Standard Type Services Reference Manual

1.0

Generated by Doxygen 1.2.15

Wed Mar 12 10:05:28 2003

Contents

1 Standard Type Services Module Index

1.1 Standard Type Services Modules

Here is a list of all modules:

2 Standard Type Services Compound Index

2.1 Standard Type Services Compound List

Here are the classes, structs, unions and interfaces with brief descriptions:

3 Standard Type Services File Index

3.1 Standard Type Services File List

Here is a list of all documented files with brief descriptions:

4 Standard Type Services Module Documentation

4.1 XSTTypeEnv functions

Functions

- XSTTypeEnv XSTTypeEnvNew (Display *dpy)

  Create a default STTypeEnv and return an XID of type XSTTypeEnv.

- XSTTypeEnv XSTTypeEnvNewCopy (Display *dpy, XSTTypeEnv iEnv)

  Copy an existing STTypeEnv and return the new XID of the copied STTypeEnv.

- void XSTTypeEnvDispose (Display *dpy, XSTTypeEnv iEnv)

  Call the STTypeEnv destructor.

- XSTFontFallbackPolicy XSTTypeEnvGetFontFallbackPolicy (Display *dpy, XSTTypeEnv iEnv)

  Returns the global font fallback policy.

- void XSTTypeEnvSetFontFallbackPolicy (Display *dpy, XSTTypeEnv iEnv, XSTFontFallbackPolicy iPolicy)

  Sets the global font fallback policy.

- XSTFont * XSTTypeEnvGetFontFallbacks (Display *dpy, XSTTypeEnv iEnv, int *oFontCount)

  Returns the global font fallback array.

- void XSTTypeEnvSetFontFallbacks (Display *dpy, XSTTypeEnv iEnv, int iFontCount, XSTFont *iFontArray)

  Sets the global font fallback array.

- XSTFont XSTTypeEnvFindFont (Display *dpy, XSTTypeEnv iEnv, XSTString *iName, int iNameLength, char *iLocale, uint16_t iNameID)

  Finds a font by its name.

- XSTFont * XSTTypeEnvFindAllFonts (Display *dpy, XSTTypeEnv iEnv, XSTString *iName, int iNameLength, char *iLocale, uint16_t iNameID, int *oFontCount)

  Finds all fonts by their Unicode name and returns an array of font IDs.

- XSTFont XSTTypeEnvFindFontByPlatformName (Display *dpy, XSTTypeEnv iEnv, byte *iName, int iNameLength, uint16_t iPlatformID, uint16_t iEncodingID, uint16_t iLanguageID, uint16_t iNameID)

  Finds a font by its platform-specific name and returns the font ID.

- XSTFont * XSTTypeEnvFindAllFontsByPlatformName (Display *dpy, XSTTypeEnv iEnv, byte *iName, int iNameLength, uint16_t iPlatformID, uint16_t iEncodingID, uint16_t iLanguageID, uint16_t iNameID, int *oFontCount)

  Finds all fonts by their Platform name and returns an array of font IDs.

- XSTFontFamily XSTTypeEnvFindFontFamily (Display *dpy, XSTTypeEnv iEnv, XSTString *iName, int iNameLength, char *iLocale, XSTLocale *ioLocale)

  Finds a font family by its Unicode name and returns the first matching font family ID.

- XSTFontFamily * XSTTypeEnvFindAllFontFamilies (Display *dpy, XSTTypeEnv iEnv, int *oFontFamilyCount)

  Finds all font families available to the Type Environment and returns an array of font family IDs.

- XSTFont * XSTTypeEnvCreateFont (Display *dpy, XSTTypeEnv iEnv, int iDataCount, byte **iData, int *iDataLength, int *oFontCount)

Creates one or more new fonts from the binary data supplied by the client, and returns an array of font IDs.

- void XSTTypeEnvDestroyFont (Display *dpy, XSTTypeEnv iEnv, XSTFont iFont)

  Disposes of a font previously created with XSTCreateFont().

- XSTScaler * XSTTypeEnvFindAllScalers (Display *dpy, XSTTypeEnv iEnv, int *oScalerCount)

  Returns an array of all font scalers available to this type environment.

- XSTScaler XSTTypeEnvFindScaler (Display *dpy, XSTTypeEnv iEnv, XSTTag iScalerTag)

  Returns a scaler ID based on its tag.

- XSTLocale XSTTypeEnvGetLocale (XSTTypeEnv iEnv, char *iLocale)

  Returns a ST_LANGUAGE_* ST_LOCALE_* pair based on the iLocale value.

- char ** XSTTypeEnvGetFontFolders (Display *dpy, XSTTypeEnv iEnv, int *oFolderCount)

  Returns an array of folder names (path names) indicating the user defined folders to search for fonts.

- void XSTTypeEnvSetFontFolders (Display *dpy, XSTTypeEnv iEnv, int iFolderCount, char **iFolders)

  Sets the array of folder names (path names) to search for fonts.

- void XSTTypeEnvSetLocations (Display *dpy, XSTTypeEnv iEnv, XSTFontLocationMask iMask)

  Sets the location mask indicating which areas to search for fonts.

- XSTFontLocationMask XSTTypeEnvGetLocations (Display *dpy, XSTTypeEnv iEnv)

  Returns the location mask indicating which areas to search for fonts.

- XSTLayoutEngine * XSTTypeEnvFindAllLayoutEngines (Display *dpy, XSTTypeEnv iEnv, int *oLECount)

  Returns an array of all layout engines available to this type environment.

- XSTLayoutEngine XSTTypeEnvFindLayoutEngine (Display *dpy, XSTTypeEnv iEnv, XSTTag iLETag)

  Returns a scaler ID based on its tag.

- XSTFont * XSTTypeEnvCreateFontFromURL (Display *dpy, XSTTypeEnv iEnv, int iURLCount, char **iURL, int *oFontCount)

  Returns the set of font ids associated with a set of URLs.

- XSTFont * XSTTypeEnvFindFontByURL (Display *dpy, XSTTypeEnv iEnv, char *iURL, int *oFontCount)

  Returns the set of font ids associated with a set of URLs.

4.1.1 Function Documentation

4.1.1.1 XSTFont* XSTTypeEnvCreateFont (Display * *dpy*, XSTTypeEnv *iEnv*, int *iDataCount*, byte ** *iData*, int * *iDataLength*, int * *oFontCount*)

Creates one or more new fonts from the binary data supplied by the client, and returns an array of font IDs.

Some font formats - like TTC - allow several logical fonts within one binary font file, but most other font formats contain one logical font per binary file. These newly created fonts are not searchable unless they are permanently installed with XSTInstallFont(). This function allocates memory for the returned array of font IDs. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iDataCount number of binary data blocks

iData array of binary data blocks

iDataLength array of binary data block lengths

oFontCount number of font IDs returned

Returns:

XSTFont ID array

4.1.1.2 XSTFont* XSTTypeEnvCreateFontFromURL (Display * *dpy*, XSTTypeEnv *iEnv*, int *iURLCount*, char ** *iURL*, int * *oFontCount*)

Returns the set of font ids associated with a set of URLs.

Parameters:

dpy X display pointer

iEnv TypeEnv XID (this is added to protocol message)

iURLCount number of URL strings in iURL

iURL array of URL strings

oFontCount number of XSTFont ids returned

Returns:

an array of XSTFont

4.1.1.3 void XSTTypeEnvDestroyFont (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFont *iFont*)

Disposes of a font previously created with XSTCreateFont().

This function will return an error code if an attempt is made to destroy a font that was not previously created with XSTCreateFont(). All fonts that have not been explicitly destroyed will be destroyed when the Type Environment object from which they were created is destroyed.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iFont font ID of a previously created font

Returns:

none

4.1.1.4 void XSTTypeEnvDispose (Display * *dpy*, XSTTypeEnv *iEnv*)

Call the STTypeEnv destructor.

Free the XSTTypeEnv XID.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

Returns:

none

4.1.1.5 XSTFontFamily* XSTTypeEnvFindAllFontFamilies (Display * *dpy*, XSTTypeEnv *iEnv*, int * *oFontFamilyCount*)

Finds all font families available to the Type Environment and returns an array of font family IDs.

This function allocates memory for the returned array of font family IDs. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

oFontFamilyCount number of font family IDs returned

Returns:

XSTFontFamily ID array

4.1.1.6 XSTFont* XSTTypeEnvFindAllFonts (Display * *dpy*, XSTTypeEnv *iEnv*, XSTString * *iName*, int *iNameLength*, char * *iLocale*, uint16_t *iNameID*, int * *oFontCount*)

Finds all fonts by their Unicode name and returns an array of font IDs.

See XSTTypeEnvFindFont() for more information. This function allocates memory for the array of font IDs returned. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iName a native encoded string that specifies the name

iNameLength number of characters in the name

iLocale locale string

iNameID meaning of the name

oFontCount number of XSTFont IDs returned

Returns:

XSTFont ID array

4.1.1.7 XSTFont* XSTTypeEnvFindAllFontsByPlatformName (Display * *dpy*, XSTTypeEnv *iEnv*, byte * *iName*, int *iNameLength*, uint16_t *iPlatformID*, uint16_t *iEncodingID*, uint16_t *iLanguageID*, uint16_t *iNameID*, int * *oFontCount*)

Finds all fonts by their Platform name and returns an array of font IDs.

See XSTTypeEnvFindFontByPlatformName() for more information. This function allocates memory for the returned array of font IDs. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iName a byte encoded string that specifies the name

iNameLength number of characters in the name

iPlatformID platform ID

iEncodingID platform-specific encoding ID

iLanguageID language ID

iNameID meaning of the name

oFontCount number of font IDs returned

Returns:

XSTFont ID array

4.1.1.8 XSTLayoutEngine* XSTTypeEnvFindAllLayoutEngines (Display * *dpy*, XSTTypeEnv *iEnv*, int * *oLECount*)

Returns an array of all layout engines available to this type environment.

This function allocates memory for the returned array of layout engine IDs. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

oLECount number of layout engines returned

Returns:

XSTLayoutEngine ID array

4.1.1.9 XSTScaler* XSTTypeEnvFindAllScalers (Display * *dpy*, XSTTypeEnv *iEnv*, int * *oScalerCount*)

Returns an array of all font scalers available to this type environment.

This function allocates memory for the returned array of scaler IDs. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

oScalerCount number of scalers returned

Returns:

XSTScaler ID array

4.1.1.10 XSTFont XSTTypeEnvFindFont (Display * *dpy*, XSTTypeEnv *iEnv*, XSTString * *iName*, int *iNameLength*, char * *iLocale*, uint16_t *iNameID*)

Finds a font by its name.

If the font name (iName) is NULL, it returns the first available font. The iNameID parameter indicates the name ID of the name, as defined in TrueType/OpenType specs. A special value of iNameID (ST_NAME_ANY) indicates that ST will search for the specified font name among all names. Standard name ID values are defined in sttags.h header file. If the iLocale parameter is NULL, then it uses the value retrieved from setlocale(). An example of a proper iLocale string would be: "en_US.UTF-8".

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iName a native encoded string that specifies the name

iNameLength number of characters in the name

iLocale locale string

iNameID meaning of the name

Returns:

XSTFont ID

4.1.1.11 XSTFont XSTTypeEnvFindFontByPlatformName (Display * *dpy*, XSTTypeEnv *iEnv*, byte * *iName*, int *iNameLength*, uint16_t *iPlatformID*, uint16_t *iEncodingID*, uint16_t *iLanguageID*, uint16_t *iNameID*)

Finds a font by its platform-specific name and returns the font ID.

Platform-specific names are defined in TrueType/OpenType specs. STSF synthesizes TrueType-compliant names for non-TrueType fonts. iPlatformID, iEncodingID, iLanguageID, and iNameID have special values of ST_PLATFORM_ANY, ST_ENCODING_ANY, ST_LANGUAGE_ANY, and ST_NAME_ANY that indicate that their values do not matter. If the font name is NULL, the first font that satisfies the search criterion is returned.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iName a byte encoded string that specifies the name

iNameLength number of characters in the name

iPlatformID platform ID

iEncodingID platform-specific encoding ID

iLanguageID language ID

iNameID meaning of the name

Returns:

XSTFont ID

4.1.1.12 XSTFont* XSTTypeEnvFindFontByURL (Display * *dpy*, XSTTypeEnv *iEnv*, char * *iURL*, int * *oFontCount*)

Returns the set of font ids associated with a set of URLs.

Parameters:

dpy X display pointer

iEnv TypeEnv XID (this is added to protocol message)

iURL URL string

oFontCount number of XSTFont ids returned

Returns:

an array of XSTFont

4.1.1.13 XSTFontFamily XSTTypeEnvFindFontFamily (Display * *dpy*, XSTTypeEnv *iEnv*, XSTString * *iName*, int *iNameLength*, char * *iLocale*, XSTLocale * *ioLocale*)

Finds a font family by its Unicode name and returns the first matching font family ID.

If the font famliy name (iName) is NULL, it returns the first available font family. If the iLocale parameter is NULL, then it uses the value retrieved from setlocale(). An example of a proper iLocale string would be: "en_US.UTF-8". The language ID is the requested language support of the font family. If the requested language for the font is not available, then the language of the family name returned will be returned.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iName a byte encoded string that specifies the name

iNameLength number of characters in the name

iLocale locale string

ioLocale number of font IDs returned

Returns:

XSTFontFamily ID

4.1.1.14 XSTLayoutEngine XSTTypeEnvFindLayoutEngine (Display * *dpy*, XSTTypeEnv *iEnv*, XSTTag *iLETag*)

Returns a scaler ID based on its tag.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iLETag a tag assigned to the layout engine.

Returns:

XSTLayoutEngine ID

4.1.1.15 XSTScaler XSTTypeEnvFindScaler (Display * *dpy*, XSTTypeEnv *iEnv*, XSTTag *iScalerTag*)

Returns a scaler ID based on its tag.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iScalerTag a tag assigned to the scaler. Known tags: iFontFusionScaler - 'FFUS' - FontFusion scaler iFreeType1Scaler - 'FTY1' - FreeType v. 1 scaler iFreeType2Scaler - 'FTY2' - FreeType v. 2 scaler

Returns:

XSTScaler ID

4.1.1.16 XSTFontFallbackPolicy XSTTypeEnvGetFontFallbackPolicy (Display * *dpy*, XSTTypeEnv *iEnv*)

Returns the global font fallback policy.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

Returns:

XSTFontFallbackPolicy value

4.1.1.17 XSTFont* XSTTypeEnvGetFontFallbacks (Display * *dpy*, XSTTypeEnv *iEnv*, int * *oFontCount*)

Returns the global font fallback array.

This function allocates memory for the array and it is the responsiblity of the caller to free the memory when it is no longer used.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

oFontCount number of fonts in array

Returns:

XSTFont ID array

4.1.1.18 char** XSTTypeEnvGetFontFolders (Display * *dpy*, XSTTypeEnv *iEnv*, int * *oFolderCount*)

Returns an array of folder names (path names) indicating the user defined folders to search for fonts.

This function allocates memory for the returned array of strings. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

oFolderCount number of font folders returned

Returns:

array of strings

4.1.1.19 XSTLocale XSTTypeEnvGetLocale (XSTTypeEnv *iEnv*, char * *iLocale*)

Returns a ST_LANGUAGE_* ST_LOCALE_* pair based on the iLocale value.

If the iLocale parameter is NULL, then it uses the value retrieved from setlocale(). An example of a proper iLocale string would be: "en_US.UTF-8".

Parameters:

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iLocale locale string

Returns:

XSTLocale

4.1.1.20 XSTFontLocationMask XSTTypeEnvGetLocations (Display * *dpy*, XSTTypeEnv *iEnv*)

Returns the location mask indicating which areas to search for fonts.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

Returns:

XSTFontLocationMask bitmask

4.1.1.21 XSTTypeEnv XSTTypeEnvNew (Display * *dpy*)

Create a default STTypeEnv and return an XID of type XSTTypeEnv.

Parameters:
dpy X Display value

Returns:
XSTTypeEnv XID

4.1.1.22 XSTTypeEnv XSTTypeEnvNewCopy (Display * *dpy*, XSTTypeEnv *iEnv*)

Copy an existing STTypeEnv and return the new XID of the copied STTypeEnv.

Parameters:
dpy X Display value

iEnv original XSTTypeEnv XID of the STTypeEnv object to copy

Returns:
XSTTypeEnv XID

4.1.1.23 void XSTTypeEnvSetFontFallbackPolicy (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFontFallbackPolicy *iPolicy*)

Sets the global font fallback policy.

All objects that use this STTypeEnv object will inherit this font fallback policy.

Parameters:
dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iPolicy a value that indicates the font fallback policy

Returns:
none

4.1.1.24 void XSTTypeEnvSetFontFallbacks (Display * *dpy*, XSTTypeEnv *iEnv*, int *iFontCount*, XSTFont * *iFontArray*)

Sets the global font fallback array.

This default setting can be overridden for the text objects by calling XSTTextSetFontFallbacks(). When some glyphs are missing from fonts set by style objects, font fallbacks list will be searched for replacement fonts.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iFontCount number of fonts in array

iFontArray array of XSTFont IDs.

Returns:

none

4.1.1.25 void XSTTypeEnvSetFontFolders (Display * *dpy*, XSTTypeEnv *iEnv*, int *iFolderCount*, char ** *iFolders*)

Sets the array of folder names (path names) to search for fonts.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iFolderCount number of font folders

iFolders array of font folders

Returns:

none

4.1.1.26 void XSTTypeEnvSetLocations (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFontLocationMask *iMask*)

Sets the location mask indicating which areas to search for fonts.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iMask bitmask indicating which areas to search for fonts

Returns:
none

4.2 Enumerated types

Typedefs

- typedef enum __XSTUnicodeEncoding XSTUnicodeEncoding

  This enum includes all of the types of Unicode encoding formats that can be output from the following functions: XSTFontGetUnicodeName XSTFontGetUnicodeTypefaceName XSTFontFamilyGetUnicodeName.

- typedef enum __XSTOutputType XSTOutputType

  This enum includes the types of output that can be used by ST.

- typedef enum __XSTEncoding **XSTEncoding**

Enumerations

- enum __XSTUnicodeEncoding { fUTF7 = 0, fUTF8, fUCS2, fUTF16, fUCS4, fUTF32 }

  This enum includes all of the types of Unicode encoding formats that can be output from the following functions: XSTFontGetUnicodeName XSTFontGetUnicodeTypefaceName XSTFontFamilyGetUnicodeName.

- enum __XSTOutputType { **fRasterOutput** = 1, **fVectorOutput** = 2, **fOutputTypeNotSet** = -1 }

  This enum includes the types of output that can be used by ST.

- enum **__XSTEncoding** { **Default_Encoding** = 0, **DEFAULT_ENCODING** = Default_Encoding, **UTF7** = 1, **UTF8**, **UTF16**, **UTF32**, **UCS2**, **UCS4**, **Glyph8**, **Glyph16**, **Glyph32**, **ISO8859_1**, **ISO8859_2**, **ISO8859_3**, **ISO8859_4**, **ISO8859_5**, **ISO8859_6**, **ISO8859_7**, **ISO8859_8**, **ISO8859_9**, **ISO8859_10**, **ISO8859_11**, **ISO8859_12**, **ISO8859_13**, **ISO8859_14**, **ISO8859_15**, **ISO8859_16**, **ISO8859_17**, **ISO8859_18**, **ISO8859_19**, **ISO8859_20**, **ISO8859_21**, **ISO8859_22**, **ISO8859_23**, **ISO8859_24**, **ISO8859_25**, **ISO8859_26**, **ISO8859_27**, **ISO8859_28**, **ISO8859_29**, **ISO8859_30**, **US_ASCII**, **EUC**, **eucJP**, **PCK**, **ANSI1251**, **KOI8_R**, **TIS620**, **GB18030**, **GBK**, **BIG5HK**, **BIG5** }

4.2.1 Typedef Documentation

4.2.1.1 typedef enum __XSTOutputType XSTOutputType

This enum includes the types of output that can be used by ST.

Generally all applications will use Raster which is the default.

4.2.2 Enumeration Type Documentation

4.2.2.1 enum __XSTOutputType

This enum includes the types of output that can be used by ST.

Generally all applications will use Raster which is the default.

```
75                              {
76 #ifdef __STTYPES_H
77     fRasterOutput     = ST_DEVICE_RASTER,
78     fVectorOutput     = ST_DEVICE_VECTOR,
79 #else
80     fRasterOutput     = 1,
81     fVectorOutput     = 2,
82 #endif
83     fOutputTypeNotSet = -1
84 } XSTOutputType;
```

4.2.2.2 enum __XSTUnicodeEncoding

This enum includes all of the types of Unicode encoding formats that can be output from the following functions: XSTFontGetUnicodeName XSTFontGetUnicodeTypefaceName XSTFontFamilyGetUnicodeName.

Enumeration values:

fUTF7 basic ascii encoding.

fUTF8 UTF-8 encoding.

fUCS2 UCS-2 encoding.

fUTF16 UTF-16 encoding.

fUCS4 UCS-4 encoding.

fUTF32 UTF-32 encoding.

```
58                                      {
59     fUTF7 = 0,
60     fUTF8,
61     fUCS2,
62     fUTF16,
63     fUCS4,
64     fUTF32
65 } XSTUnicodeEncoding;
```

4.3 XSTFont functions

Functions

- XSTNameTag XSTFontGetNameTags (Display *dpy, XSTTypeEnv iEnv, XSTFont iFont, int *oNameCount)

  Returns an array of name tag values extracted from a specified font.

- byte * XSTFontGetNameString (Display *dpy, XSTTypeEnv iEnv, XSTFont iFont, XSTNameTag iNameTag, int *oStringLength)

  Returns an array of bytes containing the name of the specified font.

- XSTString * XSTFontGetUnicodeName (Display *dpy, XSTTypeEnv iEnv, XSTFont iFont, XSTUnicodeEncoding iEnc, uint16_t iNameID, XSTLocale *ioLocale, int *oCharCount)

  Returns an array of Unicode characters containing the name of the specified font.

- XSTFontType XSTFontGetExtInfo (Display *dpy, XSTTypeEnv iEnv, XSTFont iFont, const char **oFontName, const char **oPSName, XSTFontInfoFlags *oFlags, int *oSbitCount, int **oSbitArray, XSTFontMetrics *oFontMetrics)

  Returns the type of the font along with extended font information from a font.

- XSTDesignBaselines * XSTFontGetBaselines (Display *dpy, XSTTypeEnv iEnv, XSTFont iFont)

  Returns the offsets of all baselines defined in the font from the Roman baseline.

- XSTFeatureTag * XSTFontGetFeatures (Display *dpy, XSTTypeEnv iEnv, XSTFont iFont, int *oFeatureCount)

  Returns an array of features available in the font.

- char * XSTFontGetURL (Display *dpy, XSTTypeEnv iEnv, XSTFont iFont)

  Retrieves the URL/Filename of the font.

