

Universal Multiple-Octet Coded Character Set
International Organization for Standardization
Organisation internationale de normalisation
Международная организация по стандартизации

Doc Type: Working Group Document

Title: Further discussion on ordering and the proposed Duployan script

Source: Irish National Body

Status: National Body Contribution

Action: For consideration by JTC1/SC2/WG2 and UTC

Date: 2010-10-03

1. Introduction. Document N3922 “Response to Irish NB comments N3908” discusses some of the concerns raised in N3908 “On ordering and the proposed Duployan script for shorthands and Chinook” about aspects of encoding Duployan. Essentially there are two issues N3908 raises, one of which has essentially been settled by N3922.

2. Sorting by shape and ordering within shape-classes. As in N3908, we note that Mr Anderson rightly constructs an ordering based on shape. Although what might be termed “Unified Duployan” may never have had a “native” ordering either as a whole or in any of the language variants, in the context of the UCS, an ordering which has an intrinsic logic to it will certainly be of benefit to end users, whether the operating system is ordering Duployan filenames or whether the users are using word-processing or database software to sort their data.

The issues which were raised in N3908 have been resolved thus:

2.1. Long arcs re-ordered. The long arcs have been moved to after the triple of half-circle, half-circle-with-stroke, and half-circle-with-dot:



A slight error in the collation specification in the revised N3895 (which ought to have been re-numbered N3895R) should be noted; it has been underscored here:

Primary collation: Initial vowel cluster < H < X < P < B < P N < T < TH < SLOAN DH < D < DH < D S < F < V < F N < K < KK < G < SLOAN J < K M < L < HL < LH < R < RH < R S < M < M N < M WITH DOT < M S < M N S < N < N M < N WITH DOT < N S < N M S < J < J M < J WITH DOT < J WITH DOTS INSIDE AND ABOVE < J N < J S < J M S < J S WITH DOT < J N S < S < S J < S WITH DOT < S WITH DOT BELOW < S S < S J S < S T < S T R < S P < S P R < T S < T R S < W < WH < W R < S N < S M < K R S < G R S < S K < S K R < medial/final vowel cluster.

Another slight error in the collation specification should also be noted; it has been underscored here:

Vowel order: AOU < O < WO < WOW < WA < WI < WEI < OU < OW < Romanian U < OA < A < Sloan OW < I < E < Sloan EH < Romanian I < Sloan EE < IE < Short I < EE < UI < YE < Long I < U < EU < XW < U N < LONG U < UH < Sloan U < OOH < Vocalic M < Nasal U < Sloan EN < Sloan AN < Nasal O < Pernin AN < Nasal I < Nasal A < Pernin AM < Sloan ON;

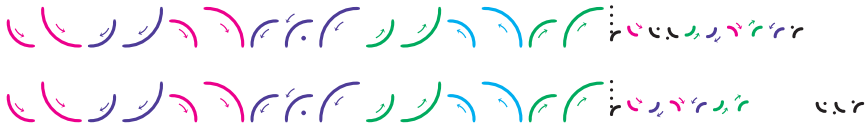
2.2. Dotted half-circles re-ordered. The relative order of the SLOAN EH and ROMANIAN I has been reversed so that they follow the same pattern as S WITH DOT and S WITH DOT BELOW:



2.3. Invariant half-circle vowels re-ordered. The half-circle vowels I, EE, IE, and UI have been re-ordered so that they follow the same pattern as M, N, J and S:



2.4. Small quarter-circle vowels re-ordered. The small quarter circles have been re-ordered according to the relative order of the downward-sloping quarter-arc consonants, even where some of them are upward-sloping. This is slightly inconsistent, but since slope directionality has not been indicated for the dotted vowels it is not clear how this category should be further improved. Is directionality inherent in the dotted vowels? The top line below shows the current ordering; the second line implements the directionality but leaves the dotted vowels to the side as their directionality is uncertain. Clarification from Mr Anderson would be welcome, though we have no objection to the first line if no logical improvement can be made.



2.5. Circles re-ordered. The circle vowels have been re-ordered according to size with the internal components of the w- vowels ordered according to the value of the second component. Thus AOU, O, WO, WOW, WA, WI, WEI, OU, OW, U, ROMANIAN U, OA, A, SLOAN OW, with the w- vowels following the pattern $o = o+o$, $o+o$, $o+ow$, $o+o$, $o+a$, $o+i$, $o+e+i$:



As stated, we believe that Mr Anderson’s ordering based on shape will very likely prove to be useful to users of Duployan. It is our view that users will learn the sequence quickly and have no trouble finding things by shape. In unifying the Diployan characters used for different languages, it made good sense to devise a coherent and logical ordering for them.

The outstanding issue for the encoding of Duployan relates to the functionality of the organization of the code chart.

3. Organization of the code chart. Having determined that the collation proposed is useful, elegant, and sensible, we believe that the ordering of the characters as presented in the code chart will prove to work *against* the end user in environments where ISO/IEC 14651 and the UCA are not properly implemented. In effect, the end user *will not be able to rely on a consistent result* in instances where a binary sort is applied.

Document N3922 makes a number of claims, not all of which are supportable, in our view. In the first place, it argues that the issues raised in document N3908 had been raised previously, and had been discussed by the UTC. N3908 was published in September 2010, and so could not have been discussed in the spring of this year. If the UTC visited similar issues, they did so without our input, and in any case we do not believe that the matter has been closed. We are bringing forward our concerns for WG2 and the UTC to consider, and expect the technical issues we raise to be examined, not to be brushed aside as N3922 attempts to do.

3.1. The nature of the Duployan collation order. It is stipulated that Duployan has no “native collation”. Nevertheless, a “Unified Duployan” encoding must have a collation, and Mr Anderson has devised an excellent one. It is no matter that the order was “invented” by Mr Anderson. Quite the opposite: his unique contribution in this matter is a valuable contribution to the history of the script itself. It doesn’t matter that “to sort Duployan characters... could take many different forms and still be just as faithful to the source documents”—the “sorting by shape” collation proposed is excellent, and so there is no reason that it should not be present in the code table as well as in the UCA and ISO/IEC 14651. The implication in N3922 seems to be that “inventing” a collation is somehow illegitimate. The practice of the UCS has shown otherwise. The encodings for Myanmar, Vai, and Tai Tham, for instance are all “comprise” orderings involving a superset of characters that goes beyond attested orders; current work on other African scripts (Kpelle, Mende, Loma) involves devising sorting orders where none had previously existed. Current UCS practice *is* to arrange the code chart more or less according to those orderings. We do not believe that Duployan should deviate from this practice.

3.2 Duployan code chart organization. Mr Anderson cites Ken Whistler in defence of “the organizing principles of the current allocation”:

“[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 function of a code chart is not to “invite comparison and contrast”. *That* functionality belongs in a *User’s Guide* to the script. The function of a code chart is to provide a structure which is *technically useful*, not a structure which has educative value.

