Subject: IDL FAQ - new maintainer
Posted by Mike Schienle on Tue, 03 Jun 1997 07:00:00 GMT
View Forum Message <> Reply to Message

IDL (Interactive Data Language) FAQ

Frequently Asked Questions about the
Interactive Data Language (IDL).
---------- ----------=

Archive-name: idl-faq
Last-modified: 1997 May 30
Version: 4.0

Latest IDL FAQ: See Appendix A02.
FAQ maintainer: Mike Schienle
Email: mgs@sd.cybernex.net

Changes in version 4.00:

New FAQ Maintainer and FAQ Location - A02
Updates to Introduction
Change to email address for J-F Pitot de La Beaujardiere - A03
Gary Kushner's IDL data acquisition web page location update - G13
Change to current version of IDL - G04
Change to next version of IDL - G11
Added David Fanning's FTP site - G07
Added David Fanning's training info - G12
Clarification to "Does case matter" - T13

Previous changes may be found in Appendix A04.
---------- ----------=

Introduction

This is a list of Frequently Asked Questions about the Interactive Data
Language or IDL. These questions pop up fairly regularly in the newsgroup
comp.lang.idl-pvwave. This list is an attempt to cut down on net traffic
regarding commonly asked questions. Users are encouraged to read through
this list before posting a query to the newsgroup.

As of 1997 May 15 Mike Schienle has been maintaining the IDL FAQ. Ray

Sterner was handling this previously, and Mike Schienle before him and Patrick Ryan before him; much of the material and many of the comments here
were compiled by them.

A Note from the editor about PV~WAVE:
I have nearly as much experience with PV~WAVE as I do with IDL, and was
maintaing the PV-WAVE FAQ at one time. I felt obligated to cease
maintaining the PV-WAVE FAQ when I began some contract work for RSI. I
will try to minimize comments which appear to favor one package or the
other. If someone is interested in maintaing the PV-WAVE FAQ please
contact me.

Contents

This list is roughly divided into two categories: general questions and
technical questions. General question numbers are prefixed with a G and
technical ones with a T

General questions

Technical questions

---------- ----------=

GENERAL QUESTIONS:

---------- ----------=

G01. What is IDL?

IDL is the Interactive Data Language. It is a product of Research
Systems,
Inc. (RSI).

The following is quoted from the README file at
rsinc.com:/pub/idl/README.
As such, it describes IDL's capabilities in an understandably
subjective
manner. ;-)

    IDL, Interactive Data analysis Language, is a complete package
for
    the interactive reduction, analysis, and visualization of
    scientific data and images. Optimized for the workstation
    environment, IDL integrates a responsive array oriented language
    with numerous data analysis methods and an extensive variety of
    two and three dimensional displays into a powerful tool for
    researchers.

    IDL supports an extensive data import capability, publication
    quality hard copy output, and user-defined Motif graphical user
    interfaces.

    Users can create complex visualizations in hours instead of weeks
    with the aid of IDL's high level capabilities and interactive
    environment.

IDL is useful in physics, astronomy, image and signal processing, mapping, medical imaging, statistics, and other technical disciplines requiring visualization of large amounts of data.

Here is a short history of RSI:
[attributed to ali@rsinc.com (Ali Bahrami)]

IDL is a product of Research Systems, Inc., founded in 1977 by David Stern. The origins of IDL were developed at the Laboratory for Atmospheric and Space Physics (LASP) at the University of Colorado. David was one of the people involved in efforts to make computers easier to use for the physicists at the Lab. The first program in the evolutionary chain to IDL was named Rufus (named after Dave's dog). Rufus was a very simple vector oriented calculator that ran on the PDP-12. It accepted 2 letter codes that specified (1) An arithmetic operation (2) The input registers to serve as operands, and (3) the destination register. The next version was the Mars Mariner Spectrum Editor (MMED) which was a version of Rufus that ran on the PDP-8.

The next program in this line was named SOL, and it also ran on the PDP-8. Unlike its predecessors, SOL was a real computer language with a real syntax (no more 2 letter codes). It was an APL influenced array oriented language with some primitive graphics capabilities. The resemblance to IDL was there, but very faintly.

In 1977, Dave left LASP to start Research Systems Inc. (RSI) with the intention of building on the ideas contained in SOL. The initial result of this endeavor was PDP-11 IDL, which was much more capable than SOL. Graphics was usually done on Tektronix terminals and outboard raster graphics displays. I used this version at LASP in 1981 on a PDP11/34 under RSX-11M in 1981 (I worked as a student at LASP from 1981 to 1987). I didn't use it for very long though, because 1981 was the year that Dave released the VAX/VMS version of IDL. This version, which was written in VAX-11 MACRO and FORTRAN, took advantage of the VAX virtual memory and 32-bit address space, and was a huge step beyond the PDP-11 version. It used essentially the same sort of graphics hardware as the PDP-11.

In 1987, Dave decided that Unix workstations were the direction in

which IDL should progress, but porting the current VAX IDL to
Unix
didn't make much sense because of its MACRO and FORTRAN
implementation. I had just finished my Masters degree and was
looking for work. Dave hired me and together we wrote the current
version of IDL for Unix on the Sun 3 taking advantage of the
re-write to extend and improve the language. Since then, we've
ported it to many Unix machines and moved it back to VMS. RSI has
many other employees now, but our focus is still the continued
development of IDL. Recently, IDL was ported to PC class systems
running Microsoft Windows.

---------- ----------=

G02. Where can I contact them?

Their address is:

Research Systems, Inc.
2995 Wilderness Place
Boulder, CO 80301
(303) 786-9900 (Voice)
(303) 786-9909 (Fax)
=A0
Email:
=A0
info@rsinc.com or support@rsinc.com
=A0
World Wide Web
http://www.rsinc.com/
=A0

Research Systems' Inc., International Distributor Offices

Austria, Germany, Lichtenstein, Luxembourg,
Switzerland, The Netherlands
CREASO, GmbH
Talhof Str. 30
D82205 Gilching
Germany
Telephone: 49 8105 25055
Fax: 49 8105 25623
Contact: Bernhard Kortmann
email: 100137.2421@compuserve.com
=A0
Brazil
SulSoft

Rua Dom Pedro II, 1220 cj 515
90550-141 Porto Alegre - RS
Brazil
Telephone/Fax 55 51 337 38 91
Contact: Michael Steinmayer
email: mis@inf.ufrgs.br
=A0
China, Hong Kong
3- Link Systems Pte Ltd.
Room 801 Stone Plaza
No 2 Haidian Street
Beijing 100080 PR China
Telephone: 8610-261-0161
Fax: 8610-261-0163
Contact: Qunli Ren
email: linkbj@ox1.ios.ac.cn
=A0
Singapore
3- Link Systems Pte Ltd.
140 Robinson Road
#05-03 Chow House
Singapore 068907
Telephone: 65 227-8671
Fax: 65 227-8679
Contact: Harry Lee
email: zdlee@singnet.com.sg
=A0
France, Belgium
Fast Parallel Solutions France
1 Place Gustave Eiffel
Silic 267
94578 Rungis Cedex, France
Telephone: 33 1 46 87 25 22
Fax: 33 1 46 87 71 38
Contact: Louis Tauziet
email: 100347.1577@compuserve.com
=A0
Italy
Alliant Computer Systems SRL
Centro Direzionale Colleoni
Palazzo Taurus
Ingresso 3
20041 Agrate Brianza MI, Italy
Telephone: 39 39 6091766
Fax: 39 39 6091779
Contact: Chris Stuart & Alberto Meroni
email: chris@specialnet.cmt.it
=A0

