Subject: Re: IDL 8.0 compile_opt changes
Posted by Maarten[1] on Fri, 08 Jan 2010 12:19:45 GMT

View Forum Message <> Reply to Message

On Jan 7, 6:56 pm, mgalloy <mgal...@gmail.com> wrote:
> On 1/7/10 8:27 AM, Maarten wrote:
>
>
>
>> On Jan 6, 10:25 pm, mgalloy<mgal...@gmail.com> wrote:
>>> On 1/6/10 7:01 AM, Maarten wrote:
>
>>>> I do love the idea of negative indices, although I'd like them to mean
>>>> the same as in Python. The samples Ive seen so far are off by one.
>
>>> My understanding of the negative indices proposal in IDL was that they
>>> would be the same as in Python:
>
>>>> >> a = [1, 2, 3, 4]
>>>> >> a[-1]
>>> 4
>>>> >> a[-2]
>>> 3
>
>> Yes, in this case it is the same. The (subtle) difference comes in for
>> ranges.
>
>> Python:
>>>> > a = [1,2,3,4]
>>>> > a[-1]
>> 4
>>>> > a[-2]
>> 3
>>>> > a[1:-1]
>> [2, 3]
>
>> The last one is the one I'm concerned about, as python does not
>> include the last index in the range.
>
>> IDL> a = [1,2,3,4]
>> IDL> print, a[3]
>>   4
>> IDL> print, a[1:3]
>>   2   3   4
>
>> (index 3 is equivalent to index -1).
>
>> Now you could say that Python and IDL already disagree here, but the

>> off-by-one is worth mentioning anyway.
>
> I think we are agreeing here, but just to be sure: Python and IDL would
> be specifying the endpoints of the range in the same way, it's just that
> Python always includes the start index and excludes the end index (even
> if not using negative indices):
>
>>>> a = [1, 2, 3, 4]
>>>> a[1:3]
>   [2, 3]

Yes. Although this is a fundamental difference that is the result of a
choice both language developers made. Thinking about it a bit longer,
I don't think the two can be made to act the same: IDL always includes
the end index of the range, while Python always excludes it. Some
emphasis on this in the documentation may be needed, as Python
probably is the most widespread programming language that offers the
facility of negative indices.

Maarten