",19,3);
sncopy(Y.sitename,"UV Software",30,3);
sncopy(Y.regname,"Owen Townsend",30,3);
```

We will update the version/license,& change sitename/regname to
the new customer info.

#5a. vi src/uvcopy.c <-- update version,license,site,name in uvcopy
#5b. vi src/uvsort.c <-- same for uvsort, etc

#6. Recompile to ensure we did not create any errors in our updates

#6a. ccuvall LNX H32 uvlib32.a disamLNX32.a
- for any Linux 32 machine with 32 bit longs & pointers

#6b. ccuvall LNX H64 uvlib64.a disamLNX64.a
- for my AMD Opteron 64 bit HP xw9400 at UV Software

L2. Creating Vancouver Utility distribution CD

** Write VU to CD on Linux **

#7. Create tar & zip files in /home2/uvbak/... in the uvbak homedir
above our current working dir /home2/uvbak/b9/...

#7a. tar cvf ../uvadm.tar .

#7b. zip -r ../uvadm.zip .

8. cd .. <-- change back up to /home2/uvbak

9. mkisofs -o uvadm.iso uvadm.tar uvadm.zip
- create an ISO9660 CD image containing uvadm.tar & uvadm.zip

10a. Login as root (must be root to run 'cdrecord')

10b. cd /home2/uvbak

10c. cdrecord -v uvadm.iso <-- write the ISO9660 image to CD
- the CD device is determined by CDR-DEVICE in /etc/cdrecord.conf

11. Verify the write by reading back (into /home2/uvbak/b10)

11a. mount -r -t iso9660 /dev/cdrom /mnt
- mount the CD (to verify write OK)

#12a. logoff root
#12a. login uvadm --> /home/uvadm
#12c. cd /home2/uvbak/b10 <-- change into b10 subdir to read CD
#12d. rm -rf * <-- clear all files from b10 subdir

#13. tar xvf /mnt/uvadm.tar <-- read back (with tar) to verify CD write OK

#13a. unzip /mnt/uvadm.zip <-- OR read back with UNZIP to verify CD write OK

```
*************************************************************************
N1.                Compiling Subfunctions & Archiving
*************************************************************************

The procedures on this page are not required if you successfully ran the
'ccuvall' script (see page 'C1') which compiles & archives all subfunctions,
and then compiles most programs (linking with subfunctions).

These instructions to compile & archive individual subfunctions would only
be needed if you had some problems with the ccuvall. If interested you can
see the subfunction compile scripts (ccuvf & ccuvfa) in the 'sf' subdir.

       ** compiling subfunctions & archiving to link with programs **

Example: compile & archive the UVstring.c subroutine

#1. cc -c -DLNX -DH64 srcf/UVstring.c <-- compile 1 subfunction (on Linux)
    ================================   - writes UVstring.o to current dir

#2. ar -r lib/uvlib64.a UVstring.o    <-- archive to lib/uvlib64.a
    ==============================

           --- or ---

#1. ccuvf UVstring LNX H64 uvlib64.a  <-- compile & archive any subfunctn
    ================================

       -- or much better --

#1. rm lib/uvlib64.a           <-- remove existing archive before recompile
    ================

#2. ccuvfa LNX H64 uvlib64.a   <-- compile & archive all subfunctions on Linux64
    ========================       from dir srcf to archive lib/uvlib64.a

#2a. ccuvfa AIX H64 uvlib64.a -q64 <-- example for AIX 64 bit machine
     ============================
```

N2. scripts to COMPILE 'uv' Programs

The procedures on this page are not required if you successfully ran the 'ccuvall' script (see page 'C1') which compiles & archives all subfunctions, and then compiles most programs (linking with subfunctions).

** Compiling Programs Individually (uvcopy example) **

```
cc -Ihdr src/uvcopy.c lib/uvlib32.a lib/disamLNX32.a -obin/uvcopy
   -DLNX -DH32 -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64
```

--- OR easier, using script 'ccuv' ---

```
ccuv uvcopy LNX H32 uvlib32.a disamLNX32.a  <-- see script 'ccuv' listed on 'N5'
```

```
ccuv uvcopy AIX H64 uvlib64.a disamAIX64.a -q64  <-- example for AIX 64 bit
```

--- choices for ccuv script above ---

