Script - Test Server .bash_profile

This script needs to be modified for the test server login

```
vi /etc/skel/.bash_profile
```

Add or change the following to this

```
PATH=$PATH:$HOME/.local/bin:$HOME/bin
export PATH
/KIWI/corp/bin/bash_profile
#Uncomment if you want the shell to exit when user exists
exit
```

Script to be added/created to the $CORPBIN of the server

Edit the following script

```
vi /KIWI/corp/bin/bash_profile
```

Add the following

```
#!/bin/bash
#Type:Utility
############################################################
# Script:
#
# This is used for logging into a test server
#
# Created
# Steve Ling
############################################################
# Revision
#
# 2025-01-26 SFL Initial revision
```

```
#
############################################################
#
# Environment:
#
umask 002
#trap 'echo "Caught Ctrl+C"; exit 0' SIGINT
export KIWIBASE=/KIWI
DBHOST=localhost
MYSQLUSER=kiwisql
MYSQLPASS=800486kiwi
#
# Variables
REVS=/KIWI/rev/map
MREVS=/KIWI/rev/mes
DATASETS=/KIWI/backups
#
# Define colors
RED='\033[1;31m'
GREEN='\033[1;32m'
YELLOW='\033[1;33m'
BLUE='\033[1;36m'
NC='\033[0m' # No Color
BOLD='\033[1m'
FLASH='\033[5m'
#
# Define log file
LOG_FILE="Log$(date +"%Y%m%d_%H%M%S").log"
#
# Logging function
log() {
  local message="$1"
  local level="$2" # Optional: INFO, WARNING, ERROR
  local color="$3"

  # Timestamp
  timestamp=$(date +"%Y-%m-%d %H:%M:%S")

  # Log message with level and timestamp
  echo -e "${color}${message}${NC}" | tee -a "$LOG_FILE"
```

```
  #echo -e "${color}${message}${NC}" >> "$LOG_FILE"
  echo "[${timestamp}] [${level}] ${message}" | logger
}
#
# Display banner
banner(){
clear
log "==========================================" "INFO" "${GREEN}${BOLD}"
log "|  Welcome to the Test/Dev Environment  |" "INFO" "${GREEN}${BOLD}"
log "|                                       |" "INFO" "${GREEN}${BOLD}"
log "| You are in the server:                |" "INFO" "${GREEN}${BOLD}"
log "  `hostname` " "INFO" "${YELLOW}${BOLD}"
log "==========================================" "INFO" "${GREEN}${BOLD}"
echo ""
}
#
checkVariables() {
if [ ! -d $REVS ];then
        log "$REVS does not exist" "ERROR" "${RED}${BOLD}${FLASH}"
        return 1
fi
#
if [ ! -d $MREVS ];then
        log "$MREVS does not exist" "ERROR" "${RED}${BOLD}${FLASH}"
        return 1
fi
#
if [ ! -d $DATASETS ];then
        log "$DATASETS does not exist" "ERROR" "${RED}${BOLD}${FLASH}"
        return 1
fi
}
#
main_menu(){
# Main Menu
  banner
  echo ""
  log "A) Add Site" "INFO" ${GREEN}${BOLD}
  log "D) Delete Site." "INFO" ${GREEN}${BOLD}
  log "R) Restore Data" "INFO" ${GREEN}${BOLD}
```

```
  log "S) Select Site" "INFO" ${GREEN}${BOLD}
  echo ""
  log "X) Exit" "INFO" ${BLUE}${BOLD}
  echo ""
  echo -en ${YELLOW}${BOLD} "Default [S] : "
  read -t 10 choice
  log choice
  case "$choice" in
    "A"|"a")
          log "Chose option A" "INFO" ${YELLOW}${BOLD}
          site_add
    ;;
    "D"|"d")
          log "Chose option D" "INFO" ${YELLOW}${BOLD}
          site_delete
    ;;
    "R"|"r")
          log "Chose option R" "INFO" ${YELLOW}${BOLD}
          data_restore
    ;;
    "S"|"s")
          log "Chose option S" "INFO" ${YELLOW}${BOLD}
          site_select
    ;;
    "X"|"x")
          log "Exiting..." "INFO" ${YELLOW}${BOLD}
          return 0
    ;;
    *)
      site_select
      return 0
    ;;