4.3.1 Function Documentation

4.3.1.1 XSTDesignBaselines* XSTFontGetBaselines (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFont *iFont*)

Returns the offsets of all baselines defined in the font from the Roman baseline.

This function allocates memory for the baselines. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

iFont font ID

Returns:

XSTDesignBaselines struct

4.3.1.2 XSTFontType XSTFontGetExtInfo (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFont *iFont*, const char ** *oFontName*, const char ** *oPSName*, XSTFontInfoFlags * *oFlags*, int * *oSbitCount*, int ** *oSbitArray*, XSTFontMetrics * *oFontMetrics*)

Returns the type of the font along with extended font information from a font.

Any of the output parameters can be NULL pointers, in that case ST will skip over them when returning values. This function allocates memory for the name strings, and the sbit array. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

iFont font ID

oFontName Copy of the "C" locale font name

oPSName Copy of the PostScript name of the font

oFlags an OR-ed mask of font flags

oSbitCount the number of the embedded bitmaps the font has

oSbitArray array of sbit sizes

oFontMetrics font metric information

Returns:
XSTFontType value

4.3.1.3 XSTFeatureTag* XSTFontGetFeatures (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFont *iFont*, int * *oFeatureCount*)

Returns an array of features available in the font.

This function allocates memory for the feature array. It is the callers responsiblity to free up this memory.

Parameters:
dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

iFont font ID

oFeatureCount number of features in the returned array

Returns:
XSTFeatureTag array

4.3.1.4 byte* XSTFontGetNameString (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFont *iFont*, XSTNameTag *iNameTag*, int * *oStringLength*)

Returns an array of bytes containing the name of the specified font.

This function allocates memory for the returned string. It is the callers responsiblity to free up this memory.

Parameters:
dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

iFont font ID

iNameTag NameTag that specifies the string

oStringLength number of bytes in the string

Returns:
byte array

4.3.1.5 XSTNameTag XSTFontGetNameTags (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFont *iFont*, int * *oNameCount*)

Returns an array of name tag values extracted from a specified font.

This function allocates memory for the array of name tags returned. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

iFont font ID

oNameCount number of retrieved string tags is stored here

Returns:

XSTNameTag array

4.3.1.6 XSTString* XSTFontGetUnicodeName (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFont *iFont*, XSTUnicodeEncoding *iEnc*, uint16_t *iNameID*, XSTLocale * *ioLocale*, int * *oCharCount*)

Returns an array of Unicode characters containing the name of the specified font.

The iEnc parameter specifies the Unicode encoding format to use for the returned font name. The ioLocale parameter allows the application to specify the desired language to return. If this language is not available, then the language of the returned string is returned in this space. This function allocates memory for the returned string. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

iFont font ID

iEnc Unicode encoding format to return

iNameID NameTag that specifies the string

ioLocale Locale/Language/Script of string

oCharCount number of bytes in the string

Returns:

Unicode character array

4.3.1.7 char* XSTFontGetURL (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFont *iFont*)

Retrieves the URL/Filename of the font.

This function allocates memory for the url name. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

iFont font ID

Returns:

array of characters

4.4 XSTFontFamily functions

Functions

- XSTFont * XSTFontFamilyGetFonts (Display *dpy, XSTTypeEnv iEnv, XSTFontFamily iFontFamily, int *oFontCount)

  Returns an array of XSTFont IDs that are members of the font family.

- XSTFontFamily XSTFontGetFontFamily (Display *dpy, XSTTypeEnv iEnv, XSTFont iFont)

  Returns a XSTFontFamily ID that the specified XSTFont (iFont) is a member.

- XSTString * XSTFontGetUnicodeTypefaceName (Display *dpy, XSTTypeEnv iEnv, XSTFont iFont, XSTUnicodeEncoding iEnc, XSTLocale *ioLocale, int *oCharCount, XSTFontWeightClass *oWeight, XSTFontWidthClass *oWidth, XSTFontStyle *oStyle)

  Returns the Typeface name of the font.

- XSTString * XSTFontFamilyGetUnicodeName (Display *dpy, XSTTypeEnv iEnv, XSTFontFamily iFontFamily, XSTUnicodeEncoding iEnc, XSTLocale *ioLocale, int *oCharCount)

  Returns an array of Unicode characters containing the name of the specified font family.

4.4.1 Function Documentation

4.4.1.1 XSTFont* XSTFontFamilyGetFonts (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFontFamily *iFontFamily*, int * *oFontCount*)

Returns an array of XSTFont IDs that are members of the font family.

This function allocates memory for the XSTFont array. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

iFontFamily font family ID

oFontCount number of fonts in the font family

Returns:

XSTFont ID array

4.4.1.2 XSTString* XSTFontFamilyGetUnicodeName (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFontFamily *iFontFamily*, XSTUnicodeEncoding *iEnc*, XSTLocale * *ioLocale*, int * *oCharCount*)

Returns an array of Unicode characters containing the name of the specified font family.

The iEnc parameter specifies the Unicode encoding format to use for the returned font name. The ioLocale parameter allows the application to specify the desired language to return. If this language is not available, then the language of the returned string is returned in this space. This function allocates memory for the returned string. It is allocates memory for the XSTFont array. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

iFontFamily font family ID

iEnc Unicode encoding format to use for returned string

ioLocale Locale/Language/Script of string

oCharCount number of Unicode characters in the returned string

Returns:
Unicode string

4.4.1.3 XSTFontFamily XSTFontGetFontFamily (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFont *iFont*)

Returns a XSTFontFamily ID that the specified XSTFont (iFont) is a member.

Parameters:
dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

iFont font ID

Returns:
XSTFontFamily ID

4.4.1.4 XSTString* XSTFontGetUnicodeTypefaceName (Display * *dpy*, XSTTypeEnv *iEnv*, XSTFont *iFont*, XSTUnicodeEncoding *iEnc*, XSTLocale * *ioLocale*, int * *oCharCount*, XSTFontWeightClass * *oWeight*, XSTFontWidthClass * *oWidth*, XSTFontStyle * *oStyle*)

Returns the Typeface name of the font.

Examples of this would be Regular, Bold, etc. The iEnc parameter specifies the Unicode encoding format to use for the returned font name. The ioLocale parameter allows the application to specify the desired language to return. If this language is not available, then the language of the returned string is returned in this space. This function allocates memory for the returned string. It is the callers responsiblity to free up this memory.

Parameters:
dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to delete

iFont font ID

iEnc Unicode encoding format to use for returned string

ioLocale Locale/Language/Script of string

oCharCount number of Unicode characters in the returned string

oWeight font wieght

oWidth font width

oStyle font style

Returns:
Unicode string

4.5 XSTGC functions

Functions

- void XSTGCSetColors (Display *dpy, GC gc, XSTGraphicsMask iMask, XSTRGBAColors colors)

  Sets graphics colors.

- XSTRGBAColor * XSTGCGetColors (Display *dpy, GC gc, XSTGraphicsMask *oMask)

  Returns an array of graphics colors.

- XSTAlphaMask XSTGCGetAlphaRange (Display *dpy, GC gc, XSTAlpha *oTextAlpha, XSTAlpha *oHighlightAlpha, XSTAlpha *oUnderlineAlpha, XSTAlpha *oStrikeThroughAlpha)

  Returns a mask of the alpha values that have been previously set.

- void XSTGCSetAlphaRange (Display *dpy, GC gc, XSTAlphaMask iMask, XSTAlpha iTextAlpha, XSTAlpha iHighlightAlpha, XSTAlpha iUnderline-Alpha, XSTAlpha iStrikeThroughAlpha)

  Sets the min and max alpha ranges.

- void XSTGCSetOutputFormat (Display *dpy, GC gc, XSTOutputType iType, XSTOutputMode iMode)

  Set the output type and mode.

- XSTOutputMode XSTGCGetOutputFormat (Display *dpy, GC gc, XSTOutputType *oType)

  Get the output type and mode.

- void XSTGCSetMatrix (Display *dpy, GC gc, XSTMatrix *iMatrix)

  Replaces the current transformation matrix with a new one.

- XSTMatrix * XSTGCGetMatrix (Display *dpy, GC gc)

  Return the current transformation matrix.

- void XSTGCConcatMatrix (Display *dpy, GC gc, XSTMatrix *iMatrix)
  Concat the current transformation matrix with another one.
- void XSTGCTranslate (Display *dpy, GC gc, double tx, double ty)
  Translate the current transformation matrix.
- void XSTGCScale (Display *dpy, GC gc, double sx, double sy)
  Scale the current transformation matrix.
- void XSTGCRotate (Display *dpy, GC gc, double theta)
  Rotate the current transformation matrix.
- void XSTGCShear (Display *dpy, GC gc, double shx, double shy)
  Shear the current transformation matrix.
- void XSTGCSetRegion (Display *dpy, GC gc, REGION *r)
  Set the clipping region for the gc.

4.5.1 Function Documentation

4.5.1.1 void XSTGCConcatMatrix (Display * *dpy*, GC *gc*, XSTMatrix * *iMatrix*)

Concat the current transformation matrix with another one.

Parameters:

dpy X Display value

gc Graphics Context

iMatrix transformation matrix to concat with current one.

Returns:

none

4.5.1.2 XSTAlphaMask XSTGCGetAlphaRange (Display * *dpy*, GC *gc*, XSTAlpha * *oTextAlpha*, XSTAlpha * *oHighlightAlpha*, XSTAlpha * *oUnderlineAlpha*, XSTAlpha * *oStrikeThroughAlpha*)

Returns a mask of the alpha values that have been previously set.

Any XSTAlpha with a pointer to NULL will not be returned.

Parameters:

dpy X Display value

gc Graphics Context

oTextAlpha Contains the min and max alpha values for the text

oHighlightAlpha Contains the min and max alpha values for the Highlight

oUnderlineAlpha Contains the min and max alpha values for the Underline

oStrikeThroughAlpha Contains the min and max alpha values for the Strike-Through

Returns:

XSTAlphaMask bitmask

4.5.1.3 XSTRGBAColor* XSTGCGetColors (Display * *dpy*, GC *gc*, XSTGraphicsMask * *oMask*)

Returns an array of graphics colors.

Parameters:

dpy X Display value

gc Graphics Context

oMask set of bitflags OR'ed together for which colors have been previously set

Returns:

XSTRGBAColor array

4.5.1.4 XSTMatrix* XSTGCGetMatrix (Display * *dpy*, GC *gc*)

Return the current transformation matrix.

This function allocates memory for the array of font IDs returned. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

gc Graphics Context

Returns:

XSTMatrix

4.5.1.5 XSTOutputMode XSTGCGetOutputFormat (Display * *dpy*, GC *gc*, XSTOutputType * *oType*)

Get the output type and mode.

The type refers to the type of monitor hooked up to the system, or format of the output to display. The mode refers to the output being raster or vector based.

Parameters:

dpy X Display value

gc Graphics Context

oType Type of output (B&W, Grayscale, LCD Optimized)

Returns:

XSTOutputMode value

4.5.1.6 void XSTGCRotate (Display * *dpy*, GC *gc*, double *theta*)

Rotate the current transformation matrix.

Parameters:

dpy X Display value

gc Graphics Context

theta value to rotate the matrix

Returns:

none

4.5.1.7 void XSTGCScale (Display * *dpy*, GC *gc*, double *sx*, double *sy*)

Scale the current transformation matrix.

Parameters:

dpy X Display value

gc Graphics Context

sx X factor to scale by

sy Y factor to scale by

Returns:

none

4.5.1.8 void XSTGCSetAlphaRange (Display * *dpy*, GC *gc*, XSTAlphaMask *iMask*, XSTAlpha *iTextAlpha*, XSTAlpha *iHighlightAlpha*, XSTAlpha *iUnderlineAlpha*, XSTAlpha *iStrikeThroughAlpha*)

Sets the min and max alpha ranges.

The iMask value is a bitmask OR'ed together to indicate which of the alpha ranges to set. Any XSTAlpha parameter that is NULL will not be set.

Parameters:

dpy X Display value

gc Graphics Context

iMask bitmask of alpha ranges to set

iTextAlpha Contains the min and max alpha values for the text

iHighlightAlpha Contains the min and max alpha values for the Highlight

iUnderlineAlpha Contains the min and max alpha values for the Underline

iStrikeThroughAlpha Contains the min and max alpha values for the Strike-Through

Returns:

none

4.5.1.9 void XSTGCSetColors (Display * *dpy*, GC *gc*, XSTGraphicsMask *iMask*, XSTRGBAColors *colors*)

Sets graphics colors.

See Xst_colors.h for a list of color positions

Parameters:

dpy X Display value

gc Graphics Context

iMask set of bitflags OR'ed together for which colors to change

colors array of colors

Returns:

none

4.5.1.10 void XSTGCSetMatrix (Display * *dpy*, GC *gc*, XSTMatrix * *iMatrix*)

Replaces the current transformation matrix with a new one.

Parameters:

dpy X Display value

gc Graphics Context

iMatrix new transformation matrix

Returns:

none

4.5.1.11 void XSTGCSetOutputFormat (Display * *dpy*, GC *gc*, XSTOutputType *iType*, XSTOutputMode *iMode*)

Set the output type and mode.

The type refers to the type of monitor hooked up to the system, or format of the output to display. The mode refers to the output being raster or vector based.

Parameters:

dpy X Display value

gc Graphics Context

iType Type of output (B&W, Grayscale, LCD Optimized)

iMode Output mode (raster, vector)

Returns:

none

4.5.1.12 void XSTGCSetRegion (Display * *dpy*, GC *gc*, REGION * *r*)

Set the clipping region for the gc.

Parameters:

dpy X Display value

gc Graphics Context

r This is the region to clip

Returns:

none

4.5.1.13 void XSTGCShear (Display * *dpy*, GC *gc*, double *shx*, double *shy*)

Shear the current transformation matrix.

Parameters:

dpy X Display value

gc Graphics Context

shx X amount to shear matrix

shy Y amount to shear matrix

Returns:

none

4.5.1.14 void XSTGCTranslate (Display * *dpy*, GC *gc*, double *tx*, double *ty*)

Translate the current transformation matrix.

Parameters:

dpy X Display value

gc Graphics Context

tx X translate value

ty Y translate value

Returns:

none

4.6 XSTLine functions

Functions

- XSTLine XSTLineNew (Display *dpy, XSTText iText, int iPosition, int iCharCount, int *oCharCount)

  Creates a new STLine object and returns the XSTLine XID referring to it.

- XSTLine XSTLineNewForWidth (Display *dpy, XSTText iText, int iPosition, Rational iWidth, int *oCharCount)

  Creates a new STLine object with as many characters as can fit within the width specified by iWidth and returns the XSTLine XID referring to it.

- void XSTLineDispose (Display ∗dpy, XSTLine iLine)

  Destroys the STLine and frees the XSTLine XID.

- void XSTLineGrow (Display ∗dpy, XSTLine iLine, Bool iAppend, int iCharCount)

  Adds characters to the beginning or to the end of the line.

- void XSTLineShrink (Display ∗dpy, XSTLine iLine, Bool iFromEnd, int iCharCount)

  Removes characters to the beginning or to the end of the line.

- int XSTLineGetPosition (Display ∗dpy, XSTLine iLine, int ∗oCharCount)

  Returns the line position within the STText object.

- void XSTLineSetMetrics (Display ∗dpy, XSTLine iLine, XSTExtLineMetrics ∗iLineMetrics, XSTBaselines ∗iBaselines)

  Imposes metrics on a line.

- XSTExtLineMetrics ∗ XSTLineGetMetrics (Display ∗dpy, XSTLine iLine, XSTBaselines ∗∗oBaselines)

  Returns metrics previously imposed on a line.

- XSTExtLineMetrics ∗ XSTLineGetDesignMetrics (Display ∗dpy, XSTLine iLine)

  Retrieves design metrics for the line based on scaled font design metrics.

- XSTTrapezoid ∗ XSTLineMeasureText (Display ∗dpy, GC gc, XSTLine iLine)

  Calculates a typographic bounding rectangle in user space of a line ignoring all imposed metrics and attributes.

- XSTTrapezoid ∗ XSTLineMeasureTextImage (Display ∗dpy, GC gc, XSTLine iLine)

  Calculates a standard bounding rectangle of a laid-out line of text after all attributes ((justification, alignment, etc.) have been applied in user space.

- XSTTrapezoid ∗∗ XSTLineGetGlyphBounds (Display ∗dpy, GC gc, XSTLine iLine, int iMaxBoundsCount, XSTBounds iBounds, int ∗oBoundsCount)

  Calculates a boundaries of glyphs of a final laid-out line in device coordinates.

- Bool XSTLineHitTest (Display ∗dpy, GC gc, XSTLine iLine, Rational iX, Rational iY, int ∗oPrimaryOffset, int ∗oSecondaryOffset)

  Converts a pair of coordinates to the logical character offset.

- Bool XSTLinePositionToCaret (Display *dpy, GC gc, XSTLine iLine, int iPosition, Bool iIsLeading, XSTCaret *oStrongCaret, XSTCaret *oWeakCaret)

  Returns one or two carets (strong and weak) for a specified logical character position If the gc parameter is specified, then the affine transform is applied to the caret and the resulting device space caret is returned.

- int XSTLineMoveCaret (Display *dpy, XSTLine iLine, int iPosition, XSTCaretDirection iDirection, XSTCaretMovement iMovement)

  Calculates a new position of the caret.

- void XSTLineRender (Display *dpy, Drawable d, GC gc, XSTLine iLine)

  Renders text represented by an STLine object onto the Drawable.

- void XSTLineAddHighlight (Display *dpy, XSTLine iLine, int iPosition, int iCharCount)

  Adds a highlighted region identified by the first character and the number of characters to an STLine object.

- void XSTLineRemoveHighlight (Display *dpy, XSTLine iLine, int iPosition, int iCharCount)

  Removes highlighting from a region of characters in an STLine object.

- void XSTLineGetHighlight (Display *dpy, XSTLine iLine, int *oRegionCount, int **oPosition, int **oCharCount)

  Queries the currently defined highlighted regions.

4.6.1 Function Documentation

4.6.1.1 void XSTLineAddHighlight (Display * *dpy*, XSTLine *iLine*, int *iPosition*, int *iCharCount*)

Adds a highlighted region identified by the first character and the number of characters to an STLine object.

STLineRender() needs to be called to display it.

Parameters:

dpy X Display value

iLine an XSTLine XID referring to the STLine object to use

iPosition first character to highlight

iCharCount number of characters in the highlighted region.

Returns:
none

4.6.1.2 void XSTLineDispose (Display * *dpy*, XSTLine *iLine*)

Destroys the STLine and frees the XSTLine XID.

Parameters:
dpy X Display value
iLine an XSTLine XID referring to the STLine object to delete

Returns:
none

4.6.1.3 XSTExtLineMetrics* XSTLineGetDesignMetrics (Display * *dpy*, XSTLine *iLine*)

Retrieves design metrics for the line based on scaled font design metrics.

Returned metrics is in user space. This function allocates memory for the returned metrics structure. It is the callers responsiblity to free up this memory.

Parameters:
dpy X Display value
iLine an XSTLine XID referring to the STLine object to use

Returns:
XSTExtLineMetrics struct

4.6.1.4 XSTTrapezoid** XSTLineGetGlyphBounds (Display * *dpy*, GC *gc*, XSTLine *iLine*, int *iMaxBoundsCount*, XSTBounds *iBounds*, int * *oBoundsCount*)

Calculates a boundaries of glyphs of a final laid-out line in device coordinates.

The array of trapezoidal bounding boxes corresponds to text regions within the line. This function allocates memory for the returned trapezoid structure. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

gc Graphics Context

iLine an XSTLine XID referring to the STLine object to use

iMaxBoundsCount specifies the precision of the output - the maximal number of trapezoids the function can allocate to store glyph bounds. Set it to 1 if only one trapezoid for the entire line is requested

iBounds specifies the type of origin used for calculating bounds

oBoundsCount number of trapezoids in the oBounds array

Returns:

XSTTrapezoid struct

4.6.1.5 void XSTLineGetHighlight (Display * *dpy*, XSTLine *iLine*, int * *oRegionCount*, int ** *oPosition*, int ** *oCharCount*)

Queries the currently defined highlighted regions.

This function allocates memory for the returned position and count arrays. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iLine an XSTLine XID referring to the STLine object to use

oRegionCount number of highlighted regions

oPosition array of first first character offsets of highlighted regions

oCharCount array of lengths of highlighted regions

Returns:

none

4.6.1.6 XSTExtLineMetrics* XSTLineGetMetrics (Display * *dpy*, XSTLine *iLine*, XSTBaselines ** *oBaselines*)

Returns metrics previously imposed on a line.

This function allocates memory for the returned metrics and baseline structures. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iLine an XSTLine XID referring to the STLine object to use

oBaselines baselines of this line of text

Returns:

XSTExtLineMetrics struct

4.6.1.7 int XSTLineGetPosition (Display * *dpy*, XSTLine *iLine*, int * *oCharCount*)

Returns the line position within the STText object.

Parameters:

dpy X Display value

iLine an XSTLine XID referring to the STLine object to use

oCharCount returns a number of characters in the line

Returns:

the first character offset of the line within the STText

4.6.1.8 void XSTLineGrow (Display * *dpy*, XSTLine *iLine*, Bool *iAppend*, int *iCharCount*)

Adds characters to the beginning or to the end of the line.

This function is used for creating customized line-break algorithms.

Parameters:

dpy X Display value

iLine an XSTLine XID referring to the STLine object to use

iAppend true if characters are appended, false if prepended

iCharCount specifies the number of added characters

Returns:

none

4.6.1.9 Bool XSTLineHitTest (Display * *dpy*, GC *gc*, XSTLine *iLine*, Rational *iX*, Rational *iY*, int * *oPrimaryOffset*, int * *oSecondaryOffset*)

Converts a pair of coordinates to the logical character offset.

If the gc parameter is specified, then the affine transform is applied in reverse to change the iX and iY values from device coordinates to user coordinates.

Parameters:

dpy X Display value

gc Graphics Context

iLine an XSTLine XID referring to the STLine object to use

iX X coordinate relative to the UL corner of the line bounding rectangle

iY Y coordinate relative to the UL corner of the line bounding rectangle

oPrimaryOffset logical offset of the glyph closest to (iX, iY) coordinates

oSecondaryOffset if (iX, iY) are on a line direction boundary this gives the second logical offset.

Returns:

is hit leading or trailing value

4.6.1.10 XSTTrapezoid* XSTLineMeasureText (Display * *dpy*, GC *gc*, XSTLine *iLine*)

Calculates a typographic bounding rectangle in user space of a line ignoring all imposed metrics and attributes.

If the gc parameter is specified, then the affine transform is applied to the rectangular bounding box and the resulting trapezoid is returned. If not, then the rectangle is returned in the trapezoid structure with all four of its vetices specified. This function allocates memory for the returned trapezoid structure. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

gc Graphics Context

iLine an XSTLine XID referring to the STLine object to use

Returns:

XSTTrapezoid struct

4.6.1.11 XSTTrapezoid* XSTLineMeasureTextImage (Display * *dpy*, GC *gc*, XSTLine *iLine*)

Calculates a standard bounding rectangle of a laid-out line of text after all attributes ((justification, alignment, etc.) have been applied in user space.