```
ccuv uvcopy INT/HP/SUN/AIX/LNX/SFU H32/H64 uvlib32/64.a disam___32/64.a [optns]
```

Note - '___' could be AIX,HP,LNX,SUN,etc
- options could be '-g' for debugging, '-Ae' for HP ANSI C compiler,
'-m32' compile 32 bit mode on 64 bit machine
'-q32' or '-q64' compile 32/64 bit mode on AIX

** compile for debugging **

```
ccuv uvcopy LNX H64 uvlib64.a disamLNX64.a -g   <-- arg6 '-g' for debugging
```

--- OR easier (if Linux) ---

```
ccdebugLNX64 uvcopy    <-- script similar to above with hard-coded options
```
for debugging on Linux 64 bit

** Compiling 'ux' programs for Variable Length Indexed files **

See instructions on page 'C5' for script 'ccuvcob' (listed on page 'N7')

```
ccuvcob uxcopy SUN H64 uvlib64.a disamSUN64.a  <-- example for uxcopy
```
(or uxcp,uxsort,uxqrpg)

N3. scripts to Compile ALL 'uv' Programs

```
#!/bin/ksh
# ccuvall - Korn shell script from UVSI stored in: /home/uvadm/sf/adm/
# ccuvall  - script to compile all Vancouver Utility programs
#          - by Owen Townsend, UV Software, Mar1993 - Jan2009
#          - see more details in the 'ccuv' script called by this ccuvall
# June2008 - see 'ccjclall' to compile the JCL conversion utilities
#          - separate since JCL converters only in class A pkg (not class B)
#
echo "compile all Vancouver Utility programs"
os="$1"; hw="$2"; uvlib="$3"; disam="$4"; opts="$5";  # capture args
if [[ -d src && -d srcf && -d lib && -d bin && -d hdr && $# -gt 3 ]]; then :
else
 echo " "
 echo "ERROR - must be in \$UV, usually /home/uvadm/ OR /opt/uvs/uvadm/"
 echo "- with subdirs src,srcf,hdr,lib,bin & must specify arguments as follows:"
 echo " "
 echo "usage: ccuvall OStype H32/H64 uvlib32/64.a disam__?__.a [options]"
 echo "       ============================================================"
 echo "       ccjclall LNX H32 uvlib32.a disamRHEL6.a       # RHEL 6.xx 64 bit"
 echo "       ccjclall LNX H64 uvlib64.a disamRHEL7.a       # RHEL 7.xx 64 bit"
 echo "       ccjclall LNX H64 uvlib64.a disamCentOS.a      # CentOS 64 bit"
 echo "       ccjclall LNX H64 uvlib64.a disamUB16.a        # UBuntu 16.04 64 bit"
 echo "       ccjclall AIX H64 uvlib64.a disamAIX64.a -q64  # AIX 64 bit"
 echo "       ccjclall AIX H32 uvlib32.a disamAIX32.a -q64  # AIX 32 bit"
 echo "- arg1 must be LNX,AIX,HP,SUN,etc"
 echo "- arg2 must be H32 or H64 for longs & pointers 32 bits or 64 bits"
 echo "      (programs use software long long for 64 bit integer acums)"
 echo "- arg3 must be subfunctions archive lib/uvlib32/64.a"
 echo "- arg4 must be D-ISAM archive lib/disam__?__.a (match 1 of above)"
 echo "- arg5 options (-q64 AIX 64 bit, -Ae HP ANSI, -g debug, etc"
 exit 91; fi;
#
rm -f lib/$uvlib                          # remove old archive (for linker)
ccuvfa        $os $hw $uvlib $opts        # compile all subfunctions & archive
echo "subfunctions compiled, now compile Vancouver Utility programs"
ccuv uvcopy    $os $hw $uvlib $disam $opts  # compile programs
ccuv uvcp      $os $hw $uvlib $disam $opts
ccuv uvhd      $os $hw $uvlib $disam $opts
ccuv uvhdcob   $os $hw $uvlib $disam $opts
ccuv uvlist    $os $hw $uvlib $disam $opts
ccuv uvqrpg    $os $hw $uvlib $disam $opts
ccuv uvsort    $os $hw $uvlib $disam $opts
ccuv uvtime    $os $hw $uvlib $disam $opts
echo "Vancouver utility compiles completed on $(date)"
echo "ls -l bin should show: uvcopy,uvcp,uvhd,uvhdcob,uvlist,uvqrpg,uvsort"
echo " - use 'ccjclall' to compile: jclproc41/51,jclunix41/51"
echo " - use 'ccuvcob' to compile: uxcopy,uxcp,uxqrpg,uxsort"
echo " - see install.doc page C4 to compile ux.... (requires Micro Focus COBOL)"
exit
```

N4. scripts to Compile All JCL conversion programs