  esac

}

site_add(){
  echo -en ${GREEN}${BOLD} "Enter 4 character site name : "
  read NEW_SITE
  if [ -z "$NEW_SITE" ]
```

```
then
  exit 1
  main_menu
else
  REVS=$(find /KIWI/rev/map/* -maxdepth 0 -type d| sort)
  PS3="Select MAP revision for $NEW_SITE : "
  select REV in $REVS
  do
    if [ ! -d "/KIWI/rev/site_$NEW_SITE" ]; then
      ln -s $REV $KIWIBASE/rev/site_$NEW_SITE
    fi
    mkdir -p $KIWIBASE/site_$NEW_SITE
    ln -s $KIWIBASE/rev/site_$NEW_SITE/bin $KIWIBASE/site_$NEW_SITE/bin
    ln -s $KIWIBASE/rev/site_$NEW_SITE/progs $KIWIBASE/site_$NEW_SITE/progs
    ln -s $KIWIBASE/rev/site_$NEW_SITE/scp $KIWIBASE/site_$NEW_SITE/scp
    ln -s $KIWIBASE/rev/site_$NEW_SITE/sql $KIWIBASE/site_$NEW_SITE/sql
    break
  done
install_mysql new
gen_kidds
mkdir -p $KIWIBASE/site_$NEW_SITE/site/bin
mkdir -p $KIWIBASE/site_$NEW_SITE/site/dat
if [ ! -d $KIWIBASE/services  ];then
   mkdir -p $KIWIBASE/services/
fi
cd $KIWIBASE
if [ ! -d $KIWIBASE/java  ];then
   ln -s $KIWIBASE/services java
fi
mkdir -p $KIWIBASE/services/sites/$NEW_SITE/
MREVS=$(find $KIWIBASE/rev/mes/* -maxdepth 0 -type d| sort)
PS3="Select MES revision for $NEW_SITE : "
select MREV in $MREVS
do
  cd /$KIWIBASE/services/sites/$NEW_SITE/
  mkdir -p `basename $MREV`
  ln -s `pwd`/`basename $MREV` current
  mkdir -p $KIWIBASE/services/sites/$NEW_SITE/current/conf
  mkdir -p $KIWIBASE/services/sites/$NEW_SITE/current/conf/roadgrids
  mkdir -p $KIWIBASE/services/sites/$NEW_SITE/current/conf/kiwiplan/roadgrids
```

```
    ln -s $KIWIBASE/services/maps/roadgrids/osm-gh
$KIWIBASE/services/sites/$NEW_SITE/current/conf/roadgrids/osm-gh
    ln -s $KIWIBASE/services/maps/roadgrids/osm-gh
$KIWIBASE/services/sites/$NEW_SITE/current/conf/kiwiplan/roadgrids/osm-gh
    mkdir -p $KIWIBASE/services/sites/$NEW_SITE/current/logs
    break
  done
  fi
  log "Added $NEW_SITE to the environment" "INFO" ${BLUE}${BOLD}
  log "Setting the $NEW_SITE environment" "INFO" ${YELLOW}${BOLD}
  export KIWI=${KIWI:-$KIWIBASE/site_$NEW_SITE}
  export CORPBIN=${CORPBIN:-${KIWIBASE}/corp/bin}
  export CORPDAT=${CORPDAT:-${KIWIBASE}/corp/dat}
  export SITEBIN=${SITEBIN:-${KIWI}/site/bin}
  export SITEDAT=${SITEDAT:-${KIWI}/site/dat}
  export BIN=${BIN:-${KIWI}/bin}
  export WORK=${WORK:-${KIWI}/work/${LOGNAME}}
  export DATA=${DATA:-${KIWI}/data_$NEW_SITE}
  export PROGS=${PROGS:-${KIWI}/progs}
  export SCP=${SCP:-${KIWI}/scp}
  export SQL=${SQL:-${KIWI}/sql}
  export UFD=${UFD:-${KIWI}/ufd}
  export KIWISEA=".:$WORK:$HOME:$DATA:$SITEDAT:$SITEBIN:$CORPDAT:$CORPBIN:$BIN:$PROGS:$SCP:$SQL"
  export PATH=".:$WORK:$HOME:$SITEBIN:$CORPBIN:$BIN:$PROGS:/bin:/usr/bin:/usr/sbin"
  export JAVA_HOME=/usr/lib/jvm/jre-11
  log "Now will add the MES environment for $NEW_SITE" "INFO" ${BLUE}${BOLD}
  cd $MREV
  REVBASE=`basename $MREV`
  log "This will take some time to decompress and will start the install" "INFO" ${BLUE}${BOLD}${FLASH}
  tar zxvf mes_$REVBASE.tar.gz
  ./mes-$REVBASE.sh
  log "Running the importdata script for the $NEW_SITE environment" "INFO" ${GREEN}${BOLD}
  $KIWIBASE/corp/bin/importdata
  main_menu
}

site_delete(){
  if [ ! -d $KIWIBASE/site_*  ];then
     log "Sites do not exists"
     sleep 2
```

```
    main_menu
  fi
  PS3="Choose site : "
  SITES=`ls -d $KIWIBASE/site_* | cut -b 12-`
  select SITE in $SITES
  do
    echo -n "Are you sure you want to delete /KIWI/site_$SITE (Y/[N])?"
