

3.3 Release Notes: C Library Support for Localization of Date, Time, and Currency

This document assumes you are familiar with the ANSI C draft standard discussion of routines **setlocale()**, **localeconv()**, and **strftime()** and the defaults database.

You can customize the date, time, and currency information returned by **strftime()** and **localeconv()** using the defaults mechanism. You must first set the locale to "", the default locale, using **setlocale()**. This is automatically done for applications by the Application Kit at startup time. You must also enable the use of these defaults by setting the value of NXUseDateTimeCurrencyDefaults:

```
dwrite GLOBAL NXUseDateTimeCurrencyDefaults YES
```

(Note: Setting this default causes **strftime()** and **localeconv()** to consult the

defaults database for date and time customizations rather than use the values supplied in the **international.strings** files in the language projects of **/NextApps/Preferences**. For currency defaults, these functions always consult the database without regard to the setting of NXUseDateTimeCurrencyDefaults.)

These steps having been taken, subsequent calls to **strftime()** and **localeconv()** access information from the defaults data base. The defaults for **strftime()** are equivalent to:

Short days of the week:

```
dwrite GLOBAL NXShortDays "Sun Mon Tue Wed Thu Fri Sat"
```

Long days of the week:

```
dwrite GLOBAL NXLongDays "Sunday Monday Tuesday Wednesday  
Thursday Friday Saturday"
```

Short months:

```
dwrite GLOBAL NXShortMonths "Jan Feb Mar Apr May Jun Jul Aug Sep  
Oct Nov Dec"
```

Long months:

```
dwrite GLOBAL NXLongMonths "January February March April May  
June July August September October November December"
```

AM and PM indication:

```
dwrite GLOBAL NXAmPm "AM PM"
```

Default date and time representation format string for **strftime()**:

```
dwrite GLOBAL NXDateAndTime "%a %b %d %H:%M:%S %Z %Y"
```

Default date representation format string for **strftime()**:

```
dwrite GLOBAL NXDate "%a %b %d %Y"
```

Default time representation format string for **strftime()**:

```
dwrite GLOBAL NXTime "%H:%M:%S %Z"
```

You can modify defaults to reflect the local spelling and preferences of these items.

Number and currency formats are returned in the **lconv** structure (declared in **locale.h**):

```
struct lconv {
    char *decimal_point;
    char *thousands_sep;
    char *grouping;
    char *int_curr_symbol;
    char *currency_symbol;
    char *mon_decimal_point;
    char *mon_thousands_sep;
    char *mon_grouping;
    char *positive_sign;
    char *negative_sign;
    char int_frac_digits;
    char frac_digits;
    char p_cs_precedes;
    char p_sep_by_space;
    char n_cs_precedes;
    char n_sep_by_space;
    char p_sign_posn;
    char n_sign_posn;
};
```

The values are taken from the defaults data base as follows:

dwrite GLOBAL NXNumberGroup '". " " 0'

Stores these strings:

```
char *decimal_point;
```

```
char *thousands_sep;  
char *grouping;
```

```
dwrite GLOBAL NXCurrencyGroup '"" "" 0'
```

Stores these strings:

```
char *mon_decimal_point;  
char *mon_thousands_sep;  
char *mon_grouping;
```

```
dwrite GLOBAL NXCurrencyFormat '"" "" "" "" 255 255 255 255 255 255  
255 255'
```

Stores these strings and integers:

```
char *int_curr_symbol;  
char *currency_symbol;  
char *positive_sign;  
char *negative_sign;  
char int_frac_digits;  
char frac_digits;  
char p_cs_precedes;  
char p_sep_by_space;  
char n_cs_precedes;  
char n_sep_by_space;  
char p_sign_posn;  
char n_sign_posn;
```

Note that multiple strings within a string (e.g., "" "" 0) are separated using spaces. You can use quotes to specify empty string or spaces. You can use \ to escape quotes. The last number in the example "" "" 0 stores an empty string

terminated by a null. You can use additional numbers to specify arbitrary 8-bit unsigned integers with are stored in the string representing grouping. The single quotes enclosing the strings in the **dwwrite** commands above are required so that the strings are correctly parsed by the shell.