

L2/10-352  
2010-09-21

Title: Response to Irish NB comments N3908  
Source: Van Anderson  
Re: Duployan proposal N3895  
Action: For consideration by JTC1/SC2/WG2 and UTC

**1. Introduction.** Michael Everson's assistance, logistic support, and review of the various stages of the Duployan script proposal over the past two years have been critical to the successful completion of this project. His comments and concerns have invariably led to improvements in the proposal. He has raised the issues contained in document N3908 before, and these have been considered with deference to his copious and inestimable experience and the valid technical concerns he raises. The issue of reordering the Duployan allocation to follow the proposed collation order was brought to the UTC at its May, 2010 meeting at Adobe in San Jose, CA, USA, and the conclusion, after significant discussion, was that the allocation order should not be changed to reflect the collation order, as is proposed in document N3908.

**2. Binary sort order and code chart organization.** Michael is correct that a simple binary sort of the current allocation will result in a different result than the proposed collation algorithm. This is due to the organizing principle of the current allocation, namely that

“The current allocation groups letters by usage, and orders them to invite comparison and contrast, shedding light on the proper usage of the characters. A collation-based order hides the relationships between a basic letter and its variants.” – UTC consensus as expressed by Ken Whistler.

The current Duployan allocation provides the general letter-forms in the first two columns. The third, fourth, and fifth contain every additional letter variant and affix sign needed for encoding French Duployan texts. Several of these letter-forms are used by the English Duployan systems of Pernin, Perrault, and Sloan, and also Romanian. The sixth and seventh columns

contain the Chinook letter variants, and similarly composed letter-forms for Romanian and the English Duployan shorthands. The eighth, ninth, and tenth columns contain characters used solely by the English Duployan shorthands.

The current allocation builds upon itself, with the most general shorthand - containing more universal and fewer unique characters - completed earliest in the allocation. On the other hand, the most proprietary shorthands - containing the fewest shared and most unique characters - complete their allocation last.

The isolation of only English Duployan characters beyond the 128 code point boundary (U+1BC7F) also reflects a conscious attempt to best comply with P&P, D.2.1: Block assignment starting on half-row boundary, and WG2 resolution M33.11 (Block assignment guideline). This is the “optimization” that is, admittedly inartfully, referred to in N3895, page 6, which Mr. Everson calls into question in section 4 of N3908.

**3. Rebuttal of collation based allocation.** There seem to be two statements which are at the heart of Mr. Everson's arguments, both contained in section 3 of N3908:

“The code chart sequence arranges the characters according to quite different criteria than those which are used for ordering, and we do not believe that the proposer’s chart ordering could be or should be considered useful to the end user.”

and

“A look at the code table in N3895 shows that it is organized on a horizontal axis, This makes for an “attractive” code table with some paedagogical features—but this does not make a useful code table in terms of technical functionality. Having acknowledged the usefulness of Mr Anderson’s proposed ordering scheme, we suggest that the code chart order also be

based on the principles of Mr Anderson's sorting order, rather than on a secondary classification as it currently is."

The first of these statements contains a single premise, which I consider to be unfounded, namely that an "end user" would not and could not find the current allocation order to be useful. More specifically, it relies on a concept of "end user" that I believe to be fundamentally at odds with the reality of the current and foreseeable future of Duployan implementation. Adding in his second statement, his point seems to be that the current allocation has no functional use and that the current order is based on inessential considerations.

I disagree with both of these statements. For Duployan, I believe it is essential to consider as "end users" not only those individuals who will simply be typing characters into a text program of some sort, but also the set of amateur programmers and problem-solvers who will undoubtedly attempt implementing the Duployan script. Given the rare and complex nature of Duployan text, I believe this is a group that must be given primary consideration if the Duployan allocation is to be successful. This script has little commercial potential, so in the foreseeable future, it will almost certainly fall on hobbyists and scholars to develop much of the rendering software, typography, and input devices that will be used by those creating documents in Duployan.