Japan
Adam Net Ltd.
Yushimadai Bldg. 2-31-27
Yushima, Bunkyo-ku
Tokyo, 113, Japan
Telephone: 81 35802 2251
Fax: 81 35802 2249
Contact: Osamu Nakano
email: nakano@adamnet.co.jp
=A0
Korea
Intersys
373-1 KuSung-Dong
YuSung-Gu
Tae Jun 305-701
Korea
Telephone: 82 42 869 4746
Fax: 82 42 862 9239
Contact: Jong-Sik Yoon
email: jsyoon@intersys.kaist.ac.kr
=A0
Spain
Estudio Atlas
Parque Tecnologico de Alava
C/Tecnologico 11,40
01510 Minano (Alava)
Spain
Telephone: 34 45 298 080
Fax: 34 45 298 084
Contact: Montserrat Bacaicoa
email: estuds01@sarenet.es
=A0
Taiwan
Concentrate Corporation
5th Fl., No. 8, Lane 259
Fu Kang Road
Shihlin, Taipei
Taiwan
Telephone: 886 2 883 7752
Fax: 886 2 881 7946
Contact: Peter Kang
email: conce@ms2.hinet.net
=A0
United Kingdom
Floating Point Systems UK Ltd.
Ash Court
23 Rose Street
Wokingham

Berks RG11 1XS
United Kingdom
Telephone: 44 1734 776333
Fax: 44 1734 776433
Contact: Doug Wilson
email: doug@floating.demon.co.uk
=A0

---------- ----------=

G03. How do I get IDL?

RSI's distribution scheme is unique in that all of the binaries and IDL code
needed are available via anonymous ftp. IDL binaries and code are available
at these sites:

gateway.rsinc.com (192.5.156.17)
pub/idl
boulder.colorado.edu (128.138.240.1)
pub/idl
ftp.Germany.EU.net (192.76.144.75)
shop/CreaSo/IDL

The README file describes which files are needed, how to unpack them, and
how to install them.

If you install IDL without a valid license, you will get IDL's 7 minute (10
minutes for Mac and PC) demo mode. This mode is designed for users who are
considering buying the package.

To actually get IDL running for good, you must pay for a license from RSI
and follow their instructions. You will be asked to fill out a form with
information unique to your machine. RSI will create a license key which the
license manager program (lmgrd) reads to validate your license.

As of release 3.1, upgrades to IDL are no longer free. For details about
upgrades and support contracts, contact RSI.

G04. What is the current version of IDL?

IDL 5.0 is the current version. It was released May 1997.

G05. On what systems does IDL run?

The information below is from the IDL 5.0 Pre-Release Online Help, under the
topics "What's New in IDL 5.0?", "Platforms Supported in this Release":

IDL supports the following Platforms and Operating System versions:

Platform	Vendor	Hardware	Operating System	Supported Versions
VMS	DEC	Alpha AXP	VMS	6.2
VMS	DEC	VAX	VMS	6.2
Unix	DEC	Alpha	Digital UNIX	4.0
	HP	PaRisc	HP-UX	10.1
	IBM	RS/6000	AIX	4.1
	Intel	Intel x86	Linux	2.0
	SGI	R4000 and up	Irix	5.3 / 6.2
	SUN	Sparc	Solaris1 (SunOS)	4.1.3
	SUN	Sparc	Solaris 2	2.5
	SUN	Intel x86	Solaris 2	2.5
Windows	Intel	Intel x86	Windows	3.11
	Intel	Intel x86	Windows 95	
	Intel	Intel x86	Windows NT	3.51, 4.0
	DEC	Alpha AXP	Windows NT	3.51, 4.0
Macintosh	Apple	Motorola 680x0	MacOS	7.1.2
	Apple	Motorola PowerPC	MacOS	7.1.2

In most cases, IDL will run under operating system version released later
than the supported version listed above.

This is the final release of IDL for Macintosh on Motorola 680x0 processors.

G06. What is PV~WAVE and how is it related to IDL?

Around the time that the Unix version of IDL first became available (1988),
Precision Visuals Inc. (PVI) entered into an agreement with RSI under which
they enhanced and resold IDL under the name PV~WAVE. In September of 1990,
they exercised an option in that agreement that resulted in the following:

- They received a copy of the IDL source code as it existed in September
1990 in return for a one-time payment to RSI.

- The connection between RSI and PVI was severed.

IDL and PV~WAVE are now on separate development tracks. Each company enhances, supports, and maintains its own product.

PVI has since merged with IMSL and is now Visual Numerics, Inc. (VNI).
---------- ----------=

G07. Are there anonymous FTP sites for IDL?

The sites below contain public domain IDL code. See also the list of World
Wide Web sites under G13.

JHU/APL/S1R IDL library
fermi.jhuapl.edu [128.244.147.18]
/pub/idl
=A0
Phil's IDL Library of Functions
irc.chmcc.org
/pub/idl
=A0
David Fanning's Example IDL Programs
ftp.dfanning.com
/pub/dfanning/outgoing/idl_examples/
=A0
Chris Chase's Emacs related IDL tools sites
fermi.jhuapl.edu [128.244.147.18]
/pub/idl_emacs
Now maintained by Phil Williams at
ftp://irc.chmcc.org/pub/idl_emacs/
=A0
Research Systems, Inc. Emacs/Xemacs related IDL tools
rsinc.com [192.5.156.17]

/pub/user_contrib/emacs

NASA IDL Astronomy User's Library
(VAX) uit.gsfc.nasa.gov [128.183.57.27]
Username: idluser
Password: (Contact landsman@stars.gsfc.nasa.gov for password)
(Unix) idlastro.gsfc.nasa.gov [128.183.57.82]
/

IUE RDAF library
iuesn1.gsfc.nasa.gov [128.183.57.16]
cetus.colorado.edu [128.138.238.151]
/pub

ICUR Spectral Analysis Software
ftp.astro.psu.edu [128.118.147.28]
/pub/nefftp/icur

IDL ROSAT software
legacy.gsfc.nasa.gov [128.183.8.233]
rosat/software/idl
=A0
IDLmeteo library
ftp.sma.ch (141.249.3.33)
/pub/idlmeteo
=A0
ESRG library
eos.crseo.ucsb.edu [128.111.228.1]
/pub/idl
=A0

---------- ----------=

G08. How can I get help?

RSI has excellent telephone and email support. You can contact them at:

Voice: (303) 786-9900
Fax: (303) 786-9909

Email:
(Internet)
info@rsinc.com # general questions
support@rsinc.com # technical support

(SPAN)

ORION::IDL
=A0

Keep in mind, however, that RSI's technical support is for their
paying
customers, i.e. those with current support contracts.
---------- ----------=

G09. Why are there two newsgroups for IDL?

Unfortunately, there are two very different packages with the
abbreviation
"IDL". The newsgroup comp.lang.idl is for the Interface Definition
Language.
The newsgroup for discussing issues related to RSI's IDL and VNI's
PV~WAVE
and IMSL/IDL is comp.lang.idl-pvwave.
---------- ----------=

G10. Does anyone at RSI read this group? Is anyone there listening?

[This question was included at the request of RSI. The answer was
provided
by Ali Bahrami. -pat]

Yes, many of us do. We're naturally curious what people think of
our product. We make notes about what people like and dislike and
this influences our decisions.