Further, it is hard to see how Ken Whistler’s statement actually applies. A collation-based order which places all of the half-circle consonant next to one another



is hardly *hiding* the relationships between a basic letter and its variants. When one compares the two proposed code charts, it is clear that both of them attempt in some way to display a relationship between the shapes of characters (namely, of basic letters and their variants):

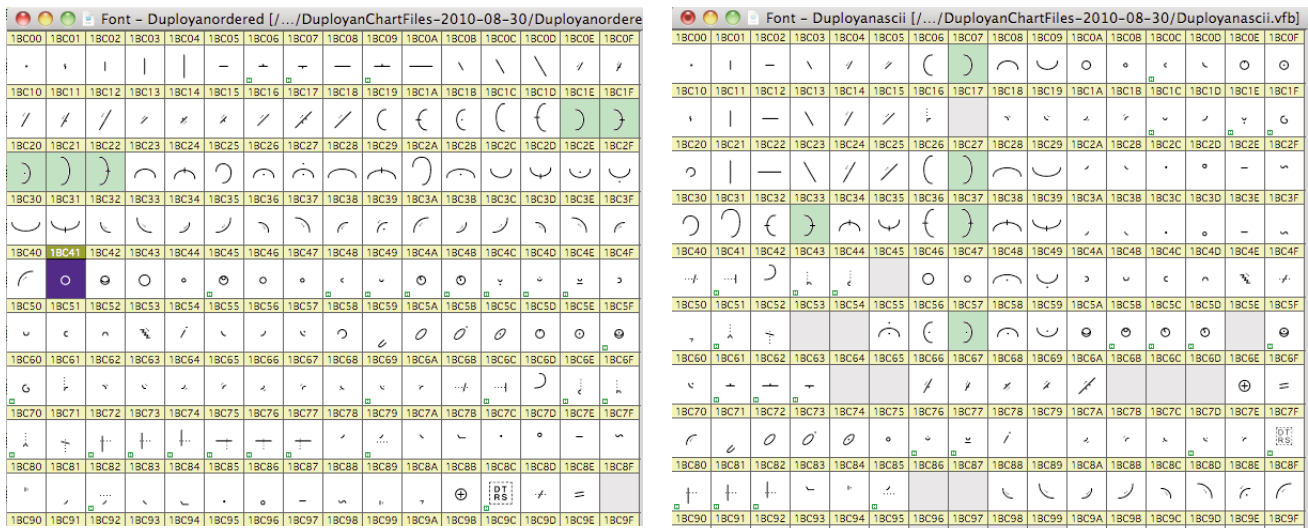
	1BC0	1BC1	1BC2	1BC3	1BC4	1BC5	1BC6	1BC7	1BC8	1BC9
0	·	/	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
1	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
2	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
3	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
4	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
5	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
6	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
7	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
8	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
9	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
A	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
B	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
C	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
D	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
E	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
F	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿

	1BC0	1BC1	1BC2	1BC3	1BC4	1BC5	1BC6	1BC7	1BC8	1BC9
0	·	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
1	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
2	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
3	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
4	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
5	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
6	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
7	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
8	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
9	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
A	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
B	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
C	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
D	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
E	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿
F	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿	⤿

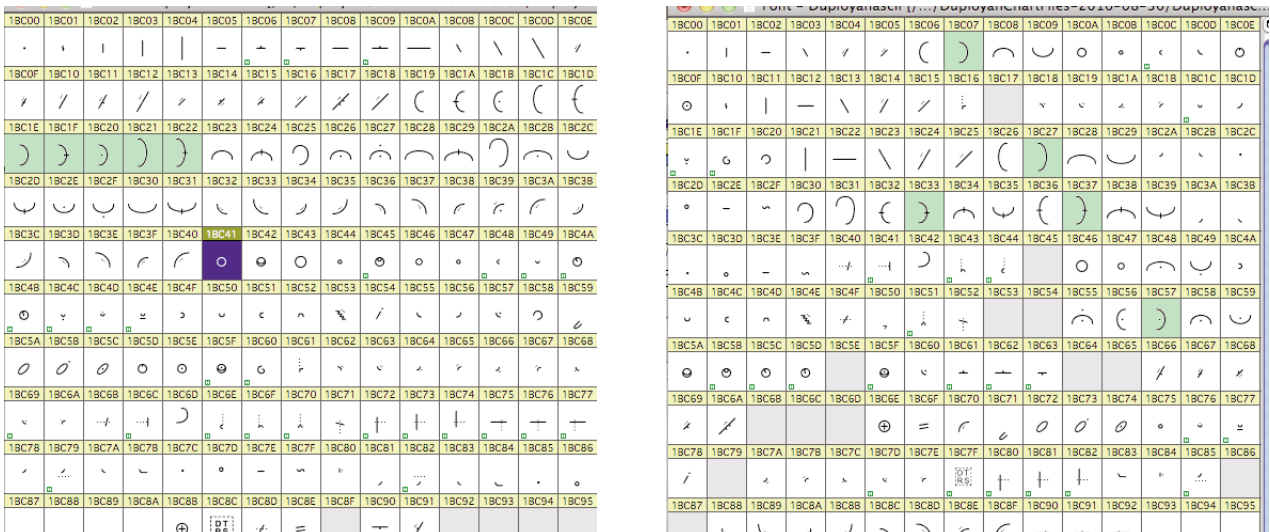
The problem is that a 16-row by 10-column display is not a constant, which renders the non-collation display unstable, since the horizontal relations are lost in other ratios.

It may be true that Mr Anderson’s proposed allocation divides up French Duployan, Romanian Stenographie, Chinook script, Pernin shorthand, Perrault shorthand, and Sloan-Duployan and that this in some way reflects historical developments. But again, that is a matter of paedagogy, which is a different thing from technical functionality.

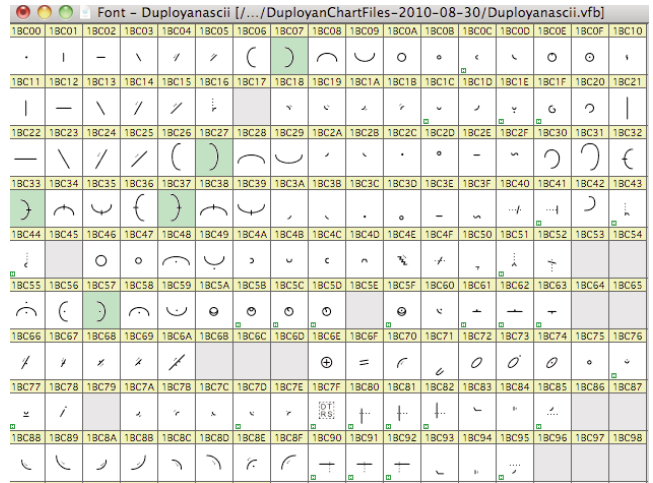
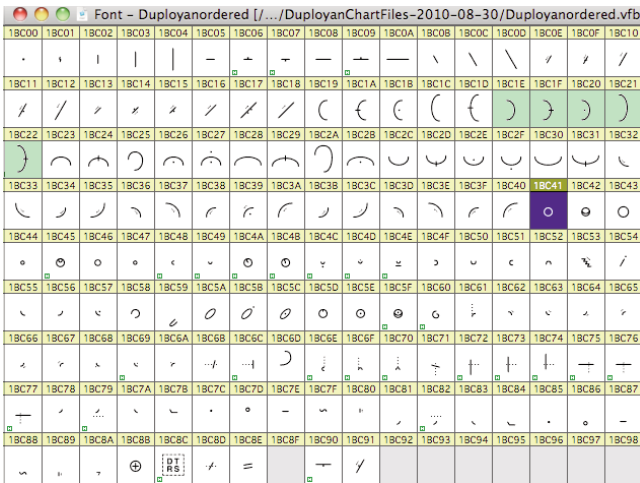
Where Mr Anderson’s proposed allocation fails to provide technical functionality is in its *instability in dynamic environments*. In a real-world user environment, the contents of a font (which typically organizes characters in code-chart order) are displayed to the end user in a variety of ways. A font developer, for example, will use a font editor such as FontLab.