If the gc parameter is specified, then the affine transform is applied to the rectangular bounding box and the resulting trapezoid is returned. If not, then the rectangle is returned in the trapezoid structure with all four of its vetices specified. This function allocates memory for the returned trapezoid structure. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

gc Graphics Context

iLine an XSTLine XID referring to the STLine object to use

Returns:

XSTTrapezoid struct

4.6.1.12 int XSTLineMoveCaret (Display * *dpy*, XSTLine *iLine*, int *iPosition*, XSTCaretDirection *iDirection*, XSTCaretMovement *iMovement*)

Calculates a new position of the caret.

The caret can be moved by characters, words or Unicode clusters. Direction can be previous, next, left or right. When the initial position of the caret is on a line direction boundary, this function calculates the new position of the primary caret. Inside an LTR text and not on the direction boundary ”next” is equivalent to ”right,” and ”previous” is equivalent to ”left.” Inside an RTL text ”next” is equivalent to ”left” and ”previous” is equivalent to ”right.” For vertical text ”left” is ”top” and ”right” is ”bottom”

Parameters:

dpy X Display value

iLine an XSTLine XID referring to the STLine object to use

iPosition initial caret position

iDirection specifies direction - left, right, next, or previous

iMovement specifies movement type - character, word, or cluster

Returns:

new caret position

4.6.1.13 XSTLine XSTLineNew (Display * *dpy*, XSTText *iText*, int *iPosition*, int *iCharCount*, int * *oCharCount*)

Creates a new STLine object and returns the XSTLine XID referring to it.

STLine is a unit of displayable text. It is derived from STText object and refers its text. A line-breaking algorithm can be implemented by sequentually adding characters to the STLine object and measuring its typographic bounds.

Parameters:

dpy X Display value

iText an XSTText XID referring to the STText object to use

iPosition an offset from the first character of the STText object

iCharCount number of characters to add to the line. A special value of ST_CHARCOUNT_TOEND adds all STText characters to this STLine object

oCharCount actual number of added characters is stored here

Returns:

XSTLine XID

4.6.1.14 XSTLine XSTLineNewForWidth (Display * *dpy*, XSTText *iText*, int *iPosition*, Rational *iWidth*, int * *oCharCount*)

Creates a new STLine object with as many characters as can fit within the width specified by iWidth and returns the XSTLine XID referring to it.

Parameters:

dpy X Display value

iText an XSTText XID referring to the STText object to use

iPosition an offset from the first character of the STText object

iWidth text width of the line in typographic points. To use text width set for the parent STText object use CWidthNotSet

oCharCount actual number of added characters is stored here

Returns:

XSTLine XID

4.6.1.15 Bool XSTLinePositionToCaret (Display * *dpy*, GC *gc*, XSTLine *iLine*, int *iPosition*, Bool *iIsLeading*, XSTCaret * *oStrongCaret*, XSTCaret * *oWeakCaret*)

Returns one or two carets (strong and weak) for a specified logical character position If the gc parameter is specified, then the affine transform is applied to the caret and the resulting device space caret is returned.

If not, then the caret is returned in user space coordinates.

Parameters:

dpy X Display value

gc Graphics Context

iLine an XSTLine XID referring to the STLine object to use

iPosition logical offset of the character within STLine's parent STText object

iIsLeading matters only if character at iOffset is at line direction boundary

oStrongCaret denotes primary (or main if oSplitCaret is false) caret

oWeakCaret denotes secondary (or main if oSplitCaret is false) caret

Returns:

if character at iOffsets is at line direction boundary

4.6.1.16 void XSTLineRemoveHighlight (Display * *dpy*, XSTLine *iLine*, int *iPosition*, int *iCharCount*)

Removes highlighting from a region of characters in an STLine object.

. STLineRender() needs to be called to display the removed highlight.

Parameters:

dpy X Display value

iLine an XSTLine XID referring to the STLine object to use

iPosition first character to remove highlighting from

iCharCount number of characters in the region.

Returns:

none

4.6.1.17 void XSTLineRender (Display * *dpy*, Drawable *d*, GC *gc*, XSTLine *iLine*)

Renders text represented by an STLine object onto the Drawable.

Parameters:

dpy X Display value

d Drawable (pixmap or window)

gc Graphics Context

iLine an XSTLine XID referring to the STLine object to use

Returns:

none

4.6.1.18 void XSTLineSetMetrics (Display * *dpy*, XSTLine *iLine*, XSTExtLineMetrics * *iLineMetrics*, XSTBaselines * *iBaselines*)

Imposes metrics on a line.

Parameters:

dpy X Display value

iLine an XSTLine XID referring to the STLine object to use

iLineMetrics sets ascent, descent and width or use NULL pointer to use the value derived from its parent STText object

iBaselines baselines of this line of text or NULL pointer to use the value derived from its parent STText object

Returns:

none

4.6.1.19 void XSTLineShrink (Display * *dpy*, XSTLine *iLine*, Bool *iFromEnd*, int *iCharCount*)

Removes characters to the beginning or to the end of the line.

This function is used for creating customized line-break algorithms.

Parameters:

dpy X Display value

iLine an XSTLine XID referring to the STLine object to use

iFromEnd true if characters are removed from line end, false if from line start

iCharCount specifies the number of added characters

Returns:
none

4.7 XSTScaler functions

Functions

- XSTTag XSTScalerGetInfo (Display *dpy, XSTTypeEnv iEnv, XSTScaler iScaler, uint32_t *oVersion, const char **oShortName, const char **oLongName, const char **oNotice, XSTScalerFlags *oSFlags, XSTScalerFontMask *oFFlags)

  Retrieves general information from the scaler.

4.7.1 Function Documentation

4.7.1.1 XSTTag XSTScalerGetInfo (Display * *dpy*, XSTTypeEnv *iEnv*, XSTScaler *iScaler*, uint32_t * *oVersion*, const char ** *oShortName*, const char ** *oLongName*, const char ** *oNotice*, XSTScalerFlags * *oSFlags*, XSTScalerFontMask * *oFFlags*)

Retrieves general information from the scaler.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iScaler Scaler ID

oVersion Scaler version

oShortName Short scaler name - up to 16 characters

oLongName Free-form scaler name

oNotice Free-form copyright notice

oSFlags Scaler flags

oFFlags Font types supported by the scaler

Returns:
scaler tag.

4.8 XSTConvert functions

Functions

- void XSTConvertUserToDeviceSpace (Display ∗dpy, GC gc, double iX, double iY, double ∗oX, double ∗oY)

  Converts a user space coordinate into a device space coordinate by applying the affine transform held in the gc to the specified coordinate.

- void XSTConvertDeviceToUserSpace (Display ∗dpy, GC gc, double iX, double iY, double ∗oX, double ∗oY)

  Converts a device space coordinate into a user space coordinate by applying the reverse of the affine transform held in the gc to the specified coordinate.

- double XSTConvertDeviceSpaceToUserDist (Display ∗dpy, GC gc, double iX, double iY)

  Converts a device space coordinate into a user space distance.

- XSTRectangle ∗ XSTConvertGetBoundingRectangle (XSTTrapezoid ∗iTrap)

  Gets the smallest rectangular that contains the supplied trapezoid.

- XSTTrapezoid ∗ XSTConvertUserToDeviceTrapezoid (Display ∗dpy, GC gc, XSTTrapezoid ∗iTrap)

  Converts a user space trapezoid into a device space trapezoid by applying the affine transform.

- XSTTrapezoid ∗ XSTConvertDeviceToUserTrapezoid (Display ∗dpy, GC gc, XSTTrapezoid ∗iTrap)

  Converts a device space trapezoid into a user space trapezoid by applying the reverse of the affine transform.

4.8.1 Function Documentation

4.8.1.1 double XSTConvertDeviceSpaceToUserDist (Display * *dpy*, GC *gc*, double *iX*, double *iY*)

Converts a device space coordinate into a user space distance.

Parameters:

dpy X Display value

gc Graphics Context

iX device space X value

iY device space Y value

Returns:

distance value

4.8.1.2 void XSTConvertDeviceToUserSpace (Display * *dpy*, GC *gc*, double *iX*, double *iY*, double * *oX*, double * *oY*)

Converts a device space coordinate into a user space coordinate by applying the reverse of the affine transform held in the gc to the specified coordinate.

Parameters:

dpy X Display value

gc Graphics Context

iX device space X value

iY device space Y value

oX returned user space X value

oY returned user space Y value

Returns:

none

4.8.1.3 XSTTrapezoid* XSTConvertDeviceToUserTrapezoid (Display * *dpy*, GC *gc*, XSTTrapezoid * *iTrap*)

Converts a device space trapezoid into a user space trapezoid by applying the reverse of the affine transform.

This function allocates memory for the returned XSTRectangle structure. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

gc Graphics Context

iTrap device space trapezoid

Returns:

XSTTrapezoid structure

4.8.1.4 XSTRectangle* XSTConvertGetBoundingRectangle (XSTTrapezoid * *iTrap*)

Gets the smallest rectangular that contains the supplied trapezoid.

This function allocates memory for the returned XSTRectangle structure. It is the callers responsiblity to free up this memory.

Parameters:

iTrap trapezoid

Returns:

XSTRectangle structure

4.8.1.5 void XSTConvertUserToDeviceSpace (Display * *dpy*, GC *gc*, double *iX*, double *iY*, double * *oX*, double * *oY*)

Converts a user space coordinate into a device space coordinate by applying the affine transform held in the gc to the specified coordinate.

Parameters:

dpy X Display value

gc Graphics Context

iX user space X value

iY user space Y value

oX returned device space X value

oY returned device space Y value

Returns:

none

4.8.1.6 XSTTrapezoid* XSTConvertUserToDeviceTrapezoid (Display * *dpy*, GC *gc*, XSTTrapezoid * *iTrap*)

Converts a user space trapezoid into a device space trapezoid by applying the affine transform.

This function allocates memory for the returned XSTRectangle structure. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

gc Graphics Context

iTrap user space trapezoid

Returns:

XSTTrapezoid structure

4.9 XSTString functions

Functions

- int XSTStringGetBytes (char *iString, XSTEncoding iEnc, int iNumCharacters)

  Returns the number of bytes in iString.

- int XSTStringGetCharacters (char *iString, XSTEncoding iEnc, int iNumBytes)

  Returns the number of characters in iString.

- int XSTStringGetPrevCharacterSize (char *iString, XSTEncoding iEnc, int iBytePos)

  Returns the number of bytes in character prior to the one beginning at iBytePos.

- int XSTStringGetNextCharacterSize (char *iString, XSTEncoding iEnc, int iBytePos)

  Returns the number of bytes in character after the one beginning at iBytePos.

- int XSTStringGetCharactersFromXtoY (char *iString, XSTEncoding iEnc, int iBytePosX, int iBytePosY)

  Returns the characters between bytes position iBytePosX and iBytePosY.

- int XSTStringGetBytesFromXtoY (char *iString, XSTEncoding iEnc, int iCharacterPosX, int iCharacterPosY)

  Returns the bytes between character position iCharacterPosX and iCharacterPosY.

- int XSTStringLength (char *iString, XSTEncoding iEnc)

  Returns the number of bytes used by the string.

4.9.1 Function Documentation

4.9.1.1 int XSTStringGetBytes (char * *iString*, XSTEncoding *iEnc*, int *iNumCharacters*)

Returns the number of bytes in iString.

Parameters:

iString pointer to string of characters

iEnc Encoding of string. see Xst_enum.h for encoding types

iNumCharacters Number of characters in the string.

Returns:

number of bytes

4.9.1.2 int XSTStringGetBytesFromXtoY (char * *iString*, XSTEncoding *iEnc*, int *iCharacterPosX*, int *iCharacterPosY*)

Returns the bytes between character position iCharacterPosX and iCharacterPosY.

Parameters:

iString pointer to string of characters

iEnc Encoding of string. see Xst_enum.h for encoding types

iCharacterPosX Character position of to begin at

iCharacterPosY Character position of to end at

Returns:

number of bytes

4.9.1.3 int XSTStringGetCharacters (char * *iString*, XSTEncoding *iEnc*, int *iNumBytes*)

Returns the number of characters in iString.

Parameters:

iString pointer to string of characters

iEnc Encoding of string. see Xst_enum.h for encoding types

iNumBytes Number of bytes in the string.

Returns:

number of characters

4.9.1.4 int XSTStringGetCharactersFromXtoY (char * *iString*, XSTEncoding *iEnc*, int *iBytePosX*, int *iBytePosY*)

Returns the characters between bytes position iBytePosX and iBytePosY.

Parameters:

iString pointer to string of characters

iEnc Encoding of string. see Xst_enum.h for encoding types

iBytePosX Byte position of the start of a character in the string

iBytePosY Byte position of the start of a character in the string

Returns:

number of characters

4.9.1.5 int XSTStringGetNextCharacterSize (char * *iString*, XSTEncoding *iEnc*, int *iBytePos*)

Returns the number of bytes in character after the one beginning at iBytePos.

Parameters:

iString pointer to string of characters

iEnc Encoding of string. see Xst_enum.h for encoding types

iBytePos Byte position of the start of a character in the string

Returns:

number of bytes

4.9.1.6 int XSTStringGetPrevCharacterSize (char * *iString*, XSTEncoding *iEnc*, int *iBytePos*)

Returns the number of bytes in character prior to the one beginning at iBytePos.

Parameters:

iString pointer to string of characters

iEnc Encoding of string. see Xst_enum.h for encoding types

iBytePos Byte position of the start of a character in the string

Returns:

number of bytes

4.9.1.7 int XSTStringLength (char * *iString*, XSTEncoding *iEnc*)

Returns the number of bytes used by the string.

Parameters:

iString pointer to string of characters

iEnc Encoding of string. see Xst_enum.h for encoding types

Returns:

number of bytes

4.10 XSTStyle functions

Functions

- XSTStyle XSTStyleNewEmpty (Display *dpy, XSTTypeEnv iEnv)

  Creates a new empty Style and returns an XID referring to the new STStyle object.

- XSTStyle XSTStyleNewDefault (Display *dpy, XSTTypeEnv iEnv)

  Creates a new Style with the attributes set to default values and returns an XID referring to the new STStyle object.

- XSTStyle XSTStyleNewCopy (Display *dpy, XSTStyle iStyle)

  Creates a new Style and copies the attributes from an existing STStyle.

- void XSTStyleDispose (Display *dpy, XSTStyle iStyle)

Delete the STStyle and free the XSTStyle XID.

- XSTStyleComparison XSTStyleCompare (Display *dpy, XSTStyle iLhs, XSTStyleMask iMask, XSTStyle iRhs)

  Compare two STStyle objects.

- Bool XSTStyleIsEmpty (Display *dpy, XSTStyle iStyle)

  Checks if an STStyle is empty.

- void XSTStyleClear (Display *dpy, XSTStyle iStyle, XSTStyleMask iMask)

  Removed specified attributes from an STStyle.

- void XSTStyleSetFont (Display *dpy, XSTStyle iStyle, XSTStyleMask iMask, XSTFont iFont, double iSize, XSTLocale iLocale, XSTBaselineFlag iBaseline)

  Sets font-related attributes of the STStyle.

- XSTFont XSTStyleGetFont (Display *dpy, XSTStyle iStyle, XSTStyleMask *oMask, double *oSize, XSTLocale *oLocale, XSTBaselineFlag *oBaseline)

  Gets font-related attributes of the STStyle.

- void XSTStyleSetScaler (Display *dpy, XSTStyle iStyle, XSTStyleMask iMask, XSTScaler iScaler, XSTHintingMode iHints, XSTSbitsMode iSbits)

  Sets scaler and its parameters for this STStyle object.

- XSTScaler XSTStyleGetScaler (Display *dpy, XSTStyle iStyle, XSTStyleMask *oMask, XSTHintingMode *oHints, XSTSbitsMode *oSbits)

  Returns the scaler and its parameters for this STStyle object.

- void * XstStyleScalerControl (Display *dpy, XSTStyle iStyle, void *iToScaler, int iToByteCount, int *oFromByteCount)

  Scaler ioctl - exchange control information with the scaler.

- void XSTStyleSetEffects (Display *dpy, XSTStyle iStyle, XSTStyleMask iMask, XSTStyleEffects iStyleEffects, XSTStrikeThroughEffects iStrikeThroughEffects, XSTUnderlineEffects iUnderlineEffects)

  Sets some effects related attributes for the STStyle object.

- void XSTStyleGetEffects (Display *dpy, XSTStyle iStyle, XSTStyleMask *oMask, XSTStyleEffects *oStyleEffects, XSTStrikeThroughEffects *oStrikeThroughEffects, XSTUnderlineEffects *oUnderlineEffects)

  Retrieves some effects related attributes for the STStyle object.

- void XSTStyleResetAttributes (Display *dpy, XSTStyle iStyle, XSTStyleMask iMask)

  Resets specified STStyle attributes to their default values.

- void XSTStyleCopyAttributes (Display *dpy, XSTStyle iToStyle, XSTStyleMask iMask, XSTStyle iFromStyle)

  Copies specified attributes set in the source STStyle object to the destination STStyle object.

- void XSTStyleOverwriteAttributes (Display *dpy, XSTStyle iToStyle, XSTStyleMask iMask, XSTStyle iFromStyle)

  Copies specified attributes set in the source STStyle object to the destination STStyle object.

- void XSTStyleUnderwriteAttributes (Display *dpy, XSTStyle iToStyle, XSTStyleMask iMask, XSTStyle iFromStyle)

  Copies specified attributes set in the source STStyle object to the destination STStyle object.

- XSTBaselines * XSTStyleGetBaselines (Display *dpy, XSTStyle iStyle, XSTBaselineFlag iBaseline)

  Retrieves baselines position for a style.

- XSTLineMetrics * XSTStyleGetDesignMetrics (Display *dpy, XSTStyle iStyle)

  Retrieves design metrics for the style based on scaled font design metrics.

- void XSTStyleSetOptions (Display *dpy, XSTStyle iStyle, XSTStyleMask iMask, int iImposeWidth, int iBeforeWithStreamShift, int iAfterWithStreamShift, int iCrossStreamShift, int iHangingInhibitFactor, int iKerningInhibitFactor, int iDecompositionInhibitFactor)

  Sets additional style options.

- XSTStyleMask XSTStyleGetOptions (Display *dpy, XSTStyle iStyle, int *oImposeWidth, int *oBeforeWithStreamShift, int *oAfterWithStreamShift, int *oCrossStreamShift, int *oTracking, int *oHangingInhibitFactor, int *oKerningInhibitFactor, int *oDecompositionInhibitFactor)

  Retrieves additional style options.

- void XSTStyleSetLayoutEngine (Display *dpy, XSTStyle iStyle, XSTStyleMask iMask, XSTLayoutEngine iLayoutEngine)

  Sets layout engine and its parameters for this STStyle object.

- XSTLayoutEngine XSTStyleGetLayoutEngine (Display *dpy, XSTStyle iStyle, XSTStyleMask *oMask)

  Returns the layout engine and its parameters for this STStyle object.

4.10.1 Function Documentation

4.10.1.1 void XSTStyleClear (Display * *dpy*, XSTStyle *iStyle*, XSTStyleMask *iMask*)

Removed specified attributes from an STStyle.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

iMask a set of XSTStyleMasks values OR'ed together or ST_SM_ALL to clear all values

Returns:

none

4.10.1.2 XSTStyleComparison XSTStyleCompare (Display * *dpy*, XSTStyle *iLhs*, XSTStyleMask *iMask*, XSTStyle *iRhs*)

Compare two STStyle objects.

Parameters:

dpy X Display value

iLhs XSTStyle XID of the first STStyle object to use

iMask mask of attributes to include in comparison

iRhs XSTStyle XID of the second STStyle object to use

Returns:

XSTStyleComparison value

4.10.1.3 void XSTStyleCopyAttributes (Display * *dpy*, XSTStyle *iToStyle*, XSTStyleMask *iMask*, XSTStyle *iFromStyle*)

Copies specified attributes set in the source STStyle object to the destination STStyle object.

Attributes not set in the source object are unset in the destination object.

Parameters:

dpy X Display value

iToStyle XSTStyle XID of the STStyle object to copy to

iMask a set of XSTStyleMasks values OR'ed together or ST_SM_ALL to copy all values

iFromStyle XSTStyle XID of the STStyle object to copy from

Returns:

none

4.10.1.4 void XSTStyleDispose (Display * *dpy*, XSTStyle *iStyle*)

Delete the STStyle and free the XSTStyle XID.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

Returns:

XSTStyle XID

4.10.1.5 XSTBaselines* XSTStyleGetBaselines (Display * *dpy*, XSTStyle *iStyle*, XSTBaselineFlag *iBaseline*)

Retrieves baselines position for a style.

Baseline positions are measured as offsets from the baseline specified by iBaseline. The function allocates memory for the array of baselines returned. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

iBaseline baseline from which offsets are measured

Returns:
XSTBaselines array

4.10.1.6 XSTLineMetrics* XSTStyleGetDesignMetrics (Display * *dpy*, XSTStyle *iStyle*)

Retrieves design metrics for the style based on scaled font design metrics.

Returned metrics are in users space. The function allocates memory for the returned metrics. It is the callers responsiblity to free up this memory.

Parameters:
dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

Returns:
XSTLineMetrics structure

4.10.1.7 void XSTStyleGetEffects (Display * *dpy*, XSTStyle *iStyle*, XSTStyleMask * *oMask*, XSTStyleEffects * *oStyleEffects*, XSTStrikeThroughEffects * *oStrikeThroughEffects*, XSTUnderlineEffects * *oUnderlineEffects*)

Retrieves some effects related attributes for the STStyle object.

The oMask parameter indicates which of the fields has been previously changed.

Parameters:
dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

oMask a set of XSTStyleMasks values OR'ed together

oStyleEffects a set of STStyleEffects values OR'ed together

oStrikeThroughEffects a set of STStrikeThrough values OR'ed together

oUnderlineEffects a set of STUnderline values OR'ed together

Returns:
none

4.10.1.8 XSTFont XSTStyleGetFont (Display * *dpy*, XSTStyle *iStyle*, XSTStyleMask * *oMask*, double * *oSize*, XSTLocale * *oLocale*, XSTBaselineFlag * *oBaseline*)

Gets font-related attributes of the STStyle.

The iMask parameter indicates which of the attributes have been previously set.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

oMask a set of XSTStyleMasks values OR'ed together

oSize font size measured in typographic points

oLocale Locale/Language/Script of the style

oBaseline The dominant baseline for this style

Returns:

XSTFont XID

4.10.1.9 XSTLayoutEngine XSTStyleGetLayoutEngine (Display * *dpy*, XSTStyle *iStyle*, XSTStyleMask * *oMask*)

Returns the layout engine and its parameters for this STStyle object.

