

**NAME**

XCreateFontSet, XFreeFontSet - create and free an international text drawing font set

**SYNTAX**

```
XFontSet XCreateFontSet(Display *display, _Xconst char *base_font_name_list, char  
***missing_charset_list_return, int *missing_charset_count_return, char **def_string_return);
```

```
void XFreeFontSet(Display *display, XFontSet font_set);
```

**ARGUMENTS**

*display* Specifies the connection to the X server.

*base\_font\_name\_list*  
Specifies the base font names.

*def\_string\_return*  
Returns the string drawn for missing charsets.

*font\_set* Specifies the font set.

*missing\_charset\_count\_return*  
Returns the number of missing charsets.

*missing\_charset\_list\_return*  
Returns the missing charsets.

**DESCRIPTION**

The **XCreateFontSet** function creates a font set for the specified display. The font set is bound to the current locale when **XCreateFontSet** is called. The font set may be used in subsequent calls to obtain font and character information and to image text in the locale of the font set.

The *base\_font\_name\_list* argument is a list of base font names that Xlib uses to load the fonts needed for the locale. The base font names are a comma-separated list. The string is null-terminated and is assumed to be in the Host Portable Character Encoding; otherwise, the result is implementation-dependent. White space immediately on either side of a separating comma is ignored.

Use of XLFD font names permits Xlib to obtain the fonts needed for a variety of locales from a single locale-independent base font name. The single base font name should name a family of fonts whose members are encoded in the various charsets needed by the locales of interest.

An XLFD base font name can explicitly name a charset needed for the locale. This allows the user to specify an exact font for use with a charset required by a locale, fully controlling the font selection.

If a base font name is not an XLFD name, Xlib will attempt to obtain an XLFD name from the font properties for the font. If this action is successful in obtaining an XLFD name, the **XBaseFontNameListOfFontSet** function will return this XLFD name instead of the client-supplied name.

Xlib uses the following algorithm to select the fonts that will be used to display text with the **XFontSet**.

For each font charset required by the locale, the base font name list is searched for the first appearance of one of the following cases that names a set of fonts that exist at the server:

- ⊕ The first XLFD-conforming base font name that specifies the required charset or a superset of the required charset in its **CharSetRegistry** and **CharSetEncoding** fields. The implementation may use a base font name whose specified charset is a superset of the required charset, for example, an ISO8859-1 font for an ASCII charset.
- ⊕ The first set of one or more XLFD-conforming base font names that specify one or more charsets that can be remapped to support the required charset. The Xlib implementation may recognize various mappings from a required charset to one or more other charsets and use the fonts for those charsets. For example, JIS Roman is ASCII with tilde and backslash replaced by yen and overbar; Xlib may load an ISO8859-1 font to support this character set if a JIS Roman font is not available.
- ⊕ The first XLFD-conforming font name or the first non-XLFD font name for which an XLFD font name can be obtained, combined with the required charset (replacing the **CharSetRegistry** and **CharSetEncoding** fields in the XLFD font name). As in case 1, the implementation may use a charset that is a superset of the required charset.
- ⊕ The first font name that can be mapped in some implementation-dependent manner to one or more fonts that support imaging text in the charset.

For example, assume that a locale required the charsets:

ISO8859-1  
JISX0208.1983  
JISX0201.1976  
GB2312-1980.0

The user could supply a `base_font_name_list` that explicitly specifies the charsets, ensuring that specific fonts are used if they exist. For example:

```
"-JIS-Fixed-Medium-R-Normal--26-180-100-100-C-240-JISX0208.1983-0,\
-JIS-Fixed-Medium-R-Normal--26-180-100-100-C-120-JISX0201.1976-0,\
-GB-Fixed-Medium-R-Normal--26-180-100-100-C-240-GB2312-1980.0,\
-Adobe-Courier-Bold-R-Normal--25-180-75-75-M-150-ISO8859-1"
```

Alternatively, the user could supply a `base_font_name_list` that omits the charsets, letting Xlib select font charsets required for the locale. For example:

```
"-JIS-Fixed-Medium-R-Normal--26-180-100-100-C-240,\
-JIS-Fixed-Medium-R-Normal--26-180-100-100-C-120,\
-GB-Fixed-Medium-R-Normal--26-180-100-100-C-240,\
-Adobe-Courier-Bold-R-Normal--25-180-100-100-M-150"
```

Alternatively, the user could simply supply a single base font name that allows Xlib to select from all available fonts that meet certain minimum XLFD property requirements. For example:

```
"-*-*-*R-Normal--*-180-100-100-*-*"
```

If **XCreateFontSet** is unable to create the font set, either because there is insufficient memory or because the current locale is not supported, **XCreateFontSet** returns NULL, `missing_charset_list_return` is set to NULL, and `missing_charset_count_return` is set to zero. If fonts exist for all of the charsets required by the current locale, **XCreateFontSet** returns a valid **XFontSet**, `missing_charset_list_return` is set to NULL, and `missing_charset_count_return` is set to zero.

If no font exists for one or more of the required charsets, **XCreateFontSet** sets `missing_charset_list_return` to a list of one or more null-terminated charset names for which no font exists and sets `missing_charset_count_return` to the number of missing fonts. The charsets are from the list of the required charsets for the encoding of the locale and do not include any charsets to which Xlib may be able to remap a required charset.

If no font exists for any of the required charsets or if the locale definition in Xlib requires that a font exist for a particular charset and a font is not found for that charset, **XCreateFontSet** returns NULL. Otherwise, **XCreateFontSet** returns a valid **XFontSet** to `font_set`.

When an Xmb/wc drawing or measuring function is called with an **XFontSet** that has missing charsets, some characters in the locale will not be drawable. If `def_string_return` is non-NULL, **XCreateFontSet** returns a pointer to a string that represents the glyphs that are drawn with this **XFontSet** when the

charsets of the available fonts do not include all font glyphs required to draw a codepoint. The string does not necessarily consist of valid characters in the current locale and is not necessarily drawn with the fonts loaded for the font set, but the client can draw and measure the default glyphs by including this string in a string being drawn or measured with the **XFontSet**.

If the string returned to `def_string_return` is the empty string (""), no glyphs are drawn, and the escapement is zero. The returned string is null-terminated. It is owned by Xlib and should not be modified or freed by the client. It will be freed by a call to **XFreeFontSet** with the associated **XFontSet**. Until freed, its contents will not be modified by Xlib.

The client is responsible for constructing an error message from the missing charset and default string information and may choose to continue operation in the case that some fonts did not exist.

The returned **XFontSet** and missing charset list should be freed with **XFreeFontSet** and **XFreeStringList**, respectively. The client-supplied `base_font_name_list` may be freed by the client after calling **XCreateFontSet**.

The **XFreeFontSet** function frees the specified font set. The associated base font name list, font name list, **XFontStruct** list, and **XFontSetExtents**, if any, are freed.

## SEE ALSO

`XExtentsofFontSet(3)`, `XFontsOfFontSet(3)`, `XFontSetExtents(3)`

*Xlib - C Language X Interface*