    read choice
      case "$choice" in
      "Y"|"y")
        echo "Deleting the Classic Site Folder"
        find $KIWIBASE/site_$SITE/ -name kwsql -exec mysql -e "drop database $SITE_classic" \;;
        echo "Deleting the Kiwi Site Folder"
        rm -rf $KIWIBASE/site_$SITE
        echo "Deleting the Rev Site Folder"
        rm -f $KIWIBASE/rev/site_$SITE
        echo "Deleting the Jave Site Folder"
        rm -rf $KIWIBASE/services/sites/$SITE
        echo "Deleting the Web Folder"
        rm -rf $KIWIBASE/services/web/$SITE
      ;;
      *)
        main_menu
      ;;
      esac
    break
  done
  main_menu
}

data_restore() {
  if [ ! -d $KIWIBASE/site_*  ];then
     log "Sites do not exists"
     sleep 2
     main_menu
  fi
  PS3="Choose site : "
  SITES=`ls -d $KIWIBASE/site_* | cut -b 12-`
  select NEW_SITE in $SITES
  do
```

```
    echo -n "Are you sure you want to overwrite data for /KIWI/site_$NEW_SITE (Y/[N])?"
    read choice
     case "$choice" in
     "Y"|"y")
       rm -rf $KIWIBASE/site_$SITE/data_$NEW_SITE/*

       echo "Select data source : "
       echo ""
       echo "I) ISAM"
       echo "M) MySQL"
       echo ""
       echo -n "Default [I] : "
       read choice
       case "$choice" in
       "M"|"m")
         install_mysql restore
       ;;
       *)
         install_isam restore
       ;;
      esac
      ;;
      *)
        main_menu
      ;;
     esac
    break
  done
  main_menu
}

install_isam() {
  if [ ! -f $KIWIBASE/backups/*.tar.gz ];then
     log "ISAM Backup files do not exist"
     sleep 2
     return
  fi
  DATASETS=`find $KIWIBASE/backups/*.tar.gz`
  PS3="Select data for $NEW_SITE : "
  select DATASET in $DATASETS
```

```
    do
      if [ $1 == "new" ]
        then
        tar -zxvf $DATASET -C $KIWIBASE/site_$NEW_SITE/
        mv -vf `find $KIWIBASE/site_$NEW_SITE/* -type d -maxdepth 0`
$KIWIBASE/site_$NEW_SITE/data_$NEW_SITE
      fi
      if [ $1 == "restore" ]
        then
        find $KIWIBASE/site_$NEW_SITE/ -name kwsql -exec rm {} \;;
        tar -zxvf $DATASET -C $KIWIBASE/site_$NEW_SITE/
        SCR_DIR=`basename \`tar -ztf $DATASET | head -n 1\``
        mv -vf $KIWIBASE/site_$NEW_SITE/$SCR_DIR $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE
      fi
      break
    done
}

install_mysql() {
if [ -f $KIWIBASE/backups/*.sql.gz ];then
   DATASETS=`find $KIWIBASE/backups/*.sql.gz`
   PS3="Select data for $NEW_SITE : "
   select DATASET in $DATASETS
   do
      if [ $1 == "restore" ];then
         log "Checking to ensure the database is not in use" "INFO" ${YELLOW}${BOLD}
         sleep 2
         mysqladmin processlist | grep -v _master | egrep -q $DATASET
         if [ $? -eq 1 ] ; then
           mysql -e "drop database `$NEW_SITE_classic`"
         logo "Dropping old databases""INFO" ${YELLOW}${BOLD}
         sleep 2
         else
           echo "Destination database is in use. Please try again later."