However, you usually won't get a direct response from us from a
posting to this group. There are many reasons for this. Here are
a
couple of the more important ones:

[] We believe that this group should belong solely to the user
community, and should be free of vendor bias and marketing. It
should be noted that both RSI and Visual Numerics (formerly PVI)
have shown great restraint in this matter, and that this group is
largely left to the actual users. (Long term readers will recall
some notable exceptions to this, but in general it is true.)

It could be argued that as long as the topic stays technical,
vendor postings are OK. The problem with this is that one persons
technical posting is another's blatant product plug, and the line
between them is not always obvious.

[] We provide support for our customers via the phone and email.
If you would like an answer from us, you should call us directly.
We have no objection to you sharing the information you get in
this manner with the newsgroup as long as you quote us accurately
and separate fact from conjecture.

In other words, you should view this newsgroup as a way to share
questions and information with other users, not as a way to
contact the vendor.

---------- ----------=

G11. When is the next version of IDL due out?

* IDL 5.0, the current version of IDL, was released in May 1997.
* IDL 5.1 (?)

---------- ----------=

G12. Are there training courses available for IDL?

[This question included at the request of RSI.]

RSI offers a number of IDL training courses for beginning,
intermediate, and advanced IDL users. IDL courses are scheduled
monthly at RSI's training facility in Boulder. On-site IDL
courses
are also available. Contact RSI at 303-786-9900 and ask for
"training" or send e-mail to training@rsinc.com for complete
scheduling and price information.

Fanning Software Consulting also offers completely customized
on-site IDL programming courses for beginning to advanced users.
For information, contact David Fanning at 970-221-0438 or at
davidf@dfanning.com. Additional information about IDL training
courses is available on the Coyote's Guide to IDL Programming web
page at http://www.dfanning.com.

---------- ----------=

G13. Is there a World Wide Web server for IDL or IDL based projects?

See also the list of ftp sites under G07.

RSI has WWW pages on IDL in general:

http://www.rsinc.com/

=A0

Coyote's Guide to IDL Programming:
http://www.dfanning.com/

=A0

Pete Riley's IDL Home Page at the Lunar and Planetary Lab:
http://xlr8.lpl.arizona.edu/idl.html

=A0

Wayne Landsman's IDL Astronomy Library World Wide Web home page:
http://idlastro.gsfc.nasa.gov/homepage.html

=A0

The JHU/APL/S1R IDL library WWW page:
http://fermi.jhuapl.edu/s1r/idl/idl.html

=A0

University of Darmstadt, Germany, IDL page (in German):
http://wwwpc.hrz.th-darmstadt.de/prog/grafik/idl-4_0.htm

=A0

Gary Kushner's IDL data acquisition web page:
http://casa.colorado.edu/~kushner/

=A0

R. Sterner's Color Shaded Releif Maps made by IDL:
http://fermi.jhuapl.edu/states/states.html

=A0

E. Loren Buhle, Jr. Ph.D. made a page on AVS IN MEDICAL TREATMENT
PLANNING
which also discusses IDL:
http://archive.xrt.upenn.edu/0h/buhle/manuscripts/avs94_pape r.html=

=A0

Liam Gumley has several IDL related web pages:

Frame Tools | Image Mapping Tool | HDF SDS Tool | Underground Guide to
IDL
http://cimss.ssec.wisc.edu/~gumley/index.html

=A0

Phil Williams' web page of IDL functions:
http://www.irc.chmcc.org/idl/philsIDL.html

---------- ----------=

G14. How can I find if a routine to do what I want already exists?

One of the most useful tools to find an available routine is Pete Riley's

Searchable List of all IDL Routines.

This list is available from Pete's IDL page at http://xlr8.lpl.arizona.edu/idl.html.

If you would like to add your IDL library to Pete's list contact him at
uk2@lpl.arizona.edu.

---------- ----------=

G15. Where can I find online manuals and tutorials?

General

http://www.dfanning.com/
Coyote's Guide to IDL Programming. A growing list of tips and example programs.

http://www.va.ucsf.edu/mrs/IDL/idl_docs.htm/
IDL Help for Advanced Users. A web page of helpful information by E. Scott
Claflin.

Online IDL manuals in Postscript

http://consult.ncsa.uiuc.edu/docs/viz/Idl/index.html#psdocs
National Center for Supercomputing Applications, University of Illinois.

http://www.tac.dk:80/idl_manuals/
The Astronomy Group, Institute of Physics and Astronomy, Aarhus University.

IDL graphics

http://www.sljus.lu.se/stm/IDL/Surf_Tips/
Struan Gray's excellent tutorial on Extending IDL's Surface Plotting Routines

Very brief IDL examples

http://consult.ncsa.uiuc.edu/docs/viz/Idl/Training/
National Center for Supercomputing Applications.

Selected topics

ftp://gemsrws.med.ge.com//sageidl/window_resize
Tips on window resize by JBob Brown. Discusses how to resize widget windows.

ftp://fermi.jhuapl.edu/www/s1r/idl/s1rlib/local_idl.html
Tutorials on some of the JHU/APL/S1R IDL Library routines. Some of these are
pretty well up to date, some need a lot of work.

Additions to this section are welcome
---------- ----------=

TECHNICAL QUESTIONS:

---------- ----------=

Note for Sun users:

There is a mini FAQ for Sun IDL in $IDL_DIR/notes, and the following questions are answered in the file sun.doc:

1. IDL complains about missing fonts when creating widgets.
2. IDL complains about missing bitmap files when creating widgets.
3. How to make IDL work with OpenWindows version 3.0.
4. 3-D appearance for OpenLook IDLwidgets.
5. Why does one of the widgets appear red?
6. Why does pressing the left button while pointing at a pulldown menu

button cause the menu to pop up with the pushpin in? Older versions of
IDL didn't to that.
7. The font used by the list widget is much larger than it used to be, and
I don't like the result on my program.
8. Why do I get "Cannot allocate colormap entry" errors, and why do they
kill IDL?
9. I've changed my .Xdefaults file, but nothing different happened. Why?

---------- ----------=

T01. Why doesn't polycontour fill open contours??

This problem is described in the POLYCONTOUR manual page.

RESTRICTIONS:

This routine will NOT draw open contours. To eliminate open contours in your dataset, surround the original array with a 1-element border on all sides. The border should be set to a value
less than or equal to the minimum data array value.

For example, if A is an (N,M) array enter:

```
 B =3D REPLICATE(MIN(A), N+2, M+2)  ;Make background
 B(1,1) =3D A                       ;Insert original data
 CONTOUR, B, PATH=3DFilename ...    ;Create the contour file.
=A0
```

[This problem was fixed in IDL 3.1.]

The following is from Ray Sterner at Johns Hopkins University:
Here is a very simple algorithm that might be a useful addition to the section of the FAQ on filled contours. It is for evenly spaced contours
only.

```
  Z  is an array to be contoured,
  CI is the desired contour interval,
  C0 is the desired starting color index,
  D  is the desired step between colors.
=A0
  T =3D fix(Z/CI)
```

M =3D T - smooth(T,3)
F =3D (C0 + T*D)*(1-M)
is an array with filled contours with the contours
plotted with color 0. For contours of a different color
simply add M*CC where CC is the desired contour color index.
=A0

---------- ----------=

T02. How do I increase the number of commands stored in the history
buffer?

The system variable !EDIT_INPUT controls command recall. By default,
it is
set to 1, causing the last 20 commands to be saved. If it is 0, no
commands
are saved. To save more than 20 commands, just put !EDIT_INPUT=3D50
(or other
large number) in your startup file.