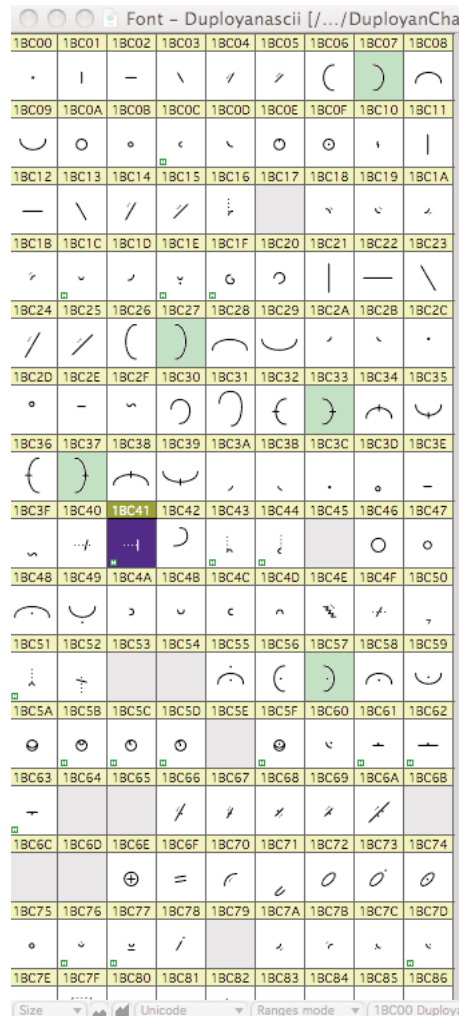
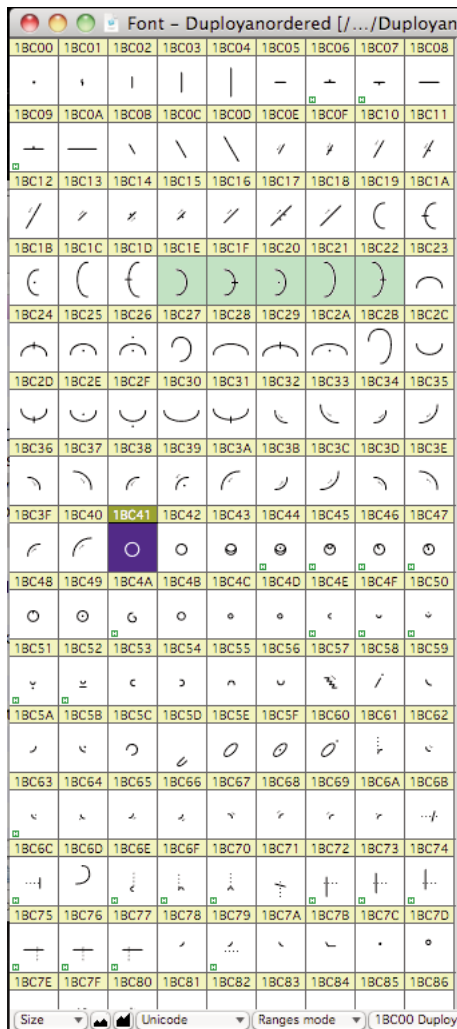
Here we see a FontLab window open to a width of sixteen characters, with five related characters highlighted in green. Unlike the UCS code charts, display is horizontal rather than vertical, but both show a similarity to the UCS code charts.



Here we see a FontLab window open to a width of fifteen characters, with the same five related characters highlighted in green. Here, the characters in the collation-based code chart are still right next to each other, but the linear relationship of the characters in the non-collation-based code chart are now scattered and more difficult to find at a glance.

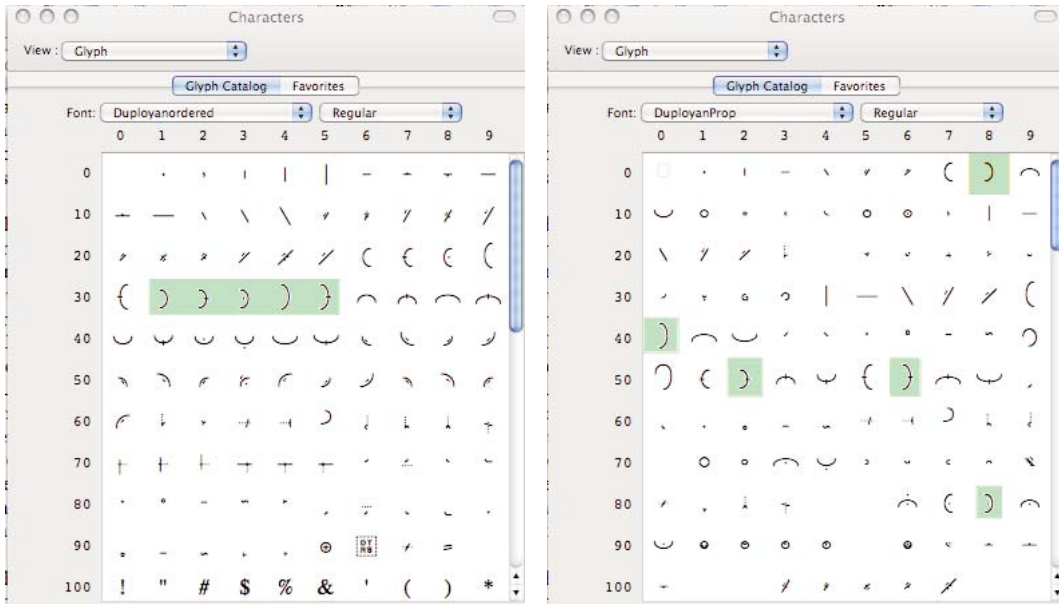


Here we see a FontLab window open to a width of seventeen characters, with the same five related characters highlighted in green. As above the linear relationship of the characters in the non-collation-based code chart are now scattered and more difficult to find at a glance.

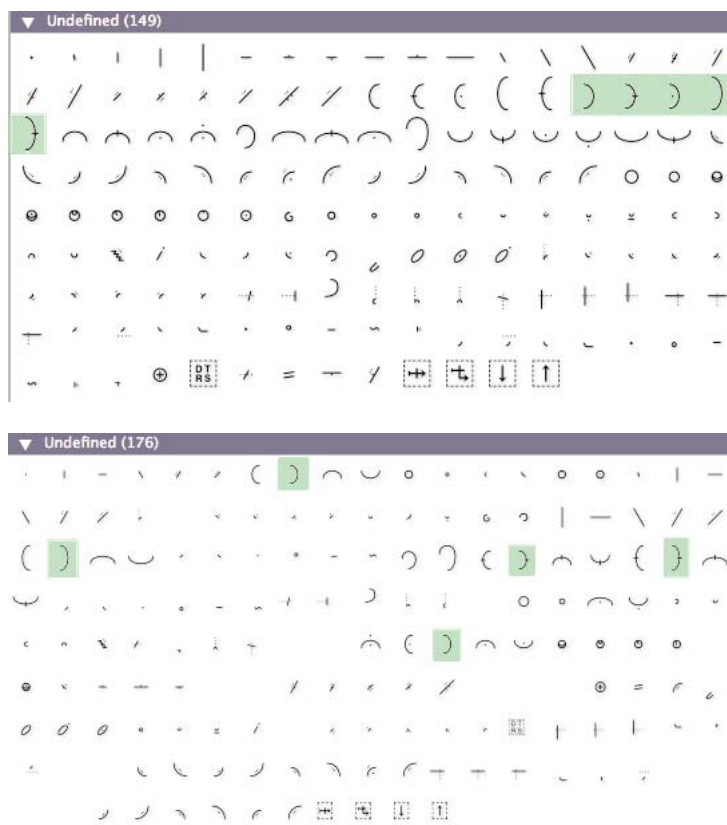


And here we see a FontLab window open to a width of only eight characters, with the same five related characters highlighted in green. Here, the scattering is even more pronounced. We do not believe that this arrangement can be considered “user-friendly”, particularly when the users will quickly become familiar with shape-based ordering in other contexts. What *advantage* is there for having two disparate orders— one which is stable in a dynamic environment and one which is not?