```
#!/bin/ksh
# ccjclall - Korn shell script from UVSI stored in: /home/uvadm/sf/adm/
# ccjclall - script to compile Vancouver Utility JCL converters
#          - separate script since converters only in class A pkg (not class B)
#          - by Owen Townsend, UV Software, Mar1993 - Jan2009
#
echo "compile all Vancouver Utility JCL converters"
echo "- must have compiled subfunctions (part of prior ccuvall)"
os="$1"; hw="$2"; uvlib="$3"; disam="$4"; opts="$5";  # capture args
if [[ -d src && -d srcf && -d lib && -d bin && -d hdr && $# -gt 3 ]]; then :
else
 echo " "
 echo "ERROR - must be in \$UV, usually /home/uvadm/ OR /opt/uvs/uvadm/"
 echo "- with subdirs src,srcf,hdr,lib,bin & must specify arguments as follows:"
 echo " "
 echo "usage: ccjclall OStype H32/H64 uvlib32/64.a disam__?__.a [options]"
 echo "       ============================================================="
 echo "       ccjclall LNX H32 uvlib32.a disamRHEL6.a       # RHEL 6.xx 64 bit"
 echo "       ccjclall LNX H64 uvlib64.a disamRHEL7.a       # RHEL 7.xx 64 bit"
 echo "       ccjclall LNX H64 uvlib64.a disamCentOS.a      # CentOS 64 bit"
 echo "       ccjclall LNX H64 uvlib64.a disamUB16.a        # UBuntu 16.04 64 bit"
 echo "       ccjclall AIX H64 uvlib64.a disamAIX64.a -q64  # AIX 64 bit"
 echo "       ccjclall AIX H32 uvlib32.a disamAIX32.a -q64  # AIX 32 bit"
 echo "       ccjclall SUN H64 uvlib64.a disamSUN64.a       # SUN 64 bit"
 echo "       ccjclall HP  H64 uvlib64.a disamHP64.a  -Ae   # HP ANSI C"
 echo "- arg1 must be LNX,AIX,HP,SUN,etc"
 echo "- arg2 must be H32 or H64 for longs & pointers 32 bits or 64 bits"
 echo "      (programs use software long long for 64 bit integer acums)"
 echo "- arg3 must be subfunctions archive lib/uvlib32/64.a"
 echo "- arg4 must be D-ISAM archive lib/disam_____.a (match 1 of above)"
 echo "- arg5 options (-q64 AIX 64 bit, -Ae HP ANSI, -g debug, etc"
 exit 91; fi;
#
ccuv jclproc41 $os $hw $uvlib $disam $opts
ccuv jclproc51 $os $hw $uvlib $disam $opts
ccuv jclunix41 $os $hw $uvlib $disam $opts
ccuv jclunix51 $os $hw $uvlib $disam $opts
echo "Vancouver JCL converter compiles completed on $(date)"
echo "ls -l bin should show: jclproc41/51,jclunix41/51"
exit
```

N5. scripts to Compile each 'uv' Program