It is important to realize that IDL looks at the value of !EDIT_INPUT
the
first time it reads anything from the keyboard, and the size of the
history
buffer is fixed after that. Hence, the command must be in a startup
file
because entering it at the keyboard is too late.
---------- ----------=

T03. How do I get IDL to call routines in language X, running under
system
Y?

Jeff Valenti has written a sizable document about calling external
FORTRAN
routines from IDL. You can find it at
eos.crseo.ucsb.edu:/pub/idl/idl-fortran.Z.
---------- ----------=

T04. Why does XPALETTE edit my color table incorrectly?

Here is the answer from RSI support:

> The color applications such as xpalette and xloadct use a common
> block called "colors" to keep track of the color vectors. When

you
call tvlct, your vectors are loaded into the colormap, but they
are not put into the colors common block.

When xpalette starts, it checks to see if the colors have been
defined in the common block and uses them if they have. Otherwise
it sets them to the standard black and white colormap, in which
r,g,b are each linear ramps. (The colors are as you expect
because
there is only one colormap.)

Admittedly, this is not the most desirable situation. You would
like your colors which you loaded with TVLCT to be recognized by
xpalette. We modified xpalette (and xloadct) to use the current
colormap when they start up by getting the current vectors with
TVLCT.

Another alternative would be to use the following lines to define
the common block prior to calling your current version of
xpalette.

[assume here that you have a routine called "restore" which
reads
colors from a file somewhere and creates vectors r, g, and b.
-pat
]

IDL> restore, file=3D'ryan.sav', r, g, b ;get the vectors
from somewhere
IDL> tvlct, r, g, b
IDL> common colors,r_orig,g_orig,b_orig,r_curr,g_curr,b_curr
IDL> r_orig =3D r & r_curr =3D r
IDL> g_orig =3D g & g_curr =3D g
IDL> b_orig =3D b & b_curr =3D b
IDL> xpalette

T05. Is there on-line help for IDL?

Try ? at the IDL prompt.

T06. I run IDL under X in SunOS 4.x, and after I logout, the screen

becomes
completely blank. Typing in login names and passwords blindly logs you
in
again with the correct colors. How to prevent this?

[Note: This is only a problem under OpenWindows2. -pat]

Add the following to your .Xdefaults:

Idl*colors: -5

which reserves some colors for the colormap so that IDL does not
exhaust all
the available colors. (For a nice summary of Sun IDL interactions with
OpenWindows, see $IDL_DIR/notes/sun.doc)

Another solution is to put a call to clear_colormap in your .login
file to
be executed after OpenWindows start up.
 ---------- ----------=

T07. Sometimes my variables seem to disappear. Why is this?

Quoting the IDL User's Guide, page 10-8:

> IDL users may find that all their variables have seemingly
> disappeared after an error occurs inside a procedure or function.
> The misunderstood subtlety is that after the error occurs, IDL's
> context is inside the called procedure, not in the main level.
> Typing RETALL or RETURN will make the lost variables reappear.
>
> RETALL is best suited for use when an error is detected in a
> procedure and it is desired to return immediately to the main
> program level despite nested procedure calls. RETALL issues
RETURN
> commands until the main program level is reached.
>
> The HELP command can be used to see the current call stack (i.e.,
> which program unit IDL is in and which program unit called it).

 ---------- ----------=

T08. Are there any editors that support IDL programming?

Yes. Emacs has a major mode for editing IDL code, idl.el, written by
Chris

Chase. This is now maintained by Phil Williams
(williams@irc.chmcc.org), bug
reports should be sent to him. The most up to date version of idl.el
may be
obtained from Phil's ftp site at

ftp://irc.chmcc.org/pub/idl_emacs/

Other possible sources are

ftp://eos.crseo.ucsb.edu/pub/idl/
ftp://fermi.jhuapl.edu/pub/idl_emacs/

Get the files: idl.el and idl-shell.el

From John E. Davis, davis@space.mit.edu:

Also available is JED, an extensible programmer's editor that is
available
for Unix, VMS, OS/2, MSDOS, and MS Windows. The size of the executable
is
only slightly larger than vi. The extension language resembles C. It
provides emacs, EDT, wordstar, and brief editor emulations (Many claim
that
JED's EDT emulation is the best around). It is the only freely
available
editor that can perform color syntax highlighting on ordinary
character-cell
color terminals (e.g., MS-Kermit, etc.) as well as under XWindows
(Xjed).
Currently supported programming modes include: C, FORTRAN, (La)TeX and
BiBTeX, DCL, IDL, NROFF, SH, HTML, and SLANG. Other extensions include
mail,
rmail, compile, as well as the ability to read GNU info files. It is
available from

ftp://space.mit.edu/pub/davis/jed/

See Appendix A02 for details on using URLs.
---------- ----------=

T09. How do I get 3-D widgets under OpenLook 3.0?

This subject is discussed on page 53 of the OpenWindows Version 3
User's

Guide Release Manual.

OpenWindows 3.0 added 3-D appearance for widgets. In order for the 3-D look
to work, it must be enabled and the background color must be a medium tone
color such as "grey" or "wheat". Add the following resources to your
~/.Xdefaults file:

For plain IDL:

```
Idl*threeD:     TRUE
Idl*background: PeachPuff2
```

For IMSL/IDL:

```
Imslidl*threeD:     TRUE
Imslidl*background: PeachPuff2
```

T10. Why does one of the widgets appear red under OpenWindows?

This subject is discussed on page 53 of the OpenWindows Version 3 User's
Guide Release Manual.

The OLIT widget toolkit, which is used by IDL, added the concept of
"mouseless focus" under version 3.0. The red widget indicates where the
current mouseless focus is. Use the arrow keys to traverse the widgets, and
the spacebar to make a selection.

T11. Where are all the IDL routines and userlib procedures?

The basic routines are not accessible, for obvious reasons. The userlib,
statlib and widget procedures are in $IDL_DIR/lib/. The procedure XDL also
displays the full pathname. The system variable !path also contains the
directory names for all accessible IDL procedures.

T12. Does anybody know how to put multiple image plots on one page in PostScript?

Because PostScript has scalable pixels, you must specify the xsize and ysize
parameters, as well as the position parameter, in TV or TVSCL. I don't know
about PV-Wave, but the following works in IDL:

```
  ; Display four images in a 2x2 grid
  ; Assume data(x,y,4) =3D array containing the 4 images

  set_plot, 'ps'          ;request PostScript output
  device, ...             ;modify page size, orientation, etc. as
desired
  ximsize =3D 0.5*!d.x_size  ;define output image size
  yimsize =3D 0.5*!d.y_size  ;note: 0.5 assumes 2x2 grid
  for i=3D0,3 do begin       ;display the 4 images, using i as
position index
     tv, data(*,*,i), i, xsize=3Dximsize, ysize=3Dyimsize
  endfor
```

T13. Does case matter in IDL?

No.

Compiled routines are case insensitive. The only catch is that, on Unix
systems, when executing a script via the .RUN command, the file name
argument must exactly match the file name as it appears on the disk. Once
the routines in the script are compiled, their names can be written in any
case. This is not a problem in operating systems such as VMS that do not
distinguish case for file names. This is usually not a problem under Unix
either since, by convention, most people use lower case file names.

Sometimes.

