Subject: Translating a Python program to IDL
Posted by wlandsman on Fri, 27 Mar 2015 18:29:16 GMT
View Forum Message <> Reply to Message

Below are some notes I took while translating a moderately complex Python program into IDL. The program reads several hundred files into a single large hash with four nested levels, and then performs computations on the hash and writes output files.

1. As others have mentioned, the Python code is somewhat cleaner because of auto-instantiation. (You don't have to define an empty hash before assigning a value.)
https://groups.google.com/d/msg/comp.lang.idl-pvwave/GMkr1ve 0_ss/tY90_t26HnYJ

2. Occasionally I would need to test for null values in IDL when there was no need in Python. For example, you can't use null values with STRPOS() (or var.INDEXOF()). I think -- but I am not absolutely certain -- that it would be good for IDL to recognize null values here.

3. The Python translation was certainly made easier by some of the new features in IDL 8.4. Unfortunately, my code also has to be able to work in 8.2 and 8.3. So I found myself writing small wrapper programs, like the following one to implement the string Contains() method.

```
function str_contains, var, substring
; Implementation of the IDL_String::Contains() method introduced in IDL V8.4 that works in earlier
; IDL versions

if !VERSION.RELEASE GE '8.4' then return, var.contains(substring) else $
                            return, strpos(var, substring) GE 0
end
```

4. I realize that one should not use hashes when compute speed is a priority, and in fact the program speed was not important. But I was surprised to find the IDL program was almost a factor of 10 slower than the Python. PROFILER told me that IDL was spending two-thirds of its time in HASH::_OVERLOADPLUS. As I was reading one file at a time and building the hash, I would have statements like the following

```
data[fil,act,inst] += hash('EXP-ONLY',param)
```

adding a new hash to the fourth nested level.

Speculating from what previous posters have written
https://groups.google.com/d/msg/comp.lang.idl-pvwave/CX3CQZe pz0U/7HSa0QGwunEJ
I'd guess that the double-linked list implementation of hash in IDL has some advantages over the dynamic arrays of Python, but this is not one of them. Since I am translating from idiomatic Python, I am not coding IDL to its advantage. Still, it would be nice to have a writeup on how to efficiently use hashes, dictionaries and lists in IDL.

5. Finally I am a little embarrassed to say that my biggest problem in translating the Python code

was keeping track of the Python indentation. I am sure there is a Python-friendly editor that would let me match up indentation levels so I would know where to place an ENDIF or ENDELSE in IDL. (I used TextWrangler.) But I was being a macho programmer, and thought I should be able to locate the end of code blocks even when there were a dozen nested levels. I was wrong. --Wayne