```
#!/bin/ksh
# ccuv - script to compile C prgm & link functions (/home/uvadm/sf/adm/ccuv)
#     - by Owen Townsend - UV Software, Aug1992 - Jan2009
# Must execute in /home/uvadm/...
# Specify options arg#6: -g debug, -Ae ANSI HP-UX, -q64 AIX 64bit, -m32 on 64
echo "----->compile C program, script ccuv $*"
# set compiler - symbol UVCC if defined, else gcc if OS Windows*, else cc
if [[ -n "$UVCC" ]]; then UVCC=$UVCC; else UVCC=cc; fi
pgm="$1"; os="$2"; hw="$3"; uvlib="$4"; disam="$5"; opts="$6"  #capture args
if [[ -f src/$pgm.c && -d src && -d srcf && -d lib && -d bin && -d hdr && $# -gt 4 ]]
 then :
 else echo " "
 echo "ERROR - must be in \$UV (usually /home/uvadm/ or /opt/uvs/uvadm)"
 echo "- with subdirs src,srcf,hdr,lib,bin - specify arguments as follows:"
 echo "  arg1 must be a program in src/... (uvcopy for example, omit '.c')"
 echo " "
 echo "ccuv program OStype H32/H64 uvlib32/64.a disam__?__.a [options]"
 echo "     ================================================================="
 echo "     ccuv uvcopy LNX H32 uvlib32.a disamRHEL6.a -m64  #RHEL 6.xx 64 bit"
 echo "     ccuv uvcopy LNX H64 uvlib64.a disamRHEL6.a       #RHEL 7.xx 64 bit"
 echo "     ccuv uvcopy LNX H64 uvlib64.a disamCentOS.a      #CentOS 64 bit"
 echo "     ccuv uvcopy LNX H64 uvlib64.a disamUB16.a        #Ubuntu 64 bit"
 echo "     ccuv uvcopy AIX H64 uvlib64.a disamAIX64   -q64  # AIX 64 bit"
 echo "arg1 program to be compiled in subdir src/... (do not specify .c ext)"
 echo "arg2 must be LNX,AIX,HP,SUN,etc"
 echo "arg3 must be H32 or H64 for longs & pointers 32 bits or 64 bits"
 echo "     (H32 uses software long long for 64 bit integer acums)"
 echo "arg4 must be subfunctions archive uvlib32/64.a in lib/..."
 echo "arg5 must be D-ISAM archive disamAIX64.a  disamRH6xx.a disamRH7xx.a etc"
 echo "arg6 options: -q64 AIX 64bit, -Ae HP ANSI, -g debug, -m32 compile on 64"
 exit 91; fi;
#
if [[ "$os" = "HP" || "$os" = "SUN" || "$os" = "AIX" || "$os" = "DEC" ||\
      "$os" = "INT" || "$os" = "LNX" || "$os" = "SCO" ||\
      "$os" = "CWIN" || "$os" = "UWIN" || "$os" = "SFU" ]]; then :
else echo "arg2 OS types: HP,SUN,AIX,DEC,INT,LNX,UWIN,CWIN,SFU"; exit 92; fi
if [[ "$hw" = "H32" || "$hw" = "H64" ]]; then :
else echo "arg3 must be H32/H64, 32/64 bits in C longs&pointers"; exit 93; fi
# verify that arg4 is the subfunctions archive lib/uvlib32/64.a
if [[ ! -f lib/$uvlib ]]; then
   echo "arg4 must be the subfunctions archive uvlib32/64.a"; exit 94; fi
# verify that arg5 is the D-ISAM archive lib/disam___32/64.a
if [[ ! -f lib/$disam ]]; then
   echo "arg5 must be the D-ISAM archive lib/disam___32/64.a"; exit 95; fi
# combine -Dsymbols to shorten command line
syms="-D$os -D$hw"
# setup Large File options (always as of Jan 2008)
LFS="-D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64"
#
$UVCC $opts $syms $LFS -Ihdr src/${pgm}.c lib/$uvlib lib/$disam -o bin/${pgm}
#=============================================================================
exit 0
```

N6. Compile subfunctions to link with programs

```
#!/bin/ksh
# ccuvfa - Korn shell script from UVSI stored in: /home/uvadm/sf/adm/
# ccuvfa - compile All C subfunctions in /home/uvadm/srcf/...
#          & add to the archive (lib/uvlib32/64.a)
#        - by Owen Townsend, UV Software, Aug1992 - Jan2009
#
# set compiler from $UVCC if defined, else gcc if OS Windows*, else cc
if [[ -n "$UVCC" ]]; then UVCC=$UVCC
else UVCC=cc; fi
os="$1"; hw="$2"; uvlib="$3" opts="$4"
if [[ -d srcf && -d hdr && -d lib && $# -gt 2 ]]; then :
else
 echo " "
 echo "ERROR - must be in \$UV, usually /home/uvadm/ OR /opt/uvs/uvadm/"
 echo "- with subdirs srcf,hdr,lib & must specify arguments as follows:"
 echo " "
 echo "ccuvfa OStype H32/H64 uvlib32/64.a [options]"
 echo "    ========================================="
 echo "    ccuvfa LNX H32 uvlib32.a -m32 # Linux 32 bit"
 echo "    ccuvfa LNX H64 uvlib64.a      # Linux 64 bit"
 echo "    ccuvfa AIX H64 uvlib64.a -q64 # AIX 64 bit"
 echo "arg1 must be LNX,AIX,HP,SUN,etc"
 echo "arg2 must be H32 or H64 for longs & pointers 32 bits or 64 bits"
 echo "     (programs use software long long for 64 bit integer acums)"
 echo "arg3 must be subfunctions archive lib/uvlib32/64.a"
 echo "arg4 options: -q64 AIX 64bit, -Ae HP ANSI, -g debug, -m32 compile on 64"
 exit 91; fi;
#
if [[ "$os" = "HP" || "$os" = "SUN" || "$os" = "AIX" || "$os" = "DEC" ||\
      "$os" = "INT" || "$os" = "LNX" || "$os" = "SCO" ||\
      "$os" = "CWIN" || "$os" = "UWIN" || "$os" = "SFU" ]]; then :
else echo "OS types: HP,SUN,AIX,DEC,INT,LNX,UWIN,CWIN,SFU"; exit 92; fi
#
if [[ "$hw" = "H32" || "$hw" = "H64" ]]; then :
else echo "arg2 must be H32/H64, 32/64 bits in C longs&pointers"; exit 93; fi
#
# if AIX 64 bit option (-q64), set 'ar' option -X64
if [[ ("$opts" == *-q64*) || ("$opts" == *aix64*) ]]; then X64="-X64"; fi
#
# setup Large File options (always as of Jan2008)
LFS="-D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64"
x=0
rm -f lib/$uvlib;  # remove old lib/uvlib__.a
for i in srcf/*.c
do
   f=${i##*/}
      $UVCC -D$os -D$hw $opts $LFS -c -Ihdr srcf/$f
   ar $X64 r lib/$uvlib ${f%.c}.o
   let x=x+1
done
rm -f *.o                # remove all objects from current directory
echo "$x files compiled from srcf & archived to lib/$uvlib"
echo "ar tv lib/$uvlib  - to display archive table of contents"
exit
```

N7. scripts to COMPILE 'ux' Programs