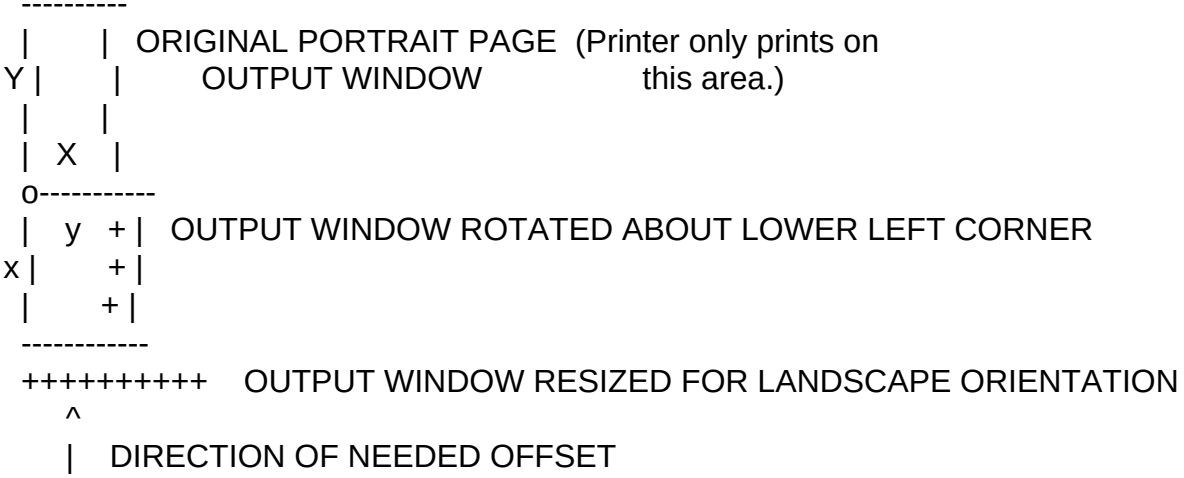
Case matters when you are doing string comparisons. Comparing "IDL" to "idl"
will return a false.

---------- ----------=

T14. How do I set up IDL to get precise control over plot window and
text
positioning with either portrait or landscape page orientation on a
PostScript or HP-GL printer?

(This answer only applies to PostScript and HP-GL printers -- other
printers
may differ in having the X and Y offsets measured from the upper left
corner
of the portrait page instead of the lower left corner.)

IDL uses portrait page orientation as a default. (The x axis is along
the
shorter dimension of the paper.) In portrait orientation the lower
left
corner of the page is the origin for the XOFFSET and YOFFSET page
offsetting
keywords of the DEVICE command that determine the origin (lower left
corner)
of the output window. (Normally one uses XOFFSET=3D0 and YOFFSET=3D0
for
portrait orientation.) Size of the output window is determined by the
XSIZE
and YSIZE keywords of the DEVICE command. The origin for graph
positioning
variables !P.POSITION and !P.REGION is the output window origin. X and
Y
coordinates for portrait page orientation are shown on the sketch
below as
upper case X and Y.



=A0

If device,/landscape is specified, then the output window is rotated
90 deg.
clockwise about the lower left corner of the page. In this condition
nothing
will be plotted on the page, since the rotation has carried the output
window entirely off the paper as shown in the sketch above. To correct
this
mismatch, the rotated output window must be offset. XOFFSET AND
YOFFSET are
still measured in the X and Y coordinates of the portrait page, but
now
represent the position of the lower left corner of the rotated (and
resized)
output window (marked by an o above) with respect to the lower left
corner
of the portrait (actual) page. Hence, one generally uses XOFFSET=3D0
and
YOFFSET=3Dlong_dimension_of_page for landscape orientation.

In landscape orientation, the coordinates for graph positioning
variables
!P.POSITION and !P.REGION are the lower case x and y coordinates shown
in
the sketch above and having origin marked by the letter o. Position of
output window origin o on the page is, of course, affected by the
setting of
XOFFSET and YOFFSET, as explained before. The XSIZE and YSIZE (output
window
size) keywords of the DEVICE command are also measured in the x and y
directions when in landscape orientation. The resizing of the
landscape page
generally means interchanging the values of XSIZE and YSIZE
appropriate for
the portrait page.

Example of settings for a portrait page:

  XPAGE=3D8.5 & YPAGE=3D11. & XOFFS=3D0. & YOFFS=3D0. ;Inches
   DEVICE,/INCHES,XSIZE=3DXPAGE,YSIZE=3DYPAGE,XOFFSET=3DXOFFS,Y
OFFSET=3D=
YOFFS
=A0

Example of settings for a landscape page:

  XPAGE=3D11. & YPAGE=3D8.5 & XOFFS=3D0. & YOFFS=3DXPAGE

;Inches
  DEVICE,/LANDSCAPE,/INCHES,XSIZE=3DXPAGE,YSIZE=3DYPAGE,XOFFSE T=3DXOFFS=
, $
     YOFFSET=3DYOFFS
=A0

Example of setting position and size of a plot window:

  X0=3D1.374 & Y0=3D1.283 & XLEN=3D3.622 & YLEN=3D6.157 ;Inches
  !P.POSITION=3D[X0/XPAGE,Y0/YPAGE,(X0+XLEN)/XPAGE,(Y0+YLEN)/Y PAGE]
=A0

Example of setting position and orientation of a text string:

  x0=3D.35 & y0=3D.37 ;Inches
  xyouts,x0/xpage,y0/ypage,!stime,orient=3D90,/normal   ;Date, time
=A0

 -- -----------=

T15. I get the error message "Code Area Full". What do I do?

IDL sets aside a certain amount of memory area for compiling programs.
The
current code and area sizes can be seen with the HELP command, e.g.

        IDL> help
        % At $MAIN$.
        Code area used: 0% (0/16384), Symbol area used: 0% (2/4096)

These sizes can be increased with the .SIZE command. Quoting the IDL
User's
Manual, page 2-11:

    These sizes represent a compromise between an unlimited program
    space and conservation of memory. User procedures and functions
    are compiled in this large program area. After successful
    compilation, a new memory are of the required size is allocated
to
    contain the newly compiled program unit.

    Resizing the code and data areas erases the currently compiled
    main program and all mail program variables. For example, to
    extend the code and data areas to 30000 and 5000 bytes
    respectively:

            .SIZE 30000 5000

=A0

Getting "Code Area Full" is often an indication that the routine is large,
and would benefit by decomposition into sub-procedures/functions. It's better to avoid use of .SIZE because your code will always work on other
systems where the users don't use a large .SIZE setting.
---------- ----------=

T16. Sometimes I get the following error message:

% Unable to allocate memory: to make array.
not enough core

RSI support replies:

The circumstances described happen when memory becomes fragmented.
Unfortunately, there is nothing you can do except use less memory in your application, or get more for the system to work with.

---------- ----------=

T17. How can I set the cursor to a crosshair on my display?

Ray Sterner of Johns Hopkins University has written a procedure for managing
this called crossi. It is in the usr.tar file of the JHU/APL IDL library
mentioned in question G07.

Joel Parker has written a procedure called rdplot which manages this using
an XOR graphics function and provides additional functionality. It is located in the NASA IDL Astronomy User's Library mentioned in question G07.
---------- ----------=

T18. How can I vectorize an equation of two different arrays?

From the user community:
I have two different arrays, (8) of float and (300,8) of float. I want
to vectorize the equation and therefore I need to use both arrays

in
the same equation. For example :

newarray=3Dcos(small_array)*sin(large_array)

where I want the data in small_array to be used over and over 300 times
in this calculation.