The oMask parameter indicates which fields were previously set

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

oMask a set of XSTStyleMasks values OR'ed together

Returns:

XSTLayoutEngine value

4.10.1.10 XSTStyleMask XSTStyleGetOptions (Display * *dpy*, XSTStyle *iStyle*, int * *oImposeWidth*, int * *oBeforeWithStreamShift*, int * *oAfterWithStreamShift*, int * *oCrossStreamShift*, int * *oTracking*, int * *oHangingInhibitFactor*, int * *oKerningInhibitFactor*, int * *oDecompositionInhibitFactor*)

Retrieves additional style options.

Returns the mask of style options that have been previously set.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

oImposeWidth weight factor for font-defined glyph widths

oBeforeWithStreamShift weight factor for with-stream shift applied before each glyph

oAfterWithStreamShift weight factor for with-stream shift applied after each glyph

oCrossStreamShift cross-stream shift weight factor

oTracking weither factor for distance between glyphs

oHangingInhibitFactor hanging glyph weight factor

oKerningInhibitFactor kerning weight factor

oDecompositionInhibitFactor ligature decomposition weight factor

Returns:

XSTStyleMask value

4.10.1.11 XSTScaler XSTStyleGetScaler (Display * *dpy*, XSTStyle *iStyle*, XSTStyleMask * *oMask*, XSTHintingMode * *oHints*, XSTSbitsMode * *oSbits*)

Returns the scaler and its parameters for this STStyle object.

The oMask parameter indicates which fields were previously set

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

oMask a set of XSTStyleMasks values OR'ed together

oHints Specifies if the scaler should use hints

oSbits Specifies if the scaler should use embedded bitmaps

Returns:

XSTScaler value

4.10.1.12 Bool XSTStyleIsEmpty (Display * *dpy*, XSTStyle *iStyle*)

Checks if an STStyle is empty.

IE. no attributes were assigned to it.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

Returns:

Bool value

4.10.1.13 XSTStyle XSTStyleNewCopy (Display * *dpy*, XSTStyle *iStyle*)

Creates a new Style and copies the attributes from an existing STStyle.

It returns an XID referring to the new STStyle object.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

Returns:

XSTStyle XID

4.10.1.14 XSTStyle XSTStyleNewDefault (Display * *dpy*, XSTTypeEnv *iEnv*)

Creates a new Style with the attributes set to default values and returns an XID referring to the new STStyle object.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

Returns:

XSTStyle XID

4.10.1.15 XSTStyle XSTStyleNewEmpty (Display * *dpy*, XSTTypeEnv *iEnv*)

Creates a new empty Style and returns an XID referring to the new STStyle object.

Style is a unit of formatting information applied to a sequence of characters in an STText object. It has several attributes that can be set and copied from one Style to another.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

Returns:

XSTStyle XID

4.10.1.16 void XSTStyleOverwriteAttributes (Display * *dpy*, XSTStyle *iToStyle*, XSTStyleMask *iMask*, XSTStyle *iFromStyle*)

Copies specified attributes set in the source STStyle object to the destination STStyle object.

Attributes must both be in the iMask and be set in both the source and destination objects to be copied.

Parameters:

dpy X Display value

iToStyle XSTStyle XID of the STStyle object to copy to

iMask a set of XSTStyleMasks values OR'ed together or ST_SM_ALL to copy all values

iFromStyle XSTStyle XID of the STStyle object to copy from

Returns:

none

4.10.1.17 void XSTStyleResetAttributes (Display * *dpy*, XSTStyle *iStyle*, XSTStyleMask *iMask*)

Resets specified STStyle attributes to their default values.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

iMask a set of XSTStyleMasks values OR'ed together or ST_SM_ALL to reset all values

Returns:
none

4.10.1.18 void* XstStyleScalerControl (Display * *dpy*, XSTStyle *iStyle*, void * *iToScaler*, int *iToByteCount*, int * *oFromByteCount*)

Scaler ioctl - exchange control information with the scaler.

This function may allocate memory for the data returned by the scaler. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

iToScaler This data is sent to the scaler

iToByteCount Number of bytes in iToScaler

oFromByteCount Number of bytes returned from the scaler

Returns:
pointer to data returned by the scaler

4.10.1.19 void XSTStyleSetEffects (Display * *dpy*, XSTStyle *iStyle*, XSTStyleMask *iMask*, XSTStyleEffects *iStyleEffects*, XSTStrikeThroughEffects *iStrikeThroughEffects*, XSTUnderlineEffects *iUnderlineEffects*)

Sets some effects related attributes for the STStyle object.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

iMask a set of XSTStyleMasks values OR'ed together or ST_SM_ALL to change all values

iStyleEffects a set of STStyleEffects values OR'ed together

iStrikeThroughEffects a set of STStrikeThrough values OR'ed together

iUnderlineEffects a set of STUnderline values OR'ed together

Returns:
none

4.10.1.20 void XSTStyleSetFont (Display * *dpy*, XSTStyle *iStyle*, XSTStyleMask *iMask*, XSTFont *iFont*, double *iSize*, XSTLocale *iLocale*, XSTBaselineFlag *iBaseline*)

Sets font-related attributes of the STStyle.

The iMask parameter indicates which of the attributes are to be set.

Parameters:
dpy X Display value
iStyle XSTStyle XID of the STStyle object to use
iMask a set of XSTStyleMasks values OR'ed together or ST_SM_ALL to change all values
iFont font ID
iSize font size measured in typographic points
iLocale Locale/Language/Script of the style
iBaseline The dominant baseline for this style

Returns:
none

4.10.1.21 void XSTStyleSetLayoutEngine (Display * *dpy*, XSTStyle *iStyle*, XSTStyleMask *iMask*, XSTLayoutEngine *iLayoutEngine*)

Sets layout engine and its parameters for this STStyle object.

Parameters:
dpy X Display value
iStyle XSTStyle XID of the STStyle object to use
iMask a set of XSTStyleMasks values OR'ed together or ST_SM_ALL change all values
iLayoutEngine XSTLayoutEngine XID of the STLayoutEngine object to use

Returns:
none

4.10.1.22 void XSTStyleSetOptions (Display * *dpy*, XSTStyle *iStyle*, XSTStyleMask *iMask*, int *iImposeWidth*, int *iBeforeWithStreamShift*, int *iAfterWithStreamShift*, int *iCrossStreamShift*, int *iHangingInhibitFactor*, int *iKerningInhibitFactor*, int *iDecompositionInhibitFactor*)

Sets additional style options.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

iMask a set of XSTStyleMasks values OR'ed together or ST_SM_ALL to copy all values

iImposeWidth weight factor for font-defined glyph widths

iBeforeWithStreamShift weight factor for with-stream shift applied before each glyph

iAfterWithStreamShift weight factor for with-stream shift applied after each glyph

iCrossStreamShift cross-stream shift weight factor

iHangingInhibitFactor hanging glyph weight factor

iKerningInhibitFactor kerning weight factor

iDecompositionInhibitFactor ligature decomposition weight factor

Returns:

none

4.10.1.23 void XSTStyleSetScaler (Display * *dpy*, XSTStyle *iStyle*, XSTStyleMask *iMask*, XSTScaler *iScaler*, XSTHintingMode *iHints*, XSTSbitsMode *iSbits*)

Sets scaler and its parameters for this STStyle object.

Parameters:

dpy X Display value

iStyle XSTStyle XID of the STStyle object to use

iMask a set of XSTStyleMasks values OR'ed together or ST_SM_ALL change all values

iScaler XSTScaler XID of the STScaler object to use

iHints Specifies if the scaler should use hints

iSbits Specifies if the scaler should use embedded bitmaps

Returns:
none

4.10.1.24 void XSTStyleUnderwriteAttributes (Display * *dpy*, XSTStyle *iToStyle*, XSTStyleMask *iMask*, XSTStyle *iFromStyle*)

Copies specified attributes set in the source STStyle object to the destination STStyle object.

Attributes must both be in the iMask and must not be set in the destination object in order to be copied.

Parameters:

dpy X Display value

iToStyle XSTStyle XID of the STStyle object to copy to

iMask a set of XSTStyleMasks values OR'ed together or ST_SM_ALL to copy all values

iFromStyle XSTStyle XID of the STStyle object to copy from

Returns:
none

4.11 XSTText functions

Functions

- XSTText XSTTextNewEmpty (Display *dpy, XSTTypeEnv iEnv)

  Creates a new STText - Text object and returns an XSTText XID refering to it.

- XSTText XSTTextNew (Display *dpy, XSTTypeEnv iEnv, XSTString *iChars, int iCharCount, char *iLocale)

  Creates a new STText and assign text to it.

- XSTText XSTTextNewCopy (Display *dpy, XSTText iText)

  Duplicates an existing STText.

- void XSTTextDispose (Display *dpy, XSTText iText)

  Destroys an existing STText object and frees the XSTText XID associated with it.

- void XSTTextClear (Display *dpy, XSTText iText)

  Restores the state of the STText to its initial state - removes all assigned text removes all layout attributes.

- void XSTTextSetMetrics (Display *dpy, XSTText iText, XSTExtLineMetrics *iLineMetrics, XSTBaselines *iBaselines)

  Sets one or more STText metrics attributes.

- XSTExtLineMetrics * XSTTextGetMetrics (Display *dpy, XSTText iText, XSTBaselines **oBaselines)

  Returns the metrics previously set for the STText object.

- void XSTTextSetControls (Display *dpy, XSTText iText, XSTTextMask iMask, XSTDirection iDirection, XSTJustification iJustification, XSTFlushFactor iFlushFactor, XSTLocale iLocale, XSTFontFallbackPolicy iPolicy, XSTLayoutOptions iLayoutOptions)

  Sets one or more STText controls.

- XSTTextMask XSTTextGetControls (Display *dpy, XSTText iText, XSTDirection *oDirection, XSTJustification *oJustification, XSTFlushFactor *oFlushFactor, XSTLocale *oLocale, XSTFontFallbackPolicy *oPolicy, XSTLayoutOptions *oLayoutOptions)

  Retrieves previously set STText controls.

- void XSTTextSetFontFallbacks (Display *dpy, XSTText iText, int iFontCount, XSTFont *iFontArray)

  Specifies a list of fonts for displaying characters missing from fonts set by Style objects for the STText object.

- XSTFont * XSTTextGetFontFallbacks (Display *dpy, XSTText iText, int *oFontCount)

  Returns a list of substitution fonts set for this text layout object.

- void XSTTextCopyAttributes (Display *dpy, XSTText iToText, XSTTextMask iMask, XSTText iFromText)

  Copies specified attributes set by XSTTextSetMetrics() and XSTTextSetControls() and the font fallback list from one STText object to another.

- void XSTTextResetAttributes (Display *dpy, XSTText iText, XSTTextMask iMask)

  Restores specified attributes to default values.

- void XSTTextSetText (Display *dpy, XSTText iText, XSTString *iChars, int iCharCount, char *iLocale)

Assigns text to an STText object.

- XSTString * XSTTextGetText (Display *dpy, XSTText iText, int *oCharCount, char *iLocale)

  Retrieves source text from the STText object.

- void XSTTextUpdate (Display *dpy, XSTText iText, XSTTextChangedType iChange, int iTextOffset, int iTextLength)

  Informs the STText object that the text has been inserted or deleted.

- void XSTTextAugmentStyle (Display *dpy, XSTText iText, XSTStyle iStyle, int iFirstChar, int iCharCount)

  Takes the supplied Style and modifies all styles covered in the supplied range.

- void XSTTextOverwriteStyle (Display *dpy, XSTText iText, XSTStyle iStyle, int iFirstChar, int iCharCount)

  Takes the supplied Style and modifies all styles covered in the supplied range.

- void XSTTextUnderwriteStyle (Display *dpy, XSTText iText, XSTStyle iStyle, int iFirstChar, int iCharCount)

  Takes the supplied Style and modifies all styles covered in the supplied range.

- void XSTTextSetStyle (Display *dpy, XSTText iText, XSTStyle iStyle, int iFirstChar, int iCharCount)

  Assigns a Style to a sequence of characters of an STText object.

- XSTStyle XSTTextGetStyle (Display *dpy, XSTText iText, int iPosition, int *oFirstChar, int *oCharCount)

  Returns an XID of a copy of the STStyle object assigned to character at offset iTextOffset in the STText object.

- XSTStyle XSTTextGetCommonStyle (Display *dpy, XSTText iText, int iPosition, int iCharCount)

  Returns an XID of a new STStyle object that represents all common style characteristics of a set of characters in an STText.

- XSTFont * XSTTextFindMissingChars (Display *dpy, XSTText iText, int *oSegCount, int **oSegOffset, int **oSegLength)

  Finds all text segments that contain missing characters and returns font IDs for the substituted fonts used to render these segments.

4.11.1 Function Documentation

4.11.1.1 void XSTTextAugmentStyle (Display * *dpy*, XSTText *iText*, XSTStyle *iStyle*, int *iFirstChar*, int *iCharCount*)

Takes the supplied Style and modifies all styles covered in the supplied range.

The function will break styles as necessary. Any attributes that are present in the supplied Style are added or replaced in the iText styles. STText creates a copy of the Style by calling its copy constructor, that it deallocates when STText's destructor is called.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

iStyle XSTStyle XID referring to the STStyle to use

iFirstChar the first character that iStyle applies to

iCharCount number of characters iStyle applies to

Returns:

none

4.11.1.2 void XSTTextClear (Display * *dpy*, XSTText *iText*)

Restores the state of the STText to its initial state - removes all assigned text removes all layout attributes.

- * -

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to clear

Returns:

none

4.11.1.3 void XSTTextCopyAttributes (Display * *dpy*, XSTText *iToText*, XSTTextMask *iMask*, XSTText *iFromText*)

Copies specified attributes set by XSTTextSetMetrics() and XSTTextSetControls() and the font fallback list from one STText object to another.

Parameters:

dpy X Display value

iToText XSTText XID referring to the destination STText to use

iMask set of bitflags OR'ed together showing which fields to copy

iFromText XSTText XID referring to the source STText to use

Returns:

none

4.11.1.4 void XSTTextDispose (Display * *dpy*, XSTText *iText*)

Destroys an existing STText object and frees the XSTText XID associated with it.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to free

Returns:

none

4.11.1.5 XSTFont* XSTTextFindMissingChars (Display * *dpy*, XSTText *iText*, int * *oSegCount*, int ** *oSegOffset*, int ** *oSegLength*)

Finds all text segments that contain missing characters and returns font IDs for the substituted fonts used to render these segments.

This function allocates memory for three arrays (oSegOffset, oSegLength, and the returned array of font IDs) which the caller is required to free.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

oSegCount number of segments with missing characters. oSegOffset, oSegLength, and the returned font ID array all contain oSegCount elements

oSegOffset array of offsets to first characters in segments with missing characters

oSegLength number of characters in each segment

Returns:
XSTFont XID array

4.11.1.6 XSTStyle XSTTextGetCommonStyle (Display * *dpy*, XSTText *iText*, int *iPosition*, int *iCharCount*)

Returns an XID of a new STStyle object that represents all common style characteristics of a set of characters in an STText.

The caller is responsible for calling the Style's destructor when it is no longer needed.

Parameters:
dpy X Display value

iText XSTText XID referring to the STText to use

iPosition offset of the character being queried

iCharCount number of characters being queried

Returns:
XSTStyle XID

4.11.1.7 XSTTextMask XSTTextGetControls (Display * *dpy*, XSTText *iText*, XSTDirection * *oDirection*, XSTJustification * *oJustification*, XSTFlushFactor * *oFlushFactor*, XSTLocale * *oLocale*, XSTFontFallbackPolicy * *oPolicy*, XSTLayoutOptions * *oLayoutOptions*)

Retrieves previously set STText controls.

Returns a bitmask indicating which of the fields was previously set.

Parameters:
dpy X Display value

iText XSTText XID referring to the STText to use

oDirection specifies text directionality

oJustification specifies text justification

oFlushFactor specifies text alignment

oLocale specifies text language/locale/script

oPolicy specifies font fallback policy

oLayoutOptions specifies the ICU layout options to use

Returns:
XSTTextMask value

4.11.1.8 XSTFont* XSTTextGetFontFallbacks (Display * *dpy*, XSTText *iText*, int * *oFontCount*)

Returns a list of substitution fonts set for this text layout object.

This function allocates memory for the returned font array. It is the callers responsiblity to free up this memory.

Parameters:
dpy X Display value

iText XSTText XID referring to the STText to use

oFontCount number of fonts in the array

Returns:
XSTFont ID array

4.11.1.9 XSTExtLineMetrics* XSTTextGetMetrics (Display * *dpy*, XSTText *iText*, XSTBaselines ** *oBaselines*)

Returns the metrics previously set for the STText object.

This function allocates memory for the returned metrics and baseline structures. It is the callers responsiblity to free up this memory.

Parameters:
dpy X Display value

iText XSTText XID referring to the STText to use

oBaselines baselines imposed on lines of text

Returns:
XSTExtLineMetrics structure

4.11.1.10 XSTStyle XSTTextGetStyle (Display * *dpy*, XSTText *iText*, int *iPosition*, int * *oFirstChar*, int * *oCharCount*)

Returns an XID of a copy of the STStyle object assigned to character at offset iTextOffset in the STText object.

The caller is responsible for calling the Style's destructor when it is no longer needed.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

iPosition offset of the character being queried

oFirstChar offset of the first character belonging to the Style

oCharCount number of characters in the Style

Returns:

XSTStyle XID

4.11.1.11 XSTString* XSTTextGetText (Display * *dpy*, XSTText *iText*, int * *oCharCount*, char * *iLocale*)

Retrieves source text from the STText object.

If the iLocale parameter is NULL, then it uses the value retrieved from setlocale(). An example of a proper iLocale string would be: "en_US.UTF-8". This value is used to convert the encoding of the string from UTF-16 into the local encoding. This function allocates memory for the returned string. It is the callers responsiblity to free up this memory.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

oCharCount number of characters in a paragraph of text

iLocale locale string.

Returns:

text string

4.11.1.12 XSTText XSTTextNew (Display * *dpy*, XSTTypeEnv *iEnv*, XSTString * *iChars*, int *iCharCount*, char * *iLocale*)

Creates a new STText and assign text to it.

Return the XSTText XID value. If the iLocale parameter is NULL, then it uses the value retrieved from setlocale(). An example of a proper iLocale string would be: "en_US.UTF-8".

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iChars a pointer to the beginning of a paragraph of text

iCharCount number of characters in a paragraph of text

iLocale locale string

Returns:

XSTText XID

4.11.1.13 XSTText XSTTextNewCopy (Display * *dpy*, XSTText *iText*)

Duplicates an existing STText.

Returns the XSTText XID referring to the new STText.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

Returns:

XSTText XID

4.11.1.14 XSTText XSTTextNewEmpty (Display * *dpy*, XSTTypeEnv *iEnv*)

Creates a new STText - Text object and returns an XSTText XID refering to it.

STText is an opaque object that represents a unit of text and styles associated with it.

STText object does not duplicate source text, instead it stores pointer to it. It is the user's responsibility to maintain the source text buffer, insert and remove text from it and notify STText of it by calling XSTTextChangeText.

STText uses coordinates relative to the upper-left corner of its typographic bounding box.

STText consists of one or more lines of text (STLine object.) A line of text is an atomic displayable unit of text in ST.

Styles (STStyle object) are assigned to STText. If none are assigned a default style is taken from STTypeEnv.

STText has several attributes:

- Width
- Global Direction
- Justification
- Alignment
- Locale

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

Returns:

XSTText XID

4.11.1.15 void XSTTextOverwriteStyle (Display * *dpy*, XSTText *iText*, XSTStyle *iStyle*, int *iFirstChar*, int *iCharCount*)

Takes the supplied Style and modifies all styles covered in the supplied range.

The function will break styles as necessary. Any attributes that are present in both the supplied style and the iText style are replaced. STText creates a copy of the Style by calling its copy constructor, that it deallocates when STText's destructor is called.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

iStyle XSTStyle XID referring to the STStyle to use

iFirstChar the first character that iStyle applies to

iCharCount number of characters iStyle applies to

Returns:

none

4.11.1.16 void XSTTextResetAttributes (Display * *dpy*, XSTText *iText*, XSTTextMask *iMask*)

Restores specified attributes to default values.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to reset

iMask set of bitflags OR'ed together showing which fields to set

Returns:

none

4.11.1.17 void XSTTextSetControls (Display * *dpy*, XSTText *iText*, XSTTextMask *iMask*, XSTDirection *iDirection*, XSTJustification *iJustification*, XSTFlushFactor *iFlushFactor*, XSTLocale *iLocale*, XSTFontFallbackPolicy *iPolicy*, XSTLayoutOptions *iLayoutOptions*)

Sets one or more STText controls.

The iMask parameter is used to specify which of the controls to set.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

iMask set of bitflags OR'ed together showing which fields to set

iDirection specifies text directionality

iJustification specifies text justification

iFlushFactor specifies text alignment

iLocale specifies text language

iPolicy specifies font fallback policy

iLayoutOptions specifies the ICU layout options to use

Returns:

none

4.11.1.18 void XSTTextSetFontFallbacks (Display * *dpy*, XSTText *iText*, int *iFontCount*, XSTFont * *iFontArray*)

Specifies a list of fonts for displaying characters missing from fonts set by Style objects for the STText object.

The order of fonts in the list matters: the search starts from the first font in the list and continues until all missing characters are found.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

iFontCount number of fonts in the array

iFontArray array of font IDs

Returns:

none

4.11.1.19 void XSTTextSetMetrics (Display * *dpy*, XSTText *iText*, XSTExtLineMetrics * *iLineMetrics*, XSTBaselines * *iBaselines*)

Sets one or more STText metrics attributes.

STText itself does not use any of these attributes but all lines derived from an STText object inherit them.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

iLineMetrics line metrics imposed on lines of text or NULL

iBaselines baselines imposed on lines of text or NULL

Returns:

none

4.11.1.20 void XSTTextSetStyle (Display * *dpy*, XSTText *iText*, XSTStyle *iStyle*, int *iFirstChar*, int *iCharCount*)

Assigns a Style to a sequence of characters of an STText object.

STText creates a copy of the Style by calling its copy constructor, that it deallocates when STText's destructor is called.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

iStyle XSTStyle XID referring to the STStyle to use

iFirstChar the first character that iStyle applies to

iCharCount number of characters iStyle applies to

Returns:

none

4.11.1.21 void XSTTextSetText (Display * *dpy*, XSTText *iText*, XSTString * *iChars*, int *iCharCount*, char * *iLocale*)

Assigns text to an STText object.

STText object does not create a private copy of this text, it keeps a pointer to it, so it is responsibility of the caller to maintain the text buffer, insert and delete text from it, etc. If the iLocale parameter is NULL, then it uses the value retrieved from setlocale(). An example of a proper iLocale string would be: "en_US.UTF-8".

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

iChars a pointer to the beginning of a paragraph of text

iCharCount number of characters in a paragraph of text

iLocale locale string.

Returns:

none

4.11.1.22 void XSTTextUnderwriteStyle (Display * *dpy*, XSTText *iText*, XSTStyle *iStyle*, int *iFirstChar*, int *iCharCount*)

Takes the supplied Style and modifies all styles covered in the supplied range.

