L2/01-416
2 November 2001

Issues with GB 18030 to Unicode Mapping Tables

John H. Jenkins
Apple Computer, Inc.
jenkins@apple.com

This issue is probably best addressed through three emails:

Email #1:

From: Ken Lunde <lunde@adobe.com>
Date: Mon Oct 15, 2001 10:22:45 AM US/Mountain
To: "John H. Jenkins" <jenkins@apple.com>
Cc: rscook@socrates.berkeley.edu, Tom BISHOP <tb@wenlin.com>, Alex SIMAGIN
    <al@legion.ru>, Dirk MEYER <dmeyer@adobe.com>, emuller@adobe.com
Subject: Re: GB 18030

John,

You wrote:

>I'll make sure it gets on the agenda. Meanwhile, I'm confused about one point. How
>are these characters handled in the GB 18030 -> Unicode mapping tables out there?

They are mapped to the user-defined space as follows (the first field is the Adobe-GB1-4 CID, the second field is the GB 18030-2000 code point, and the third field is the Unicode UTF-16 code point):

22048	fe51	e816
22049	fe52	e817
22050	fe53	e818
22056	fe59	e81e
22064	fe61	e826
22069	fe66	e82b
22070	fe67	e82c
22075	fe6c	e831
22076	fe6d	e832
22085	fe76	e83b
22093	fe7e	e843
22110	fe90	e854
22111	fe91	e855
22126	fea0	e864

-- Ken

Email #2:

From: Kenneth Whistler <kenw@sybase.com>
Date: Wed Oct 17, 2001 08:53:44 PM US/Mountain
To: rscook@socrates.berkeley.edu
Cc: unicore@unicode.org, kenw@sybase.com
Subject: Re: Is it too late for WG2?

Richard,

>This is the list, with the ones that we (myself, Ken and Tom Bishop)
>have already mapped:

Thanks for the list. These are, indeed, all from the famous "Level 4" excreta in GB 18030, p. 81 in the original standard, most of which got mapped into Extension A forms eventually.

They all, in fact, look like oddball radical forms.
See 2E80..2EF3 for the rest of this kind of stuff. They don't show up in the IRG trail of documents (dating back to June 6, 1992, WG2 N824) on CJK radicals, however, and I suspect were culled as defective in some way, or added to their candidate list of components, rather than radicals. But that didn't keep them out of GB 18030, apparently, even though the repertoire of GB 18030 is nominally supposed to be aligned with GB 13000.1.

01.) [U+20087] <- RC
02.) [U+20089] <- RC
03.) [U+200CC] <- RC
04.) [U+

This is the top half of U+770B, a component, rather than a radical, although I suspect someone, sometime, has listed it as a radical or radical variant.
Cf. the top half of U+7740 and the CJK radical variant U+2EB6, which is sometimes used as an alternative lookup path for U+7740, instead of U+2F6C.

05.) [U+
06.) [U+
07.) [U+

Similar examples can be turned up for these.

08.) [U+215D7] <- KL
09.) [U+
10.) [U+2298F] <- RC
11.) [U+20509] <- TB
12.) [U+2099D] <- KL
13.) [U+241FE] <- KL
14.) [U+470c] <- RC <-*

This guy is a pseudo-radical for characters like U+81E0 and U+883B.

>Note that #14 should be a compatibility ideograph ... not quite sure if
>any of the others are compatibility ... items 4..7 and 9 are as yet
>unmapped (by us). Here's the original PDF Lunde gave us:

I think it is pretty clear that these are all detritus from character lookup indices somewhere -- character component fragments that have been used at some point or other to look up characters.

For now, the ones that can be meaningfully mapped to Plane 2 Extension B characters probably should be nailed down to that, as you have shown above.

The rest can be appended after U+2EF4, if no match shows up, as more "CJK radicals supplement". But that would take action on an amendment to 10646.

