
Subject: Re: IDL 5.2 crashes with Linux Mandrake 6.0
Posted by [Christian Marquardt](#) on Tue, 15 Jun 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I just ran into the same stuff today... I believe a couple of people have suggested a solution in the past, targeted at the original Redhat 6.0 distribution.

Here's what *seems* (see below) works for Mandrake:

First, get the following two packages:

- compat-glibc-5.2-2.0.7.1.i386.rpm
- compat-libs-5.2-1.i386.rpm

and install them as root. You'll find them on any redhat mirror (like ftp.redhat.com or ftp.uni-bayreuth.de) in the RPMS directory of the distribution CD. For some reason, they are not included in the Mandrake distribution.

Then, it is necessary to ensure that a) the correct (old) version of the libraries is used and the correct (old) version of the loader is used to start up the idl-binary (not the shell-script in /usr/local/rsi/idl_5.2/bin). I have patched that shell script to ensure both the correct setting for LD_LIBRARY_PATH and the startup of the idl binary with the old loader, and attach both the new shell script as well as a diff to this mail. Be careful not to overwrite the old version!

Finally, this pathed script brings idl (and idlde and idlhelp and idldemo) up, *BUT* - as Laurent Chardon stated a while ago - breaks everything inside IDL which uses functions of the operating system, e.g. findfile and spawning shell command from inside IDL. This can be circumvented by adding the line

```
setenv, 'LD_LIBRARY_PATH='
```

in idlstartup.pro - which resets the environment again to make operating system accesses possible. (NB: It might be necessary to let /usr/local/rsi/idl_5.2/bin/bin.linux in the library search path - any comments???)

This workaround seems to work for me, but: I have only tested this for an hour or so, I have not used widget programs (apart from xloadct), I have not used live tools or insight. Accessing netCDF's, however, works as well (or so it seems to me)

Could someone verify if this works? And also let people at RSI (and Creaso, who

sells IDL in Germany) know that it would really be nice to sell something that works?

Hope this helps,

Chris.

--

Christian Marquardt

GFZ Potsdam | Tel.: (+49) 331-288-1168
Am Telegrafenberg C2 | Fax.: (+49) 331-288-1169
D-14473 Potsdam | Email: marq@gfz-potsdam.de

*** /usr/local/rsi/idl_5.2/bin/idl.orig Mon Nov 2 18:23:09 1998

--- /usr/local/rsi/idl_5.2/bin/idl Tue Jun 15 22:29:41 1999

*** 310,316 ****

exit 1

fi

! exec \$IDL_DIR/bin/bin.\$OS\$OSVER\$ARCH/\$APPLICATION \$* \$APP_ARGS

We shouldn't get here unless there was an error.

echo "\$APPLICATION is not available for this system (\$OS/\$ARCH)"

--- 310,320 ----

exit 1

fi

! if ["\$APPLICATION" = "idlhelp"]; then

! exec \$IDL_DIR/bin/bin.\$OS\$OSVER\$ARCH/\$APPLICATION \$* \$APP_ARGS

! else

! LD_LIBRARY_PATH=/usr/i386-glibc20-linux/lib:\$LD_LIBRARY_PATH exec

/usr/i386-glibc20-linux/lib/ld-linux.so.2 \$IDL_DIR/bin/bin.\$OS\$OSVER\$ARCH/\$APPLICATION \$* \$APP_ARGS

! fi

We shouldn't get here unless there was an error.

echo "\$APPLICATION is not available for this system (\$OS/\$ARCH)"

#!/bin/sh

#

\$Id: startapp,v 1.49 1998/09/28 17:28:42 kirk Exp \$

#

This script determines the operating system and hardware combination

and overlays itself with the correct binary for the desired program.