The function will break styles as necessary. Any attributes that are present in the supplied style, but not in the iText style are added. STText creates a copy of the Style by calling its copy constructor, that it deallocates when STText's destructor is called.

Parameters:

dpy X Display value

iText XSTText XID referring to the STText to use

iStyle XSTStyle XID referring to the STStyle to use

iFirstChar the first character that iStyle applies to

iCharCount number of characters iStyle applies to

Returns:
none

4.11.1.23 void XSTTextUpdate (Display * *dpy*, XSTText *iText*, XSTTextChangedType *iChange*, int *iTextOffset*, int *iTextLength*)

Informs the STText object that the text has been inserted or deleted.

Parameters:
dpy X Display value

iText XSTText XID referring to the STText to use

iChange indicates whether characters were inserted or removed

iTextOffset position where text has been inserted or removed

iTextLength number of added or removed characters

Returns:
none

4.12 General typedefs

Compounds

- struct XSTAlphaStruct

  XSTAlpha is a structure containing the min and max alpha values to associate with a particular XST drawable object.

- struct XSTBaselines

  XSTBaselines contains baseline offsets from the default baseline.

- struct XSTCaret

  XSTCaret contains two end points of the caret line.

- struct XSTNameTagStruct

  XSTNameTagStruct is a structure containing the four elements that comprise a TrueType name tag.

- struct **XSTStyledGlyph**

Typedefs

- typedef XID XSTTypeEnv

  XID for STTypeEnv object.

- typedef unsigned int XSTFont

  STFont value.

- typedef unsigned int XSTScaler

  STScaler value.

- typedef unsigned int XSTLayoutEngine

  STLayoutEngine value.

- typedef unsigned int XSTTag

  A Tag assigned to the scaler.

- typedef unsigned int XSTFontFamily

  STFontFamily value.

- typedef XSTNameTagStruct * XSTNameTag

  pointer to structure, allows NULLs.

- typedef uint32_t XSTFeatureTag

  OpenType/TrueTypeGX features.

- typedef STFontFallbackPolicy XSTFontFallbackPolicy

  Fallback Policy to use globally for Type Environment or locally for Text.

- typedef STFontServerFontType XSTFontType

  Contains the type of the Font.

- typedef STBaselineFlag XSTBaselineFlag

  Contains which of the standard baselines to use.

- typedef STFontInfoFlags XSTFontInfoFlags

Contains bitmask values indicating features that the font supports.

- typedef STFontMetrics XSTFontMetrics

  This structure contains metric information about the font.

- typedef STFontWeightClass XSTFontWeightClass

  Contains the weight descriptor for the font.

- typedef STFontWidthClass XSTFontWidthClass

  Contains the width descriptor for the font.

- typedef STFontStyle XSTFontStyle

  Contains the inclination descriptor for the font.

- typedef uint32_t XSTFontLocationMask

  Contains a bitmask of the areas to search for fonts.

- typedef Rational XSTDesignBaselines [32]

  Metrics in design coordinates.

- typedef XID XSTStyle

  XID refering to STStyle object.

- typedef unsigned int XSTFontFeatureTag

  Contains a value of an OpenType or TrueTypeGX features that is available for the particular font.

- typedef STStyleComparison XSTStyleComparison

  Contains the equality value for the style.

- typedef STStrikeThrough XSTStrikeThroughEffects

  This bitmask indicates the type of strike through to use.

- typedef STStyleEffects XSTStyleEffects

  This bitmask indicates the various style effects to use during this style run.

- typedef STUnderline XSTUnderlineEffects

  This bitmask indicates the type of underline to use.

- typedef STStyleMask XSTStyleMask

  This bitmask indicates the values of the Style that are desired to change, or that have been previously changed.

- typedef STLineMetrics XSTLineMetrics

  XSTLineMetrics contains basic metric information.

- typedef XID XSTGlyphVector

  STGlyphVector.

- typedef STPoint **XSTPoint**
- typedef unsigned int XSTRGBAColor

  Single color in RGBA format.

- typedef XSTRGBAColor XSTRGBAColors [9]

  Array of changeable colors.

- typedef STMatrix XSTMatrix

  Transformation matrix.

- typedef XSTAlphaStruct * XSTAlpha

  This pointer to the structure is used to allow users to pass NULL to the function indicating that no change is necessary or desired for that alpha range.

- typedef STOutputMode XSTOutputMode

  Contains the format of the output device along with a bitmask of the flags to use for output.

- typedef STGraphicsMask XSTGraphicsMask

  This bitmask represents which colors to set or which colors have been previously set.

- typedef unsigned int XSTLine

  XID referring to an STLine object.

- typedef STTrapezoid XSTTrapezoid

  XSTTrapezoid contains four corners of the trapezoid bounding the text.

- typedef STRectangle XSTRectangle

  XSTRectangle contains two points indicating the bounding corners around untranslated text.

- typedef STCaretDirection XSTCaretDirection

  Contains the caret direction type.

- typedef STCaretMovement XSTCaretMovement

  Contains the caret movement type.

- typedef STLayoutEngineFlags XSTLayoutEngineFlags

  This bitmask indicates the capabilities of the layout engine.

- typedef STScalerFlags XSTScalerFlags

  This bitmask indicates the capabilities of the scaler.

- typedef STHintingMode XSTHintingMode

  Contains the mode for applying hints.

- typedef STSbitsMode XSTSbitsMode

  Contains the mode for using embedded bitmaps.

- typedef STFontServerFontMask XSTScalerFontMask

  This bitmask indicates the font type that the font server is using for the font.

- typedef XID XSTText

  XID referring to an STText object.

- typedef STTextChanged XSTTextChangedType

  Contains text change notification indicators of what type of change is occurring.

- typedef STBounds XSTBounds

  Contains information on the caret origin and glyph orgin setting.

- typedef STTextMask XSTTextMask

  This bitmask indicates the values of the Text that are desired to change, or that have been previously changed.

- typedef STDirection XSTDirection

  Contains the direction of the text.

- typedef STJustification XSTJustification

  Contains the justification of the text.

- typedef STFlushFactor XSTFlushFactor

  Contains the flush factor of the line.

- typedef STLayoutOptions XSTLayoutOptions

  Contains the layout options for ICU.

- typedef STExtLineMetrics XSTExtLineMetrics

  XSTExtLineMetrics contains basic and extended metric information.

- typedef uint32_t XSTLocale

  Contains the language/locale information.

- typedef uint32_t XSTString

  This is a type cast for all strings passed into and out of XST.

4.12.1 Typedef Documentation

4.12.1.1 typedef STBaselineFlag XSTBaselineFlag

Contains which of the standard baselines to use.

See STBaselineFlag for more information on the valid values of XSTBaselineFlag.

4.12.1.2 typedef STBounds XSTBounds

Contains information on the caret origin and glyph orgin setting.

See STBounds for more information on the valid values.

4.12.1.3 typedef STCaretDirection XSTCaretDirection

Contains the caret direction type.

See STCaretDirection in sttypes.h for more information about valid values.

4.12.1.4 typedef STCaretMovement XSTCaretMovement

Contains the caret movement type.

See STCaretMovement in sttypes.h for more information about valid values.

4.12.1.5 typedef STDirection XSTDirection

Contains the direction of the text.

See STDirection in sttypes.h for more information about the valid values.

4.12.1.6 typedef STFlushFactor XSTFlushFactor

Contains the flush factor of the line.

See STFlushFactor in sttypes.h for more information about the valid values.

4.12.1.7 typedef STFontFallbackPolicy XSTFontFallbackPolicy

Fallback Policy to use globally for Type Environment or locally for Text.

See STFontFallbackPolicy for information on values to use.

4.12.1.8 typedef unsigned int XSTFontFeatureTag

Contains a value of an OpenType or TrueTypeGX features that is available for the particular font.

See STFontFeatureTag in sttypes.h for more information about the valid values.

4.12.1.9 typedef STFontInfoFlags XSTFontInfoFlags

Contains bitmask values indicating features that the font supports.

These features include kerning support and embedded bitmaps.

4.12.1.10 typedef uint32_t XSTFontLocationMask

Contains a bitmask of the areas to search for fonts.

Bitmasks include system, local and user. See STFontLocationsMask for more information about valid values.

4.12.1.11 typedef STFontMetrics XSTFontMetrics

This structure contains metric information about the font.

See STFontMetrics in sttypes.h for more information about the structure, it's members, and their values.

4.12.1.12 typedef STFontStyle XSTFontStyle

Contains the inclination descriptor for the font.

This value ranges from Normal to Slanted. See STFontStyle for more information about valid values.

4.12.1.13 typedef STFontServerFontType XSTFontType

Contains the type of the Font.

See STFontType in sttypes.h for more information on the valid values of XSTFontType.

4.12.1.14 typedef STFontWeightClass XSTFontWeightClass

Contains the weight descriptor for the font.

This value ranges from Thin to Black. See STFontWeightClass in sttypes.h for more information about the valid values.

4.12.1.15 typedef STFontWidthClass XSTFontWidthClass

Contains the width descriptor for the font.

This value ranges from Ultra condensed to ultra expanded. See STFontWidthClass in sttypes.h for more information about valid values.

4.12.1.16 typedef STGraphicsMask XSTGraphicsMask

This bitmask represents which colors to set or which colors have been previously set.

See STGraphicsMask in sttypes.h for more information about valid values.

4.12.1.17 typedef STHintingMode XSTHintingMode

Contains the mode for applying hints.

See STHintingMode in sttypes.h for more information.

4.12.1.18 typedef STJustification XSTJustification

Contains the justification of the text.

See STJustification in sttypes.h for more information about the valid values.

4.12.1.19 typedef STLayoutOptions XSTLayoutOptions

Contains the layout options for ICU.

Currently this includes treating text as an array of glyph IDs. See STLayoutOptions in sttypes.h for more information on valid values.

4.12.1.20 typedef STLineMetrics XSTLineMetrics

XSTLineMetrics contains basic metric information.

See STLineMetrics in sttypes.h for more information on the structure.

4.12.1.21 typedef uint32_t XSTLocale

Contains the language/locale information.

This is used by Styles for the langauge of the text, and for requesting font information in a particular language. See stlangdef.h for valid values.

4.12.1.22 typedef STMatrix XSTMatrix

Transformation matrix.

x' = xx * x + xy * y + tx y' = yx * x + yy * y + ty

See STMatrix in sttypes.h for more information about the structure

4.12.1.23 typedef STOutputMode XSTOutputMode

Contains the format of the output device along with a bitmask of the flags to use for output.

See STOutputMode in sttypes.h for more information about valid values.

4.12.1.24 typedef STSbitsMode XSTSbitsMode

Contains the mode for using embedded bitmaps.

See STSbitsMode in sttypes.h for more information.

4.12.1.25 typedef STScalerFlags XSTScalerFlags

This bitmask indicates the capabilities of the scaler.

Abilities include hinting, support for various output formats, algorithmic styles, etc. See STScalerFlags in sttypes.h for more information about valid values.

4.12.1.26 typedef STFontServerFontMask XSTScalerFontMask

This bitmask indicates the font type that the font server is using for the font.

This value differs slightly from the STFontType value. See STFontServerFontMask in sttypes.h for more information about the valid values.

4.12.1.27 typedef STStrikeThrough XSTStrikeThroughEffects

This bitmask indicates the type of strike through to use.

Strike throughs may be single, or double lines and may be thick. See STStrikeThrough in sttypes.h for more information.

4.12.1.28 typedef uint32_t XSTString

This is a type cast for all strings passed into and out of XST.

By casting the string to an XSTString, then all formats and encoding can be converted by ICU without problems of bounding.

4.12.1.29 typedef STStyleComparison XSTStyleComparison

Contains the equality value for the style.

These values range from equal, unequal, to subset and superset. See STStyleComparison in sttypes.h for more information on the valid values.

4.12.1.30 typedef STStyleEffects XSTStyleEffects

This bitmask indicates the various style effects to use during this style run.

Style effects include ligature splitting, algorithmic styles, vertical text, etc. See STStyleEffects in sttypes.h for more information.

4.12.1.31 typedef STStyleMask XSTStyleMask

This bitmask indicates the values of the Style that are desired to change, or that have been previously changed.

See STStyleMask in sttypes.h for more information on the valid values.

4.12.1.32 typedef unsigned int XSTTag

A Tag assigned to the scaler.

Scaler values are:

4.12.1.33 typedef STTextChanged XSTTextChangedType

Contains text change notification indicators of what type of change is occurring.

From inserted text to removed text. See STTextChanged for more information on the valid values.

4.12.1.34 typedef STTextMask XSTTextMask

This bitmask indicates the values of the Text that are desired to change, or that have been previously changed.

See STTextMask in sttypes.h for more information on the valid values.

4.12.1.35 typedef STTrapezoid XSTTrapezoid

XSTTrapezoid contains four corners of the trapezoid bounding the text.

See STTrapezoid for more information about the structure.

4.12.1.36 typedef STUnderline XSTUnderlineEffects

This bitmask indicates the type of underline to use.

Underlines may be single, double, dashes, or wavy and may be thick. Additionally, there may be two underlines where one underline is represented by the lower half of the bitmask and the other by the upper half. See STUnderline in sttypes.h for more information.

4.13 STDevice class and its children

4.14 XSTGlyphVector functions

Functions

- XSTGlyphVector XSTGlyphVectorNew (Display *dpy, XSTTypeEnv iEnv, XSTStyledGlyph *iGlyphs, int iGlyphCount)

  Creates a new STGlyphVector object and returns the XSTGlyphVector XID referring to it.

- XSTGlyphVector XSTGlyphVectorNewCopy (Display *dpy, XSTGlyphVector iGlyph)

  Creates a copy of the STGlyphVector and returns the new XSTGlyphVector XID referring to it.

- void XSTGlyphVectorDispose (Display *dpy, XSTGlyphVector iGlyph)

  Destroys the STGlyphVector and frees the XSTGlyphVector XID.

- void XSTGlyphVectorSetGlyphs (Display *dpy, XSTGlyphVector iGlyph, XSTStyledGlyph *iGlyphs, int iGlyphCount)

  Sets the array of glyphs associated with the STGlyphVector.

- XSTStyledGlyph * XSTGlyphVectorGetGlyphs (Display *dpy, XSTGlyphVector iGlyph, int iPosition, int *ioGlyphCount)

  Returns the list of styled glyphs associated with the STGlyphVector.

- void XSTGlyphVectorReplaceGlyphs (Display *dpy, XSTGlyphVector iGlyph, int iPosition, int iGlyphCount, XSTStyledGlyph *iGlyphs, int iGlyphArrayCount)

  Replaces a specified range of glyphs with a supplied array of styled glyphs.

- void XSTGlyphVectorAdjustPositions (Display ∗dpy, XSTGlyphVector iGlyph, int iPosition, int iGlyphCount, XSTPoint iDelta)

  Changes the positions of a specified range of glyphs by moving them by an amount equal to the x and y values in the XSTPoint structure.

- XSTRectangle ∗ XSTGlyphVectorMeasure (Display ∗dpy, XSTGlyphVector iGlyph, int iPosition, int iGlyphCount)

  Returns the rectangular box containing the specified glyphs.

- XSTTrapezoid ∗ XSTGlyphVectorGetBounds (Display ∗dpy, GC gc, XSTGlyphVector iGlyph, int iPosition, int iGlyphCount, int iMaxBoundsCount, int iBoundsType, int ∗oBoundsCount)

  Returns an array of XSTTrapezoids containing bounding information for the specified glyphs.

- void XSTGlyphVectorRender (Display ∗dpy, Drawable d, GC gc, XSTGlyphVector iGlyph, int iPosition, int iGlyphCount)

  Draws the glyphs on the specified drawable.

4.14.1 Function Documentation

4.14.1.1 void XSTGlyphVectorAdjustPositions (Display ∗ *dpy*, XSTGlyphVector *iGlyph*, int *iPosition*, int *iGlyphCount*, XSTPoint *iDelta*)

Changes the positions of a specified range of glyphs by moving them by an amount equal to the x and y values in the XSTPoint structure.

Parameters:

dpy X Display pointer

iGlyph an GlyphVector XID.

iPosition Index position of first glyph to adjust.

iGlyphCount Number of glyphs in vector to adjust

iDelta Delta to change each specified glyph by.

Returns:

none

4.14.1.2 void XSTGlyphVectorDispose (Display * *dpy*, XSTGlyphVector *iGlyph*)

Destroys the STGlyphVector and frees the XSTGlyphVector XID.

Parameters:

dpy X Display pointer

iGlyph an GlyphVector XID.

Returns:

none

4.14.1.3 XSTTrapezoid* XSTGlyphVectorGetBounds (Display * *dpy*, GC *gc*, XSTGlyphVector *iGlyph*, int *iPosition*, int *iGlyphCount*, int *iMaxBoundsCount*, int *iBoundsType*, int * *oBoundsCount*)

Returns an array of XSTTrapezoids containing bounding information for the specified glyphs.

Parameters:

dpy X Display pointer

gc X Graphics Context

iGlyph an GlyphVector XID.

iPosition Index position for first glyph to get bounds for

iGlyphCount Number of glyphs to include in bounds

iMaxBoundsCount Maximum number of trapezoids that can be output.

iBoundsType Specifies the type of origin used

oBoundsCount The number of trapezoids returned.

Returns:

XSTTrapezoid array

4.14.1.4 XSTStyledGlyph* XSTGlyphVectorGetGlyphs (Display * *dpy*, XSTGlyphVector *iGlyph*, int *iPosition*, int * *ioGlyphCount*)

Returns the list of styled glyphs associated with the STGlyphVector.

This function allocates space for the array of styled glyphs. It is the users responsibility to free this array.

Parameters:

dpy X Display pointer

iGlyph an XSTGlyphVector XID.

iPosition Index position of first glyph to get.

ioGlyphCount Number of glyphs to get and number in the returned array.

Returns:

XSTStyledGlyph array

4.14.1.5 XSTRectangle* XSTGlyphVectorMeasure (Display * *dpy*, XSTGlyphVector *iGlyph*, int *iPosition*, int *iGlyphCount*)

Returns the rectangular box containing the specified glyphs.

Parameters:

dpy X Display pointer

iGlyph an GlyphVector XID.

iPosition Index position of first glyph to measure

iGlyphCount Number of glyphs in vector to measure

Returns:

XSTRectangle

4.14.1.6 XSTGlyphVector XSTGlyphVectorNew (Display * *dpy*, XSTTypeEnv *iEnv*, XSTStyledGlyph * *iGlyphs*, int *iGlyphCount*)

Creates a new STGlyphVector object and returns the XSTGlyphVector XID referring to it.

An STGlyphVector is an array of styled glyphs and positions.

Parameters:

dpy X Display pointer

iText an XSTTypeEnv XID.

iGlyphs XSTStyledGlyph array

iGlyphCount Number of items in the array.

Returns:

XSTGlyphVector XID

4.14.1.7 XSTGlyphVector XSTGlyphVectorNewCopy (Display * *dpy*, XSTGlyphVector *iGlyph*)

Creates a copy of the STGlyphVector and returns the new XSTGlyphVector XID referring to it.

Parameters:

dpy X Display pointer

iGlyph an XSTGlyphVector XID.

Returns:

XSTGlyphVector XID

4.14.1.8 void XSTGlyphVectorRender (Display * *dpy*, Drawable *d*, GC *gc*, XSTGlyphVector *iGlyph*, int *iPosition*, int *iGlyphCount*)

Draws the glyphs on the specified drawable.

Parameters:

dpy: X Display pointer

d Drawable to draw on.

gc: X GC pointer

iGlyph: an GlyphVector XID.

iPosition: Index position for first glyph to draw

iGlyphCount: Number of glyphs to draw

Returns:

none

4.14.1.9 void XSTGlyphVectorReplaceGlyphs (Display * *dpy*, XSTGlyphVector *iGlyph*, int *iPosition*, int *iGlyphCount*, XSTStyledGlyph * *iGlyphs*, int *iGlyphArrayCount*)

Replaces a specified range of glyphs with a supplied array of styled glyphs.

Parameters:

dpy X Display pointer

iGlyph an GlyphVector XID.

iPosition Index position of first glyph to replace.

iGlyphCount Number of glyphs in vector to replace.

iGlyphs Array of glyphs to use as replacements.

iGlyphArrayCount Number of glyphs in new array.

Returns:
none

4.14.1.10 void XSTGlyphVectorSetGlyphs (Display * *dpy*, XSTGlyphVector *iGlyph*, XSTStyledGlyph * *iGlyphs*, int *iGlyphCount*)

Sets the array of glyphs associated with the STGlyphVector.

This will effectively delete any other glyphs previously associated with the vector.

Parameters:
dpy X Display pointer

iGlyph an GlyphVector XID.

iGlyphs Glyph array

iGlyphCount Number of glyphs in the array

Returns:
none

4.15 XSTLayoutEngine functions

Functions

- XSTTag XSTLayoutEngineGetInfo (Display *dpy, XSTTypeEnv iEnv, XSTLayoutEngine iLayoutEngine, uint32_t *oVersion, const char **oShortName, const char **oLongName, const char **oNotice, XSTLayoutEngineFlags *oLEFlags)

  Retrieves general information from the scaler.

4.15.1 Function Documentation

4.15.1.1 XSTTag XSTLayoutEngineGetInfo (Display * *dpy*, XSTTypeEnv *iEnv*, XSTLayoutEngine *iLayoutEngine*, uint32_t * *oVersion*, const char ** *oShortName*, const char ** *oLongName*, const char ** *oNotice*, XSTLayoutEngineFlags * *oLEFlags*)

Retrieves general information from the scaler.

Parameters:

dpy X Display value

iEnv XSTTypeEnv XID of the STTypeEnv object to use

iLayoutEngine LayoutEngine ID

oVersion LayoutEngine version

oShortName Short scaler name - up to 16 characters

oLongName Free-form scaler name

oNotice Free-form copyright notice

oLEFlags LayoutEngine flags

Returns:

scaler tag.

4.16 Enumerated Type

Typedefs

- typedef enum __XSTAlphaMask XSTAlphaMask

  XSTAlphaMask is an enumerated type of the different alpha values that can be set by the user application.

Enumerations

- enum __XSTAlphaMask { **mAlphaNotSet** = 0x00000000, **mTextAlpha** = 0x00000001, **mHighlightAlpha** = 0x00000002, **mUnderlineAlpha** = 0x00000004, **mStrikeThroughAlpha** = 0x00000008 }

  XSTAlphaMask is an enumerated type of the different alpha values that can be set by the user application.

5 Standard Type Services Class Documentation

5.1 FMatrix Struct Reference

16.16 transformation matrix with tx, ty.

`#include <sttypes.h>`

Public Attributes

- F16Dot16 xx
  transformation matrix value.
- F16Dot16 xy
  transformation matrix value.
- F16Dot16 yx
  transformation matrix value.
- F16Dot16 yy
  transformation matrix value.
- F16Dot16 tx
  transformation matrix value for x translation.
- F16Dot16 ty
  transformation matrix value for y translation.