=A0

From Dan Carr (dan@rsinc.com):
IDL> arr1 =3D Findgen(8)
IDL> arr2 =3D Findgen(300, 8)
IDL> newarr =3D (Replicate(1.0, 300) # Cos(arr1)) * Sin(arr2)

=A0

From Dave Landers (landers@tsunami.dseg.ti.com)
to convert an array1(M) to array2(n,M) :
array2 =3D array1(Lindgen(n,M) / n)
or array2 =3D replicate(1,n) # array1

to convert an array1(M) to array2(M,n) :
array2 =3D array1(Lindgen(M,n) MOD M)
or array2 =3D array1 # replicate(1,n)

=A0

From Chris Chase (chase@grant.jhuapl.edu)
Instead of using matrix multiply, one can use REBIN:

To make array1(M) to array2(M,n):
array2 =3D rebin(array1,M,n,/samp)
To make array1(M) to array2(n,M):
array2 =3D rebin(reform(array1,1,M),M,n,/samp)

Note: use of /overwrite in REFORM avoids copying array1, but changes
array1.

REBIN is a little faster than the matrix multiply method. The
additional advantage of REBIN is that it will work with higher
dimensions, e.g. to make array1(M,N) into array2(M,N,L):

array2 =3D rebin(array1,M,N,L,/samp)

---------- ----------=

T19. How can I get IDL and MacX to work without crashing?

Using MacX v1.2 and IDL cause the Mac to crash quite often. This happens
especially during allocation of color resources or display windows. You can
get around the problem by downgrading to MacX v1.1.7 (apparently Apple will
supply this if you can prove to them that you rightfully own v1.2). Another
solution is to purchase White Pine's eXodus software. Rumors are that eXodus
is an overall better product than MacX. White Pine can be contacted at:

White Pine Software
40 Simon St. Suite 201 Nashua, NH 03060-3043
phone: 603-886-9050 Fax: 603-886-9051

Note: MacX 1.5 works correctly with IDL. Also, Tenon Intersystems
<http://www.tenon.com> makes a very fast X Server called XTen, and a full
UNIX implementation which runs alongside MacOS called MachTen. MachTen
includes the functionality and speed of XTen.

---------- ----------=

T20. How can I determine if a variable is defined?

It is often useful to determine if an IDL variable is defined. This is
easily done using the n_elements function which returns 0 if the given
variable is undefined. This is especially useful for setting defaults for
keyword parameters. Here are several examples:

```
if n_elements(start) eq 0 then start=3D0

if n_elements(dir) eq 0 then cd, current=3Ddir
```

---------- ----------=

T21. Why should KEYWORD_SET not be used to check if a variable is defined?

From Bill Thompson:

The IDL function KEYWORD_SET() is only designed to be used with logical
variables, i.e. those which can be either True (usually signalled with the
value 1) or False (0). It has the property that if a variable is undefined,
then it returns False, so people often make the mistake of using it to test
whether a variable is defined or not.

To test whether a variable is defined or not, use N_ELEMENTS() instead. This
will return 0 if a variable is undefined, or some positive number otherwise.
Only use KEYWORD_SET for truly Boolean (True/False) variables.
 ---------- ----------=

T22. What is the undocumented routine TVRDC?

From a comp.lang.idl-pvwave post by William Thompson:

The reason that TVRDC is undocumented is because it's not needed any more.
It doesn't do anything that CURSOR doesn't do. TVRDC is only retained for
compatibility with older programs.

In the old days, before X-windows, CURSOR was used to read coordinates off
of line graphics terminals, and TVRDC was used to read coordinates off of
image display devices. With the advent of IDL 2.0, the distinctions between
different kinds of graphics devices were mostly removed, and both of these
functions were merged into CURSOR.
 ---------- ----------=

T23. How can IDL be used to generate dynamic GIFs for display on the World
Wide Web?

By Jason Mathews, NASA/ Goddard Space Flight Center:

Common Gateway Interface (CGI) Perl scripts are invoked via submiting

a HTML
form , which execute IDL in batch mode, runs a IDL routine to make a
plot of
the selected parameters, and writes the results to a GIF. The perl
program
output is displayed on the WWW browser as a HTML document with an
inline GIF
image.

The IDL program must use the 'Z' device and write the output results
to a
GIF file as in the example below:

```
SET_PLOT, 'Z'                   ; Select the Z buffer output device
PLOT, x, y, ...                 ; Draw the plot
image =3D TVRD()                 ; Copy the device contents into an
image
WRITE_GIF, 'filename.gif', image ; Write image to a GIF file
EXIT                            ; Exit IDL routine
=A0
```

Examples of various perl programs that use IDL on the web and the
corresponding HTML forms are available via the following URL:
http://coney.gsfc.nasa.gov/Mathews/misc/idl-www.html

Some example WWW-based Data Browsing and Retrieval Systems using IDL:

* NSSDC OMNIWeb: http://nssdc.gsfc.nasa.gov/omniweb/ow.html
* NSSDC COHOWeb: http://nssdc.gsfc.nasa.gov/cohoweb/cw.html
* WWW/IDL Interface Demo: http://www.gsfc.nasa.gov/idl/idl_web.html

---

T24. How can IDL be used to save an IDL window in a specified graphics
format file (GIF, TIFF, ...)?

By Mark Rivers, CARS, Univ. of Chicago (slightly edited)

```
IDL> plot,x,y,title=3D'This is my plot'  ; Make a plot (or display an
image).
IDL> image =3D tvrd()                    ; Read image into an array.
IDL> tvlct, r, g, b, /get               ; Read color table.
IDL> write_gif, file, image, r, g, b    ; Save in a GIF image file.
=A0
```

This sequence should work on any windowing display (X, Windows, Mac).
For

better speed and more flexibility, first issue the command
SET_PLOT, 'Z'
to write to the Z-buffer pseudo-device. You can set its resolution,
and its
write and readback performance is very fast.

[Additional notes by editor] The screen image may be saved in other
graphics
formats by using the correct write routine. Some of the routines of
interest
are:
write_gif, tiff_write, write_jpeg, write_bmp, ...
Note the different name pattern for tiff_write, check the manuals for
the
calling syntax for each routine.

There are also corresponding routines to read the images back into
IDL.

Which format is best? GIF images are compressed so take less space
than
uncompressed images. JPEG images are also compressed but with a lossy
compression, that is, the image when read back into IDL is not
identical to
the image written. However JPEG images can look very good and also
often can
be more highly compressed than GIF. Some images may be larger (much
larger)
using JPEG as compared to GIF, try both and compare. GIF images are
widely
used on the World Wide Web, JPEG is also supported by some web
browsers.
TIFF may be needed for publishing purposes.
 -- -----------=

T25. Why don't my desktop PC applications recognize the preview portion=
 of
IDL EPSI files?

By Troy Klein, Johns Hopkins Applied Physics Lab

Quoting the IDL version 4.0 online help for the PREVIEW keyword, "Set
this
keyword to add a 'device independent screen preview' to the PostScript
output file, in encapsulated PostScript interchange format (EPSI). Use
this
keyword only with encapsulated output. Many, but certainly not all,

desktop
publishing and word processing programs will display this screen
preview
when the file is imported into a document."

The EPSI format is not compatible with some of the more popular
Windows and
Macintosh applications. Windows applications expect the preview image
to be
in TIFF format and Macintosh applications expect the preview image to
be in
PICT format and in the resource fork of the EPS file. I have found the
EPSI
file to be compatable with Framemaker for UNIX/X windows.