           read error
         fi
      else
        mkdir -p $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE
      fi
      logo "Creating new database." "INFO" ${YELLOW}${BOLD}
```

```
      sleep 2
      mysql -e "create database $NEW_SITE_classic"
      log "Restoring database now." "INFO" ${YELLOW}${BOLD}
      sleep 2
      zcat $DATASET | grep -v "CREATE DATABASE" | mysql $NEW_SITE_classic
      mysql -e "grant all on $NEW_SITE.* to 'kiwisql'@'localhost' identified by '800486kiwi'"
     break
   done
else
   log "SQL Backup files do not exist" "INFO" ${YELLOW}${BOLD}
fi
   if [ ! -d $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE ]; then
        echo "Creating the site Data Folder"
        mkdir -p $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE
   fi
   touch $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE/kwsql
   echo "DATA="${NEW_SITE}"_classic" > $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE/kwsql
   echo "HOST=localhost" >> $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE/kwsql
   echo "INTERFACE=sql" >> $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE/kwsql
   echo "USER=kiwisql" >> $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE/kwsql
   echo "PASSWORD=800486kiwi" >> $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE/kwsql
#   echo "PASSWORDX=2o5tP2P8uX4WMp4xsjVnK8DGX5uYrwBz" >>
/KIWI/site_$NEW_SITE/data_$NEW_SITE/kwsql
   echo "LOG=error" >> $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE/kwsql
   echo "LOGPID=1" >> $KIWIBASE/site_$NEW_SITE/data_$NEW_SITE/kwsql
}

gen_kidds() {
  rm -f $KIWIBASE/site_$NEW_SITE/KIDSENV
  touch $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "SHELL=/bin/sh" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "TERM=vt100" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "KIDSDEBUG=4" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "KIDSLOGLEVEL=4" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "KIWISITE=$NEW_SITE" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "KIWI=/KIWI/site_$NEW_SITE" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "DATA=\$KIWI/data_$NEW_SITE" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "WORK=\$KIWI/work/$LOGNAME" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "BIN=\$KIWI/bin/" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
```

```
  echo "PROGS=\$KIWI/progs/" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "SCP=\$KIWI/scp/" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "SITEBIN=\$KIWI/site/bin" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "SITEDAT=\$KIWI/site/dat" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "SQL=\$KIWI/sql" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "KIDSLOGDIR=\$KIWI/work/$LOGNAME" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "KWSQL_COMMONDIR=\$KIWI/data_$NEW_SITE" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "LD_LIBRARY_PATH=\$PROGS" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "KWSQL_USER=kiwisql" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "KWSQL_PASS=800486kiwi" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "" >> $KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "KIWISEA=.:\$DATA:\$SITEDAT:\$SITEBIN:\$PROGS:\$BIN:\$SCP:\$SQL"  >>