5.1.1 Detailed Description

16.16 transformation matrix with tx, ty.

The documentation for this struct was generated from the following file:

- sttypes.h

5.2 NameRecord Struct Reference

Expanded STNameTag which includes the string along with the tag desciptor.

`#include <sttypes.h>`

Public Attributes

- uint16 platformID
  Platform ID.
- uint16 encodingID
  Platform-specific encoding ID.
- uint16 languageID
  Language ID.
- uint16 nameID
  Name ID.
- uint16 slen
  String length in bytes.
- byte * sptr
  Pointer to string data (not zero-terminated!).

5.2.1 Detailed Description

Expanded STNameTag which includes the string along with the tag desciptor.

The documentation for this struct was generated from the following file:

- sttypes.h

5.3 STBezierCurve Struct Reference

A Bezier curve.

`#include <sttypes.h>`

Public Attributes

- double x0
  ???
- double y0

???

- double x1
  ???
- double y1
  ???
- double x2
  ???
- double y2
  ???
- double x3
  ???
- double y3
  ???

5.3.1 Detailed Description

A Bezier curve.

The documentation for this struct was generated from the following file:

- sttypes.h

5.4 STBitBltRec Struct Reference

Private data sent to bitblt function.

`#include <sttypes.h>`

Public Attributes

- STRenderingParams **rparams**
- int x
  X coordinate to render at.

- int y
  Y coordinate to render at.
- int w
  width to render.
- int h
  height to render.
- int sx
  source X within passed data.
- int sy
  source Y within passed data.
- int bytesPerRow
  bytes per row of passed data.
- int highlighted
  is text highlighted.

5.4.1 Detailed Description

Private data sent to bitblt function.

The documentation for this struct was generated from the following file:

- sttypes.h

5.5 STBSpline Struct Reference

A B-spline.

`#include <sttypes.h>`

Public Attributes

- double x0
  ???

- double y0
  ???
- double x1
  ???
- double y1
  ???
- double x2
  ???
- double y2
  ???

5.5.1 Detailed Description

A B-spline.

The documentation for this struct was generated from the following file:

- sttypes.h

5.6 STCaret Struct Reference

Defines a text caret.

`#include <sttypes.h>`

Public Attributes

- double ax
  X Origin of the caret.
- double ay
  Y Origin of the caret.
- double bx
  X End Point of the caret.

- double by

  Y End point of the caret.

5.6.1 Detailed Description

Defines a text caret.

The documentation for this struct was generated from the following file:

- sttypes.h

5.7 stdevice Struct Reference

STDevice structure.

`#include <sttypes.h>`

Public Attributes

- STDeviceType devType

  Class type.

- STMatrix matrix

  Current transformation matrix.

5.7.1 Detailed Description

STDevice structure.

The documentation for this struct was generated from the following file:

- sttypes.h

5.8 STExtLineMetrics Struct Reference

STExtLineMetrics struct.

`#include <sttypes.h>`

Public Attributes

- double ascent
  Ascent, positive direction - up.
- double descent
  Descent, positive direction - down.
- double leading
  Leading, if > 0 - make lines farther apart, if < 0 makes lines closer.
- double width
  Text width in horizontal mode or height in vertical.

5.8.1 Detailed Description

STExtLineMetrics struct.

The documentation for this struct was generated from the following file:

- sttypes.h

5.9 STFontMetrics Struct Reference

STFontMetrics struct.

`#include <sttypes.h>`

Public Attributes

- double hAscent
  horizontal ascender.
- double hDescent
  horizontal descender.
- double hLeading
  horizontal leading.
- double vAscent

vertical ascender.

- double vDescent
  vertical descender.
- double vLeading
  vertical leading.
- double italicAngle
  italic angle in radians.
- double xMin
  top left corner of bounding box.
- double yMin
  top left corner of bounding box.
- double xMax
  bottom right corner of bounding box.
- double yMax
  bottom right corner of bounding box.
- int widthClass
  value from 1 to 9.
- int weightClass
  value from 100 to 900.

5.9.1 Detailed Description

STFontMetrics struct.

The documentation for this struct was generated from the following file:

- sttypes.h

5.10 STGlyphMetrics Struct Reference

???

`#include <sttypes.h>`

Public Attributes

- F16Dot16 xAdvH
  X Advance vector for horizontal writing mode.
- F16Dot16 yAdvH
  Y Advance vector for horizontal writing mode.
- F16Dot16 xAdvV
  X Advance vector for vertical writing mode.
- F16Dot16 yAdvV
  Y Advance vector for vertical writing mode.
- F16Dot16 topH
  top Vector from glyph origin to top-left corner in horisontal mode.
- F16Dot16 leftH
  left Vector from glyph origin to top-left corner in horisontal mode.
- F16Dot16 topV
  top Vector from glyph origin to top-left corner in vertical mode.
- F16Dot16 leftV
  left Vector from glyph origin to top-left corner in vertical mode.

5.10.1 Detailed Description

???

The documentation for this struct was generated from the following file:

- sttypes.h

5.11 stgraphics Struct Reference

STGraphics structure.

`#include <sttypes.h>`

Public Attributes

- STDevice dev
  Device associated with Graphics.
- STOutputMode outputMode
  Output Mode and Format values.
- STRGBAColor textColor
  Color of unhighlighted text.
- STRGBAColor highLightTextColor
  Color of highlighted text.
- STRGBAColor highLightBackgroundColor
  Color of highlight background.
- STRGBAColor strikeThroughColor
  Color of strike through.
- STRGBAColor highlightStrikeThroughColor
  Color of highlighted strike through.
- STRGBAColor underlineColor
  Color of underline.
- STRGBAColor highlightUnderlineColor
  Color of highlighted underline.
- STRGBAColor underlineColor2
  Color of second underline.
- STRGBAColor highlightUnderlineColor2
  Color of highlighted second underline.
- STGraphicsMask attributesSet
  values that have been changed by user.

5.11.1 Detailed Description

STGraphics structure.

The documentation for this struct was generated from the following file:

- sttypes.h

5.12 STKernData Struct Reference

???

`#include <sttypes.h>`

Public Attributes

- double xh
  X Kerning vector for horizontal writing mode.
- double yh
  Y Kerning vector for horizontal writing mode.
- double xv
  X Kerning vector for vertical writing mode.
- double yv
  Y Kerning vector for vertical writing mode.

5.12.1 Detailed Description

???

The documentation for this struct was generated from the following file:

- sttypes.h

5.13 STLineMetrics Struct Reference

STLineMetrics struct.

`#include <sttypes.h>`

Public Attributes

- double ascent

  Ascent, positive direction - up.

- double descent

  Descent, positive direction - down.

- double leading

  Leading, if > 0 - make lines farther apart, if < 0 makes lines closer.

5.13.1 Detailed Description

STLineMetrics struct.

The documentation for this struct was generated from the following file:

- sttypes.h

5.14 STLineSeg Struct Reference

A line.

`#include <sttypes.h>`

Public Attributes

- double x0

  X Coordinate value start point of the line.

- double y0

  Y Coordinate value start point of the line.

- double x1

X Coordinate value end point of the line.

- double y1

  Y Coordinate value end point of the line.

5.14.1 Detailed Description

A line.

The documentation for this struct was generated from the following file:

- sttypes.h

5.15 STLookupTable Struct Reference

???

`#include <sttypes.h>`

Public Attributes

- int n

  Number of entries in c array.

- uint32 c [256]

  ???

5.15.1 Detailed Description

???

The documentation for this struct was generated from the following file:

- sttypes.h

5.16 STMatrix Struct Reference

Transformation matrix.

`#include <sttypes.h>`

Public Attributes

- double xx
  transformation matrix value.
- double xy
  transformation matrix value.
- double yx
  transformation matrix value.
- double yy
  transformation matrix value.
- double tx
  transformation matrix value for x translation.
- double ty
  transformation matrix value for y translation.

5.16.1 Detailed Description

Transformation matrix.

x' = xx ∗ x + xy ∗ y + tx y' = yx ∗ x + yy ∗ y + ty

The documentation for this struct was generated from the following file:

- sttypes.h

5.17 STNameTag Struct Reference

STNameTag struct.

`#include <sttypes.h>`

Public Attributes

- uint16 platformID
  TrueType platform ID.
- uint16 encodingID
  TrueType encoding ID.
- uint16 languageID
  TrueType language ID.
- uint16 nameID
  TrueType name ID.

5.17.1 Detailed Description

STNameTag struct.

5.17.2 Member Data Documentation

5.17.2.1 uint16 STNameTag::encodingID

TrueType encoding ID.

See TT_ENCODING_

5.17.2.2 uint16 STNameTag::languageID

TrueType language ID.

See TT_LANGUAGE_

5.17.2.3 uint16 STNameTag::nameID

TrueType name ID.

See TT_NAME_

5.17.2.4 uint16 STNameTag::platformID

TrueType platform ID.

See TT_PLATFORM_

The documentation for this struct was generated from the following file:

- sttypes.h

5.18 STPath Struct Reference

A path.

`#include <sttypes.h>`

Public Attributes

- int count
  Number of elements in path.
- STPathElement * type
  Array of element types in path.
- void * element
  ???

5.18.1 Detailed Description

A path.

The documentation for this struct was generated from the following file:

- sttypes.h

5.19 STPoint Struct Reference

A point.

`#include <sttypes.h>`

Public Attributes

- double x
  X Coordinate value.
- double y
  Y Coordinate value.

5.19.1 Detailed Description

A point.

The documentation for this struct was generated from the following file:

- sttypes.h

5.20 STRasterDeviceStruct Struct Reference

STRasterDevice is a class that specifies raster devices.

`#include <sttypes.h>`

Public Attributes

- STDeviceType devType
  fRasterDevice.
- STMatrix matrix
  transformation matrix associated with device.
- uint32 width
  Width.
- uint32 height
  Heihgt.
- uint16 ppix
  Horizontal points per inch resolution.
- uint16 ppiy

Vertical points per inch resolution.

- uint16 bpp
  bits per pixel.
- void * p
  private data.
- BitBltF bitblt
  blit function.
- HighlightF highlight
  blit function.
- UnderlineF underline
  blit function.
- StrikeThroughF strikethrough
  blit function.

5.20.1 Detailed Description

STRasterDevice is a class that specifies raster devices.

The documentation for this struct was generated from the following file:

- sttypes.h

5.21 STRectangle Struct Reference

Rectangular bounding box.

`#include <sttypes.h>`

Public Attributes

- double ax
  One corner of the bounding box.
- double ay

One corner of the bounding box.

- double bx

  Opposite corner of the bounding box.

- double by

  Opposite corner of the bounding box.

5.21.1 Detailed Description

Rectangular bounding box.

The documentation for this struct was generated from the following file:

- sttypes.h

5.22 STTrapezoid Struct Reference

Defines a trapezoid for bounding boxes.

`#include <sttypes.h>`

Public Attributes

- double urx

  Upper-right X coordinate.

- double ury

  Upper-right Y coordinate.

- double lrx

  Lower-right X coordinate.

- double lry

  Lower-right Y coordinate.

- double ulx

  Upper-left X coordinate.

- double uly

Upper-left Y coordinate.

- double llx
  Lower-left X coordinate.
- double lly
  Lower-left Y coordinate.

5.22.1 Detailed Description

Defines a trapezoid for bounding boxes.

The documentation for this struct was generated from the following file:

- sttypes.h

5.23 STVectorDeviceStruct Struct Reference

STVectorDevice is a class that specifies vector devices.

`#include <sttypes.h>`

Public Attributes

- STDeviceType devType
  fVectorDevice.
- STMatrix matrix
  transformation matrix associated with device.
- uint16 ppix
  Horisontal resolution for hinted outlines.
- uint16 ppiy
  Vertical resolution for hinted outlines.
- void * p
  private data.
- CopyOutlineF copyoutl

vector function.

5.23.1 Detailed Description

STVectorDevice is a class that specifies vector devices.

The documentation for this struct was generated from the following file:

- sttypes.h

5.24 TMatrix Struct Reference

16.16 tranformation matrix.

`#include <sttypes.h>`

Public Attributes

- F16Dot16 xx
  transformation matrix value.
- F16Dot16 xy
  transformation matrix value.
- F16Dot16 yx
  transformation matrix value.
- F16Dot16 yy
  transformation matrix value.

5.24.1 Detailed Description

16.16 tranformation matrix.

The documentation for this struct was generated from the following file:

- sttypes.h

5.25 TPoint Struct Reference

16.16 point.

`#include <sttypes.h>`

Public Attributes

- F16Dot16 x
  Coordinate value.
- F16Dot16 y
  Coordinate value.

5.25.1 Detailed Description

16.16 point.

The documentation for this struct was generated from the following file:

- sttypes.h

5.26 uint16pair Struct Reference

???

`#include <sttypes.h>`

Public Attributes

- uint16 s
  ???
- uint16 d
  ???

5.26.1 Detailed Description

???

The documentation for this struct was generated from the following file:

- sttypes.h

5.27 XSTAlphaStruct Struct Reference

XSTAlpha is a structure containing the min and max alpha values to associate with a particular XST drawable object.

`#include <Xst_typedef.h>`

Public Attributes

- double min_alpha

  minimum alpha value.

- double max_alpha

  maximum alpha value.

5.27.1 Detailed Description

XSTAlpha is a structure containing the min and max alpha values to associate with a particular XST drawable object.

The valid objects are: Text, Highlight, Underline, and Strikethrough. The range for the values goes from 0.0 to 1.0. Min must be less than max.

5.27.2 Member Data Documentation

5.27.2.1 double XSTAlphaStruct::max_alpha

maximum alpha value.

< 1.0 is semi-transparent

The documentation for this struct was generated from the following file:

- Xst_typedef.h

5.28 XSTBaselines Struct Reference

XSTBaselines contains baseline offsets from the default baseline.

`#include <Xst_typedef.h>`

Public Attributes

- Rational base [32]
  distance from the default baseline in points.

5.28.1 Detailed Description

XSTBaselines contains baseline offsets from the default baseline.

The documentation for this struct was generated from the following file:

- Xst_typedef.h

5.29 XSTCaret Struct Reference

XSTCaret contains two end points of the caret line.

`#include <Xst_typedef.h>`

Public Attributes

- Rational ax
  Upper end point X value of caret.
- Rational ay
  Upper end point Y value of caret.
- Rational bx
  Lower end point X value of caret.

- Rational by

  Lower end point Y value of caret.

5.29.1 Detailed Description

XSTCaret contains two end points of the caret line.

The documentation for this struct was generated from the following file:

- Xst_typedef.h

5.30 XSTNameTagStruct Struct Reference

XSTNameTagStruct is a structure containing the four elements that comprise a TrueType name tag.

`#include <Xst_typedef.h>`

Public Attributes

- uint16_t platformID

  TrueType platformID.

- uint16_t encodingID

  TrueType encodingID.

- uint16_t languageID

  TrueType languageID.

- uint16_t nameID

  TrueType nameID.

5.30.1 Detailed Description

XSTNameTagStruct is a structure containing the four elements that comprise a TrueType name tag.

5.30.2 Member Data Documentation

5.30.2.1 uint16_t XSTNameTagStruct::encodingID

TrueType encodingID.

See TT_ENCODING_* in sttypes.h

5.30.2.2 uint16_t XSTNameTagStruct::languageID

TrueType languageID.

See TT_LANGUAGE_* in sttypes.h

5.30.2.3 uint16_t XSTNameTagStruct::nameID

TrueType nameID.

See TT_NAME_* in sttypes.h

5.30.2.4 uint16_t XSTNameTagStruct::platformID

TrueType platformID.

See TT_PLATFORM_* in sttypes.h

The documentation for this struct was generated from the following file:

- Xst_typedef.h

6 Standard Type Services File Documentation

6.1 sttypes.h File Reference

STSF Type definitions.

```
#include <sys/types.h>
```

Compounds

- struct FMatrix

16.16 transformation matrix with tx, ty.

- struct **FUnitBBox**
- struct **FUnitGlyphMetrics**
- struct NameRecord

  Expanded STNameTag which includes the string along with the tag desciptor.

- struct STBezierCurve

  A Bezier curve.

- struct STBitBltRec

  Private data sent to bitblt function.

- struct STBSpline

  A B-spline.

- struct STCaret

  Defines a text caret.

- struct stdevice

  STDevice structure.

- struct STExtLineMetrics

  STExtLineMetrics struct.

- struct STFontMetrics

  STFontMetrics struct.

- struct STGlyphMetrics

  ???

- struct stgraphics

  STGraphics structure.

- struct STKernData

  ???

- struct STLineMetrics

  STLineMetrics struct.

- struct STLineSeg

  A line.

- struct STLookupTable

  ???

- struct STMatrix

  Transformation matrix.

- struct STNameTag

  STNameTag struct.

- struct **STNameTagsRec**
- struct STPath

  A path.

- struct STPoint

  A point.

- struct STRasterDeviceStruct

  STRasterDevice is a class that specifies raster devices.

- struct STRectangle

  Rectangular bounding box.

- struct **STRenderingParams**
- struct **STStyledGlyph**
- struct STTrapezoid

  Defines a trapezoid for bounding boxes.

- struct STVectorDeviceStruct

  STVectorDevice is a class that specifies vector devices.

- struct TMatrix

  16.16 tranformation matrix.

- struct TPoint

  16.16 point.

- struct uint16pair

  ???

Defines

- #define **GM_NBUCKETS** 13
- #define **RM_NBUCKETS** 13
- #define **DEFAULT_FONT_SIZE** 12.0
- #define **BITMAP_BUFFER_SIZE** 131072
- #define **FI_CACHE_COUNT** 4000
- #define **CScalerNotSet** 0
- #define ST_DEFAULT_SCALER 0

  Default scaler ID.

- #define ST_BEGINNING 0

  STPosition: index of the first character or glyph of an object.

- #define ST_END 16777216

  STCount: Index of the last character or glyph.

- #define ST_ALL 16777216

  STCount or STCharCount: Value that specifies all remaining characters or glyphs.

- #define **STGLYPH_SIZE** 16
- #define TRUE ((STBoolean) (1 == 1))

  Definition of TRUE.

- #define FALSE ((STBoolean) (1 == 0))

  Definition of FALSE.

- #define CWidthNotSet -1.0

  XXX DEPRECATED.

- #define ST_FROM_TEXT_BEGINNING 0

  ???

- #define ST_TO_TEXT_END -1

  ???

- #define ST_ALL_TEXT -1

  ???

- #define ST_SM_FONTID 1

  Change or changed FontID.

- #define ST_SM_SIZE (1 << 1)

Change or changed Size.

- #define ST_SM_LANGUAGE (1 << 2)
  Change or changed Language.
- #define ST_SM_BASELINE (1 << 3)
  Change or changed Baseline.
- #define ST_SM_EFFECTS (1 << 5)
  Change or changed Style Effects.
- #define ST_SM_STRIKETHROUGH (1 << 6)
  Change or changed Strike Through.
- #define ST_SM_UNDERLINE (1 << 7)
  Change or changed Underline.
- #define ST_SM_SCALER (1 << 8)
  Change or changed Scaler.
- #define ST_SM_HINTINGMODE (1 << 9)
  Change or changed Hinting mode.
- #define ST_SM_SBITSMODE (1 << 10)
  Change or changed Use Embedded Bitmaps.
- #define ST_SM_LAYOUTENGINE (1 << 11)
  Change or changed layout engine.
- #define ST_SM_ALL 0xFFFFFFFF
  Change or changed All applicable fields.
- #define STMaxStyleBitMask ST_SM_LAYOUTENGINE
  Largest STStyleMask value.
- #define ST_GM_DEVICE 1
  Change device STGraphics is to use.
- #define ST_GM_OUTPUTMODE (1 << 1)
  Change output mode, LCD, B&W, etc.
- #define ST_GM_COLOR_TEXT (1 << 2)

Normal text color.

- #define ST_GM_COLOR_HLTEXT (1 << 3)
  Highlighted text color.
- #define ST_GM_COLOR_HLBG (1 << 4)
  Highlighted background color.
- #define ST_GM_COLOR_ST (1 << 5)
  Strikethrough color.
- #define ST_GM_COLOR_HLST (1 << 6)
  Highlighted strikethrough color.
- #define ST_GM_COLOR_UL1 (1 << 7)
  First underline color.
- #define ST_GM_COLOR_HLUL1 (1 << 8)
  Highlighted first underline color.
- #define ST_GM_COLOR_UL2 (1 << 9)
  Second underline color.
- #define ST_GM_COLOR_HLUL2 (1 << 10)
  Highlighted second underline color.
- #define ST_GM_ALL 0xFFFFFFFF
  Change all graphics fields.
- #define ST_TM_DIRECTION 1
  Change direction of text flow.
- #define ST_TM_JUSTIFICATION (1 << 1)
  Change text justification.
- #define ST_TM_FLUSHFACTOR (1 << 2)
  Change text flush factor.
- #define ST_TM_LANGUAGE (1 << 3)
  Change text language.
- #define ST_TM_FALLBACKPOLICY (1 << 4)

Change text fallback policy.

- #define ST_TM_LINEMETRICS (1 << 5)

  Change text imposed line metrics.

- #define ST_TM_BASELINES (1 << 6)

  Change text baseline.

- #define ST_TM_TEXTWIDTH (1 << 7)

  Change text imposed metric width.

- #define ST_TM_ALL 0xFFFFFFFF

  Change all applicable text fields.

- #define ST_SE_LIGATURESPLIT 1

  Specifies if caret can be positioned inside ligatures.

- #define ST_SE_SLANTEDCARET (1 << 1)

  Specifies if caret and selections should be parallel to the angle of the text or perpendicular to the baseline.

- #define ST_SE_OPTALIGNMENT (1 << 2)

  Automatic adjustment of optical positions at the beginning and end of lines.

- #define ST_SE_NO_CROSSKERN (1 << 3)

  Disabled cross-stream kerning defined in a font.

- #define ST_SE_NO_KERN (1 << 4)

  Disable kerning.

- #define ST_SE_EMBOLDEN (1 << 5)

  Alogirthmically embolden the font.

- #define ST_SE_ITALICIZE (1 << 6)

  Algorithmically italicezes the font.

- #define ST_SE_CONDENSED (1 << 7)

  Decreases the horizontal distance between all glyphs.

- #define ST_SE_EXTENDED (1 << 8)

  Increases the horizontal distance between all glyphs.

- #define ST_SE_VERTICALTEXT (1 << 9)

Specifies vertical glyph orientation.