There are two workarounds available for Mac users and two for Windows
users,
all are similar. If you are using IDL on a UNIX machine, there is a
package
available (as of 8/2/95) on the popular University of Michigan Mac
shareware/freeware archive (use the mirror site at
ftp://mirrors.aol.com)
titled "ps2epsmac", which uses Ghostscript and NetPBM on a UNIX
machine to
interpret the postscript and create a new EPS file in Macbinary format
with
the PICT preview. There are several Mac programs available
(StuffitExpander,
Fetch, etc.) to interpret the Macbinary file. I have had great success
in
using this method to import IDL EPS files into Word, Powerpoint, and
MacDraw
Pro. There is a similar package at the archive titled "ps2epsplus"
which
does the same thing, but all on a Mac.

For UNIX IDL users trying to import IDL EPS files into Windows
applications,
there is a package called "ps2epsf" which can be located using an
Archie
search. This package essentially does the same thing as the UNIX/Mac
package
but produces an EPS file with a TIFF preview. The second solution for
Windows users is the Windows applicationi GSview (which can also be
found
with archie). Version 1.2 claims to be capable of adding preview
images to
EPS files.

---------- ----------=

T26. How can I fix widgets that broke with IDL version 4.x.x?

In IDL version 4, the sizing and layout of IDL's Motif widgets changed
in
several ways. This can cause problems. Widget changes are discussed in
a
file available from RSI at

ftp://boulder.colorado.edu/pub/rsi/idl/notes/widgets.txt.

---------- ----------=

T27. Why is memory not released back to the operating system after an
array
is deleted?

By Eric Korpela of Berkeley

This is a result of IDL being written in C and using the C library
functions
(malloc and free) for memory allocation. In most C libraries, memory
that is
freed is NOT returned to the operating system. The C program retains
this
memory and will reuse it for future calls to malloc (assuming that the
new
allocation will fit in the freed block).

Another way of considering it is in terms of how memory allocation is
done
under UNIX. New memory is allocated using brk() or sbrk() which
control the
size of the data segment. These routines are called by malloc().

Suppose you allocate 3 1 MB regions of memory under C.
=A0
char *p1=3D(char *)malloc(3*1024*1024);
char *p2=3D(char *)malloc(3*1024*1024);
char *p3=3D(char *)malloc(3*1024*1024);
=A0
Here's what your data segment would look like assuming malloc had to
call
sbrk().
=A0

```
 ------------------------------------------------------------ ---
prev stuff | overhead | 3MB | overhead | 3MB | overhead | 3MB |
 ------------------------------------------------------------ ---
                ^            ^            ^    ^
                p1            p2            p3    end of
segment.
=A0
Now we free(p1).
=A0
 ------------------------------------------------------------ ----
prev stuff | overhead | free | overhead | 3MB | overhead | 3MB |
 ------------------------------------------------------------ ----
                             ^            ^    ^
                             p2            p3    end of
segment
=A0
```

Notice that the free memory is still in the data segment. If free had called
brk to reduce the size of the segment, the 3MB pointed to my p3 would be
outside the data segment! SIGSEGV city! If free had moved the allocated
memory to lower addresses so the segment size could be reduced without
losing data, then p2 and p3 would point to invalid addresses, and we'd be
forced to use handles rather than pointers and call GetPointerFromHandle()
every time we wanted to access the memory. Ick! Just like Windows!

```
 ------------------------------------------------------------ -----------=
-----
```

T28. Why do color bars in PostScript not look as good as on the screen?

By default IDL uses 4 bits per pixel when displaying images in PostScript.
This gives only 16 possible gray shades or colors as may be seen by the
following example commands:

```
a =3D rebin(bindgen(256),256,50)        ; Create a color bar.
set_plot,'ps'                          ; Set PostScript mode.
device,/color                          ; Specify color PostScript.
loadct,4                               ; Load a color table.
tv,a,0,0,xs=3D12.8,ys=3D2.5,/centimeters  ; Display bar.
device,/close                          ; Close PostScript file
```

(idl.ps).
$gs idl.ps ; Send to a color PS printer or
use
; a PostScript viewer to see
results.
=A0

Replace first call to device above by:

device,/color,bits_per_pixel=3D8
=A0

This simple change may fix a number of problems with color PostScript.
By
the way, you will need to use /inches or /centimeters on a tv or tvscl
command to get the desired results. This applies to both the position
and
sizes of the image. By default these are in pixels and PostScript has
a lot
of pixels as may be seen by doing help,/structure,!d after
set_plot,'ps'.
Also try help,/device for other useful info when in PS mode.

---------- ----------=

T29. Does anyone know how to ensure vector-drawn fonts look good (and
similar) across different X11 servers?

By Liam Gumley, Space Science and Engineering Center, University of
Wisconsin-Madison

The fix I came up with goes as follows. Put the following commands in
an IDL
startup file (they must be executed immediately after IDL startup):

device,retain=3D2,pseudo=3D8 ; 8 bit display with backing
store
window,/free,/pixmap,colors=3D-5 ; Create window to allocate colors
plot,[0] ; Might not be needed, but won't hurt
wdelete,!d.window ; Delete the window
device,set_character_size=3D[6,9] ; Set the vector font size
print, 'Number of colors allocated is ', !d.n_colors
=A0

If you use this in your startup file, then you should not need any
entries
in your $HOME/.Xdefaults file. The only command I have in

$HOME/.Xdefaults
is

Idl*fontlist: screen16
=A0

which sets the font used for widgets (unrelated to the graphics vector
font).

As RSI technical support explained it to me, the default graphics
vector
font size is 6x9 at startup. However as soon as you open a graphics
window,
that font size may be changed (why, I don't know). So the only way to
guarantee that you get 6x9 vector fonts is to use the commands above
immediately after startup. Please note that you are not guaranteed to
get
256 colors allocated to IDL 4.01 if you use the colors=3D-5 keyword (I
usually
get about 170 colors on my SGI). On SGI boxes with 24 bit graphics,
setting
colors=3D256 usually gets you 256 colors, but this won't work on all
Unix
boxes. The IDL 5.0 pre-release seems to be a bit smarter (at least on
an
SGI), and even if you set colors=3D-5, it allocates 256 colors.

---------- ----------=

APPENDIX

---------- ----------=

A01. Disclaimer:

Questions and answers in this document are culled from the user
community,
except where noted otherwise. No warranty, express or implied exists
regarding this document. Permission to copy all or part of this work
is
granted, provided that the copies are not made or distributed for
resale.
---------- ----------=

A02. Obtaining the latest IDL FAQ

The current IDL FAQ may be accessed at http://ww2.sd.cybernex.net in directory /home/mgs/WWW-export/faq/

HTML version: idl_faq.html
The plain text version will be provided at a later time, or use your browser to convert HTML to text.

The URL (Uniform Resource Locator) for this file is:
http://ww2.sd.cybernex.net/faq/idl_faq.html

How to interpret the URL

Using a WWW (World Wide Web) Browser, for example mosaic:
mosaic http://ww2.sd.cybernex.net/faq/idl_faq.html
Save the file using the Save as ... option.

=A0

Using anonymous ftp:
Anonymous FTP is not currently available, but should be soon. Please
email me if you need other access.

