Internationalization and CDE

B

This is a guide for software developers writing internationalized Solaris CDE applications. It presents the differences between OpenWindows and CDE internationalization. Any steps that must be performed to internationalize a Solaris CDE application that are not mentioned in this appendix remain unchanged from the OpenWindows guidelines.

This appendix assumes familiarity with Xt and Motif programming.

Ensure Correct CDE NLS Environment	*68*
Message Catalog Functions	*68*
Locale Announcement	*68*
Character Strings and XmStrings	*68*
Include app-defaults File	*69*
Localize Motif Resources	*69*
Deliver Message Catalog	*70*
Fonts	*70*

B

Ensure Correct CDE NLS Environment

You must make sure that the `NLSPATH` environment variable is set properly. This ensures that your application will find message catalogs.

If your application calls `DtInitialize()` before `catopen()`, then `NLSPATH` is set. Otherwise, make sure that it understands the `NLSPATH` variable. For example, you can set it explicitly in your code.

Message Catalog Functions

Solaris CDE applications must use the `catopen()`, `catclose()`, and `catgets()` family of messaging functions (instead of `gettext()` and so on). See the man pages for information on how to use these functions.

Also, if the system on which you are writing your application does not define `NL_CAT_LOCALE` in a header file, you must define it in your application. This ensures that the application is portable to other UNIX XPG4-compliant platforms.

Locale Announcement

So that your application knows what locale it is running in, use:

- `XtSetLanguageProc()`—If your application is Motif/Xt GUI-based
- `setlocale()`—If your application is not GUI-based

Character Strings and XmStrings

This section describes how to convert between character strings and `XmStrings`.

To Convert from Character String to XmString

In Motif, you must make an explicit call to convert a string to `XmString`, an internal representation of the string. If you internationalize your application, use `XmStringCreateLocalized()` to perform this conversion.

In CDE Motif—unlike in Motif 1.2.3—`XmStringCreateLocalized()` recognizes `\n` as a line separator, so a string can have multiple lines.

To Convert from XmString to Character String

To convert from XmString to a character string, you must traverse the compound string and retrieve each segment individually. See section 19.3.3 of the *Motif Programming Manual*, Volume 6A, which contains a code sample that performs this conversion.

Do not use `XmStringGetLtoR()` to convert from XmString to a character string. This function assumes the text is oriented left-to-right.

Include app-defaults File

If your application uses resources, make sure to include an `app-defaults` file. Refer to the *X Toolkit Intrinsics Programming Manual* (Volume 4, Section 2.3.3) for details on setting up an `app-defaults` file.

Localize Motif Resources

Localize any Motif resource that represents something that must be customized for a particular locale by putting the resource in your `app-defaults` file. These include, but are not limited to:

- XmNmnemonic
- XmNfontList
- XmNcolumns
- XmNinputMethod
- XmNpreeditType

B

Deliver Message Catalog

Every SVR4 UNIX platform contains the `gencat` utility. Run `gencat` on the translated message catalog file to produce a binary form of the message catalog. The resulting file is a formatted message database.

CDE and gencat

Each platform has its own implementation of `gencat`. Follow these rules for message catalogs to ensure that each message catalog's format does not cause different `gencat` utilities to break:

1. Use a `$quote` directive if you have trailing spaces (see the `gencat` man page).
2. Do not put more than one space between the message id and the message string.
3. Insert a space between `$` and the comments.
4. Message IDs need to be sorted in ascending order.

.msg Files

A `.msg` file is a message catalog containing translatable text that appears in your application code. It is the file that you give to a translator for translation. See the `gencat` man page for its format. Use `gencat` to turn a translated message catalog to a message database.

.cat Files

A .cat file is a message catalog that results when you run `gencat` on a `.msg` file. This is a binary file.

Fonts

In the OpenWindows environment, font alias names differ depending on the application locale. In Solaris CDE, the font alias names are independent of locale. Refer to *CDE Internationalization Programmer's Guide* or *CDE Programmer's Guide* for information on CDE font names.