$Revision: 1.2 $ $Author: death $ $Date: 93/04/04 23:44:39 $

MacToNeXTText

INHERITS FROM TextConverter

DECLARED IN MacToNeXTText.h

CLASS DESCRIPTION

The MacToNeXTText class serves to convert raw text using the standard Mac character set to the standard NeXT character set. It accomplishes this by overriding the methods provided by TextConverter class, and introducing its own conversion algorithms instead. See the TextConverter class document for details about the way one of these classes works.

There are zillions of fonts on the Mac. Apple introduced a set with the Macintosh originally. The general layout for these type families is defined in Inside Macintosh Volume 1 page 221 (many of the fancy families, like San Francisco, do not have all the characters listed in the table). With the LaserWriter Plus came several new type families, and these used a superset of the character set defined in Inside Mac. This class understands the superset used by faces such as Avant Garde, Bookman, etc. This is just to say that it will do a reasonable job of converting Mac text that was created using any of Apple's type families to NeXT text. When they extend themselves beyond the basic ASCII character set, third parties, both commercial

and non commercial, often create their own character mappings which may or may not have any relation to Apple's. This class can not garantee to process those correctly. Recognizing that some fonts may have stuck strictly to the table in Inside Mac, and then done their own thing in the undefined character spaces, this object allows one to toggle a mode with SetMode:. If passed YES, it will convert only those characters defined in the Inside Mac table. If NO, it will use the full table as shown in the LaserWriter families. The default is NO.

A table listing the conversions done by this class is provided at the end.

INSTANCE VARIABLES

Declared in MacToNeXTText	Boolean	strictIM

METHOD TYPES

	Initalizing	- init
Converting characters	- ConvertCharacter: - ConvertString:WithLength:	
Setting conversion type	- SetMode:	

CLASS METHODS

None

INSTANCE METHODS

ConvertCharacter:

- (Character) **ConvertCharacter:** (Character) *macChar*
    0 Result codes and test
    1 The returned character

This returns a single character in the NeXTstep character encoding set that most closely matches the specified Macintosh character. If the match can not be made exactly, this sets the error to errCANTMAPTOONE.

ConvertString:WithLength:

- (Pointer) **ConvertString:** (Pointer) *theData* **WithLength:** (Integer) *length*
    0 Result codes and test
    1 The returned Pointer
    2 The lenght of the returning data

This behaves as documented in the TextConverter class: it converts data from the source string to a destination string it creates, and it then returns this string with its length. See the description above, and the table below, for encoding details.

init

- (Instance) **init**
    0 Result codes and test

This initalizes the object, including setting its strictIM flag to NO.

UseIM1:

- (Instance) **UseIM1:** (Boolean) *doItStrictly*
    0 Result codes and test

This allows the caller to specify whether future conversions should use only the table found in Inside Macintosh Volume I, p. 221. or use the character set (which is a superset of the former) used by the families that are used with the LaserWriters. The only reason to specify YES (i.e. use the restricted IM V1 table) is if one is

converting some older text which made use of those upper characters in other ways, and one doesn't want them to be shuffled around. This is just to say: the odds are REAL good that you won't ever use this.

BUGS AND PROBLEMS

none yet

ENHANCEMENT IDEAS

none

CONSTANT, DEFINED TYPES AND ERROR CODES

```
#define errCANTMAPTOONE             1001
```

CONVERSION TABLE

Characters from 0x00 to 0x79 are converted exactly as the superclass converts them, with the following exceptions (the second is to convert the line termination character)

Character name	**Mac code**	**NeXT code**
NUL	0x00	0x00
CR/LF	0x0C	0x0A

Characters above 0x79 are converted as detailed in the following columns. Note that an entry of `ÐÐ' means that the specified Mac character could not be converted. When this occurrs, the object returns the Mac character code, and sets an error value. If the strict Inside Mac V. 1 flag is set, then all characters above D8 are returned as themselves with no error (as shown in column 5). If one uses the ConvertString method, the object returns the values detailed in

column 4. Note that these never return a value of `ÐÐ', but instead return the name of the character it could not convert. (the values with `ÐÐ' in the final column will get the strings showin the ConvertString column if this flag is set when calling that method). Finally note that the first four characters are only actually found in Chicago. However, they shouldn't be turning up, in general, in other fonts, so I took the liberty of including them here.

