Subject: Re: renaming a variable without making a copy
Posted by Kenneth P. Bowman on Tue, 08 Dec 2009 21:51:17 GMT
View Forum Message <> Reply to Message

In article <MPG.2587fff16b170da398968a@news.giganews.com>,
 David Fanning <news@dfanning.com> wrote:

> newName = Temporary(oldName)

Can anyone explain to me what TEMPORARY actually does? The documentation says

> The TEMPORARY function returns a temporary copy of a variable, and sets
> the original variable to "undefined".

which makes no sense to me at all. Doesn't making a "temporary copy of a variable" occupy memory? Perhaps I am confused by the use of the name "TEMPORARY".

My concept of an IDL variable (which could easily be wrong) is: some metadata that describes the variable (what you get with the SIZE function) and the actual data that comprises the variable. These things could be in different places in memory, and the metadata could contain, for example, a pointer to the actual data. Most of the time, I don't need to know.

Does TEMPORARY wipe out the old metadata (replacing it with "undefined") and create new "unnamed" metadata that points to the data part of the destroyed variable?

The example in the Docs is not very revealing.

Why does

```
A = TEMPORARY(A) + 1
```

use less memory than

```
A = A + 1
```

??

I suppose there is a good reason that the latter example "creates a new array for the result of the addition, places the sum into the new array, assigns it to A, and then frees the old allocation of A", although it just seems to me like the interpreter is being obtuse.

Cheers, Ken