

# IDL 2019

Ronn Kling  
KRS, inc  
April 15, 2009

# Why 2019?

- Far enough away that :
  - Allows for some pie in the sky thinking
  - Time for open source 4GL's to evolve
- Close enough that
  - most of us will still be using IDL
  - Can make some reasonable extrapolations
- Why talk about it here?

# Future of 4GL's

- Good news, there will always be a 4GL language out there
  - Just way too efficient to ever go away
- Bad news, 4GL's will evolve and the ones that exist now, may not exist in the future
  - But I doubt that many new 4GL's will appear
  - Just too hard to start over and be able to compete
- I expect that we will still have open source and commercial 4GL's coexisting

# Open Source 4GL's

- Advantages
  - FREE
  - Ability to rapidly integrate new algorithms (but not always done)
    - Possible to have a very quick release cycle
- Disadvantages
  - “You get what you pay for”
  - Help isn't always that great
  - *Training isn't free*
  - Relies on a pool of geeks to continually maintain and update the language
    - Computer science majors are declining

# Commercial 4GL's

- Advantages
  - “You get what you pay for”
    - History of a good product and user trust
  - Ability to have outstanding help
  - Training can be subsidized by product sales
- Disadvantages
  - Not Free
  - Long development cycle
  - Tend to be slow in releasing new algorithms

# How does IDL stack up?

- Pretty Good!
- Long history and lots of code out there
  - But this is true of FORTRAN also
- Extremely intelligent user community
  - Matlab would kill for the IDL user demographics
- User community is developing and starting to interact
- The rest of this briefing is my slant on how IDL should evolve to become the 4GL language of choice

# The Keys to World Domination

- Keep IDL as inexpensive as possible
- Keep training as inexpensive as possible
- Help has to be the very best there is
- Make IDL easier to use “out of the box”
  - Facebook generation
- BUT, have language features that encourage power users to extend IDL
  - Power users are the best evangelists and are the ones that have the ability to add new analysis features to the language quickly
- Borrow good ideas from competitors
- Encourage and support the user community

# Price

- IDL is cheaper than Matlab
  - Especially when you start looking at the matlab toolboxes
    - I am not convinced that toolboxes are a good idea. It feels like they are trying to nickel and dime me to death.
  - Having independent third parties selling modules may make this a better sell
  - Little rant – I think you should give the student edition out for free
- Training is reasonable, but should always be measured against the “free” 4GL training courses
  - Little rant – Training is a marketing expense, not a profit center



# New user experience

- IDL 7.1 is a HUGE step in this direction
- But, does anyone do this right in the visualization arena?
- Look at how many people use Excel to plot, why?
- Goals;
  - Only choose a file to display an image
  - Only choose a file to make a line plot, contour or surface
  -

# .idl data file

- Special extension so that IOPEN understands it
- 1 column is a line plot
- 2 column is an x-y plot
- 3 column is either an x-y-z plot, contour or surface
- Multi-column is either a surface or contour
- Two optional lines at the top
  - Dim = [ x,y,z]
  - Command, x title, y title
- Goal is to create a simple data file and get it plotted.

# Help!

- Help is not bad right now, but it is language focused
- Needs to also be analysis focused
  - How do I make a plot, smooth an image, etc.
  - Click on a section and code shows up in the editor
- Find out who has the best help and mimic it!

# Matlab Community Example

[Files](#)

[Files by Product](#)

[Tags](#)

[Authors](#)

**Comments and Ratings**

[Submit a File](#)

**Search Comments and Ratings**

[View All Comments and Ratings by:](#) [Most Recent](#) | [Rank](#)

## Comments and Ratings

[E-mail this View](#)

1 - 50 of 37717

Date	File	Comment by	Comment	Rating
03 Mar 2009	<a href="#">Toolbox Graph</a> A toolbox to perform computations on graph. Author: <a href="#">Gabriel Peyre</a>	<a href="#">S, David</a>	Just wanted to add that the error I mentioned is caused by the flipping of the sign of the surface normal of just one face by rearranging the order of the vertices in the face matrix	<input type="button" value="comment"/>
03 Mar 2009	<a href="#">hs: a pedestrian history search engine</a> hs searches the command history for patterns Author: <a href="#">us</a>	<a href="#">Peshkin, Leon</a>	should have become a policeman ...	★☆☆☆☆
03 Mar 2009	<a href="#">OFDM with 16-QAM</a> OFDM Implementation using 16-QAM Modulation Author: <a href="#">Muhammad Nadeem Khan</a>	<a href="#">qwe, asd</a>	hi can u send the .m and .mdl files please i need to refer, so it can help me understand where im going wrong crisis_666@yahoo.com thanks	<input type="button" value="comment"/>
03 Mar 2009	<a href="#">non-rigid b-spline grid image registration</a>	<a href="#">Garcia, Maria</a>	Error using all example_3d : ??? Seven	<input type="button" value="comment"/>

# New language features

- Truth in advertising – I stole these from Python
- Slicing
  - `A = indices[-3:-1]`, - means pull from the end
- TRY BEGIN block for error handling
  - Practically speaking, as of now we can only have one catch handler per procedure. This means that I find myself writing simple wrappers for one line function calls
  - Add a new control type

```
TRY BEGIN
  triangulate,x,y,tr,b
ENDTRY CATCH BEGIN
  print,'Collinear points'
ENDCATCH
```

# New language features

- Lists – arrays of mixed types
  - [ 50, !pi, 'hello world']
  - Simpler than an anonymous structure
- New type of FOR loop
  - Names = ['a','b','c']  
FOR name in Names DO BEGIN  
    print, name  
ENDFOR
  - ages = [10,11,12]  
FOR name,age in [names,ages] DO BEGIN  
    print,name, age  
ENDFOR

# New language features

- This new FOR loop would also allow object based iterators

```
olterate = obj_new('testlter',0,9)
FOR t in olterate DO BEGIN
  print,t
ENDFOR
```

- The testlter object would have these two methods

```
testlter::__iter__
return,self ;returns an object that has a next method
end
```

```
testlter::__next__
self.value++
if self.value gt 10 then return,!stopiteration $
  else return, self.value
end
```

# New language features

- Operator Overloading  
pro test::\_\_ge\_\_, other  
if self.flag ge other->getFlag() then return,1 \$  
else return,0  
return & end
- Used like this  
IF testObj ge otherObj then ....
- Addition would be very similar  
pro test::\_\_add\_\_, other  
return, self.posVal + other->getPosVal()
- Used like  
newVal = obj1 + obj2
- Could be done for all the operators, ge,gt,le,le,eq, ne,+,-,/,\*



# New language features

- ICC color management
  - Guarantees color consistency for an application
  - Matlab doesn't have this
  - Johnson & Johnson has incorporated this into their IDL tools
- Implied print at the command line
  - $2 + 5$  instead of `print, 2 + 5`

# User Community Support

- Regularly release alpha/beta versions of new algorithms with source code (both IDL and C/C++)
  - Doesn't have to be for all platforms
  - Doesn't have to be completely generic. If it solves one problem then let the users extend it.
    - Number of downloads will give ITTVIS an idea of how important it is
- “You learn the most from your un-happiest customer” - Bill Gates
- Continue User Group meetings
  - And keep them as cheap as possible

# Pie in the Sky

- Paradigm shift – instead of starting with a visualization start with new technology and see what kind of visualizations you can make
  - Wii, Iphone, Itouch are good examples
- 3D mouse support
- Bumptop video <http://bumptop.com/#TEDVideo>
- What would physics based interaction with my visualizations look like and how would it help?
- Ability to run matlab .m files
  - If that is a legal issue than run Octave files