Character name	**Mac code**	**NeXT code**	**ConvertString**	**With In. Mac flag**
commandsymbol	0x11	ÐÐ	[commandsymbol]	ÐÐ
check	0x12	ÐÐ	[check]	ÐÐ
diamond	0x13	ÐÐ	[diamond]	ÐÐ
apple	0x14	ÐÐ	[apple]	ÐÐ
Adieresis	0x80	0x85	0x85	0x85
Aring	0x81	0x86	0x86	0x86
Ccedilla	0x82	0x87	0x87	0x87
Eacute	0x83	0x89	0x89	0x89
Ntilde	0x84	0x91	0x91	0x91
Odieresis	0x85	0x96	0x96	0x96
Udieresis	0x86	0x9A	0x9A	0x9A
aacute	0x87	0xD6	0xD6	0xD6
agrave	0x88	0xD5	0xD5	0xD5
acircumflex	0x89	0xD7	0xD7	0xD7
adieresis	0x8A	0xD9	0xD9	0xD9
atilde	0x8B	0xD8	0xD8	0xD8
aring	0x8C	0xDA	0xDA	0xDA
ccedilla	0x8D	0xDB	0xDB	0xDB
eacute	0x8E	0xDD	0xDD	0xDD
egrave	0x8F	0xDC	0xDC	0xDC
ecircumflex	0x90	0xDE	0xDE	0xDE
dieresis	0x91	0xDF	0xDF	0xDF
iacute	0x92	0xE2	0xE2	0xE2
igrave	0x93	0xE0	0xE0	0xE0
icircumflex	0x94	0xE4	0xE4	0xE4
idieresis	0x95	0xE5	0xE5	0xE5
ntilde	0x96	0xE7	0xE7	0xE7
oacute	0x97	0xED	0xED	0xED
ograve	0x98	0xEC	0xEC	0xEC

ocircumflex	0x99	0xEE	0xEE	0xEE
odieresis	0x9A	0xF0	0xF0	0xF0
otilde	0x9B	0xEF	0xEF	0xEF
uacute	0x9C	0xF3	0xF3	0xF3
ugrave	0x9D	0xF2	0xF2	0xF2
ucircumflex	0x9E	0xF4	0xF4	0xF4
udieresis	0x9F	0xF6	0xF6	0xF6
dagger	0xA0	0xB2	0xB2	0xB2
degree	0xA1	ÐÐ	[degrees]	ÐÐ
cent	0xA2	0xA2	0xA2	0xA2
sterling	0xA3	0xA3	0xA3	0xA3
section	0xA4	0xA7	0xA7	0xA7
bullet	0xA5	0xB7	0xB7	0xB7
paragraph	0xA6	0xB6	0xB6	0xB6
germandbls	0xA7	0xFB	0xFB	0xFB
registerserif	0xA8	0xB0	0xB0	0xB0
copyrightserif	0xA9	0xA0	0xA0	0xA0
trademarkserif	0xAA	ÐÐ	[trademarkserif]	ÐÐ
acute	0xAB	0xC2	0xC2	0xC2
dieresis	0xAC	0xC8	0xC8	0xC8
notequal	0xAD	ÐÐ	[notequal]	ÐÐ
AE	0xAE	0xE1	0xE1	0xE1
Oslash	0xAF	0xE9	0xE9	0xE9
infinity	0xB0	ÐÐ	[infinity]	ÐÐ
plusminus	0xB1	0xD1	0xD1	0xD1
lessequal	0xB2	ÐÐ	[lessequal]	ÐÐ
greaterequal	0xB3	ÐÐ	[greaterequal]	ÐÐ
yen	0xB4	0xA5	0xA5	0xA5
mu	0xB5	0x9D	0x9D	0x9D
partialdiff	0xB6	ÐÐ	[partialdiff]	ÐÐ
summation	0xB7	ÐÐ	[summation]	ÐÐ
product	0xB8	ÐÐ	[product]	ÐÐ
pi	0xB9	ÐÐ	[pi]	ÐÐ
integral	0xBA	ÐÐ	[integral]	ÐÐ
ordfeminine	0xBB	0xE3	0xE3	0xE3
ordmasculine	0xBC	0xEB	0xEB	0xEB