The program is determined from the name by which the script is invoked.

```

APPLICATION=`basename $0`
APP_ARGS=
IDL_VERSION=5.2
ENVI_VERSION=3.1
ION_VERSION=1.1
LICENSE_NAME=license.dat

INSTALL_DIR=/usr/local/rsi

INS_TYPE="IDL"

# Use tr in /bin or /usr/bin if available
TR=tr
if [ -x /usr/bin/tr ]; then
    TR=/usr/bin/tr
fi

PRODUCT=`echo $INS_TYPE | $TR '[A-Z]' '[a-z]'`

if [ "$PRODUCT" = "envi" ]; then
    PROD_VERSION=$ENVI_VERSION
fi

if [ "$PRODUCT" = "ion" ]; then
    PROD_VERSION=$ION_VERSION
fi

if [ "$PRODUCT" != "idl" ]; then
    IDLDIRECTORY_VERS=${PRODUCT}_${PROD_VERSION}/idl_${IDL_VERSION}
    IDLDIRECTORY=${PRODUCT}/idl
else
    IDLDIRECTORY_VERS=idl_${IDL_VERSION}
    IDLDIRECTORY=${PRODUCT}/idl
fi

# Some applications can be invoked with or without a version suffix.
# Recognise the versioned ones and remove the version.
case $APPLICATION in
    "idl_${IDL_VERSION}") APPLICATION=idl;;
    "idlde_${IDL_VERSION}") APPLICATION=idlde;;
    "idlhelp_${IDL_VERSION}") APPLICATION=idlhelp;;
    "idldemo_${IDL_VERSION}") APPLICATION=idldemo;;
    "envihelp_${ENVI_VERSION}") APPLICATION=envihelp;;
    "iond_${ION_VERSION}") APPLICATION=iond;;
    "ionstat_${ION_VERSION}") APPLICATION=ionstat;;
    "iondown_${ION_VERSION}") APPLICATION=iondown;;

```

```
"ion_httpd_$ION_VERSION") APPLICATION=ion_httpd;;
esac
```

```
if [ "$APPLICATION" = "envihelp" ]; then
```

```
# Find the main ENVI directory
```

```
if [ "$ENVI_DIR" = "" ]; then
```

```
    TMP_DIR=`echo $IDL_DIR | awk -FV '{for(i=1;i<NF;i++){if(i>1)printf("%s", "/"); printf("%s", $i); }printf("\n")}'`
```

```
    for DIR in $INSTALL_DIR/envi_$ENVI_VERSION $TMP_DIR
    /usr/local/rsi/envi_$ENVI_VERSION ../idl /usr/local/lib/envi /usr/local/envi /usr/local/bin/envi
    do
        if [ -d $DIR ]; then
            if [ -f $DIR/save/envi_m.sav ]; then
                ENVI_DIR=$DIR
                break
            fi
        fi
    done
fi
```

```
if [ "$ENVI_DIR" = "" ]; then
```

```
    echo "Unable to access $APPLICATION. You will have to define the ENVI_DIR
environment variable to point at the main ENVI directory before
it will work."
```

```
    exit 1
```

```
elif [ "$IDL_DIR" = "" ]; then
```

```
    IDL_DIR=$ENVI_DIR/idl_$IDL_VERSION
```

```
fi
```

```
fi
```

```
if [ "$APPLICATION" = "iond" -o "$APPLICATION" = "ionstat" -o "$APPLICATION" = "
iondown" -o "$APPLICATION" = "ion_httpd" ]; then
```

```
# Find the main ION_DIR directory
```

```
if [ "$ION_DIR" = "" ]; then
```

```
    TMP_DIR=`echo $IDL_DIR | awk -FV '{for(i=1;i<NF-1;i++){if(i>1)printf("%s", "/"); printf("%s", $i); }printf("\n")}'`
```

```
    for DIR in $INSTALL_DIR/ion_$ION_VERSION $TMP_DIR/ion_$ION_VERSION
    /usr/local/rsi/ion_$ION_VERSION ../i on_$ION_VERSION /usr/local/lib/ion /usr/local/ion
    /usr/local/bin/ion
```

```

do
  if [ -d $DIR ]; then
    if [ -f $DIR/bin/iond ]; then
      ION_DIR=$DIR
      break
    fi
  fi
done
fi

if [ "$ION_DIR" = "" ]; then
  echo "Unable to access $APPLICATION. You will have to define the ION_DIR
environment variable to point at the main ION directory before
it will work."
  exit 1
elif [ "$IDL_DIR" = "" ]; then
  IDL_DIR=$ION_DIR/idl_$IDL_VERSION
fi
fi

# Find the main IDL directory
if [ "$IDL_DIR" = "" ]; then
  for DIR in $INSTALL_DIR/$IDLDIRECTORY_VERS /usr/local/rsi/$IDLDIRECTORY_VERS .
./idl_$IDL_VERSION /usr/local/lib/$IDLDIRECTORY /usr/local/$IDLDIRECTORY
/usr/local/bin/$IDLDIRECTORY
do
  if [ -d $DIR ]; then
    if [ -f $DIR/resource/fonts/hersh1.chr ]; then
      IDL_DIR=$DIR
    break
  fi
  fi
done
fi

if [ "$IDL_DIR" = "" ]; then
  echo "Unable to access $APPLICATION. You will have to
define the IDL_DIR environment variable to point at the main
IDL directory before it will work."
  exit 1
fi

# If LM_LICENSE_FILE is not defined and one of the following exists
# $IDL_DIR/../../license/license.dat (license dir from base idl product)
# $IDL_DIR/../../license/license.dat (license dir from bundled product)
# then define LM_LICENSE_FILE. If file is not found, leave it unset.
# If LM_LICENSE_FILE is already set, then respect that and leave it alone.
if [ \( "$LM_LICENSE_FILE" = "" \) ]; then