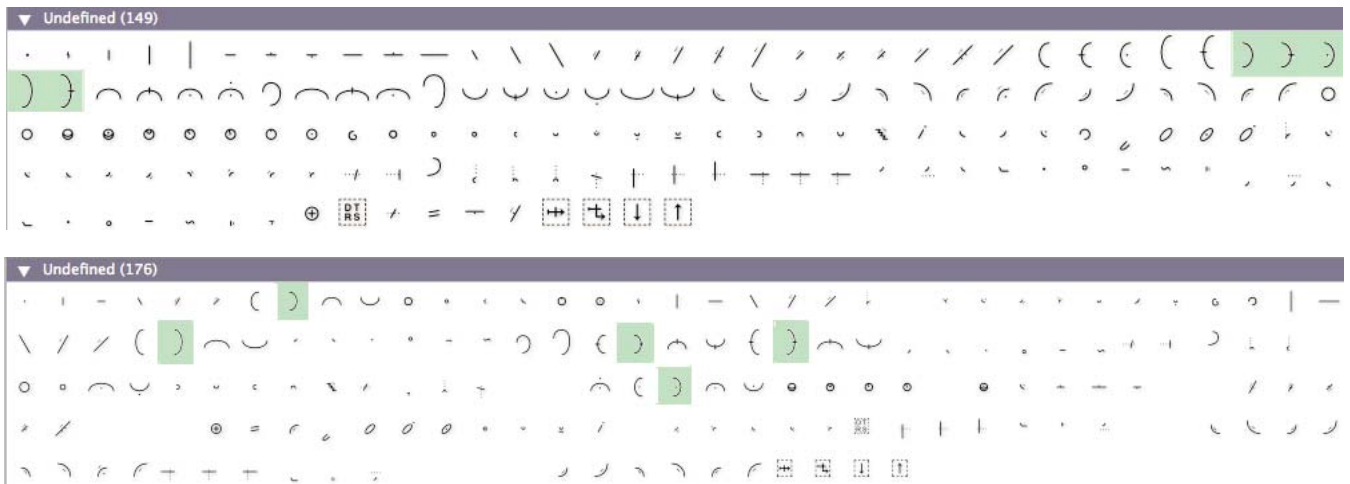
Another common usage environment is the “character picker”. A variety of these are available. One use would be the kind of use to which Mr Anderson refers in N3922, where an amateur developer of keyboards has to actually find the characters he or she wishes to insert. Any *User Guide to the Duployan Script* can list the characters needed for French Duployan or Chinook. They can be listed by name and code position, or by graphically highlighting them in a chart. But when actually finding them in a character picker, a collation-based code table has more stability and enables easier identification and selection of the desired characters.



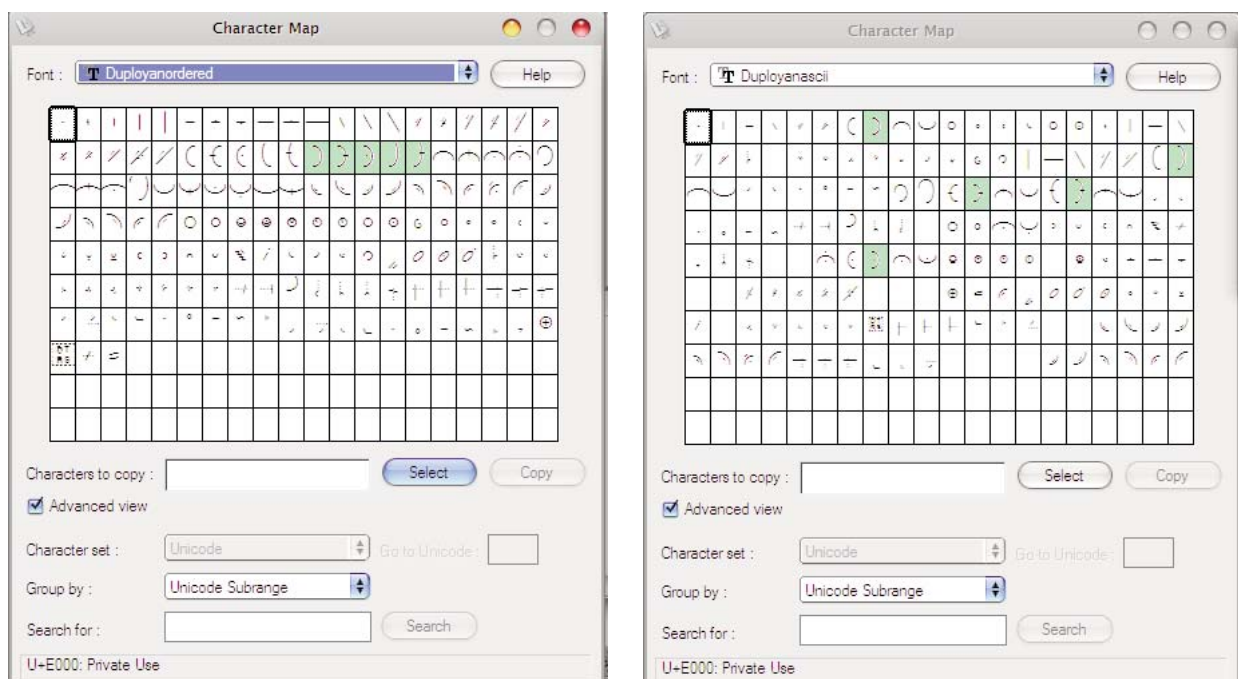
Here we see the glyph palette in Apple’s Character Viewer, which seems to be hard-coded to display in rows of ten glyphs.



Here we see the glyph palette in PopChar, which can be re-sized to various widths. The width here is 17 characters. On the top is the collation-based ordering; on the bottom the non-collation-based ordering.



Here we see the glyph palette in PopChar, re-sized to 35 characters. Again, a stable ordering above, a chaotic ordering below.



Here we see the Character Map in Windows XP, which is hard-coded to 20 characters. Glyphs are easy to find on the left, and not so easy to find on the right.

Now in order to actually make a keyboard layout, or to insert the occasional character into text when a Duployan keyboard is not being used, software applications such as Character Viewer or PopChar or Character Map are routinely used. *Finding* characters in such applications is a *key functionality* which benefits from a collation-based ordering in the code table.

4. “Advantages”. Document N3922 suggests that there are three advantages to the non-collation-based allocation.

4.1. The 128-character block “rule”. It is true that the Principles and Procedures mention that all things being equal it’s reasonable to put higher-frequency characters toward the front of a block with more than 128 characters in it. A number of things should be noted, however. First, the Principles and Procedures are *guidelines*, not *laws*. They are there to help standardizers, not to tie their hands. Second, that rule was approved in 1997; computer processing power is much greater now than it was at that time. Third, it appears that Mr Anderson is citing only part of P&P D.2.1, which reads in full:

When allocating code space to a block requiring fewer than 128 positions, these positions should not cross a 128-code position (half row) boundary. Wherever possible, if the number of positions is close to 128, it is preferable to start the collection at the half-row boundary. For blocks slightly larger than 128 positions the highest frequency characters should all be allocated within the first 128 positions. *This highest frequency allocation principle may be overridden when there is justification to do otherwise. The purpose of this guideline is to insure greater compression ratios for run-length compression techniques.* (See resolution M33.11). Further, for blocks requiring closer to 128 positions it is desirable to start at a half-row boundary.

Note the sentences which we have italicized. With regard to Duployan, the first of these sentences leaves us *free* to choose to prefer a collation-based ordering if there are good, functional, technical reasons to do so (which there are). D.2.1 does not prevent us from doing so. The second sentence gives the explanation for the recommendation, and here it seems that the recommendation is of little consequence given the very small user community and relatively small amount of data in Duployan.

Many script blocks are greater than 8 columns and have common letters after the block end. Myanmar, for instance. Cyrillic, Greek, Latin, Canadian Syllabics, and Coptic have characters all over the place in the UCS. It would be one thing if Duployan were a national script used in massive databases handling personal names and addresses for the purposes of taxation and social insurance. In such a scenario such compression might well be valuable. But Duployan is in a minority of minority scripts, and it is our view that the value such optimization might have does not outweigh the functional advantages of a collation based code-chart order.

Only eleven characters would be affected by this, all of them affixes or punctuation:

DUPLOYAN AFFIX LOW ACUTE
DUPLOYAN AFFIX LOW GRAVE
DUPLOYAN AFFIX LOW DOT
DUPLOYAN AFFIX LOW CIRCLE
DUPLOYAN AFFIX LOW LINE
DUPLOYAN AFFIX LOW WAVE
DUPLOYAN AFFIX LOW ARROW
DUPLOYAN SIGN O WITH CROSS
DUPLOYAN THICK LETTER SELECTOR
DUPLOYAN DOUBLE MARK
DUPLOYAN PUNCTUATION CHINOOK FULL STOP

4.2. Helping amateur developers to select language subsets. As we have indicated above, we do not believe that Mr Anderson's assertions about this advantage stand up to scrutiny. In the first place, how many keyboard layouts would one need? French Duployan, Romanian Stenographie, Chinook script, Pernin shorthand, Perrault shorthand, and Sloan-Duployan, that's 6, multiplied by three platforms (Keyman, MSKLC, Apple) is just 9. Double that, and it's still 18. It simply does not make sense to suggest that this rarified activity should be considered to be more technically functional than an ordering based on a new (and excellent) collation sequence which users of all varieties of Duployan will be encountering anyway.

