

APL Characters and Their Aliases

**Bob Smith
Sudley Place Software
Originally Written
14 Dec 2013
Updated
22 Jun 2016**

The advent of Unicode solved many problems with dealing with APL characters, however there was still some wiggle room as to which Unicode characters were to be used in a Unicode implementation of APL, and different implementors made different choices. This paper is an attempt to raise awareness of these issues because the differences impede transfer of information.

The relevant document for the APL character set is the “APL Character Repertoire”, a.k.a. ISO-IEC / JTC 1 / SC 22 N 3067 (called herein ACR). For whatever reasons, that document never became a standard, but it does provide some guidance, and is better than each implementor making separate choices.

Different Strokes

There are a surprising number of similar APL characters in Unicode and in a number of cases some implementors went one way, others the other way. The following table lists the characters in question along with the way APL2, Dyalog, GNU APL, MicroAPL, NARS2000, and ngn/apl behave. I'm told by APL2000 that “Generally the default code point scheme for the VisualAPL product follows the IBM APL2 workstation scheme”.

When there are differences among APL implementations, users can become confused. They type something into one APL system, copy it to another and are greeted by a SYNTAX ERROR or the like.

The whole basis for the confusion in the recent lengthy thread on comp.lang.apl entitled “caret vs and” is that in some implementations the symbol for the logical “and” function is U+005E only, in some implementations it's U+2227 only, and in some both characters work. The original poster encountered some APL text from the APL wiki that had been produced by a system that supports U+005E and copied it into a system that uses U+2227 only and fails on U+005E.

When our systems differ in the set of acceptable characters for the same function, it serves only to confuse the end user to the detriment of us all.

Unicode Name, Symbol & Code as per ACR	Alternate Name, Symbol & Code	APL2 Notes	Dyalog Notes	GNU APL Notes	Micro APL Notes	NARS 2000 Notes	ngn/apl Notes
Minus sign (-) U+2212	Hyphen-minus (-) U+002D	2	2	2	1 2	1 2	1 2
Divides () U+2223	Vertical line () U+007C	1 2	1 2	1 2	1 2	1 2	1 2
White vertical rectangle (▯) U+25AF	Apl functional symbol quad (▯) U+2395	1 2	2	2	1 2	1 2	2
Star operator (*) U+22C6	Asterisk (*) U+002A	2	2	1 2	1 2	1 2	1 2
Tilde operator (~)	Tilde (~) U+007E	2	2	1 2	1 2	1 2	2

U+223C							
Apl functional symbol down caret tilde (∇) U+2371	Nor ($\bar{\vee}$) U+22BD	1	1	1	1	1 2	1
Apl functional symbol up caret tilde (\wedge) U+2372	Nand ($\bar{\wedge}$) U+22BC	1	1	1	1	1 2	1
Apl functional symbol alpha (α) U+237A	Greek small letter alpha (α) U+03B1		1			1 2	1
Apl functional symbol omega (ω) U+2375	Greek small letter omega (ω) U+03C9		1			1 2	1
Logical and (\wedge) U+2227	Circumflex accent ($\hat{}$) U+005E	2	1 2	1	1 2	1 2	1 2
Less-than or equal to (\leq) U+2264	Less-than or slanted equal to (\leqslant) U+2A7D	1	1	1	1	1 2	1
Greater-than or equal to (\geq) U+2265	Greater-than or slanted equal to (\geqslant) U+2A7E	1	1	1	1	1 2	1
White bullet	Ring operator	2	2	2	2	1 2	2

(◉) U+25E6	(◊) U+2218						
Medium white circle (◉) U+26AA	White circle (◯) U+25CB	2	2	2	2	1 2	2
Diamond (◊) U+2B26	Diamond operator (◊) U+22C4 White Diamond (◊) U+25C7 Lozenge (◊) U+25CA	2	2	3	2 3	1 2 3	2

Notes legend:

1 Functionally supports ACR codepoint (col 1), i.e. the codepoint may be used as a function

2 Functionally supports alternate codepoint (col 2), ...

3 Functionally supports U+25CA

These characters are included here because I have encountered them in APL code displayed somewhere on the Internet or in a PDF file. Thus blindly copying them into an APL session can produce an error which might well confuse the user.