$KIWIBASE/site_$NEW_SITE/KIDSENV
  echo "PATH=\$PATH:.:\$SITEDAT:\$SITEBIN:\$CORPBIN:\$PROGS:\$BIN:\$DAT" >>
$KIWIBASE/site_$NEW_SITE/KIDSENV
}

setup_counter () {
cd $KIWIBASE
rm $KIWIBASE/counter
ln -s $KIWIBASE/site_$NEW_SITE/work/mdcxmt counter
}

site_select(){
  if [ ! -d $KIWIBASE/site_*  ];then
     log "Sites do not exists" "ERROR" ${RED}${BOLD}${FLASH}
     sleep 2
     main_menu
  fi
    banner
    PS3="Choose site : "
    SITES=`ls -d $KIWIBASE/site_* | cut -b 12-`
    log "Sites that are currently running:" "INFO" ${YELLOW}${BOLD}
    for j in $SITES
        do
              RUNNING=`ps -ef | grep tomcat | grep -i $j | cut -b 133-136`
              if [ ! "$RUNNING" = "" ];then
                    log "${GREEN}${BOLD}Site ${BLUE}${BOLD}$j ${GREEN}${BOLD}is already running" "INFO"
```

```
${BLUE}${BOLD}
            fi
        done
    echo ""
    log "Site Ports" "INFO" ${YELLOW}${BOLD}
    for s in $SITES
        do
            if [ -e $KIWIBASE/services/sites/$s/current/conf/recentparametervalues.properties ];then
                PORTS=`cat $KIWIBASE/services/sites/$s/current/conf/recentparametervalues.properties|grep
OFFSET|cut -d"=" -f2`
                PORTEXT=`expr 8080 + $PORTS`
                if [ -n $PORTS ];then
                    log "${GREEN}${BOLD}Site ${BLUE}${BOLD}$s ${GREEN}${BOLD}has a port of
${BLUE}${BOLD}$PORTEXT" "INFO" ${BLUE}${BOLD}
                fi
            fi
        done
    echo ""
    log "Sites that are available:" "INFO" ${YELLOW}${BOLD}
    select SITE in $SITES
    do
      if [ -z "$SITE" ];then
        SITE=$SITES
      fi

      export PLANTID=$SITE
      export KIWI=$KIWIBASE/site_$PLANTID
      export EXEC="kiwimenu menu=support"

      # MySQL query
      QUERY="SELECT SUBSTRING(xl_body,30,39) FROM XLATEP WHERE xl_system='GEN' AND xl_prefix='EE' OR
xl_prefix='EI'"

      # Execute query and store results
      results=$(mysql -h "$DBHOST" -u "$MYSQLUSER" -p"$MYSQLPASS" -D "${SITE}_classic" -e "$QUERY" --batch
--silent)

      # Check if query was successful
      if [ $? -ne 0 ]; then
        log "Error executing MySQL query" "ERROR" ${RED}${BOLD}${FLASH}
```

```
        sleep 2
        main_menu
      fi

      # Read results line by line
      while IFS=$'\t' read -r -a columns; do

        mkdir -p ${columns[0]}
        log "Created the following folder : ${columns[0]}" "INFO" ${BLUE}${BOLD}

      done <<< "$results"

#      setup_counter
      . $KIWIBASE/corp/bin/stdprofile
      break
    done
}
#
# Execute
checkVariables
main_menu
#
#if [ -f ~/.bashrc ]; then
#        . ~/.bashrc
#fi
#
#THIS MUST BE AT THE END OF THE FILE FOR SDKMAN TO WORK!!!
#export SDKMAN_DIR="/KIWI/home/remuser//.sdkman"
#[[ -s "/KIWI/home/remuser//.sdkman/bin/sdkman-init.sh" ]] && source
"/KIWI/home/remuser//.sdkman/bin/sdkman-init.sh"
#
#exit 0
```

Script to be added/created to the $CORPBIN of the server

Edit the following script

```
vi /KIWI/corp/bin/stdprofile
```

Add the following

```
#!/bin/bash
umask 002

if [ -r $KIWI/site/KIWIENV ]; then