In the meantime, it is clear that *everybody's* tables for GB 18030 are screwed up if these aren't accounted for, and furthermore, that they will be screwed up again even *when* these "characters" are added to 10646/Unicode, since that will once again shift and bust up the range mappings for GB 18030.

What an f***ing mess this GB 18030 is!

-- (a very grumpy) Ken

Email #3:

From: Kenneth Whistler <kenw@sybase.com>
Date: Tue Oct 23, 2001 12:54:18 PM US/Mountain
To: markus.scherer@jtcsv.com
Cc: unicore@unicode.org, kenw@sybase.com
Subject: GB 18030 Mapping Problems (was Re: Is it too late for WG2?)

Markus,

>Hello all, I am sorry that I overlooked this thread
>last week (so my comments may be wasted now).

It isn't really a WG2 issue anyway.

>I would vote against _any_ changes to the GB 18030 mappings.

I am very sympathetic to this position, but unfortunately, I don't think it is a tenable position.

>The reason is simple:
>
>GB 18030 is basically a UTF on top of GBK. All Unicode code
>points (except for single surrogates) are mapped.

The rub here is the term "basically".

>Every code point on plane 2 (and everywhere else) already
>has a mapping. Each PUA code point has its mapping.
>
>You cannot change GB 18030 by assigning characters, just
>like you cannot change UTF-8 by assigning new characters.

This is where you are wrong. Neither WG2 nor the UTC control GB 18030. The Chinese standards body can change GB 18030 any time they choose by assigning characters. Say the Chinese standards body decides that for GB 18030:2003 they are going to assign the character HAIRY FLAPDOODLE to E785, and requires (by law) that all vendors support HAIRY FLAPDOODLE in order to sell software in China. The vendors will quickly add HAIRY FLAPDOODLE to their fonts and temporarily map it into user space for their Unicode implementations, disrupting the tables. They will then turn around and require that HAIRY FLAPDOODLE be encoded in Unicode and 10646 for compatibility with GB 18030. That will again disrupt the tables.

And essentially there is nothing we can do about it except plead with the Chinese national body not to do such disruptive things.

>Also, GB 18030 has been implemented and shipped based on
>the mapping table from November of 2000. Any change will
>cause incompatibilities.

E816:FE51
E817:FE52
E818:FE53

```
E81E:FE59
E826:FE61
E82B:FE66
E82C:FE67
E831:FE6C
E832:FE6D
E83B:FE76
E843:FE7E
E854:FE90
E855:FE91
E864:FEA0
```

These are the 14 mappings in question, where the Unicode value is a PUA code point, and the GB 18030 value is one of the unmapped entities.

That works for Unicode 3.0 -- in fact, you have no alternative, since the 14 entities aren't in Unicode 3.0.

But Unicode 3.1 changes things, since some portion of these were added in Extension B on Plane 2. If you don't change the mapping for the ones on Plane 2, you end up with bizarre roundtrip mapping problems for them, and things represented by the wrong code points. If you *do* change the mapping, then you end up with versioned tables that convert data differently, depending on the level of Unicode, and (eventually, presumably) the level of GB 18030.

In other words, we are stuck between a rock and a hard place.

Once people bought into this architectural nightmare of treating GB 18030 as UTF-GB, and started following GB 18030 in mapping all the user space and unassigned code points, rather than just mapping the *assigned* characters of GB 18030, you bought yourself this unending heartache.

Unless you think you can get China to agree to never add anything again to their standards and you think you can get all the vendors, including Adobe, to agree to support the 14 entities in PUA space, rather than as regular characters, I don't think a position of no change to the tables is one that can be held long against the assault of other realities.

--Ken

So, what's the best thing to do? Apple and other companies are under a 31 December 2001 deadline; either support GB 18030 or you can't sell your software in the PRC. Supporting GB 18030 through Unicode is OK. But if this support ends up mapping non-user area stuff into the Unicode user area, there are considerable risks.

Any sage advice? Any united front? Any hope?