The current allocation organizes by *usage* – a term I will be using extensively herein – meaning that someone familiar with one of the Duployan adaptations can look at the chart and see logical groupings of the characters for their orthography. This is a critical orienting task that a collation based allocation will make nearly impossible. I believe a collation based allocation will result in non-standard texts being created by "end users", because another end user – the person who programmed their input device – was unable to properly ascertain which characters are used for a particular Duployan shorthand.

I consider this to be one of the critical concerns of script encoding that

absolutely cannot be overlooked. Unicode exists, first and foremost, to standardize the documents created by end users, and no script allocation should be accepted that thwarts this basic architectural goal, which is exactly what I believe a collation based allocation would do. An allocation that sheds light onto the identity and usage of its characters certainly should not be discarded for the payoff of a somewhat simplified collation algorithm that will, quite frankly, rarely be used by the “end users” which Mr. Everson rightly concerns himself with, but too narrowly defines.

In short, having acknowledged the usefulness of Mr Everson’s proposed ordering scheme, I believe that the code chart order should be based on actual character usage, rather than on the collation order, which not only obfuscates the structure of the script and impedes the process of implementation, but also slavishly defers to a property of limited utility to end users of all stripes.

**4. Comparison of collation algorithms.** Even though I believe that the usage based allocation is necessary, the issue of the collation algorithm cannot be summarily dismissed. By comparing the proposed Duployan specification to other scripts, I can only conclude that a collation algorithm based on a different ordering than the block allocation does not present a barrier to implementation.

Except for a single logic branch for initial vowels – which cannot be addressed short of an allocation with duplicate vowel characters – the current Duployan collation is logically equivalent to and, due to being uncased, simpler than the Latin collation. Latin has 1) a set of basic, shared, universal letter-forms (in two cases), with a universally recognized order, and 2) a set of letter variants and precomposed letters-with-diacritics, allocated outside the basic Latin block, that are sorted after their base characters. Duployan, on the other hand, has 1) a set of basic, shared, universal letter-forms (no cases), with a standard order, and 2) a set of variant letter-forms allocated separate from the base characters, that are sorted after their base characters. The same is true for the Greek script, Cyrillic, Arabic, &c. The conclusion must follow that if the current non-binary collation of >1200 Latin character variants and precomposed characters with diacritics with case distinctions is not onerous, then the non-binary collation of <150 Duployan character variants without

case distinctions should be accommodated.

**5. Conclusion.** Even though the suggestion of a collation based allocation was serious and logical enough that I personally raised it in discussion with the UTC, the conclusion of the professionals at Unicode, many of whom have dedicated 2 decades of their life to this Standard and represent companies that will have the onus of implementing the collation algorithm, was that the current usage based allocation offered myriad advantages to a large segment of users, while a collation based allocation offered a single simplification for a small audience whose technical backgrounds will mediate the complexities of a non-collation allocation. The fact that the collation algorithm is probably the one aspect of Duployan that will not be implemented by amateur programmers, only by professionals, decreases the advantage of a collation based allocation to negligibility. If a goal of an allocation is to make life easier for the largest group of users with the greatest need for that simplicity and clarity, then a usage based allocation is the only acceptable organization of the Duployan block.

**6. Other issue #1.** In section 5 of N3908, Mr. Everson notes that there are gaps in the seventh column (U+1BC64 and U+1BC65). This gap was intended to accommodate any dotted line consonant based on the T/D, F/V, or K/G that is found in, possibly, one of the currently undocumented English Duployan orthographies. The three character gap U+1BC6B..U+1BC6D is intended to accommodate dotted L/R line consonants, as well as any extra W-vowels from the Interior Salish languages that use the Chinook script. This leaves perfectly intuitive space for dotted line consonants, except those based on P/B, and accommodates overflow of W-vowels from the previous column.

