PRESSFROMNS

By: Tad Hogg (Hogg.pa@Xerox.com)

INTRODUCTION

This module is a patch to allow Press printers to print NS characters by translating them to appropriate Press fonts. Before loading this file, make sure there are no open press streams (i.e. no hardcopy in progress to a Press printer).

CONTROLLING CHARACTER SET TRANSLATIONS

The translations are controlled by a number of variables and functions described below. These variables can be modified to provide additional or different translations.

Global character set translations

NSTOASCIITRANSLATIONS [Variable]

an ASSOC list whose elements have the form (charset translationArrayName). This specifies which translation array is to be used when translating the specified character set.

Example: `((0 ASCIIFROM0ARRAY) (38 ASCIIFROM38ARRAY))` specifies that ASCIIFROM38ARRAY is bound to the array to be used for translating charset 38.

The translationArrayName is bound to an array whose index ranges from 0 to 255. Each element of the array specifies the translation to use for the corresponding charcode in this charset.

Translations are of one of the following forms:

1. NIL -- no translation specified which will use the font as is for charset 0 and otherwise print a black box to indicate the NS character could not be translated

2. an integer in the range 0 to 255 which indicates that this value is to be used as the translated charcode, but that no font translation is required [This is mainly useful for converting NS to ASCII in charset 0.]

3. a two element translation list of the form (fontFamily charcode) which indicates that this character should be translated to charcode in a font whose family is fontFamily. Note that charcode should be in the range 0 to 255. fontFamily can also be a font descriptor or a font specification list (e.g. (Gacha 10)) of a form acceptable to FONTCREATE.

PRESSFONTFAMILIES [Variable]

a list whose elements are of the form (FAMILY . specialTranslations). The optional list specialTranslations specifies translations to use for Press fonts in charset 0. Each element is (charcode translation) which specifies that translation is to be used for charcode in this family.

Example: `((GACHA (50 (HELVETICA 40)) (51 (TIMESROMAN 45))) (TIMESROMAN) (SYMBOL))`

Note: these translations are cached in the font descriptor when PRESS fonts are created so any changes to these variables will not change the translations in previously created fonts.

Translations in individual fonts

The following two functions provided detailed control over the translations used in individual fonts.

(GETCHARPRESSTRANSLATION *CHARCODE FONT*) [Function]

returns the translation (a two element list) used for CHARCODE in FONT

(PUTCHARPRESSTRANSLATION *CHARCODE FONT NEWTRANSLATION*) [Function]

sets the translation to be used for CHARCODE in FONT. NEWTRANSLATION should be a translation in one of the forms described above.

Additional functions

(PRESS.NSARRAY *CHARSET FAMILY ASCIIARRAY*) [Function]

This function returns a suitable translation array built as the inverse of a translation array from Press to NS characters. Such arrays are used to print Press fonts on Interpress printers and are listed in the variable ASCIITONSTRANSLATIONS. CHARSET is the character set for which to create a translation. FAMILY is the press font family for which ASCIIARRAY is the translation to NS characters. If ASCIIARRAY is not specified, the function looks through all arrays included on ASCIITONSTRANSLATIONS to fill in the translation array.he following two functions provided detailed control over the translations used in individual fonts.

CONTROLLING PRESS FONT COERCIONS

There is also a mechanism for determining which press font is actually used for the translation. For example, an NS character in the Modern 8 font might translate to an ASCII character in Symbol 8. If this font does not exist on the printer, a (generally incorrect) font change will be done by the printer. The following procedure changes the actual font used, e.g. to Symbol 10 in this example.

The coercion is controlled by a coercion list which is an alist indexed by device. The font coercion scans down the element on the list for the requested device (e.g. PRESS) looking for the first entry that matches the user request. If a match is found, then the entry tells how to construct an appropriate new name from the requested specification. Fields of the newname not specified in the entry are simply copied over.

FONTCOERCIONS [Variable]

This list allows the user to coerce fonts that he knows don't exist on the printer even tho the fonts-widths files doesn't indicate that (e.g. the desired size doesn't exist). FONTCOERCIONS is initialized simply to take all SYMBOL fonts of size less than 10 into 10, and size greater than 12 into 12.

MISSINGFONTCOERCIONS [Variable]

If the initial coerced (or uncoerced) lookup fails, then MISSINGFONTCOERCIONS is used. This takes MODERN into HELVETICA etc--the standard press coercions.

The procedure for determining whether a user request matches a coercion entry is straightforward. If the match-part of the coercion entry is an atomic family name, it matches if it is eq to the requested family. Otherwise, the match-part must be a list of family, size, face in standard fontname order. If a component is NIL or missing, then it is assumed to match. The only funniness is in size matching. The size component can be NIL (matches anything), a particular size (EQ matches), or a list of the form (<

n) or (> n) where n is a size number. The first matches any requested size less than n, and the second matches any requested size greater than n.

FONTCOERCIONS for PRESS starts out as

```
((SYMBOL (< 10) ) (SYMBOL 10)) (SYMBOL (> 12))(SYMBOL 12))
```

MISSINGFONTCOERCIONS for PRESS starts out as

```
((MODERN HELVETICA)(CLASSIC TIMESROMAN) etc.)
```