- #define ST_SE_FORCEHANGING (1 << 10)
  Should hanging glyphs extend into the margins.
- #define ST_FL_SYSTEM 1
  Search the system-specific place (/usr/X11/lib/fonts).
- #define ST_FL_LOCAL (1 << 1)
  Search the local folder like (/usr/local/lib/fonts).
- #define ST_FL_USER (1 << 2)
  Search the user folder (~/.fonts).
- #define ST_STRIKETHROUGH_SINGLE 1
  Draws a horisontal line through text.
- #define ST_STRIKETHROUGH_DOUBLE 2
  Draws two horisontal lines through text.
- #define ST_STRIKETHROUGH_THICK 0x80000000
  Makes it thick.
- #define ST_UNDERLINE_SINGLE 1
  Draws a single line parallel to the baseline.
- #define ST_UNDERLINE_DOT (1 << 1)
  Draws a line of dots.
- #define ST_UNDERLINE_DASH (1 << 2)
  Draws a line of dashes.
- #define ST_UNDERLINE_LONGDASH (1 << 3)
  Draws a line of long dashes.
- #define ST_UNDERLINE_DOTDASH (1 << 4)
  Draws a line of dot-dash pairs.
- #define ST_UNDERLINE_WAVE (1 << 5)
  Underlines with a wave.
- #define **ST_UNDERLINE_WAVE_MS** 33
- #define **ST_UNDERLINE_SHORT_SQUARE_WAVE** 34

- #define **ST_UNDERLINE_LONG_SQUARE_WAVE** 35
- #define ST_UNDERLINE_DOUBLE (1 << 14)

  ORing this flag with the previously selected underline style doubles the underline.

- #define ST_UNDERLINE_THICK (1 << 15)

  ORing this flag with the previously selected underline style makes it thick.

- #define ST_COMBINE_UNDERLINE(first, second) ((second) << 16 | (first))

  Use this macro to combine two underline values into a single bitmask to set for the style.

- #define **ST_GET_FIRST_UNDERLINE**(first) (((first) & 0x0000FFFF));
- #define **ST_GET_SECOND_UNDERLINE**(first) (((first) & 0xFFFF0000) >> 16);
- #define **ST_GET_FIRST_UNDERLINE_TYPE**(first) (((first) & 0x00003FFF));
- #define **ST_GET_SECOND_UNDERLINE_TYPE**(first) (((first) & 0x3FFF0000) >> 16);
- #define ST_OM_MASK 0x0000FFFF

  Output mode portion of mask.

- #define ST_OF_MASK 0xFFFF0000

  Output format portion of mask.

- #define ST_OM_RASTER_MONO 1

  Two-color rendering.

- #define ST_OM_RASTER_GRAYSCALE 2

  Anti-aliased mode.

- #define ST_OM_RASTER_GREYSCALE ST_OUTPUT_RASTER_GRAYSCALE

  Anti-aliased mode.

- #define ST_OM_RASTER_LCD 3

  Optimized for LCD output.

- #define ST_OM_RASTER_TV 4

  Optimized for TV output.

- #define ST_OM_VECTOR 32

  Vector based output.

- #define ST_OF_SUPPRESS_FRACTIONAL_METRICS 0x00010000
  Do not use fractional metrics.
- #define ST_OF_SUPPRESS_FRACTIONAL_DELTAS 0x00020000
  Do not use fractional deltas.
- #define ST_OF_HINT_OUTLINES 0x00040000
  Grid-fitted outlines.
- #define ST_IGNORE_HINTS 1
  Ignore font hints.
- #define ST_APPLY_HINTS 2
  Apply font hints.
- #define ST_AUTOHINT 3
  Turn autohinting on.
- #define ST_IGNORE_SBITS 1
  Do not use embedded glyph bitmaps.
- #define ST_USE_SBITS 2
  Use embedded glyph bitmaps.
- #define ST_BASELINE_ROMAN 0
  The standard Roman baseline.
- #define ST_BASELINE_CENTERED 1
  The baseline used in ideographic scripts like Chinese, Japanese and Korean.
- #define ST_BASELINE_HANGING 2
  The baseline used in Devanagiri and similar scripts.
- #define ST_BASELINE_LOWCENTERED 3
  Similar to STCenteredBaseline but with the glyphs lowered.
- #define ST_BASELINE_MATH 4
  The baseline for setting mathematics.
- #define ST_BASELINE_LAST ST_BASELINE_MATH
  The ordinal number of the last defined baseline.

- #define ST_BOUNDS_CARET_ORIGINS 0

  The caret origin is halfway between two characters.

- #define ST_BOUNDS_DEVICE_ORIGINS 1

  Glyph origins in device space.

- #define ST_BOUNDS_FRACTIONAL_ORIGINS 2

  Glyph origins in user space.

- #define ST_FB_NOSUBSTITUTION 0

  Do not substitute fonts.

- #define ST_FB_USEFONTLIST 1

  Search only fonts on a substitution font list.

- #define ST_FB_SEARCHALL 2

  Search all available fonts.

- #define ST_CM_BYCHARACTER 0

  Move caret one character at a time.

- #define ST_CM_BYWORD 1

  Move caret one word at a time.

- #define ST_CM_BYUNICODECLUSTER 2

  Move caret one unicode cluster at a time.

- #define ST_CD_LEFT 0

  Move caret left.

- #define ST_CD_RIGHT 1

  Move caret right.

- #define ST_CD_PREVIOUS 2

  Move caret to the previous movement value.

- #define ST_CD_NEXT 3

  Move caret to the next movement value.

- #define ST_TC_INSERTED 0

  Text has been inserted.

- #define ST_TC_REMOVED 1

  Text has been removed.

- #define ST_TC_REPLACED 2

  Text has been replaced with an equal amount of text.

- #define ST_SC_EQUAL 0

  Two styles are equal.

- #define ST_SC_UNEQUAL 1

  Two styles are not equal, nor subsets or superset of each other.

- #define ST_SC_SUBSET 2

  First style is a subset of second style.

- #define ST_SC_SUPERSET 3

  First style is a superset of second style.

- #define ST_LO_GLYPHVECTOR 1

  Treat text as an array of glyph IDs.

- #define ST_UNKNOWN_FONT 0

  Reserved.

- #define ST_TRUETYPE 1

  Classic TrueType.

- #define ST_TYPE1 2

  Type1.

- #define ST_OPENTYPE_TTF 3

  OpenType, TrueType open - TTF.

- #define ST_OPENTYPE_OTF 4

  OpenType - OTF.

- #define ST_TRUETYPE_GX 5

  TrueType with GX extensions.

- #define ST_EOT 6

  Embedded OpenType.

- #define ST_T2K 7
  FontFusion T2K format, Asian and Roman.
- #define ST_PFR 8
  TrueDoc PFR font.
- #define ST_INTELLIFONT 9
  Intellifont.
- #define ST_SPEEDO 10
  Speedo.
- #define ST_SF_SBITS 0x00000001
  scaler supports embedded bitmaps for TrueType and OpenType fonts.
- #define ST_SF_HINTS 0x00000002
  scaler supports hints.
- #define ST_SF_AUTOHINTING 0x00000004
  scaler supports autohinting.
- #define ST_SF_BITMAP 0x00000008
  scaler supports B&W single bit bitmap generation.
- #define ST_SF_GRAYSCALE 0x00000010
  scaler supports single byte antialised alpha-mask generation.
- #define ST_SF_LCDOPTIMIZED 0x00000020
  scaler supports LCD Optimized output.
- #define ST_SF_EMBOLDEN 0x00000040
  scaler supports algorithmic emboldening.
- #define ST_SF_OBLIQUE 0x00000080
  scaler supports algorithmic oblique.
- #define ST_SF_FRACDELTAS 0x00000100
  scaler supports fractional glyph positioning.
- #define ST_LF_AAT 0x00000001
  layout engine supports GX-style tables.

- #define ST_LF_OPENTYPE 0x00000002
  layout engine supports OpenType-style tables.
- #define ST_FT_UNKNOWN 0
  Font type is not known.
- #define ST_FM_UNAVAILABLE 0xFFFFFFFF
  Font is no longer available.
- #define ST_FT_TRUETYPE 1
  Classic TrueType.
- #define ST_FM_TRUETYPE (1 << (ST_FT_TRUETYPE - 1))
  TrueType font mask.
- #define ST_FT_TTC 2
  Microsoft TrueType Collection File.
- #define ST_FM_TTC (1 << (ST_FT_TTC - 1))
  Microsoft TrueType Collection font mask.
- #define ST_FT_TYPE1 3
  Adobe Type1.
- #define ST_FM_TYPE1 (1 << (ST_FT_TYPE1 - 1))
  Adobe Type1 font mask.
- #define ST_FT_OPENTYPE_TTF 4
  OpenType with TrueType glyph procedures.
- #define ST_FM_OPENTYPE_TTF (1 << (ST_FT_OPENTYPE_TTF - 1))
  OpenType with TrueType glyph procedures font mask.
- #define ST_FT_OPENTYPE_OTF 5
  OpenType with Type1 glyph procedures.
- #define ST_FM_OPENTYPE_OTF (1 << (ST_FT_OPENTYPE_OTF - 1))
  OpenType with Type1 glyph procedures font mask.
- #define ST_FT_TRUETYPE_GX 6
  TrueType with GX extensions.

- #define ST_FM_TRUETYPE_GX (1 << (ST_FT_TRUETYPE_GX - 1))
  TrueType with GX extensions font mask.
- #define ST_FT_EOT 7
  Embedded OpenType.
- #define ST_FM_EOT (1 << (ST_FT_EOT - 1))
  Embedded OpenType font mask.
- #define ST_FT_T2K 8
  FontFusion T2K format, Asian and Roman.
- #define ST_FM_T2K (1 << (ST_FT_T2K - 1))
  FontFusion T2K font mask.
- #define ST_FT_PFR 9
  TrueDoc PFR font.
- #define ST_FM_PFR (1 << (ST_FT_PFR - 1))
  TrueDoc PFR font mask.
- #define ST_FT_INTELLIFONT 10
  Intellifont.
- #define ST_FM_INTELLIFONT (1 << (ST_FT_INTELLIFONT - 1))
  Intellifont font mask.
- #define ST_FT_SPEEDO 11
  Speedo font.
- #define ST_FM_SPEEDO (1 << (ST_FT_SPEEDO - 1))
  Speedo font mask.
- #define ST_FI_SBITS 1
  Embedded bitmaps are available for this font.
- #define ST_FI_KERN 2
  Kerning data is available for this font.
- #define ST_RM_INVALID 0
  not a valid rendering mode.

- #define ST_RM_SIMPLEBITMAP 1
  1 bit per pixel bitmap.
- #define ST_RM_ALPHAMASK 2
  Anti-aliased alpha-mask 4 or 8 bits per pixel.
- #define ST_RM_LOOKUPTABLE 3
  Color lookup table, 8 bits per pixel.
- #define ST_RM_THREE_ALPHAMASKS 4
  Anti-aliased alpha-mask 4 or 8 bits per pixel.
- #define ST_DEVICE_RASTER 1
  Identifies STRasterDevice.
- #define ST_DEVICE_VECTOR 2
  Identifies STVectorDevice.
- #define F16DOT16_TO_INT(A) ((A) >> 16)
  Convert F16.16 to int.
- #define F16DOT16_TO_FLOAT(A) ((A) / 65536.0)
  Convert F16.16 to float.
- #define INT_TO_F16DOT16(A) ((A) << 16)
  Convert int to F16.16.
- #define FLOAT_TO_F16DOT16(A) ((F16Dot16) (A * 65536.0))
  Convert float to F16.16.
- #define ST_WEIGHT_THIN 1
  Font weight described as thin.
- #define ST_WEIGHT_EXTRALIGHT 2
  Font weight described as extra light.
- #define ST_WEIGHT_LIGHT 3
  Font weight described as light.
- #define ST_WEIGHT_NORMAL 4
  Font weight described as normal.

- #define ST_WEIGHT_MEDIUM 5
  Font weight described as medium.
- #define ST_WEIGHT_SEMIBOLD 6
  Font weight described as semi bold.
- #define ST_WEIGHT_BOLD 7
  Font weight described as bold.
- #define ST_WEIGHT_EXTRABOLD 8
  Font weight described as extra bold.
- #define ST_WEIGHT_BLACK 9
  Font weight described as black.
- #define ST_WEIGHT_UNKNOWN 10
  Font weight described as unknown.
- #define ST_WIDTH_ULTRA_CONDENSED 1
  Font width ultra condensed.
- #define ST_WIDTH_EXTRA_CONDENSED 2
  Font width extra condensed.
- #define ST_WIDTH_CONDENSED 3
  Font width condensed.
- #define ST_WIDTH_SEMI_CONDENSED 4
  Font width semi condensed.
- #define ST_WIDTH_NORMAL 5
  Font width normal.
- #define ST_WIDTH_SEMI_EXPANDED 6
  Font width semi expanded.
- #define ST_WIDTH_EXPANDED 7
  Font width expanded.
- #define ST_WIDTH_EXTRA_EXPANDED 8
  Font width extra expanded.

- #define ST_WIDTH_ULTRA_EXPANDED 9
  Font width ultra expanded.
- #define ST_WIDTH_UNKNOWN 10
  Font width unknown.
- #define ST_STYLE_NORMAL 1
  Font style normal.
- #define ST_STYLE_ITALIC 2
  Font style italic.
- #define ST_STYLE_OBLIQUE 3
  Font style oblique.
- #define ST_STYLE_INCLINED 4
  Font style inclined.
- #define ST_STYLE_SLANTED 5
  Font style slanted.
- #define ST_STYLE_UNKNOWN 6
  Font style unknown.
- #define ST_PLATFORM_ANY 0xFFFF
  Use any platform ID in font.
- #define ST_ENCODING_ANY 0xFFFF
  Use any encoding ID in font.
- #define ST_LANGUAGE_ANY 0xFFFF
  Use any language ID in font.
- #define ST_NAME_ANY 0xFFFF
  Use any name ID in font.
- #define ST_SCALER_DUMMY 0x444D4D59
  'DMMY' Dummy scaler for testing.
- #define ST_SCALER_FONTFUSION 0x46465553
  'FFUS' Bitstream FontFusion scaler.

- #define ST_SCALER_FREETYPE1 0x46545931
  'FTY1' FreeType version 1 scaler.
- #define ST_SCALER_FREETYPE2 0x46545932
  'FTY2' FreeType version 2 scaler.
- #define ST_SCALER_XATM 0x5841544D
  'XATM' Adobe XATM-based scaler.
- #define ST_LAYOUTENGINE_ICU 0x49435530
  'ICU0' IBM ICU Layout Engine.
- #define TT_PLATFORM_UNICODE 0
  Unicode encoding for platform ID.
- #define TT_PLATFORM_MACINTOSH 1
  Apple Mac encoding for platform ID.
- #define TT_PLATFORM_ISO 2
  ISO encoding for platform ID.
- #define TT_PLATFORM_MICROSOFT 3
  Microsoft encoding for platform ID.
- #define TT_PLATFORM_CUSTOM 4
  Custom encoding for platform ID.
- #define TT_ENCODING_UNICODE_10 0
  Unicode 1.0 semantics.
- #define TT_ENCODING_UNICODE_11 1
  Unicode 1.1 semantics.
- #define TT_ENCODING_UNICODE_ISO10646 2
  ISO 10646:1993 semantics.
- #define TT_ENCODING_UNICODE_20 3
  Unicode 2.0 and onward semantics.
- #define TT_ENCODING_MS_SYMBOL 0
  Microsoft symbol encoding.

- #define TT_ENCODING_MS_UNICODE 1
  Microsoft unicode encoding.
- #define TT_ENCODING_MS_SHIFTJIS 2
  Microsoft shift JIS encoding.
- #define TT_ENCODING_MS_PRC 3
  Microsoft PRC encoding.
- #define TT_ENCODING_MS_BIG5 4
  Microsoft BIG5 encoding.
- #define TT_ENCODING_MS_WANSUNG 5
  Microsoft WANSUNG encoding.
- #define TT_ENCODING_MS_JOHAB 6
  Microsoft JOHAB encoding.
- #define TT_ENCODING_MS_UCS4 10
  Microsoft UCS4 encoding.
- #define TT_NAME_COPYRIGHTNOTICE 0
  Copyright string.
- #define TT_NAME_FONTFAMILY 1
  Font Family name string.
- #define TT_NAME_FONTSUBFAMILY 2
  Font Typeface name string.
- #define TT_NAME_UNIQUEID 3
  Unique ID string.
- #define TT_NAME_FULLFONTNAME 4
  Full Font name string.
- #define TT_NAME_VERSIONSTRING 5
  Version string.
- #define TT_NAME_POSTSCRIPTNAME 6
  Postscript name string.

- #define TT_NAME_TRADEMARK 7
  Trademark string.
- #define TT_NAME_MANUFACTURERNAME 8
  Manufacturer name string.
- #define TT_NAME_DESIGNERNAME 9
  Designer string.
- #define TT_NAME_DESCRIPTION 10
  Description string.
- #define TT_NAME_VENDORURL 11
  Vendor URL string.
- #define TT_NAME_DESIGNERURL 12
  Designer URL string.
- #define TT_NAME_LICENSEDESCRIPTION 13
  License description string.
- #define TT_NAME_LICENSEINFOURL 14
  License info URL string.
- #define TT_NAME_PREFERREDFAMILY 16
  Preferred family name string.
- #define TT_NAME_PREFERREDSUBFAMILY 17
  Preferred family typeface name string.
- #define TT_NAME_COMPATIBLEFULL 18
  Full compatible name string.
- #define TT_NAME_SAMPLETEXT 19
  sample text string.
- #define TT_NAME_POSTSCRIPTCIDNAME 20
  Postscript CID name string.
- #define TT_LANGUAGE_MS_ARABIC_SAUDIARABIA 0x0401
  Microsoft Arabic language ID.

- #define TT_LANGUAGE_MS_ARABIC_IRAQ 0x0801
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_EGYPT 0x0c01
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_LIBYA 0x1001
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_ALGERIA 0x1401
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_MOROCCO 0x1801
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_TUNISIA 0x1c01
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_OMAN 0x2001
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_YEMEN 0x2401
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_SYRIA 0x2801
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_JORDAN 0x2c01
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_LEBANON 0x3001
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_KUWAIT 0x3401
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_UAE 0x3801
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_ARABIC_BAHRAIN 0x3c01
  Microsoft Arabic language ID.

- #define TT_LANGUAGE_MS_ARABIC_QATAR 0x4001
  Microsoft Arabic language ID.
- #define TT_LANGUAGE_MS_BULGARIAN_BULGARIA 0x0402
  Microsoft Bulgarian language ID.
- #define TT_LANGUAGE_MS_CATALAN_SPAIN 0x0403
  Microsoft Catalan language ID.
- #define TT_LANGUAGE_MS_CHINESE_TAIWAN 0x0404
  Microsoft Chinese language ID.
- #define TT_LANGUAGE_MS_CHINESE_PRC 0x0804
  Microsoft Chinese language ID.
- #define TT_LANGUAGE_MS_CHINESE_HONGKONG 0x0c04
  Microsoft Chinese language ID.
- #define TT_LANGUAGE_MS_CHINESE_SINGAPORE 0x1004
  Microsoft Chinese language ID.
- #define TT_LANGUAGE_MS_CHINESE_MACAU 0x1404
  Microsoft Chinese language ID.
- #define TT_LANGUAGE_MS_CZECH_CZECHREPUBLIC 0x0405
  Microsoft Czech language ID.
- #define TT_LANGUAGE_MS_DANISH_DENMARK 0x0406
  Microsoft Danish language ID.
- #define TT_LANGUAGE_MS_GERMAN_GERMANY 0x0407
  Microsoft German language ID.
- #define TT_LANGUAGE_MS_GERMAN_SWITZERLAND 0x0807
  Microsoft German language ID.
- #define TT_LANGUAGE_MS_GERMAN_AUSTRIA 0x0c07
  Microsoft German language ID.
- #define TT_LANGUAGE_MS_GERMAN_LUXEMBOURG 0x1007
  Microsoft German language ID.

- #define TT_LANGUAGE_MS_GERMAN_LIECHTENSTEIN 0x1407
  Microsoft German language ID.
- #define TT_LANGUAGE_MS_GREEK_GREECE 0x0408
  Microsoft Greek language ID.
- #define TT_LANGUAGE_MS_ENGLISH_US 0x0409
  Microsoft English language ID.
- #define TT_LANGUAGE_MS_ENGLISH_UK 0x0809
  Microsoft English language ID.
- #define TT_LANGUAGE_MS_ENGLISH_AUSTRALIA 0x0c09
  Microsoft English language ID.
- #define TT_LANGUAGE_MS_ENGLISH_CANADA 0x1009
  Microsoft English language ID.
- #define TT_LANGUAGE_MS_ENGLISH_NEWZEALAND 0x1409
  Microsoft English language ID.
- #define TT_LANGUAGE_MS_ENGLISH_IRELAND 0x1809
  Microsoft English language ID.
- #define TT_LANGUAGE_MS_ENGLISH_SOUTHAFRICA 0x1c09
  Microsoft English language ID.
- #define TT_LANGUAGE_MS_ENGLISH_JAMAICA 0x2009
  Microsoft English language ID.
- #define TT_LANGUAGE_MS_ENGLISH_CARIBBEAN 0x2409
  Microsoft English language ID.
- #define TT_LANGUAGE_MS_ENGLISH_BELIZE 0x2809
  Microsoft English language ID.
- #define TT_LANGUAGE_MS_ENGLISH_TRINIDAD 0x2c09
  Microsoft English language ID.
- #define TT_LANGUAGE_MS_ENGLISH_ZIMBABWE 0x3009
  Microsoft English language ID.

- #define TT_LANGUAGE_MS_ENGLISH_PHILIPPINES 0x3409

  Microsoft English language ID.

- #define TT_LANGUAGE_MS_SPANISH_SPAIN_TRADITIONALSORT 0x040a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_MEXICO 0x080a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_SPAININTERNATIONALSORT 0x0c0a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_GUATEMALA 0x100a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_COSTARICA 0x140a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_PANAMA 0x180a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_DOMINICANREPUBLIC 0x1c0a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_VENEZUELA 0x200a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_COLOMBIA 0x240a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_PERU 0x280a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_ARGENTINA 0x2c0a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_ECUADOR 0x300a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_CHILE 0x340a

  Microsoft Spanish language ID.

- #define TT_LANGUAGE_MS_SPANISH_URUGUAY 0x380a
  Microsoft Spanish language ID.
- #define TT_LANGUAGE_MS_SPANISH_PARAGUAY 0x3c0a
  Microsoft Spanish language ID.
- #define TT_LANGUAGE_MS_SPANISH_BOLIVIA 0x400a
  Microsoft Spanish language ID.
- #define TT_LANGUAGE_MS_SPANISH_ELSALVADOR 0x440a
  Microsoft Spanish language ID.
- #define TT_LANGUAGE_MS_SPANISH_HONDURAS 0x480a
  Microsoft Spanish language ID.
- #define TT_LANGUAGE_MS_SPANISH_NICARAGUA 0x4c0a
  Microsoft Spanish language ID.
- #define TT_LANGUAGE_MS_SPANISH_PUERTORICO 0x500a
  Microsoft Spanish language ID.
- #define TT_LANGUAGE_MS_FINNISH_FINLAND 0x040b
  Microsoft Finnish language ID.
- #define TT_LANGUAGE_MS_FRENCH_FRANCE 0x040c
  Microsoft French language ID.
- #define TT_LANGUAGE_MS_FRENCH_BELGIUM 0x080c
  Microsoft French language ID.
- #define TT_LANGUAGE_MS_FRENCH_CANADA 0x0c0c
  Microsoft French language ID.
- #define TT_LANGUAGE_MS_FRENCH_SWITZERLAND 0x100c
  Microsoft French language ID.
- #define TT_LANGUAGE_MS_FRENCH_LUXEMBOURG 0x140c
  Microsoft French language ID.
- #define TT_LANGUAGE_MS_FRENCH_MONACO 0x180c
  Microsoft French language ID.