We suggest that the environments of font design, or of using character pickers to find characters, are more likely to be seen by more users than the environment of amateur keyboard layout creation. Moreover, we do not believe that the annotated names list of Mr Anderson's proposal is in fact sufficient to help anybody to accurately create a language-specific subset. The only way of doing that is to give a

complete specification outside of the code chart, which would also include combining diacritics and any other characters that might be used in a particular Duployan orthography.

We do not believe that there are grounds for the allegation that a collation-based allocation “obfuscates the classes of character variants, and jumbles the characters necessary to each orthography, inviting mistaken identities, resulting in non-conformant documents”. Once again, the code chart is not the place for the specification of which characters are to be used in which orthography. A proper *User’s Guide to the Duployan Script*, whether a Unicode Technical Note or as some other sort of monograph, is the proper place for such information; the names list simply could not give it comprehensively, and so this argument too does not convince us that there are any advantages to a non-collation-based allocation.

4.3. “Inherent properties of characters from the source document”. Unfortunately there is little that can be said about this “advantage”, as there is no description given about what “properties” are “inherent” in Duployan characters. Is a usage scenario intended? The Latin letter *eð* is used in Icelandic; does this mean that “Icelandicness” is an “inherent property” of this letter? If so, it is shared by the properties “Faroese-ness” and “Old-Englishness” and “International-Phonetic-Alphabetness”. It is true that the source materials do not, by and large, offer a rigorous shape-based ordering such as that which Mr Anderson has “invented and imposed” upon Duployan. But this is no fault. Unified Duployan as a member of the UCS *requires* such an ordering, for the benefit of the end users of the script. The fact that no such order is found in the sources means only that the sources were unconcerned about sorting Duployan, since in their glossaries words are normally sorted according to their Latin transcriptions, not their Duployan constituent parts.

4.4. “Imposing a collation-based allocation order”. Neither the UTC nor WG2 makes any formal recommendation regarding the “imposition” of order in the code chart, but it is certainly the case that *in practice* a binary order which is isomorphic to the more formal collation order is usually the order which is adopted.

Getting back to functionality,

Here is the alphabetical order specified for collation (simple consonants only):

H X P B T D F V K G L R M N J S

That is what users of Duployan will get in an application that implements the UCA/14651. Here is the alphabetical order of a binary sort of the non-collation-based order:

H P T F K L M N J S X B D V G R

This is a lot of change. It is like expecting an order like

A B C D E F G H I J K L O M N P

and getting a sort based on:

A C E G I K O M N P B D F H J L

We believe that this will lead to a confusing and undesirable ambiguity for the end user of Duployan, who will learn to expect one sort order but in many environments will get another. There is no advantage to the end user here. Nor, given the usage scenario we have shown above, would there be any advantage to abandoning the shape-based ordering (which is very good) in favour of the “paedagogical” ordering.

Mr Anderson concludes with the following statement:

I reiterate that imposition of the complex collation specification onto the allocation order is inappropriate and cannot be justified by the source documents on which the Duployan proposal is based.

Why is it inappropriate, just because the source materials were not interested in collating Duployan characters? That was not a technical requirement for those who devised the various Duployan orthographies. But those intrepid innovators are long dead, and could not anticipate modern usage environments. Duployan is to be used by people *today*, on computers which will sort their data. The only sort which makes sense for the modern user is the shape-based one which Mr Anderson has expertly devised. There is therefore no advantage to the end user for having the code chart be in any *other* order, as this will simply result in a binary order which conflicts with the superior order proposed for the UCA and ISO/IEC 14651.

5. Issue: Expansion. In Mr Anderson's proposal some blanks are left here and there within the code chart, evidently for potential additions. For instance, between U+1BC63 and U+1BC66 two spaces are left. One might surmise that the following two characters are envisaged:

1BC63 $\overline{\text{r}}$
1BC64 * $\overline{\text{r}}$
1BC65 * $\overline{\text{r}}$
1BC66 $\overline{\text{r}}$

We would have no issue with re-inserting such gaps in the re-ordered chart and would welcome Mr Anderson's advice on this point, should it be agreed that a collation-based code-chart order be adopted.

The Irish National Body favours the encoding of Mr Anderson's Duployan character set, but after considerable reflection, would prefer to see an agreed reorganization of the code chart to follow the collation order *before* and *as a precondition to* a new ballot being sent out after the Busan meeting of WG2. We have attached a draft to this document.

	1BC0	1BC1	1BC2	1BC3	1BC4	1BC5	1BC6	1BC7	1BC8	1BC9
0	1BC00	1BC10	1BC20	1BC30	1BC40	1BC50	1BC60	1BC70	1BC80	
1	1BC01	1BC11	1BC21	1BC31	1BC41	1BC51	1BC61	1BC71	1BC81	
2	1BC02	1BC12	1BC22	1BC32	1BC42	1BC52	1BC62	1BC72	1BC82	
3	1BC03	1BC13	1BC23	1BC33	1BC43	1BC53	1BC63	1BC73	1BC83	
4	1BC04	1BC14	1BC24	1BC34	1BC44	1BC54	1BC64	1BC74	1BC84	
5	1BC05	1BC15	1BC25	1BC35	1BC45	1BC55	1BC65	1BC75	1BC85	
6	1BC06	1BC16	1BC26	1BC36	1BC46	1BC56	1BC66	1BC76	1BC86	
7	1BC07	1BC17	1BC27	1BC37	1BC47	1BC57	1BC67	1BC77	1BC87	
8	1BC08	1BC18	1BC28	1BC38	1BC48	1BC58	1BC68	1BC78	1BC88	
9	1BC09	1BC19	1BC29	1BC39	1BC49	1BC59	1BC69	1BC79	1BC89	
A	1BC0A	1BC1A	1BC2A	1BC3A	1BC4A	1BC5A	1BC6A	1BC7A	1BC8A	
B	1BC0B	1BC1B	1BC2B	1BC3B	1BC4B	1BC5B	1BC6B	1BC7B	1BC8B	
C	1BC0C	1BC1C	1BC2C	1BC3C	1BC4C	1BC5C	1BC6C	1BC7C	1BC8C	
D	1BC0D	1BC1D	1BC2D	1BC3D	1BC4D	1BC5D	1BC6D	1BC7D	1BC8D	
E	1BC0E	1BC1E	1BC2E	1BC3E	1BC4E	1BC5E	1BC6E	1BC7E	1BC8E	
F	1BC0F	1BC1F	1BC2F	1BC3F	1BC4F	1BC5F	1BC6F	1BC7F		

Dot consonants

- 1BC00 ∙ DUPLOYAN LETTER H
 • Chinook, Pernin, Sloan, Perrault
 • non-joining character
- 1BC01 ∙ DUPLOYAN LETTER X
 • Salishan
 • non-joining character

Vertical-line consonants

- 1BC02 | DUPLOYAN LETTER P
 • Chinook number 1
- 1BC03 | DUPLOYAN LETTER B
 → 1BC72 † duployan affix low vertical secant
 → 1BC73 † duployan affix mid vertical secant
 → 1BC74 † duployan affix high vertical secant
- 1BC04 | DUPLOYAN LETTER P N
 = Sloan B B
 → 1BC1E) duployan letter n