Additions and Corrections

Send additions and corrections to:

Mike Schienle

mgs@sd.cybernex.net

---------- ----------=

A03. Many thanks to the following for their contributions

black@breeze.rsre.mod.uk (John Black)
claflin@claes.space.lockheed.com (Scott Claflin)
edelsohn@npac.syr.edu (David Edelsohn)
fireman@iuegtc.DNET.NASA.GOV (Gwyn Fireman)
gurman@umbra.gsfc.nasa.gov (Joseph B. Gurman)
jdlb@acm.org (JF Pitot de La Beaujardiere)
kashyap@oddjob.uchicago.edu (Vinay Kashyap)
mayor@vaxine.larc.nasa.gov (Shane Mayor)
oet@maz.sma.ch (Thomas Oettli)
rmmoss@Texaco.COM (Robert M. Moss)
sterne@dublin.llnl.gov (Philip Sterne)

William.T.Thompson.1@gsfc.nasa.gov (William Thompson)
valenti@soleil.Berkeley.EDU (Jeff Valenti)
sterner@tesla.jhuapl.edu (Ray Sterner)
joel@stars.gsfc.nasa.gov (Joel Parker)
landers@tsunami.dseg.ti.com (David Landers)
dan@rsinc.com (Dan Carr)
denisef@rsinc.com (Denise Fields)
mathews@nssdc.gsfc.nasa.gov (Jason Mathews)
rivers@cars3.uchicago.edu (Mark Rivers)
klein@hobbes.jhuapl.edu (Troy Klein)
davis@space.mit.edu (John Davis)
korpela@ssl.berkeley.edu (Eric Korpela)
hahn@hrz1.hrz.th-darmstadt.de (Norbert Hahn)
chase@grant.jhuapl.edu (Chris Chase)
liam.gumley@ssec.wisc.edu (Liam Gumley)
jbob@snap.med.ge.com (JBob Brown)
kivory2@gwdg.de (Kevin Ivory)
struan.gray@sljus.lu.se (Struan Gray)
=A0
=A0

---------- ----------=

Previous IDL FAQ maintainers:

Patrick Ryan: founded the IDL FAQ
Mike Schienle: from 12/01/93 to 10/27/94
Ray Sterner: from 10/28/94 to 5/15/97
=A0
=A0

---------- ----------=

A04. IDL FAQ Versions History

Changes in version 3.21:
. Added Struan Gray's surface plotting tutorial to G15.
. Added a new question on vector-drawn fonts (T29).

Changes in version 3.20:
. Cleaned up T08 at Kevin Ivory's suggestion.
. Added Phil William's Emacs site to G07.
=A0
Changes in version 3.19:
. Corrected IDL 5 release date in G11.
. Updated the source site of the emacs major mode for editing IDL code

in T08.
=A0
Changes in version 3.18:
. Added Phil's IDL Library of Functions to G13 and G07.
=A0
Changes in version 3.17:
. Added new question: Where can I find online manuals and tutorials?
as G15
. Added Coyote's Guide to IDL Programming to questions G13 and G15.
. Added E. Scott Claflin's IDL Help for Advanced Users to G15
=A0
Changes in version 3.16:
. Added an explicit reference to Chris Chase's ftp directory on fermi
in G07
. Added Research Systems ftp site in G07
. Added a pointer to the WWW site list under the ftp site question
(G07)
. Added a pointer to the ftp site list under the WWW site question
(G13)
=A0
Changes in version 3.15:
. Slightly modified Liam Gumley's web site address
and the description in G13
. Minor rewording for the VNI section of T08
=A0
Changes in version 3.14:
. Added a new web site in G13
=A0
Changes in version 3.13:
. Added a note to a non-working link in G13
. Added additional info from Chris Chase in T18.
=A0
Changes in version 3.12:
. Updated answers to G0, G05, and G11 with information from RSI.
. Added a new IDL web page to G13,submitted by Norbert Hahn.
It is located at the University of Darmstadt, Germany, and is in
German.
. Added a new question on color PostScript (T28).
=A0
Changes in version 3.11:
. Fixed the missing end of question T02.
. Added a new question on why memory is not released when an array is
deleted (T27).
=A0
Changes in version 3.10:
. Updated RSI's home page in G13.
. Made a few minor changes in the URLs that point to fermi (the home
site

of the IDL FAQ).
. Fixed the broken link to Gary Kushner's IDL data acquisition web
page in G13.
=A0
Changes in version 3.9:
. New question on how to find IDL routines (G14).
. New question on widgets breaking with IDL version 4 (T26).
. Minor change to G04.
=A0
Changes in version 3.8:
. New question on Encapsulated Postscript Previews (T25).
. Generalized question T08 to non-emacs editors
that support IDL programming (first is JED from MIT).
. Minor addition to G11 When is the next
version of IDL due out?
=A0
Changes since version 3.7:
. New question on screen save to a graphics format image (T24).

Changes since version 3.6:
. New question on using IDL to make WWW images (T23).
. IDL 4.0 release dates added (G04, G11).
=A0
Changes since version 3.5:
. Added question T22: What is the undocumented routine TVRDC?
. Added question T21: Why should KEYWORD_SET not be used to check if
a variable is defined?
=A0
Changes since version 3.4:
. Added Pete Riley's IDL Home Page at LPL (G13).
. Added RSI's International Distributor Offices (G02).
=A0
Changes since version 3.3:
. Added Gary Kushner's IDL data acquisition WWW page (G13).
. Dropped Hal Mueller's U.S. map page (was out of service) (G13).
. Added R. Sterner's Color Shaded Relief Maps WWW page (G13).
. Added a new question on detecting undefined variables (T20).
. Added a hyperlink to the directory in question T03.
=A0
Changes since version 3.2:
. Updated IDL release dates.
. Added Wayne Landsman's IDL Astronomy Library WWW page.
. Added JHU/APL/S1R IDL Library WWW page.
=A0
Changes since version 3.1:
. Made it easier to find where to get the latest FAQ.
. Retroactively changed last version to 3.0, the first HTML version.
=A0

Changes since version 3.0:
. New FAQ maintainer.
. First HTML version.
. Added an Appendix for some previous material.
=A0
Changes since version 2.8:
. Added Power Macintosh to systems which IDL runs on.
. Updated "Training" contact information.
=A0
Changes since version 2.7:
. Updated information regarding next release dates.
. Changed reference to openwin3.doc to sun.doc.
=A0
Changes since version 2.6:
. New question T19 added regarding MacX and IDL.
. Updated information regarding current IDL versions.
. Reference to current IDL version changed from 3.5.1 to 3.6.1.
=A0
Changes since version 2.5:
. T18 responses corrected.
=A0
Changes since version 2.4:
. New question T18 added regarding vectorizing an equation.
. Address change for FAQ maintainer.
. Address change for FAQ location.
=A0
Changes since version 2.3:
. New question T17 added regarding cross-hair cursor.
. Additional info for NASA ftp site.
. Additional information provided for question T01 regarding contours.
. Question G09 referred to the "other" IDL as Interface Description
  Language, rather than the correct Interface Definition Language.
=A0
Changes since version 2.2:
. Question G13 regarding IDL World Wide Web info added.
. JHU/APL IDL library name changed from nansen to fermi. Nansen will
  still work for some time.
. Release dates for upcoming versions of IDL revised.
=A0
Changes since version 2.1:
. Due to resource problems, the lumpi.informatik.uni-dortmund.de site
  is no longer carrying IDL binaries. However, the distribution is now
  available from ftp.Germany.EU.net (192.76.144.75).
. Reference to current IDL version changed from 3.5 to 3.5.1.
. Mention of compressed version of FAQ included.
. Added ESRG to list of FTP sites.
=A0

---------- ----------=

The End

---------- ----------=