Omega	0xBD	ĐĐ	[Omega]	ĐĐ
ae	0xBE	0xF1	0xF1	0xF1
oslash	0xBF	0xF9	0xF9	0xF9
questiondown	0xC0	0xBF	0xBF	0xBF
exclamdown	0xC1	0xA1	0xA1	0xA1
logicalnot	0xC2	0xBE	0xBE	0xBE
radical	0xC3	ĐĐ	[radical]	ĐĐ
florin	0xC4	0xA6	0xA6	0xA6
approxequal	0xC5	ĐĐ	[approxequal]	ĐĐ
delta	0xC6	ĐĐ	[delta]	ĐĐ
guillemotleft	0xC7	0xAB	0xAB	0xAB
guillemotright	0xC8	0xBB	0xBB	0xBB
elipsis	0xC9	0xBC	0xBC	0xBC
nbspace	0xCA	0x80	0x80	0x80
Agrave	0xCB	0x81	0x81	0x81
Atilde	0xCC	0x84	0x84	0x84
Otilde	0xCD	0x95	0x95	0x95
OE	0xCE	0xEA	0xEA	0xEA
oe	0xCF	0xFA	0xFA	0xFA
endash	0xD0	0xB1	0xB1	0xB1
emdash	0xD1	0xD0	0xD0	0xD0
quotedblleft	0xD2	0xAA	0xAA	0xAA
quotedblright	0xD3	0xBA	0xBA	0xBA
quoteleft	0xD4	0x60	0x60	0x60
quoteright	0xD5	0x27	0x27	0x27
divide	0xD6	0x9F	0x9F	0x9F
lozenge	0xD7	ĐĐ	[lozenge]	ĐĐ
ydieresis	0xD8	0xFD	0xFD	0xFD
Ydieresis	0xD9	ĐĐ	[Ydieresis or a picture]	0xD9
fraction	0xDA	0xA4	0xA4	0xDA
currency	0xDB	0xA8	0xA8	0xDB
guilsinglleft	0xDC	0xAC	0xAC	0xDC
guilsinglright	0xDD	0xAD	0xAD	0xDD
fi	0xDE	0xAE	0xAE	0xDE
fl	0xDF	0xAF	0xAF	0xDF
daggerdbl	0xE0	0xB3	0xB3	0xE0

periodcentered	0xE1	0xB4	0xB4	0xE1
quotesinglebase	0xE2	0xB8	0xB8	0xE2
quotedblbase	0xE3	0xB9	0xB9	0xE3
perthousand	0xE4	0xBD	0xBD	0xE4
Acircumflex	0xE5	0x83	0x83	0xE5
Ecircumflex	0xE6	0x8A	0x8A	0xE6
Aacute	0xE7	0x82	0x82	0xE7
Edieresis	0xE8	0x8B	0x8B	0xE8
Egrave	0xE9	0x88	0x88	0xE9
Iacute	0xEA	0x8D	0x8D	0xEA
Icircumflex	0xEB	0x8E	0x8E	0xEB
Idieresis	0xEC	0x8F	0x8F	0xEC
Igrave	0xED	0x8C	0x8C	0xED
Oacute	0xEE	0x93	0x93	0xEE
Ocircumflex	0xEF	0x94	0x94	0xEF
apple	0xF0	ÐÐ	[apple]	0xF0
Ograve	0xF1	0x92	0x92	0xF1
Uacute	0xF2	0x98	0x98	0xF2
Ucircumflex	0xF3	0x99	0x99	0xF3
Ugrave	0xF4	0x97	0x97	0xF4
dotlessi	0xF5	0xF5	0xF5	0xF5
circumflex	0xF6	0xC3	0xC3	0xF6
tilde	0xF7	0xC4	0xC4	0xF7
macron	0xF8	0xC5	0xC5	0xF8
breve	0xF9	0xC6	0xC6	0xF9
dotaccent	0xFA	0xC7	0xC7	0xFA
ring	0xFB	0xCA	0xCA	0xFB
cedilla	0xFC	0xCB	0xCB	0xFC
hungarumlaut	0xFD	0xCD	0xCD	0xFD
ogonek	0xFE	0xCE	0xCE	0xFE
caron	0xFF	0xCF	0xCF	0xFF

MODIFICATION HISTORY

$Log: MacToNeXTText.rtf,v $Revision 1.2 93/04/04 23:44:39 deathSun Apr 4

23:44:39 PDT 1993Revision 1.1 93/01/10 15:08:05 deathSun Jan 10 15:08:05 PST 1993