Horizontal-line consonants

- 1BC05 - DUPLOYAN LETTER T
 • Chinook number 2
- 1BC06 - DUPLOYAN LETTER TH
 • Chinook, Sloan, Pernin, Perrault
- 1BC07 - DUPLOYAN LETTER SLOAN DH
- 1BC08 - DUPLOYAN LETTER D
 → 1BC75 † duployan affix left horizontal secant
 → 1BC76 † duployan affix mid horizontal secant
 → 1BC77 † duployan affix right horizontal secant
- 1BC09 - DUPLOYAN LETTER DH
 • Chinook
 → 1BC08 — duployan letter d
- 1BC0A — DUPLOYAN LETTER D S
 = Sloan D D
 → 1BC2C ∪ duployan letter s

Northwest-to-southeast diagonal-line consonants

- 1BC0B \ DUPLOYAN LETTER F
 • Chinook number 3
- 1BC0C \ DUPLOYAN LETTER V
- 1BC0D \ DUPLOYAN LETTER F N
 = Sloan V V
 → 1BC1E) duployan letter n

Northeast-to-southwest diagonal-line consonants

- 1BC0E / DUPLOYAN LETTER K
 • Chinook number 4
 • written down and to the left
- 1BC0F / DUPLOYAN LETTER KK
 • Chinook
 • written down and to the left
- 1BC10 / DUPLOYAN LETTER G
 • written down and to the left
- 1BC11 / DUPLOYAN LETTER SLOAN J
 • written down and to the left

- 1BC12 / DUPLOYAN LETTER K M
 • written down and to the left
 = Sloan G G
 → 1BC19 (duployan letter m

Southwest-to-northeast diagonal-line consonants

- 1BC13 / DUPLOYAN LETTER L
 • written up and to the right
 = Pernin letter R
- 1BC14 / DUPLOYAN LETTER HL
 • Chinook
 • written up and to the right
- 1BC15 / DUPLOYAN LETTER LH
 • Chinook
 • written up and to the right
- 1BC16 / DUPLOYAN LETTER R
 • Chinook number 5
 • French number milliards
 • written up and to the right
 = Pernin letter L
 = Pernin Reporters word repeat sign
- 1BC17 / DUPLOYAN LETTER RH
 • Chinook
 • written up and to the right
- 1BC18 / DUPLOYAN LETTER R S
 • written up and to the right
 = Sloan R R
 → 1BC2C ∪ duployan letter s

Left half-circle consonants

- 1BC19 (DUPLOYAN LETTER M
 • Chinook Number 6
- 1BC1A (DUPLOYAN LETTER M N
 • Romanian Mai mult, not Romanian Mult mai shorthand sign
 → 1BC1E) duployan letter n
- 1BC1B (DUPLOYAN SIGN M WITH DOT
 = Romanian sign Mijloc)
- 1BC1C (DUPLOYAN LETTER M S
 = Sloan shorthand letter M M
 → 1BC2C ∪ duployan letter s
- 1BC1D { DUPLOYAN LETTER M N S
 → 1BC1E) duployan letter n
 → 1BC2C ∪ duployan letter s

Right half-circle consonants

- 1BC1E) DUPLOYAN LETTER N
 • Chinook number 7
- 1BC1F } DUPLOYAN LETTER N M
 • not Romanian nu nu shorthand sign
 → 1BC19 (duployan letter m
- 1BC20) DUPLOYAN LETTER N WITH DOT
 = Chinook NG
 = Romanian sign Nici
- 1BC21) DUPLOYAN LETTER N S
 = Pernin, Sloan, Perrault letter NG
 → 1BC2C ∪ duployan letter s
- 1BC22 } DUPLOYAN LETTER N M S
 → 1BC19 (duployan letter m
 → 1BC2C ∪ duployan letter s

Top half-circle consonants

- 1BC23 ⤴ DUPLOYAN LETTER J
• Chinook number 8
= Chinook letter SH
= Pernin letter SH
- 1BC24 ⤴ DUPLOYAN LETTER J M
• not Romanian cee a ce shorthand sign
→ 1BC19 ⤴ duployan letter m
- 1BC25 ⤴ DUPLOYAN LETTER J WITH DOT
= Chinook, Romanian CH
= Sloan ZH
= Chinook, Perrault J
- 1BC26 ⤴ DUPLOYAN SIGN J WITH DOTS INSIDE AND ABOVE
= Romanian sign Ici
- 1BC27 ⤴ DUPLOYAN LETTER J N
→ 1BC1E ⤴ duployan letter n
- 1BC28 ⤴ DUPLOYAN LETTER J S
= Romanian stenographic letter Ge
= Pernin, Perrault letter ZH
= Sloan letter CH
→ 1BC2C ⤴ duployan letter s
- 1BC29 ⤴ DUPLOYAN LETTER J M S
→ 1BC19 ⤴ duployan letter m
→ 1BC2C ⤴ duployan letter s
- 1BC2A ⤴ DUPLOYAN LETTER J S WITH DOT
= Sloan letter hard CH
= Pernin, Perrault letter Ch
- 1BC2B ⤴ DUPLOYAN LETTER J N S
→ 1BC1E ⤴ duployan letter n
→ 1BC2C ⤴ duployan letter s

Bottom half-circle consonants

- 1BC2C ⤵ DUPLOYAN LETTER S
• Chinook number 9
• French Hundreds
- 1BC2D ⤵ DUPLOYAN LETTER S J
• not Romanian sa se shorthand sign
→ 1BC23 ⤴ duployan letter j
- 1BC2E ⤵ DUPLOYAN LETTER S WITH DOT
= Chinook TS
= Chinook, Romanian, Sloan Z
- 1BC2F ⤵ DUPLOYAN LETTER S WITH DOT BELOW
= Romanian Sh
- 1BC30 ⤵ DUPLOYAN LETTER S S
• French, Sloan
= Romanian stenographic letter Ts
= Pernin, Perrault letter Z
→ 1BC2C ⤴ duployan letter s
- 1BC31 ⤵ DUPLOYAN LETTER S J S
→ 1BC23 ⤴ duployan letter j

Downward-sloping quarter-arc consonants

- 1BC32 ⤵ DUPLOYAN LETTER S T
• Pernin, Perrault
• written down
= Sloan SM
- 1BC33 ⤵ DUPLOYAN LETTER S T R
• Pernin, Perrault
• written down
= Sloan SN

- 1BC34 ⤵ DUPLOYAN LETTER S P
• Pernin, Perrault
• written down
= Sloan KW
- 1BC35 ⤵ DUPLOYAN LETTER S P R
• Pernin, Perrault
• written down
= Sloan SKW
- 1BC36 ⤵ DUPLOYAN LETTER T S
• written down
• Perrault
= Sloan STD
- 1BC37 ⤵ DUPLOYAN LETTER T R S
• written down
• Perrault
= Sloan SST
- 1BC38 ⤵ DUPLOYAN LETTER W
• Sloan, Perrault, Pernin
• written down
• takes form of a hook or wave after K and G
- 1BC39 ⤵ DUPLOYAN LETTER WH
• written down
- 1BC3A ⤵ DUPLOYAN LETTER W R
• written down
• Perrault
= Sloan SW

Upward-sloping quarter-arc consonants

- 1BC3B ⤴ DUPLOYAN LETTER S N
• written up
• Perrault
= Pernin KRS
= Sloan SP
- 1BC3C ⤴ DUPLOYAN LETTER S M
• written up
• Perrault
= Pernin GRS
= Sloan SL
- 1BC3D ⤴ DUPLOYAN LETTER K R S
• written up
• Perrault
- 1BC3E ⤴ DUPLOYAN LETTER G R S
• written up
• Perrault
- 1BC3F ⤴ DUPLOYAN LETTER S K
• written up
• Perrault, Pernin
= Sloan TS
- 1BC40 ⤴ DUPLOYAN LETTER S K R
• written up
• Perrault, Pernin
= Sloan DS

Circle vowels

- 1BC41 ○ DUPLOYAN LETTER AOU
- 1BC42 ○ DUPLOYAN LETTER O
• Chinook number 0
- 1BC43 ⊙ DUPLOYAN LETTER WO
• Chinook
- 1BC44 ⊙ DUPLOYAN LETTER WOW
• Salishan