```
#!/bin/ksh
# ccuvcob - Korn shell script from UVSI stored in: /home/uvadm/sf/adm/
# ccuvcob - compile UV program with Micro Focus COBOL to use EXTFH
#           for variable length Indexed files IDXFORMAT3 & IDXFORMAT8
#         - by Owen Townsend, UV Software, Dec2002 - Jan2009
# Jan2009 - Micro Focus compile tested with uvlib32.a & disam___32.a
#           (specify option '-m32' to compile uvlib & disam on 64 bit machines)
#         - will soon test 64 bit via cobmode=64 ? where to specify ?
pgm="$1"; os="$2"; hw="$3"; uvlib="$4"; disam="$5"; opts="$6"  #capture args
if [[ -f src/$pgm.c && $# -gt 4 ]]; then :
 else
 echo "ccuvcob program OStype H32/H64 uvlib32/64.a disam___32/64.a [options]"
 echo "======================================================================"
 echo "ccuvcob uxcopy LNX H32 uvlib32.a disamRHEL32.a -m32 # RHEL 32 bit"
 echo "--> use 32 bits for ux COBOL versions 64 bits not working yet"
 echo "ccuvcob uxcopy LNX H64 uvlib64.a disamRHEL7.a      # RHEL 7.xx 64 bit"
 echo "ccuvcob uxcopy AIX H64 uvlib64.a disamAIX64.a -q64 # AIX 64 bit"
 echo "ccuvcob uxcopy HP  H64 uvlib64.a disamHP64.a  -Ae  # HP ANSI C"
 echo "ccuvcob uxcopy SUN H64 uvlib64.a disamSUN64.a      # SUN 64 bit"
 echo "- arg1 program to be compiled in subdir src/... (no .c ext)"
 echo "- arg2 must be LNX,AIX,HP,SUN,etc"
 echo "- arg3 must be H32 or H64 for longs & pointers 32 bits or 64 bits"
 echo "       (H32 uses software long long for 64 bit integer acums)"
 echo "- arg4 must be subfunctions archive lib/uvlib32/64.a"
 echo "- arg5 must be D-ISAM archive lib/disam___32/64.a (___=LNX,AIX,SUN,HP)"
 echo "- arg6 options: -q64 AIX 64, -Ae HP ANSI, -g debug, -m32 compile on 64"
 echo "--> Must specify -m32 compiling uvlib32.a/disamXXX32.a on 64 bit machine"
 echo "--> Must have Micro Focus COBOL installed"
 exit 91; fi;
#
if [[ "$os" = "HP" || "$os" = "SUN" || "$os" = "AIX" || "$os" = "DEC" ||\
      "$os" = "INT" || "$os" = "LNX" || "$os" = "SCO" ||\
      "$os" = "CWIN" || "$os" = "UWIN" || "$os" = "SFU" ]]; then :
else echo "arg2 OS types: HP,SUN,AIX,DEC,INT,LNX,UWIN,CWIN,SFU"; exit 92; fi
#
if [[ "$hw" = "H32" || "$hw" = "H64" ]]; then :
else echo "arg3 must be H32/H64, 32/64 bits in C longs&pointers"; exit 93; fi
#
# verify that arg4 is the subfunctions archive lib/uvlib32/64.a
if [[ ! -f lib/$uvlib ]]; then
   echo "arg4 must be the subfunctions archive uvlib32/64.a"; exit 94; fi
#
# verify that arg5 is the D-ISAM archive lib/disam___32/64.a
if [[ ! -f lib/$disam ]]; then
   echo "arg5 must be the D-ISAM archive lib/disam___32/64.a"; exit 95; fi
isam=${disam%.*}      # remove the .a to pass as a -Dvariable
#
# establish COBOL options for Micro Focus COBOL compile
export COBMODE=32                   # 32 bit mode required
export COBOPT=$UV/ctl/cobdirectives # Micro Focus COBOL directives (-C options)
export EXTFH=$UV/ctl/extfh.cfg      # COBOL File Handler Configuration
#
cob -x $opts -N LITLINK\"2\" -CC -Ihdr -CC -I$COBDIR/include -CC -D$os \
-CC -D$hw -CC -D$isam src/$pgm.c lib/$uvlib lib/$disam -o bin/$pgm
#=====================================================================
rm -f $pgm.o
```