```

```

if [ \( -f $IDL_DIR/./license/$LICENSE_NAME \) ]; then
#   default location for standard IDL distribution
  LM_LICENSE_FILE=$IDL_DIR/./license/$LICENSE_NAME
  export LM_LICENSE_FILE
elif [ \( -f $IDL_DIR/././license/$LICENSE_NAME \) ]; then
#   default location for standard bundled distribution (ENVI, ION, etc.)
  LM_LICENSE_FILE=$IDL_DIR/././license/$LICENSE_NAME
  export LM_LICENSE_FILE
fi
fi

# Determine the operating system, hardware architecture, and os release
# Make sure these agree with IDL's compiled in paths or online help
# won't be found.
OS=
ARCH=
OSVER=
case `uname` in

  "SunOS")
if [ -d /proc ]; then # Solaris
  OS="solaris"
  OSVER="2"
  BIN_ARCH=/usr/bin/arch
  if [ -x /usr/bin/arch ]; then
    BIN_ARCH=/usr/bin/arch
  elif [ -x /usr/ucb/arch ]; then
    BIN_ARCH=/usr/ucb/arch
  fi
  if [ ` $BIN_ARCH ` = i86pc ]; then
    ARCH=".x86"
  else
    ARCH=".sparc"
    # If there is a special version of the program for the Ultra,
    # use it. Otherwise, use the regular sparc version.
    if [ \( -f $IDL_DIR/bin/bin.$OS${OSVER}.sun4u/$APPLICATION \) \
      -a \( ` $BIN_ARCH -k ` = sun4u \) ]; then
      ARCH=".sun4u"
    fi
  fi
else # 4.1
  echo "$APPLICATION: System architecture is not supported."
  exit 1
fi
;;

  "AIX") OS="ibm";;

```

```

"HP-UX") OS="hp";;

"IRIX") OS="sgi";;

"IRIX6") OS="sgi";;

"IRIX64") OS="sgi";;

"OSF1") OS="alpha";;

"ULTRIX") OS="ultrix";;

"Linux") OS="linux";;

*)
echo "$APPLICATION: Unable to recognise system architecture."
exit 1
;;

esac

# Add the bin directory to the library search path
case $OS in
  "ibm")
    if [ "$LIBPATH" = "" ]; then
      LIBPATH="/lib:/usr/lib:$IDL_DIR/bin/bin.$OS$OSVER$ARCH:$IDL_DIR/bin/b
in.$OS$OSVER$ARCH/dm/lib "
    else
      LIBPATH="/lib:/usr/lib:$IDL_DIR/bin/bin.$OS$OSVER$ARCH:$IDL_DIR/bin/b
in.$OS$OSVER$ARCH/dm/lib:$LIBPATH "
    fi
  # Append the lib directory of oracle on if ORACLE_HOME set for Dataminer
  if [ "$ORACLE_HOME" != "" ]; then
    LIBPATH="$ORACLE_HOME/lib:$LIBPATH"
  fi
  # Append the lib directory of sybase on if Sybase set for Dataminer
  if [ "$SYBASE" != "" ]; then
    LIBPATH="$SYBASE/lib:$LIBPATH"
  fi
  export LIBPATH
  ;;

  "hp")
    if [ "$SHLIB_PATH" = "" ]; then
      SHLIB_PATH="/usr/lib/X11R5:/usr/lib/Motif1.2:$IDL_DIR/bin/bin.$OS$OSVER$
ARCH:$IDL_DIR/bin/bin.$OS$OSVER$ARCH/dm/lib "
    else
      SHLIB_PATH="/usr/lib/X11R5:/usr/lib/Motif1.2:$IDL_DIR/bin/bin.$OS$OSVER$