**7. Other issue #2.** In section 6 of N3908, Mr. Everson rightly notes seeming inconsistencies in the sub-ordering within letter classes. The reason to insert the  $\frac{3}{4}$  circle arcs in the J series, breaking up the ordering of arc, arc-with-stroke, arc-with-dot, is based, again, on usage. The  $\frac{3}{4}$  circle arcs in the J series are from French Duployan, like the J series characters with strokes. The dotted letters are from Chinook, Romanian, and English.

The order of the J series is as follows: basic J < J with stroke (French JeMe)

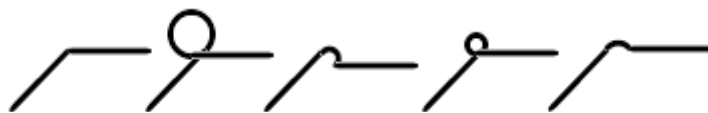
< 9-6 o'clock  $\frac{3}{4}$  circle (French JeNe) < J with dot (Chinook, English, Romanian), J with dots above and below (Romanian). Then, the large versions of J follow: long J (French JeSe, English, Romanian) < long J with stroke (French JeMeSe) < long 9-6  $\frac{3}{4}$  circle (French JeNeSe) < long J with dot (English and Romanian). The collation algorithm, like the allocation order, is based on actual usage, not theoretical perfection.

The problem with applying the order of dotted consonants to dotted vowels is that vowels change their orientation based on the adjacent letters. *The dots move around relative to the vowel*, so the dot on the Romanian I will sit below when preceded by a P, but above when preceded by an S. The vowels are treated differently from the consonants because they act differently than the consonants. As an example, vowels almost always have little effect on the overall shape of the word:



*BD, BOD, BID, BAD, BOnD*

All have the same general angular 'L' shape. Because the vowel signs end close to where they began, the vowels can be more freely interchanged without effecting the overall shape of the word. If I were to change the B's to R's, these words would look starkly different:



*RD, ROD, RID, RAD, ROnD*

The letters and their variants were ordered to reflect the form, usage, and behaviors of letters, while respecting any deeper relationships that the original source documents convey. They are not arbitrary – the nasals In, Un, On, An – are cited in the same order in French Duployan, Chinook, and Sloan primers. The combined consonants occur in the same order in all of the

English Duployan shorthands, but this order is different from the order of the nasals. The fact is, they also don't change their shape to accommodate the adjacent characters like nasal vowels. I do not feel that it is prudent or justified to impose an artificial structure at odds with the conclusions of the originators, even if it does seem inconsistent to computer-savvy script encoders today.

In the end, the collation specification is an attempt to codify what already exists in the source documents, and to resolve as unobtrusively as possible any inconsistencies between the different systems. I do not think it is a coincidence that most of the Duployan shorthands introduce letters in the order P, T, F, K, L, M, N, J, S, vowels. I don't think it's a coincidence that most introduce the long line consonants directly after the short line consonants. Lastly, I do not think it is right to impose my conceptions of logic and consistency over the determinations of Emile Duployé, J.M.R. LeJeune, Margareta Sfințescu, Helen Pernin, John Mathew Sloan, and Denis Perrault. I hope that the conclusion of the WG2 is that it is unnecessary to do so.

**8. Final Conclusion.** The Duployan block is currently ordered according to the internal logic of the script and how it will be implemented. Likewise, the collation order is specified according to the internal logic of the script and how it has been used. The differences between these two specifications reflect the dissociation between modern computer programming and archaic stenography study. It would be a coincidence of unthinkable proportions if these two starting-points led to identical conclusions. The current Duployan proposal balances the needs of these two perspectives consciously. I believe that attempts to impose either the historic ordering, or the needs of modern programmers to both the collation algorithm and the allocation order is unwarranted and will result in more harm than good. The allocation and collation orders differ for compelling reasons, and the proposal to radically alter the allocation order to reflect collation has been rejected by the UTC as non-compelling and counter-productive, and I see no reason to reverse that conclusion.