Subject: Multiple journal files on linux
Posted by markmcg4 on Fri, 14 Jun 2002 09:13:18 GMT
View Forum Message <> Reply to Message

Hey all,

Is there a way of creating a different journal file for each IDL session that I start? I tend to do a lot of command line editing first before I am happy to put code in a procedure/function and I tend to forget what I have done after several weeks. Journal files are usually small so storing them would not really be a problem (though a simple script could be written to remove old files or even archive them!).

I know about the history recall available in IDLWAVE, but here we have a linux cluster, so I could be running IDL on several different machines over a period of time (or several sessions on the same machine) and from experience I think it is the last IDLWAVE session that is ended that becomes the history file.

Obviously if the fisrt bit could be sorted then it would be trivial to incorpoate it into IDLWAVE (I guess)!

Slï¿½n, Mark.

Subject: Re: Multiple journal files on linux
Posted by Craig Markwardt on Wed, 19 Jun 2002 13:34:21 GMT
View Forum Message <> Reply to Message

markmcg4@hotmail.com (Mark McGrath) writes:

> Thanks David, that is almost what I was looking for.
>
> This routine makes an `almost' unique file. An example:
> Running IDL on a linux cluster, as we do here, if the machine clocks
> are a litle out of sync it would be possible to over write the file
> that had been opened already, no? (Not much info lost there though,
> but the more serious case of the clocks being a lot out of sync then
> large amounts of info in the journal file could be lost.)
>
> I have hunted about a bit and come across this:
> Add it to .cshrc/.tcshrc or equiv:
>

I don't understand. Why not do the logic within your IDL_STARTUP file, in which case you can execute arbitrary IDL code? You would modify David's JOURNAL_UNIQUE procedure to test that the file exists before opening it, and if it does, append a unique suffix or whatever,

and then retest. No need for alias-nasties!

Good luck,
Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: Multiple journal files on linux
Posted by Ken Mankoff on Wed, 19 Jun 2002 15:57:55 GMT
View Forum Message <> Reply to Message

On 19 Jun 2002, Mark McGrath wrote:
> This routine makes an `almost' unique file. An example:
> Running IDL on a linux cluster, as we do here, if the machine clocks
> are a litle out of sync it would be possible to over write the file
> that had been opened already, no? (Not much info lost there though,
> but the more serious case of the clocks being a lot out of sync then
> large amounts of info in the journal file could be lost.)
>
> Any ideas, anyone?

I ran into the problem of making a unique directory on a
multiprocessor machine, where a timestamp may not suffice...

Use the unique process ID. Each IDL session will have a unique PID, no matter how many processors and what the time is on each machine... I use a combination, the time (in msec), then a "_", then the PID.

I believe you can still have multiple processes assigned the same time and PID if you are using networked computers (as opposed to 1 computer with multiple processors), but the chances, especially when using msec and not sec, drop to VERY low.

-k.

Subject: Re: Multiple journal files on linux
Posted by Paul Van Delst[1] on Wed, 19 Jun 2002 16:21:19 GMT
View Forum Message <> Reply to Message

Ken Mankoff wrote:
>

> On 19 Jun 2002, Mark McGrath wrote:
>> This routine makes an `almost' unique file. An example:
>> Running IDL on a linux cluster, as we do here, if the machine clocks
>> are a litle out of sync it would be possible to over write the file
>> that had been opened already, no? (Not much info lost there though,
>> but the more serious case of the clocks being a lot out of sync then
>> large amounts of info in the journal file could be lost.)
>>
>> Any ideas, anyone?
>
> I ran into the problem of making a unique directory on a
> multiprocessor machine, where a timestamp may not suffice...
>
> Use the unique process ID. Each IDL session will have a unique PID, no
> matter how many processors and what the time is on each machine... I
> use a combination, the time (in msec), then a "_", then the PID.
>
> I believe you can still have multiple processes assigned the same time
> and PID if you are using networked computers (as opposed to 1 computer
> with multiple processors), but the chances, especially when using msec
> and not sec, drop to VERY low.