- 1BC45 ◦ DUPLOYAN LETTER WA
- Chinook
 - Not Romanian O+A
 - = Perrault letter OY
 - Chinook number 100s
- 1BC46 ◦ DUPLOYAN LETTER WI
- Chinook
- 1BC47 ◦ DUPLOYAN LETTER WEI
- Salishan
- 1BC48 ◦ DUPLOYAN LETTER OU
- should not be used for Perrault Ow
 - ≈ <initial, final> 1BC49 ◦
 - = Chinook letter oo
 - 1BC5E ◊ duployan letter uh
 - 1BC5F ◊ duployan letter sloan u
 - 1BC60 ◊ duployan letter ooh
- 1BC49 ◦ DUPLOYAN LETTER OW
- should not be used for Romanian U
 - ≈ <medial> 1BC4A ◦
- 1BC4A ◦ DUPLOYAN LETTER ROMANIAN U
- 1BC4B ◦ DUPLOYAN LETTER OA
- = Pernin letter AW
 - = Perrault letter AW
- 1BC4C ◦ DUPLOYAN LETTER A
- Chinook number 10s
- 1BC4D ◦ DUPLOYAN LETTER SLOAN OW
- reverse circle vowel

Orienting half-circle vowels

- 1BC4E ◂ DUPLOYAN LETTER I
- character rotates to match entry angle of preceding consonant
 - character has primary orientation (right and up)
 - = Perrault letter long A, short E (with accents)
 - 1BC53 ◂ duployan letter ie
 - 1BC54 ◃ duployan letter short i
 - 1BC55 ◄ duployan letter ee
 - 1BC56 ◅ duployan letter ui
 - 1BC6E ◄ duployan affix attached i hook
 - 1BC6F ◄ duployan affix attached e hook
- 1BC4F ◃ DUPLOYAN LETTER E
- character rotates to match entry angle of preceding consonant
 - character has secondary orientation (left and down)
 - = Sloan letter long A
 - = Perrault letter short I, long E (with dot accent)
 - 1BC6F ◄ duployan affix attached e hook
- 1BC50 ◄ DUPLOYAN LETTER SLOAN EH
- 1BC51 ◅ DUPLOYAN LETTER ROMANIAN I
- character rotates to match entry angle of preceding consonant, with dot maintaining relative position
 - secondary orienting (left and down)
- 1BC52 ≍ DUPLOYAN LETTER SLOAN EE

Invariant half-circle vowels

- 1BC53 ◂ DUPLOYAN LETTER IE
- Duployan shorthand
 - used as an invariant vowel and for orienting word abbreviations consisting of only vowels
 - 1BC4E ◂ duployan letter i
 - = Pernin letter A
- 1BC54 ◃ DUPLOYAN LETTER SHORT I
- Pernin, Duployan shorthand
 - used as an invariant vowel and for orienting word abbreviations consisting of only vowels
 - 1BC4E ◂ duployan letter i
 - = Consolidated Duployan letter R T R
- 1BC55 ◄ DUPLOYAN LETTER EE
- Pernin, Duployan shorthand
 - used as an invariant vowel and for orienting word abbreviations consisting of only vowels
 - 1BC4E ◂ duployan letter i
- 1BC56 ◅ DUPLOYAN LETTER UI
- Duployan shorthand
 - used as an invariant vowel and for orienting word abbreviations consisting of only vowels
 - 1BC4E ◂ duployan letter i
 - = Pernin letter E

Diagonal-line vowels

- 1BC57 ↗ DUPLOYAN LETTER YE
- 1BC58 ↘ DUPLOYAN LETTER LONG I
- Pernin
 - angles like an “F” when adjacent a K-type consonant

Quarter-circle vowels

- 1BC59 ◂ DUPLOYAN LETTER U
- character rotates to match entry angle of preceding consonant
 - character has primary orientation (right and up)
 - = Romanian stenographic letter EN
 - 1BC38 ◂ duployan letter w
 - 1BC4A ◂ duployan letter romanian u
 - 1BC5D ◂ duployan letter long u
- 1BC5A ◃ DUPLOYAN LETTER EU
- character rotates to match entry angle of preceding consonant
 - character has secondary orientation (left and down)
 - in French usage, may be rendered with a dot contextually
 - = Romanian stenographic letter AN
- 1BC5B ◄ DUPLOYAN LETTER XW
- = Perrault Uh
 - not French Eu
- 1BC5C ◅ DUPLOYAN LETTER U N
- 1BC1E ◅ duployan letter n

Oblong circle vowels

- 1BC5D ◊ DUPLOYAN LETTER LONG U
- Pernin, Perrault
 - this vowel does not rotate to match entry angle of preceding consonant
 - 1BC48 ◊ duployan letter ou

- 1BC5E ◌ DUPLOYAN LETTER UH
 • Sloan
 → 1BC48 ◌ duployan letter ou
- 1BC5F ◌ DUPLOYAN LETTER SLOAN U
 → 1BC48 ◌ duployan letter ou
- 1BC60 ◌ DUPLOYAN LETTER OOH
 • Sloan
 → 1BC48 ◌ duployan letter ou

Small quarter-circle vowels

- 1BC61 ⚡ DUPLOYAN LETTER VOCALIC M
 • primary orienting vowel
 = Perrault letters Am, Em, Im, Um (with accents)
- 1BC62 ♡ DUPLOYAN LETTER NASAL U
 • character positions diacritically, as an orienting vowel, or as an invariant vowel
 • secondary orientation
 • invariant direction down
 • French number 1
 = Pernin letter IN
 = Consolidated Duployan affix INT-R-
- 1BC63 ♡ DUPLOYAN LETTER SLOAN EN
- 1BC64 ♡ DUPLOYAN LETTER SLOAN AN
- 1BC65 ♡ DUPLOYAN LETTER NASAL O
 • character positions diacritically, as an orienting vowel, or as an invariant vowel
 • neutral nasal vowel for transcription of an ambiguous secondary orienting nasal vowel
 • secondary orientation
 • invariant direction up
 = Pernin letter OM
 = Perrault letters An, En, In, Un (with accents)
 = Pernin letter IM
 = Consolidated Duployan affix INT-R-
- 1BC66 ♡ DUPLOYAN LETTER PERNIN AN
 • written down
- 1BC67 ♡ DUPLOYAN LETTER NASAL I
 • character positions diacritically, as an orienting vowel, or as an invariant vowel
 • primary orientation
 • invariant direction down
 • Romanian multiplicative number prefix
 = Pernin letter IM
 = Consolidated Duployan affix INT-R-
- 1BC68 ♡ DUPLOYAN LETTER NASAL A
 • Perrault vocalic N - An, En, In, Un (with accents)
 • character positions diacritically, as an orienting vowel, or as an invariant vowel
 • neutral nasal vowel for transcription of an ambiguous primary orienting nasal vowel
 • primary orientation
 • invariant direction up
 = Pernin letter ON
 = Romanian stenographic letter YN
- 1BC69 ♡ DUPLOYAN LETTER PERNIN AM
 • written down
- 1BC6A ♡ DUPLOYAN LETTER SLOAN ON

