Ben Fino-Radin <bfinoradin@gmail.com>

# Korean DBCS

10 messages

**Ben Fino-Radin** <ben.finoradin@rhizome.org> Sun, Sep 9, 2012 at 1:44 PM
To: lunde@adobe.com, ken.lunde@gmail.com

Hi Ken,

I am doing some work for the Whitney Museum of American Art on restoring an old web page from 1995.

One of the main issues is that the page(s) contain a large amount of Korean text, which now does not render properly. I have tried nearly everything under the sun – from viewing the content in emulations of Windows 95 with the Korean IME installed, to manually mapping the DBCS hex values to various code pages. I have been unsuccessful in my attempts to identify the encoding or character set, and am hoping that you might be able to provide some guidance and wisdom…

Luckily, we have two pages (a small portion) of the text that were printed out in 1996, with the Hangul characters correctly rendered. I ran (and visually validated) some optical character recognition on these, so we now have the ability to know for certain what characters the DBCS values are supposed to be rendered as (that is, for only a small portion of the complete content).

For example, the following DBCS hex values: B1 DB E7 85 BC 96 C0 A3 D2 F8 E1 D5 C0 A9
Should be rendered as: 광주비엔날레여

Is this a lost cause? Any guidance would be greatly appreciated.

Best,
Ben

–
Ben Fino-Radin
Digital Conservator
**Rhizome at the New Museum**
235 Bowery New York, NY 10002
(212) 219-1288 x258
ben.finoradin@rhizome.org

**2 attachments**

**sentence5.html**
5K

**95.253_Davis-Korean Characters.pdf**
104K

**Ken Lunde** <lunde@adobe.com> Sun, Sep 9, 2012 at 1:48 PM
To: "ben.finoradin@rhizome.org" <ben.finoradin@rhizome.org>

Ben,

Give me a few minutes. I'll be home soon.

-- Ken
[Quoted text hidden]

---

**Ben Fino-Radin** <ben.finoradin@rhizome.org> Sun, Sep 9, 2012 at 2:30 PM
To: Ken Lunde <lunde@adobe.com>

Thanks Ken – Looking forward to hearing your thoughts.
[Quoted text hidden]

---

**Ken Lunde** <lunde@adobe.com> Sun, Sep 9, 2012 at 2:46 PM
To: Ben Fino-Radin <ben.finoradin@rhizome.org>

Ben,

I was not able to make any correspondences between these character codes and any known Korean encoding, but I figured out what happened. The original encoding was EUC-KR, but the individual bytes were converted from Extended ASCII to Mac Roman encoding. For example, 0xBC and 0x96 were originally 0xBA and 0xF1, which are for ㅂ|. Most of what you need to repair the bytes can be found in the table here:

http://en.wikipedia.org/wiki/Mac_OS_Roman

This is a form of encoding damage.

Does this help?

-- Ken
[Quoted text hidden]

---

**Ken Lunde** <lunde@adobe.com> Sun, Sep 9, 2012 at 3:06 PM
To: Ben Fino-Radin <ben.finoradin@rhizome.org>

Ben,

The conversion was likely from Windows 1252 to Mac Roman. Here is a Windows 1252 table:

http://en.wikipedia.org/wiki/Windows-1252

If you construct a correspondence table from Windows 1252 to Mac Roman, then reverse it, you can probably restore the text. When you restore it, I suggest converting the EUC-KR characters into NCRs (Numeric Character References), which are immune to such forms of damage, and explicitly reference a Unicode code point.

Regards…

-- Ken
[Quoted text hidden]

---

**Ben Fino-Radin** <ben.finoradin@rhizome.org> Sun, Sep 9, 2012 at 3:20 PM
To: Ken Lunde <lunde@adobe.com>

Ken,

This is fantastic – much work to be done, but this does seem to explain the mystery.

I'll let you know if I am successful!

Best,
Ben
[Quoted text hidden]

---

**Ben Fino-Radin** <ben.finoradin@rhizome.org> Sun, Sep 9, 2012 at 5:28 PM
To: Ken Lunde <lunde@adobe.com>

Initial results are looking quite promising…
[Quoted text hidden]

---

**Ben Fino-Radin** <ben.finoradin@rhizome.org> Sun, Sep 9, 2012 at 6:38 PM
To: Ken Lunde <lunde@adobe.com>

Hi Ken,

I've put together a conversion table for about half of the table, but it seems that there are a few key discrepancies.

For example, the OCRed text provides 광주비엔낱레여, which Google translates to "Gwangju Biennale open". This is very cool, as this text is in fact from the opening of the 1995 Gwangju Biennale. The Windows-1252 conversion of the damaged encoding, provides 광주비엔뿯뇫여 (claiming 뿯 rather than 낱, and 뇫 rather than 레). Google translates to "Guangzhou Biel ppwot nyokyeo".

Looking at these bytes, it would appear that a successful restoration should look like:

0xD2 = 0xB3
0xF8 = 0xB9
0xE1 = 0xB7
0xD5 = 0xB9

rather than

0xD2 = 0x97
0xF8 = 0xAF
0xE1 = 0x87
0xD5 = 0x92

Can you give me an idea of what lead you to Windows-1252 as the potential match? What should I look for in viable alternatives? If you are confident that this is truly viable, I could look into hiring you to consult further on the project (that is if you are available and interested).

Many thanks,
Ben
[Quoted text hidden]

---

**Ken Lunde** <lunde@adobe.com> Sun, Sep 9, 2012 at 7:14 PM
To: Ben Fino-Radin <ben.finoradin@rhizome.org>

Ben,

If most of the bytes are being restored correctly, then the conversion that destroyed the data is largely based on Windows-1252, which itself based on ISO 8859-1. You may need to special-case some of the mappings. Windows-1252 has been the most widely used Western encoding for web pages for years, which is why I thought to look at its tables. You may also need to double-check the tables. The only other viable alternatives are related encodings. The number of affected byte values is 128, so at least the mapping table can be small. I may have some more ideas later today or tomorrow.

Regards…

-- Ken
[Quoted text hidden]

---

**Ken Lunde** <lunde@adobe.com> Tue, Sep 11, 2012 at 12:59 PM
To: Ben Fino-Radin <ben.finoradin@rhizome.org>

Ben,

Were you able to make any more progress on this? You might find the following to be helpful:

http://harvey.nu/applescript_encoding_macroman_convert.html

I suspect the main issue will be to correctly handle the converted bytes that are outside the scope of the characters that are common between Mac Roman and Windows 1252. Most of the 128 effected byte values should convert easily, and a small number will require another method. Hopefully, the conversion that took place didn't collapse two different Windows 252 code points into the same Mac Roman one.

Please keep me posted.

-- Ken
[Quoted text hidden]