```

```

ARCH:$IDL_DIR/bin/bin.$OS$OSVER$ARCH/dm/lib:$SHLIB_PATH "
fi

# Append the lib directory of sybase on if Sybase set for Dataminer
  if [ "$SYBASE" != "" ]; then
    SHLIB_PATH="$SYBASE/lib:$SHLIB_PATH"
  fi
export SHLIB_PATH
;;

"solaris")
if [ "$LD_LIBRARY_PATH" = "" ]; then
  LD_LIBRARY_PATH="
$IDL_DIR/bin/bin.$OS$OSVER$ARCH:$IDL_DIR/bin/bin.$OS$OSVER$A
RCH/dm/lib:/usr/openwin/lib:/usr/dt/lib "
else
  LD_LIBRARY_PATH="
$IDL_DIR/bin/bin.$OS$OSVER$ARCH:$IDL_DIR/bin/bin.$OS$OSVER$A
RCH/dm/lib:/usr/openwin/lib:/usr/dt/lib:$LD_LIBRARY_PATH "
fi

# Append the lib directory of sybase on if Sybase set for Dataminer
  if [ "$SYBASE" != "" ]; then
    LD_LIBRARY_PATH="$SYBASE/lib:$LD_LIBRARY_PATH"
  fi
export LD_LIBRARY_PATH
;;

*)
if [ "$LD_LIBRARY_PATH" = "" ]; then
  LD_LIBRARY_PATH="$IDL_DIR/bin/bin.$OS$OSVER$ARCH"
else
  LD_LIBRARY_PATH="$IDL_DIR/bin/bin.$OS$OSVER$ARCH:$LD_LIBRARY_PATH "
fi
export LD_LIBRARY_PATH
;;
esac

# Users Libraries
if [ "$IDL_PATH" = "" ]; then
  IDL_PATH="+$IDL_DIR/lib:+$IDL_DIR/examples"
fi

# Add the IDL bin directory to the path so that idlde will always find idl
PATH=$IDL_DIR/bin:$PATH
export PATH IDL_DIR IDL_PATH

```

```

# HyperText help requires special setup
# Set up environment variables for HyperHelp.
HHHOME=$IDL_DIR/help
HHPATH=$IDL_DIR/bin/bin.$OS$OSVER$ARCH
XPPATH=$IDL_DIR/resource/xprinter
if [ "$XLIBI18N_PATH" = "" ]; then
  XLIBI18N_PATH=$IDL_DIR/resource/X11/lib
fi
if [ "$XFILESEARCHPATH" = "" ]; then
  XFILESEARCHPATH=$IDL_DIR/resource/X11/lib/app-defaults/%N%S:
$IDL_DIR/resource/X11/lib/app-defaults/%T/%N%S
else
  XFILESEARCHPATH=" $IDL_DIR/resource/X11/lib/app-defaults/%N%S:$IDL_DIR/resourc
e/X11/lib/app-defaults/%T/%N%S:$XFILESEARCHPATH "
fi
export HHHOME HHPATH XPPATH XLIBI18N_PATH XFILESEARCHPATH

if [ "$APPLICATION" = "idlhelp" ]; then
  APP_ARGS=$IDL_DIR/help/idl.hlp
fi

if [ "$APPLICATION" = "envihelp" ]; then
  APPLICATION=idlhelp
  APP_ARGS=$ENVI_DIR/help/envi.hlp
fi

if [ "$APPLICATION" = "iond" -o "$APPLICATION" = "ionstat" -o "$APPLICATION" = "iondown" -o
"$APPLICATION" = "ion_httpd" ]; then
  exec $ION_DIR/bin/bin.$OS$OSVER$ARCH/$APPLICATION $* $APP_ARGS

  # We shouldn't get here unless there was an error.
  echo "$APPLICATION is not available for this system ($OS/$ARCH)"
  exit 1
fi

if [ "$APPLICATION" = "idlhelp" ]; then
  exec $IDL_DIR/bin/bin.$OS$OSVER$ARCH/$APPLICATION $* $APP_ARGS
else
  LD_LIBRARY_PATH=/usr/i386-glibc20-linux/lib:$LD_LIBRARY_PATH exec
/usr/i386-glibc20-linux/lib/ld-linux.so.2 $IDL_DIR/bin/bin.$OS$OSVER$ARCH/$APPLICATION $*
$APP_ARGS
fi

# We shouldn't get here unless there was an error.
echo "$APPLICATION is not available for this system ($OS/$ARCH)"
exit 1

```

File Attachments