Functionality

To save you from some typing, I use the following function to test the functionality of the symbols (except for Alpha and Omega which are not functionally defined on all systems):

```
test
```

⊖←'1', (⊖UCS 16±2 2 1 2), '1'	⊖ Minus
⊖←'1', (⊖UCS 16±0 0 2 13), '1'	⊖ Minus
⊖←'1', (⊖UCS 16±2 2 2 3), '1'	⊖ Modulus
⊖←'1', (⊖UCS 16±0 0 7 12), '1'	⊖ Modulus
⊖←'1', (⊖UCS 16±2 2 12 6), '1'	⊖ Star
⊖←'1', (⊖UCS 16±0 0 2 10), '1'	⊖ Star
⊖← (⊖UCS 16±2 2 3 12), '1'	⊖ Tilde
⊖← (⊖UCS 16±0 0 7 14), '1'	⊖ Tilde
⊖←'1', (⊖UCS 16±2 3 7 1), '1'	⊖ Nor
⊖←'1', (⊖UCS 16±2 2 11 13), '1'	⊖ Nor
⊖←'1', (⊖UCS 16±2 3 7 2), '1'	⊖ Nand
⊖←'1', (⊖UCS 16±2 2 11 12), '1'	⊖ Nand
⊖←'1', (⊖UCS 16±2 2 2 7), '1'	⊖ And
⊖←'1', (⊖UCS 16±0 0 5 14), '1'	⊖ And
⊖←'1', (⊖UCS 16±2 2 6 4), '1'	⊖ Not More
⊖←'1', (⊖UCS 16±2 10 7 13), '1'	⊖ Not More
⊖←'1', (⊖UCS 16±2 2 6 5), '1'	⊖ Not Less
⊖←'1', (⊖UCS 16±2 10 7 14), '1'	⊖ Not less
⊖←'1', (⊖UCS 16±2 5 14 6), '.=1'	⊖ Jot
⊖←'1', (⊖UCS 16±2 2 1 8), '.=1'	⊖ Jot
⊖←'1', (⊖UCS 16±2 6 10 10), '1'	⊖ Circle
⊖←'1', (⊖UCS 16±2 5 12 11), '1'	⊖ Circle
⊖←'1', (⊖UCS 16±2 11 2 6), '1'	⊖ Diamond
⊖←'1', (⊖UCS 16±2 2 12 4), '1'	⊖ Diamond
⊖←'1', (⊖UCS 16±2 5 12 7), '1'	⊖ Diamond
⊖←'1', (⊖UCS 16±2 5 12 10), '1'	⊖ Diamond
⊖← (⊖UCS 16±2 5 10 15)	⊖ Quad
⊖← (⊖UCS 16±2 3 9 5)	⊖ Quad

Note that for the last two lines you'll need to provide a value for the Quad to complete execution.

Atomic Vector

If your $\square AV$ has no room in which to include these new characters, consider translating them on entry to the corresponding symbol that is in $\square AV$. NARS2000 even has a means of translating symbols on the way out via Copy (Ctrl-C in Windows) to various other APL systems that don't support the same set of principal characters NARS2000 uses for the functions in the above table.

Conclusion

Unicode was a great start to enabling APL characters to be used, however in order for there to be interoperability, we have to agree upon which characters are functional. It doesn't matter if your system can change the mapping of glyphs to codepoints as the vast majority of users won't change from the default behavior.

- Do you consider it worthwhile to support the above codepoints as functions?
- Are there other characters you believe should be included in the above table?

Author Contact

Please direct any comments, questions, etc. to the author at bsmith@sudleyplace.com.

Online Version

This paper is an ongoing effort and can be out-of-date the next day. To find the most recent version, goto <http://sudleyplace.com/APL> and look for the title of this paper on that page.