I do the following for multiple job submissions on a multi CPU machine in a shell script. I didn't want the PID to be involved and just use a simple counter suffix. I've only got access to 8 CPUS at once so I've never exceeded the (arbitrary) max value of 10 directories/files created at the same time.

Works quite well.

```
# --------------------------
# Get the start date and time
# --------------------------

LBLRUN_DATE=`date '+%Y%m%d'`
LBLRUN_TIME=`date '+%H%M%S%Z'`


# ------------------------------------------------
# Create a root definition for the LBLRUN_TAG variable
# ------------------------------------------------

 ROOT_LBLRUN_TAG='_'`whoami`'_'${LBLRUN_DATE}'_'${LBLRUN_TIME }


#  ------------------------------------------------------------ ----
# Make sure the directory doesn't already exist by suffixing
# the name with an _ and an integer identifier.
```

```
# Need to check this just in case the time and dates were assigned
# at the same time.
#  ------------------------------------------------------------ ----

LBLRUN_SUFFIX=0
while :
do

  # -- Suffix the directory with a number
  LBLRUN_SUFFIX=`expr ${LBLRUN_SUFFIX} + 1`
  LBLRUN_TAG=${ROOT_LBLRUN_TAG}'_'${LBLRUN_SUFFIX}
  LBL_WORK=${LBL_RUN_ROOT}'/.lblrtm'${LBLRUN_TAG}

  # -- Create the work directory
  mkdir ${LBL_WORK}

  # -- If successful, exit this loop
  if [ $? -eq 0 ]; then
    break
  fi

  # -- Otherwise keep trying to create a work directory (assuming
  # -- the creation failed since it already exists).
  # -- If we have reached a nesting of 10, maybe something is wrong?
  if [ ${LBLRUN_SUFFIX} -gt 10 ]; then
    echo "Error occurred creating work directory (based on .lblrtm${ROOT_LBLRUN_TAG})"
    date
    echo "${LBL_RUN_ROOT} directory listing follows:"
    ls -laF ${LBL_RUN_ROOT}
    exit 2
  fi

done
```

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC Beer is good.
Ph: (301)763-8000 x7274 My wife.
Fax:(301)763-8545

Subject: Re: Multiple journal files on linux
Posted by R.Bauer on Mon, 24 Jun 2002 13:59:02 GMT
View Forum Message <> Reply to Message

Mark McGrath wrote:
>

> Hey all,
>
> Is there a way of creating a different journal file for each IDL
> session that I start? I tend to do a lot of command line editing first
> before I am happy to put code in a procedure/function and I tend to
> forget what I have done after several weeks. Journal files are usually
> small so storing them would not really be a problem (though a simple
> script could be written to remove old files or even archive them!).
>
> I know about the history recall available in IDLWAVE, but here we have
> a linux cluster, so I could be running IDL on several different
> machines over a period of time (or several sessions on the same
> machine) and from experience I think it is the last IDLWAVE session
> that is ended that becomes the history file.
>
> Obviously if the fisrt bit could be sorted then it would be trivial to
> incorpoate it into IDLWAVE (I guess)!
>
> Slï¿½n, Mark.

Dear Mark,

I am not sure if it would help to have several different journals

To begin journaling to the file myjournal.pro, enter:JOURNAL,
'myjournal.pro'

To show on which machine it is done you can use the hostname and the
clock.
I believe this is different enough.

Myself I am working many times interactive too. All what I am evaluating
becomes a procedure or function with a lot of comments.
The Journal has no comments how did you remember later what's you like
to
have and what's not?

Normally I start in idlde or emacs with a procedure declaration and a
stop.
All what's I like to do are defined interactively. If it works I copy
this
to the procedure. The procedure runs again and stops somewhere else
where I like to add some lines and so on. After the routine is finished
I take a look on it if it is better a function or a procedure.

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
---------- -------
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml
========== =======