

Re: Array Functions

Source:

<http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.worksheet.functions/2004-02/1993.html>

From: Alan Beban (*alannospambeban_at_pacbell.net*)

Date: 02/09/04

Date: Mon, 09 Feb 2004 03:23:47 -0800

Harlan Grove wrote:

> "Alan Beban" <*alannospambeban@pacbell.net*> wrote...

>

>>Harlan Grove wrote:

>>

>>>Odd bits.

>>>

>>> *If TypeName(InputArray) = "Object()" Then*

>>>

>>>*You'd be assigning a range to an array of objects?! . . .*

>>

> ...

>

>>*It's not clear to me that a user needs to be protected from assigning a*

>>*range to an array of objects because you or I think it doesn't make*

>>*sense to do so. . . .*

>

>

> *As so often seems to be the case, you're missing the point. It's not your or*

> *my sensibilities that matter. This just doesn't do anything useful. The only*

> *object type a range can contain is other range objects. All your code does*

> *is replace single area, multiple cell ranges with arrays of ranges in which*

> *each entry corresponds to a single cell in the original range. What*

> *conceivable benefit would be gained from this?*

No, I didn't miss the point. Setting aside for the time being your second comment below, what it does that might conceivably be useful is provide for the "direct" assignment of a 2-D Variant() array that contains objects (which myRange can be) to an array of Object type. Unless this facility were otherwise provided for (ala, e.g., your second suggestion, below), it would be lost by the simple removal of the *If TypeName(InputArray) = "Object()"* section to remove the facility for the useless assignment of a range; though that facility might not make sense, I don't see the harm in leaving it in—i.e., letting that section apply to both ranges (though useless) and some arrays (conceivably useful).

>[Y]our code is checking **only** that InputArray is an array of objects, not
> that InputRange also contains object references. When VBA tries to set an
> object variable to a nonobject reference, it throws a runtime error. You've
> trapped runtime errors, so nothing catastrophic happens. However, since your
> function provides other types of diagnostics for unsupported argument types,
> why not check that when InputArray is an array of object references, that
> InputRange must also be an array of objects?
>

The code does not provide **any** checking to see that InputRange contains only values that are acceptable to an array of the type that InputArray is. It does not, for example, check that InputRange does not contain any "pure strings" (such as "OK") when InputArray is of type Integer or Long or Single or Double. This did not, and does not, seem straightforward to me and I left that to the errorhandler. I did not, for example, try to work out code that would exclude InputRange dates when InputArray was of Long type, or exclude InputRange integers when InputArray was of Date type, or, as you suggest above, exclude InputRange non-objects when InputArray is of Object type. I'm not sure what the code would look like in this last case, so I can't be sure that working it out for the sole purpose of removing a perhaps useless, but harmless, functionality of permitting the assigning of ranges to Object type arrays would be worth it.

Thanks again for taking time. I do appreciate the substantive comments.

Alan Beban