   COMM_DIR=`cat /KIWI/site_$PLANTID/site/KIWIENV | grep "^COMMSDIR:" | cut -d":" -f2`
   AB_DIR=`cat /KIWI/site_$PLANTID/site/KIWIENV | grep "^HOMEAB:" | cut -d":" -f2`
fi

export KIWIBASE=/KIWI
export KIWI=${KIWI:-/KIWI/site_$PLANTID}
export BACKUPS=${BACKUPS:-${KIWIBASE}/backups}
export BIN=${BIN:-${KIWI}/bin}

# COMMS
export COMMSDIR=${COMMSDIR:-${KIWI}/work/cscomm}
export COMM_DIR=${COMM_DIR:-COMMSDIR}
export COMMS_PRINTER=98

export CORPBIN=${CORPBIN:-${KIWIBASE}/corp/bin}
export CORPDAT=${CORPDAT:-${KIWIBASE}/corp/dat}
export DATA=${DATA:-${KIWI}/data_$PLANTID}
export HOMEAB=${HOMEAB:-${KIWI}/work/mdcxmt}
export HOMECO=${HOMECO:-${KIWI}/work/corrop}
export HOMECONV=${HOMECONV:-${KIWI}/work/ducker}
export PROGS=${PROGS:-${KIWI}/progs}
export SCP=${SCP:-${KIWI}/scp}
export SQL=${SQL:-${KIWI}/sql}
export UFD=${UFD:-${KIWI}/ufd}
export SITEBIN=${SITEBIN:-${KIWI}/site/bin}
export SITEDAT=${SITEDAT:-${KIWI}/site/dat}
export WORK=${WORK:-${KIWI}/work/${LOGNAME}}
export ftpdir_ord=/mnt/ducker/ordinfo
export ftpdir_seq=/mnt/ducker/corrseq
export ftpdir_ols=/mnt/ducker/ol
export ftpdir_stk=/mnt/ducker/pidnumber
export ftpdir_load=/mnt/ducker/cvm_enter
export DCV_VERSION=3.01
export USESAMBA=1
```

```
# VUE Directories
export VUE=${VUE:-${KIWIBASE}/services/sites/$PLANTID/current}
export JCSC=${VUE:-${KIWIBASE}/services/sites/$PLANTID/current}
export VBIN=${VBIN:-${KIWIBASE}/services/sites/$PLANTID/current/bin}
export VLOG=${VLOG:-${KIWIBASE}/services/sites/$PLANTID/current/logs}
export VCONF=${VCONF:-${KIWIBASE}/services/sites/$PLANTID/current/conf}
export KBIN=${KBIN:-${KIWIBASE}/services/etl/kdw/current/kiwiplan/bin}
export KLOG=${KLOG:-${KIWIBASE}/services/etl/kdw/current/kiwiplan/logs}
export KCONF=${KCONF:-${KIWIBASE}/services/etl/kdw/current/kiwiplan/conf}

export PS1="\\[$(tput bold)\\]\\[$(tput setaf 1)\\]\\[$(tput setab 8)\\]
"\\u\\[$(tput setaf 5)\\]@\\[$(tput setaf 1)\\]`hostname`"\\[$(tput setaf 2)\\] : ""`uname`"" : \\[$(tput setaf 6)\\]""\d
\t : ""`echo $PLANTID`""
"[\\w]'\$\[$(tput sgr0)\] '
export KWSQL_COMMONDIR=${KWSQL_COMMONDIR:-${DATA}}
#export LANGUAGE=${LANGUAGE:-{KIWI_LANG}}
export KWSQL_USER=${KWSQL_USER:-kiwisql}
export MYSQL_PWD=${MYSQL_PWD:-800486kiwi}
export EXEC=${EXEC:-""}

# Has to come after the $PROGS definition
export LD_LIBRARY_PATH=${PROGS}

#export HOMELD=${HOMELD:-${KIWI}/work/autotags}
#export HOMECONVM=${HOMECONVM:-${KIWI}/work/minda}
#export HOMECONVS=${HOMECONVS:-${KIWI}/work/mindas}

export KIWISEA=".:$WORK:$HOME:$DATA:$SITEDAT:$SITEBIN:$CORPDAT:$CORPBIN:$BIN:$PROGS:$SCP:$SQL"
export PATH=".:$WORK:$HOME:$SITEBIN:$CORPBIN:$BIN:$PROGS:/bin:/usr/bin:/usr/sbin"
export JAVA_HOME=/usr/lib/jvm/jre-11

#if [ -z "${WORK}" ]; then
#  echo "Variable WORK does not exist. WORK: ${WORK}"
#  sleep 2
#  exit
#fi

if [ ! -d ${WORK} ]; then
  mkdir -p ${WORK}
  if [ ! -d ${WORK} ]; then
```

```
    clear
    echo -e "ERROR: Cannot create work directory : ${WORK}"
    echo -e "Press <Enter> to continue ... \c"
    read key
    exit 1
  fi
fi

cd $WORK

if [ -z "${EXEC}" ]; then
  exit
else
  ${EXEC}
fi
```

Revision #9
Created 31 January 2025 04:10:48 by Steve Ling
Updated 6 June 2025 05:05:56 by Steve Ling