Attached affixes

- 1BC6B ↗ DUPLOYAN AFFIX ATTACHED SECANT
 • dots show position on and relative orientation to base glyph and are not rendered
 • as a prefix, takes opposite relative position to following glyph
 • generally crosses adjacent character at perpendicular, but has a bias towards SW/NE angle to contrast 1BC71 ↘
 • default neutral secant affix
 = French suffix -anse
 = Pernin prefix Pre-
 = Sloan affix Ax-/-ext
- 1BC6C ↖ DUPLOYAN AFFIX ATTACHED TANGENT
 • dots show position on and relative orientation to base glyph and are not rendered
 • as a prefix, takes opposite relative position to following glyph
 = French suffix -tan
 = Romanian shorthand letter Str-/-str
- 1BC6D ⤵ DUPLOYAN AFFIX ATTACHED TAIL
 • orienting character
 = French suffix -sionaire
- 1BC6E ⚡ DUPLOYAN AFFIX ATTACHED I HOOK
 • glyph is retrograde and opens left or right, dependent on preceding letter
 • dots show position of preceding glyph and are not rendered
 → 1BC4E ◌ duployan letter i
 = French suffix -tou
 = Sloan affix Irre-/-ary
- 1BC6F ⚡ DUPLOYAN AFFIX ATTACHED E HOOK
 • glyph is retrograde and opens up or down, dependent on preceding letter
 • dots show position of preceding glyph and are not rendered
 → 1BC4F ◌ duployan letter e
 = French suffix -te
- 1BC70 ⚡ DUPLOYAN AFFIX ATTACHED TANGENT HOOK
 • attached affix
 • dots show position on and relative orientation to base glyph and are not rendered
 = Romanian affix Ist-/-ism
 = Consolidated Duployan prefix T-R-
- 1BC71 ↘ DUPLOYAN AFFIX ATTACHED LEFT-TO-RIGHT SECANT
 • dots show position on and relative orientation to base glyph and are not rendered
 • generally crosses adjacent character at perpendicular, but has a bias towards NW/SE angle to contrast 1BC6B ↗
 • as a suffix, takes opposite relative position to following glyph
 = Pernin prefix Per-

Vertical affixes

- 1BC72 † DUPLOYAN AFFIX LOW VERTICAL SECANT
 = Pernin Reporters Sub-
 • dots show position on base glyph and are not rendered
 → 1BC03 | duployan letter b

- 1BC73 † DUPLOYAN AFFIX MID VERTICAL SECANT
= Pernin Reporters Trans-
• dots show position on base glyph and are not rendered
→ 1BC03 | duployan letter b
- 1BC74 † DUPLOYAN AFFIX HIGH VERTICAL SECANT
= Pernin Reporters Super-
• dots show position on base glyph and are not rendered
→ 1BC03 | duployan letter b

Horizontal affixes

- 1BC75 † DUPLOYAN AFFIX LEFT HORIZONTAL SECANT
= Pernin Reporters Extra-
• dots show position on base glyph and are not rendered
→ 1BC08 — duployan letter d
- 1BC76 † DUPLOYAN AFFIX MID HORIZONTAL SECANT
= Pernin Reporters Inter-
• dots show position on base glyph and are not rendered
→ 1BC08 — duployan letter d
- 1BC77 † DUPLOYAN AFFIX RIGHT HORIZONTAL SECANT
= Pernin Reporters Contra-
• dots show position on base glyph and are not rendered
→ 1BC08 — duployan letter d

High affixes

- 1BC78 ′ DUPLOYAN AFFIX HIGH ACUTE
= French suffix -ment
= Romanian suffix -mant
= Pernin Sub-
= Pernin Reporters' suffix Pro-
→ 02CA ′ modifier letter acute accent
- 1BC79 ′ DUPLOYAN AFFIX HIGH TIGHT ACUTE
= Pernin Pro-
• as a suffix, placed above and to the right of the following letter
- 1BC7A ` DUPLOYAN AFFIX HIGH GRAVE
= French suffix -ien
= Pernin suffix Con-
→ 02CB ` modifier letter grave accent
- 1BC7B ~ DUPLOYAN AFFIX HIGH LONG GRAVE
= Pernin Contra-
- 1BC7C · DUPLOYAN AFFIX HIGH DOT
• not Romanian hundreds - use U+0307 ◌̇
combining dot above and U+0308 ◌̈
combining diaeresis
• French number thousands
= French suffix -eur
= Romanian shorthand affix trans-/lui
→ 02D9 · dot above
- 1BC7D ° DUPLOYAN AFFIX HIGH CIRCLE
• Not Romanian number grade or percent suffix
• French ordinal number
= French suffix -euse
→ 00B0 ° degree sign
→ 02DA ° ring above

- 1BC7E - DUPLOYAN AFFIX HIGH LINE
= French suffix -iste
= Romanian shorthand affix -tor
= Pernin affix Dis-
→ 02C9 ˘ modifier letter macron
- 1BC7F ˘ DUPLOYAN AFFIX HIGH WAVE
= French suffix -ificatif
→ 02DC ˘ small tilde
- 1BC80 ˆ DUPLOYAN AFFIX HIGH VERTICAL
• also functions as attached affix vertical up with ZWJ
• this affix is about half as long as Duployan Letter P
• as a prefix, has falling stroke direction
= Pernin ZWJ + -ime
= Sloan Tele-
→ 1BC02 † duployan letter p


Low affixes

- 1BC81 ˘ DUPLOYAN AFFIX LOW ACUTE
= French suffix -cion
= Pernin prefix ex-
→ 02CF ˘ modifier letter low acute accent
- 1BC82 ˘ DUPLOYAN AFFIX LOW TIGHT ACUTE
= Pernin Suf-, Sug-
• as a suffix, placed under and to the right of the following letter
- 1BC83 ˘ DUPLOYAN AFFIX LOW GRAVE
= French suffix -ion
• French number millions
→ 02CE ˘ modifier letter low grave accent
- 1BC84 ˘ DUPLOYAN AFFIX LOW LONG GRAVE
= Pernin Extra-
- 1BC85 ˘ DUPLOYAN AFFIX LOW DOT
= French suffix -ie
• French iterative number
= Romanian shorthand affix Inter-
• not Romanian millions - see U+0323 ◌̣
combining dot below and U+0324 ◌̤
combining diaeresis below
- 1BC86 ◌ DUPLOYAN AFFIX LOW CIRCLE
= French suffix -iere
• French percent
→ 02F3 ◌ modifier letter low ring
- 1BC87 ˘ DUPLOYAN AFFIX LOW LINE
= French suffix -isme
= Pernin affix Mis-
→ 02D7 - modifier letter minus sign
- 1BC88 ˘ DUPLOYAN AFFIX LOW WAVE
= French suffix -ification
→ 02F7 ˘ modifier letter low tilde
- 1BC89 ˘ DUPLOYAN AFFIX LOW VERTICAL
• functions as attached affix vertical down with ZWJ
• this affix is about half as long as Duployan the letter P
• as a prefix, has rising stroke direction
= Pernin ZWJ + -ine
→ 1BC02 † duployan letter p
- 1BC8A ˘ DUPLOYAN AFFIX LOW ARROW
= Romanian prefix Sub-
• low affix

Chinook sign

- 1BC8B ⊕ DUPLOYAN SIGN O WITH CROSS
- Chinook Likalisti

Sloan R-form selector






- 1BC8C  DUPLOYAN THICK LETTER SELECTOR
- commonly abbreviated DTLS
 - Sloan R-flavored letters
 - Shape shown is arbitrary and is not visibly rendered
 - Causes previous Duployan character to be rendered bold

Shorthand double mark





- 1BC8D ≠ DUPLOYAN DOUBLE MARK
- Dots show position on and relative orientation to base glyph and are not rendered
 - Romanian, Sloan shorthands
 - Should be used with M, N, J, and S for the Romanian word signs Mai mult, Nu nu, Ceea ce, and Sa se
 - Can be doubled and tripled

Chinook punctuation

- 1BC8E = DUPLOYAN PUNCTUATION CHINOOK FULL STOP

	1BCA
0	 1BCA0
1	 1BCA1
2	 1BCA2
3	 1BCA3
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Shorthand Format Controls

- 1BCA0  SHORTHAND FORMAT LETTER OVERLAP
 - shape shown is arbitrary and is not visibly rendered
- 1BCA1  SHORTHAND FORMAT CONTINUING OVERLAP
 - shape shown is arbitrary and is not visibly rendered
- 1BCA2  SHORTHAND FORMAT DOWN STEP
 - = Romanian shorthand affix -tsion-
 - = Sloan contracted ending oo/o + ZWSP
 - shape shown is arbitrary and is not visibly rendered
- 1BCA3  SHORTHAND FORMAT UP STEP
 - = Sloan contracted ending uh/au/aui + ZWSP
 - shape shown is arbitrary and is not visibly rendered