```
exit 0
```

N8. scripts to COMPILE 'ux' Programs

```
#!/bin/ksh
# ccdisams - script to compile DISAM subfunctions & create archive to link with uv...
#          - by Owen Townsend, UV Software, 1993,
#          - update DISAM96 1998, DISAM06 1996, DISAM72 2015, 2020
# compile in disam06 superdir with subdirs: base, head, wrap, sf, lib
# base - I copied all base functions into wrap (will compile all in wrap)
# wrap - originally just 'std' functions, I added 'is' functions from base
# head - disam06 headers - isconfig.h modified for UV Software
#      - header files from base & wrap moved to head
# sf   - my compile scripts (added to disam06 superdir by OT)
# lib  - for the output archive (added to disam06 superdir by OT)
#
#usage:  sf/ccdisams sourcedir archive  [options]
# 1.  sf/ccdisams wrap lib/disamRHEL6.a   -m64 <-- RedHat 6.xx
# 2.  sf/ccdisams wrap lib/disamRHEL7.a   -m64 <-- RedHat 7.xx
# 3.  sf/ccdisams wrap lib/disamCentOS7.a -m64 <-- CentOS 7.xx
# 4.  sf/ccdisams wrap lib/disamUB16.a    -m64  <-- Ubuntu 16.04
# 5a. sf/ccdisams wrap lib/disamAIX64.a   -q64  <-- AIX 64 bit
# 5b. sf/ccdisams wrap lib/disamAIX32.a   -q32  <-- AIX 32 bit
#
# lib names reflect UNIX system (LNX,RHEL6,RHEL7,UBuntu,AIX)
# default to 'cc' but allow user to: export UVCC=gcc
if [[ -z "$UVCC" ]]; then UVCC=cc; fi
lib="$1"; disama="$2"; opts="$3";  # capture args in symbols
echo "compiling all programs in $lib and archiving to $disama"
if [[ ! -d "$lib" ]]; then
   echo "usage: ccdisams srcdir archive [options]"
   echo "       ================================="
   echo "example: ccdisams wrap lib/disamRHEL7.a -m64"
   echo "         ==================================="
   echo " - arg1 must be a directory & arg2(output) must not exist"
   exit 1; fi
if [[ -f "$disama" ]]; then
   echo "- arg2 output archive must not exist, remove before regen"
   exit 1; fi
# set ar option -X64 if AIX 64 bit
if [[ ("$opts" == *-q64*) || ("$opts" == *aix64*) ]]; then X64="-X64"; fi
x=0
# compile all subfunctions in directory & archive to library
for i in $lib/*
do
   f=${i##*/}                          # remove the directory
   b=${f%.c}                           # remove the .c to get basename
   if [[ "$f" = *.c ]]                 # if current file is C source
   then                                # compile current .c to object .o
$UVCC $opts -c -DLF64 -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -Ihead  $lib/$f
   ar $X64 r $disama $b.o              # add .o to archive
   rm $b.o                             # remove .o
   let x=x+1
   fi
done
ar $X64 tv $disama          # display archive table of contents
echo "$x files compiled from $lib & archived to $disama $opts"
exit
```