Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by Michael Wallace on Thu, 11 Mar 2004 21:07:04 GMT
View Forum Message <> Reply to Message

> just create two soft links:
>
> ln -s /usr/local/rsi/idl_5.5/bin/idl /usr/local/bin/idl55
> ln -s /usr/local/rsi/idl_6.0/bin/idl /usr/local/bin/idl60
>
> and use the idl55 and idl60 commands (idl is a script itself, which sets
> the appropriate env. variables).

In this case, you still have the issue of what IDL_PATH is set to. If left unset, the soft links solution you suggest will work. IDL will automagically determine the correct lib directory to use.

However, almost all of us have to explicitly set IDL_PATH in order to include our own libraries and other procedures. Because of this we are forced to choose between the $RSI_DIR/idl_5.5/lib directory and the $RSI_DIR/idl_6.0/lib directories. And you can't set IDL_PATH to include both directories since one will clobber the other.

So, with that in mind, the better solution is to symlink the IDL directories rather than the executables. By using a symlink that points to idl_5.5 or idl_6.0 or other directory, we don't have the IDL_PATH issue because we just have to only include $RSI_DIR/idl/lib in IDL_PATH. So, all we have to do is change the symlink and we're ready to go.

I hacked up a quick little script in the last few minutes to do the recreation of the symlink for us. I make no guarantees about the quality. I show it here as an example only.

```
#!/bin/bash

#

# Usage: switchidl version_number
#
# Example: switchidl 5.6
#
# This script will recreate the idl symlink on systems where multiple
# versions of IDL are installed. Two assumptions are made:
#   1. All IDL installations are in the $RSI_DIR directory
#   2. All IDL installations are in directories of the form idl_xxx
#      where xxx is the version number
#
```

```
# If RSI_DIR is not set, use the IDL default
if [ ! -n "$RSI_DIR" ]; then
    RSI_DIR=/usr/local/rsi
fi


# Check for the version number argument
if [ ! -n "$1" ]; then
    echo Usage: switchidl version_number
    exit 1
fi


# Check if requested IDL directory exists
if [ ! -e $RSI_DIR/idl_$1 ]; then
    echo IDL $1 could not be found
    exit 2
fi


# Check if IDL symlink already exists
if [ -e $RSI_DIR/idl ]; then


    # Delete the symlink, if possible
    if [ -w $RSI_DIR/idl ]; then
       rm $RSI_DIR/idl
    else
       echo Incorrect permissions -- could not alter symlink
       exit 3
    fi
else


    # Check if it's possible to write to $RSI_DIR
    if [ ! -w $RSI_DIR ]; then
       echo Incorrect permissions -- could not alter symlink
       exit 3
    fi
fi

# All checks are satisfied

# Move to $RSI_DIR, create the link and return
owd=`pwd`
cd $RSI_DIR
ln -s idl_$1 idl
```

```
cd $owd
```