- #define TT_LANGUAGE_MS_HEBREW_ISRAEL 0x040d
  Microsoft Hebrew language ID.
- #define TT_LANGUAGE_MS_HUNGARIAN_HUNGARY 0x040e
  Microsoft Hungarian language ID.
- #define TT_LANGUAGE_MS_ICELANDIC_ICELAND 0x040f
  Microsoft Icelandic language ID.
- #define TT_LANGUAGE_MS_ITALIAN_ITALY 0x0410
  Microsoft Italian language ID.
- #define TT_LANGUAGE_MS_ITALIAN_SWITZERLAND 0x0810
  Microsoft Italian language ID.
- #define TT_LANGUAGE_MS_JAPANESE_JAPAN 0x0411
  Microsoft Japanese language ID.
- #define TT_LANGUAGE_MS_KOREAN_KOREA_EXTENDEDWANSUNG 0x0412
  Microsoft Korean language ID.
- #define TT_LANGUAGE_MS_DUTCH_NETHERLANDS 0x0413
  Microsoft Dutch language ID.
- #define TT_LANGUAGE_MS_DUTCH_BELGIUM 0x0813
  Microsoft Dutch language ID.
- #define TT_LANGUAGE_MS_NORWEGIAN_NORWAY_BOKMAL 0x0414
  Microsoft Norwegian language ID.
- #define TT_LANGUAGE_MS_NORWEGIAN_NORWAY_NYNORSK 0x0814

  Microsoft Norwegian language ID.
- #define TT_LANGUAGE_MS_POLISH_POLAND 0x0415
  Microsoft Polish language ID.
- #define TT_LANGUAGE_MS_PORTUGUESE_BRAZIL 0x0416
  Microsoft Portuguese language ID.
- #define TT_LANGUAGE_MS_PORTUGUESE_PORTUGAL 0x0816

Microsoft Portuguese language ID.

- #define TT_LANGUAGE_MS_ROMANIAN_ROMANIA 0x0418
  Microsoft Romanian language ID.
- #define TT_LANGUAGE_MS_RUSSIAN_RUSSIA 0x0419
  Microsoft Russian language ID.
- #define TT_LANGUAGE_MS_CROATIAN_CROATIA 0x041a
  Microsoft Croatian language ID.
- #define TT_LANGUAGE_MS_SERBIAN_SERBIA_LATIN 0x081a
  Microsoft Serbian language ID.
- #define TT_LANGUAGE_MS_SERBIAN_SERBIA_CYRILLIC 0x0c1a
  Microsoft Serbian language ID.
- #define TT_LANGUAGE_MS_SLOVAK_SLOVAKIA 0x041b
  Microsoft Slovak language ID.
- #define TT_LANGUAGE_MS_ALBANIAN_ALBANIA 0x041c
  Microsoft Albanian language ID.
- #define TT_LANGUAGE_MS_SWEDISH_SWEDEN 0x041d
  Microsoft Swedish language ID.
- #define TT_LANGUAGE_MS_SWEDISH_FINLAND 0x081d
  Microsoft Swedish language ID.
- #define TT_LANGUAGE_MS_THAI_THAILAND 0x041e
  Microsoft Thai language ID.
- #define TT_LANGUAGE_MS_TURKISH_TURKEY 0x041f
  Microsoft Turkish language ID.
- #define TT_LANGUAGE_MS_URDU_PAKISTAN 0x0420
  Microsoft Urdu language ID.
- #define TT_LANGUAGE_MS_INDONESIAN_INDONESIA 0x0421
  Microsoft Indonesian language ID.
- #define TT_LANGUAGE_MS_UKRAINIAN_UKRAINE 0x0422

Microsoft Ukrainian language ID.

- #define TT_LANGUAGE_MS_BELARUSIAN_BELARUS 0x0423
  Microsoft Belarusian language ID.
- #define TT_LANGUAGE_MS_SLOVENIAN_SLOVENIA 0x0424
  Microsoft Slovenian language ID.
- #define TT_LANGUAGE_MS_ESTONIAN_ESTONIA 0x0425
  Microsoft Estonian language ID.
- #define TT_LANGUAGE_MS_LATVIAN_LATVIA 0x0426
  Microsoft Latvian language ID.
- #define TT_LANGUAGE_MS_LITHUANIAN_LITHUANIA 0x0427
  Microsoft Lithuanian language ID.
- #define TT_LANGUAGE_MS_FARSI_IRAN 0x0429
  Microsoft Farsi language ID.
- #define TT_LANGUAGE_MS_VIETNAMESE_VIETNAM 0x042a
  Microsoft Vietnamese language ID.
- #define TT_LANGUAGE_MS_ARMENIAN_ARMENIA 0x042b
  Microsoft Armenian language ID.
- #define TT_LANGUAGE_MS_AZERI_AZERBAIJAN_LATIN 0x042c
  Microsoft Azeri language ID.
- #define TT_LANGUAGE_MS_AZERI_AZERBAIJAN_CYRILLIC 0x082c
  Microsoft Azeri language ID.
- #define TT_LANGUAGE_MS_BASQUE_SPAIN 0x042d
  Microsoft Basque language ID.
- #define TT_LANGUAGE_MS_MACEDONIAN_MACEDONIA 0x042f
  Microsoft Macedonian language ID.
- #define TT_LANGUAGE_MS_AFRIKAANS_SOUTHAFRICA 0x0436
  Microsoft Afrikaans language ID.
- #define TT_LANGUAGE_MS_GEORGIAN_GEORGIA 0x0437

Microsoft Georgian language ID.

- #define TT_LANGUAGE_MS_FAEROESE_FAEROEISLANDS 0x0438
  Microsoft Faeroese language ID.
- #define TT_LANGUAGE_MS_HINDI_INDIA 0x0439
  Microsoft Hindi language ID.
- #define TT_LANGUAGE_MS_MALAY_MALAYSIA 0x043e
  Microsoft Malay language ID.
- #define TT_LANGUAGE_MS_MALAY_BRUNEIDARUSSALAM 0x083e
  Microsoft Malay language ID.
- #define TT_LANGUAGE_MS_KAZAKH_KAZAKSTAN 0x043f
  Microsoft Kazakh language ID.
- #define TT_LANGUAGE_MS_SWAHILI_KENYA 0x0441
  Microsoft Swahili language ID.
- #define TT_LANGUAGE_MS_UZBEK_UZBEKISTAN_LATIN 0x0443
  Microsoft Uzbek language ID.
- #define TT_LANGUAGE_MS_UZBEK_UZBEKISTAN_CYRILLIC 0x0843
  Microsoft Uzbek language ID.
- #define TT_LANGUAGE_MS_TATAR_TATARSTAN 0x0444
  Microsoft Tatar language ID.
- #define TT_LANGUAGE_MS_PUNJABI_INDIA 0x0446
  Microsoft Punjabi language ID.
- #define TT_LANGUAGE_MS_GUJARATI_INDIA 0x0447
  Microsoft Gujarati language ID.
- #define TT_LANGUAGE_MS_TAMIL_INDIA 0x0449
  Microsoft Tamil language ID.
- #define TT_LANGUAGE_MS_TELUGU_INDIA 0x044a
  Microsoft Telugu language ID.
- #define TT_LANGUAGE_MS_KANNADA_INDIA 0x044b

Microsoft Kannada language ID.

- #define TT_LANGUAGE_MS_MARATHI_INDIA 0x044e
  Microsoft Marathi language ID.
- #define TT_LANGUAGE_MS_SANSKRIT_INDIA 0x044f
  Microsoft Sanskrit language ID.
- #define TT_LANGUAGE_MS_KONKANI_INDIA 0x0457
  Microsoft Konkani language ID.
- #define TT_LANGUAGE_MS_KYRGYZ_KYRGYZSTAN 0x0440
  Microsoft Kyrgyz language ID.
- #define TT_LANGUAGE_MS_MONGOLIAN_MONGOLIA_CYRILLIC 0x0450
  Microsoft Mongolian language ID.
- #define TT_LANGUAGE_MS_GALICIAN_SPAIN 0x0456
  Microsoft Galician language ID.
- #define TT_LANGUAGE_MS_SYRIAC_SYRIA 0x045A
  Microsoft Syriac language ID.
- #define TT_LANGUAGE_MS_DIVEHI_MALDIVES 0x0465
  Microsoft Divehi language ID.
- #define TT_ENCODING_APPLE_ROMAN 0
  Apple Roman encoding ID.
- #define TT_ENCODING_APPLE_JAPANESE 1
  Apple Japanese encoding ID.
- #define TT_ENCODING_APPLE_TRADITIONALCHINESE 2
  Apple Chinese encoding ID.
- #define TT_ENCODING_APPLE_KOREAN 3
  Apple Korean encoding ID.
- #define TT_ENCODING_APPLE_ARABIC 4
  Apple Arabic encoding ID.
- #define TT_ENCODING_APPLE_HEBREW 5

Apple Hebrew encoding ID.

- #define TT_ENCODING_APPLE_GREEK 6
  Apple Greek encoding ID.
- #define TT_ENCODING_APPLE_RUSSIAN 7
  Apple Russian encoding ID.
- #define TT_ENCODING_APPLE_RSYMBOL 8
  Apple RSymbol encoding ID.
- #define TT_ENCODING_APPLE_DEVANAGARI 9
  Apple Devanagari encoding ID.
- #define TT_ENCODING_APPLE_GURMUKHI 10
  Apple Gurmukhi encoding ID.
- #define TT_ENCODING_APPLE_GUJARATI 11
  Apple Gujarati encoding ID.
- #define TT_ENCODING_APPLE_ORIYA 12
  Apple Oriya encoding ID.
- #define TT_ENCODING_APPLE_BENGALI 13
  Apple Bengali encoding ID.
- #define TT_ENCODING_APPLE_TAMIL 14
  Apple Tamil encoding ID.
- #define TT_ENCODING_APPLE_TELUGU 15
  Apple Telugu encoding ID.
- #define TT_ENCODING_APPLE_KANNADA 16
  Apple Kannada encoding ID.
- #define TT_ENCODING_APPLE_MALAYALAM 17
  Apple Malayalam encoding ID.
- #define TT_ENCODING_APPLE_SINHALESE 18
  Apple Sinhalese encoding ID.
- #define TT_ENCODING_APPLE_BURMESE 19

Apple Burmese encoding ID.

- #define TT_ENCODING_APPLE_KHMER 20
  Apple Khmer encoding ID.
- #define TT_ENCODING_APPLE_THAI 21
  Apple Thai encoding ID.
- #define TT_ENCODING_APPLE_LAOTIAN 22
  Apple Laotian encoding ID.
- #define TT_ENCODING_APPLE_GEORGIAN 23
  Apple Georgian encoding ID.
- #define TT_ENCODING_APPLE_ARMENIAN 24
  Apple Armenian encoding ID.
- #define TT_ENCODING_APPLE_SIMPLIFIEDCHINESE 25
  Apple Chinese encoding ID.
- #define TT_ENCODING_APPLE_TIBETAN 26
  Apple Tibetan encoding ID.
- #define TT_ENCODING_APPLE_MONGOLIAN 27
  Apple Mongolian encoding ID.
- #define TT_ENCODING_APPLE_GEEZ 28
  Apple Geez encoding ID.
- #define TT_ENCODING_APPLE_SLAVIC 29
  Apple Slavic encoding ID.
- #define TT_ENCODING_APPLE_VIETNAMESE 30
  Apple Vietnamese encoding ID.
- #define TT_ENCODING_APPLE_SINDHI 31
  Apple Sindhi encoding ID.
- #define TT_ENCODING_APPLE_UNINTERPRETED 32
  Apple Uninterpreted encoding ID.
- #define TT_LANGUAGE_APPLE_ENGLISH 0

Apple English language ID.

- #define TT_LANGUAGE_APPLE_FRENCH 1
  Apple French language ID.
- #define TT_LANGUAGE_APPLE_GERMAN 2
  Apple German language ID.
- #define TT_LANGUAGE_APPLE_ITALIAN 3
  Apple Italian language ID.
- #define TT_LANGUAGE_APPLE_DUTCH 4
  Apple Dutch language ID.
- #define TT_LANGUAGE_APPLE_SWEDISH 5
  Apple Swedish language ID.
- #define TT_LANGUAGE_APPLE_SPANISH 6
  Apple Spanish language ID.
- #define TT_LANGUAGE_APPLE_DANISH 7
  Apple Danish language ID.
- #define TT_LANGUAGE_APPLE_PORTUGUESE 8
  Apple Portuguese language ID.
- #define TT_LANGUAGE_APPLE_NORWEGIAN 9
  Apple Norwegian language ID.
- #define TT_LANGUAGE_APPLE_HEBREW 10
  Apple Hebrew language ID.
- #define TT_LANGUAGE_APPLE_JAPANESE 11
  Apple Japanese language ID.
- #define TT_LANGUAGE_APPLE_ARABIC 12
  Apple Arabic language ID.
- #define TT_LANGUAGE_APPLE_FINNISH 13
  Apple Finnish language ID.
- #define TT_LANGUAGE_APPLE_GREEK 14

Apple Greek language ID.

- #define TT_LANGUAGE_APPLE_ICELANDIC 15
  Apple Icelandic language ID.
- #define TT_LANGUAGE_APPLE_MALTESE 16
  Apple Maltese language ID.
- #define TT_LANGUAGE_APPLE_TURKISH 17
  Apple Turkish language ID.
- #define TT_LANGUAGE_APPLE_CROATIAN 18
  Apple Croatian language ID.
- #define TT_LANGUAGE_APPLE_TRADITIONALCHINESE 19
  Apple Chinese language ID.
- #define TT_LANGUAGE_APPLE_URDU 20
  Apple Urdu language ID.
- #define TT_LANGUAGE_APPLE_HINDI 21
  Apple Hindi language ID.
- #define TT_LANGUAGE_APPLE_THAI 22
  Apple Thai language ID.
- #define TT_LANGUAGE_APPLE_KOREAN 23
  Apple Korean language ID.
- #define TT_LANGUAGE_APPLE_LITHUANIAN 24
  Apple Lithuanian language ID.
- #define TT_LANGUAGE_APPLE_POLISH 25
  Apple Polish language ID.
- #define TT_LANGUAGE_APPLE_HUNGARIAN 26
  Apple Hungarian language ID.
- #define TT_LANGUAGE_APPLE_ESTONIAN 27
  Apple Estonian language ID.
- #define TT_LANGUAGE_APPLE_LATVIAN 28

Apple Latvian language ID.

- #define TT_LANGUAGE_APPLE_SAMI 29
  Apple Sami language ID.
- #define TT_LANGUAGE_APPLE_FAROESE 30
  Apple Faroese language ID.
- #define TT_LANGUAGE_APPLE_FARSIPERSIAN 31
  Apple Farsi language ID.
- #define TT_LANGUAGE_APPLE_RUSSIAN 32
  Apple Russian language ID.
- #define TT_LANGUAGE_APPLE_SIMPLIFIEDCHINESE 33
  Apple Chinese language ID.
- #define TT_LANGUAGE_APPLE_FLEMISH 34
  Apple Flemish language ID.
- #define TT_LANGUAGE_APPLE_IRISHGAELIC 35
  Apple Irish language ID.
- #define TT_LANGUAGE_APPLE_ALBANIAN 36
  Apple Albanian language ID.
- #define TT_LANGUAGE_APPLE_ROMANIAN 37
  Apple Romanian language ID.
- #define TT_LANGUAGE_APPLE_CZECH 38
  Apple Czech language ID.
- #define TT_LANGUAGE_APPLE_SLOVAK 39
  Apple Slovak language ID.
- #define TT_LANGUAGE_APPLE_SLOVENIAN 40
  Apple Solvenian language ID.
- #define TT_LANGUAGE_APPLE_YIDDISH 41
  Apple Yiddish language ID.
- #define TT_LANGUAGE_APPLE_SERBIAN 42

Apple Serbian language ID.

- #define TT_LANGUAGE_APPLE_MACEDONIAN 43
  Apple Macedonian language ID.
- #define TT_LANGUAGE_APPLE_BULGARIAN 44
  Apple Bulgarian language ID.
- #define TT_LANGUAGE_APPLE_UKRAINIAN 45
  Apple Ukrainian language ID.
- #define TT_LANGUAGE_APPLE_BYELORUSSIAN 46
  Apple Byelorussian language ID.
- #define TT_LANGUAGE_APPLE_UZBEK 47
  Apple Uzbek language ID.
- #define TT_LANGUAGE_APPLE_KAZAKH 48
  Apple Kazakh language ID.
- #define TT_LANGUAGE_APPLE_CYRILLICAZERBAIJANI 49
  Apple Azerbi language ID.
- #define TT_LANGUAGE_APPLE_ARABICAZERBAIJANI 50
  Apple Azerbi language ID.
- #define TT_LANGUAGE_APPLE_ARMENIAN 51
  Apple Armenian language ID.
- #define TT_LANGUAGE_APPLE_GEORGIAN 52
  Apple Georgian language ID.
- #define TT_LANGUAGE_APPLE_MOLDAVIAN 53
  Apple Moldavian language ID.
- #define TT_LANGUAGE_APPLE_KIRGHIZ 54
  Apple Kirghiz language ID.
- #define TT_LANGUAGE_APPLE_TAJIGI 55
  Apple Tajigi language ID.
- #define TT_LANGUAGE_APPLE_TURKMEN 56

Apple Turkmen language ID.

- #define TT_LANGUAGE_APPLE_MONGOLIAN 57
  Apple Mongolian language ID.
- #define TT_LANGUAGE_APPLE_CYRILLICMONGOLIAN 58
  Apple Mongolian language ID.
- #define TT_LANGUAGE_APPLE_PASHTO 59
  Apple Pashto language ID.
- #define TT_LANGUAGE_APPLE_KURDISH 60
  Apple Kurdish language ID.
- #define TT_LANGUAGE_APPLE_KASHMIRI 61
  Apple Kashmiri language ID.
- #define TT_LANGUAGE_APPLE_SINDHI 62
  Apple Sindhi language ID.
- #define TT_LANGUAGE_APPLE_TIBETAN 63
  Apple Tibetan language ID.
- #define TT_LANGUAGE_APPLE_NEPALI 64
  Apple Nepali language ID.
- #define TT_LANGUAGE_APPLE_SANSKRIT 65
  Apple Sanskrit language ID.
- #define TT_LANGUAGE_APPLE_MARATHI 66
  Apple Marathi language ID.
- #define TT_LANGUAGE_APPLE_BENGALI 67
  Apple Bengali language ID.
- #define TT_LANGUAGE_APPLE_ASSAMESE 68
  Apple Assamese language ID.
- #define TT_LANGUAGE_APPLE_GUJARATI 69
  Apple Gujarati language ID.
- #define TT_LANGUAGE_APPLE_PUNJABI 70

Apple Punjabi language ID.

- #define TT_LANGUAGE_APPLE_ORIYA 71
  Apple Oriya language ID.
- #define TT_LANGUAGE_APPLE_MALAYALAM 72
  Apple malayalam language ID.
- #define TT_LANGUAGE_APPLE_KANNADA 73
  Apple Kannada language ID.
- #define TT_LANGUAGE_APPLE_TAMIL 74
  Apple Tamil language ID.
- #define TT_LANGUAGE_APPLE_TELUGU 75
  Apple Telugu language ID.
- #define TT_LANGUAGE_APPLE_SINHALESE 76
  Apple Sinhalese language ID.
- #define TT_LANGUAGE_APPLE_BURMESE 77
  Apple Burmese language ID.
- #define TT_LANGUAGE_APPLE_KHMER 78
  Apple Khmer language ID.
- #define TT_LANGUAGE_APPLE_LAO 79
  Apple Lao language ID.
- #define TT_LANGUAGE_APPLE_VIETNAMESE 80
  Apple Vietnamese language ID.
- #define TT_LANGUAGE_APPLE_INDONESIAN 81
  Apple Indonesian language ID.
- #define TT_LANGUAGE_APPLE_TAGALONG 82
  Apple Tagalong language ID.
- #define TT_LANGUAGE_APPLE_ROMANMALAY 83
  Apple Malay language ID.
- #define TT_LANGUAGE_APPLE_ARABICMALAY 84

Apple Malay language ID.

- #define TT_LANGUAGE_APPLE_AMHARIC 85
  Apple Amharic language ID.
- #define TT_LANGUAGE_APPLE_TIGRINYA 86
  Apple Tiginya language ID.
- #define TT_LANGUAGE_APPLE_GALLA 87
  Apple Galla language ID.
- #define TT_LANGUAGE_APPLE_SOMALI 88
  Apple Somali language ID.
- #define TT_LANGUAGE_APPLE_SWAHILI 89
  Apple Swahili language ID.
- #define TT_LANGUAGE_APPLE_KINYARWANDARUANDA 90
  Apple Kinyarwandaruanda language ID.
- #define TT_LANGUAGE_APPLE_RUNDI 91
  Apple Rundi language ID.
- #define TT_LANGUAGE_APPLE_NYANJACHEWA 92
  Apple Nyanjachewa language ID.
- #define TT_LANGUAGE_APPLE_MALAGASY 93
  Apple Malagasy language ID.
- #define TT_LANGUAGE_APPLE_ESPERANTO 94
  Apple Esperanto language ID.
- #define TT_LANGUAGE_APPLE_WELSH 128
  Apple Welsh language ID.
- #define TT_LANGUAGE_APPLE_BASQUE 129
  Apple Basque language ID.
- #define TT_LANGUAGE_APPLE_CATALAN 130
  Apple Catalan language ID.
- #define TT_LANGUAGE_APPLE_LATIN 131

Apple Latin language ID.

- #define TT_LANGUAGE_APPLE_QUENCHUA 132
  Apple Quenchua language ID.
- #define TT_LANGUAGE_APPLE_GUARANI 133
  Apple Guarani language ID.
- #define TT_LANGUAGE_APPLE_AYMARA 134
  Apple Aymara language ID.
- #define TT_LANGUAGE_APPLE_TATAR 135
  Apple Tatar language ID.
- #define TT_LANGUAGE_APPLE_UIGHUR 136
  Apple Uighur language ID.
- #define TT_LANGUAGE_APPLE_DZONGKHA 137
  Apple Dzongkha language ID.
- #define TT_LANGUAGE_APPLE_ROMANJAVANESE 138
  Apple Javanese language ID.
- #define TT_LANGUAGE_APPLE_ROMANSUNDANESE 139
  Apple Sundanese language ID.
- #define TT_LANGUAGE_APPLE_GALICIAN 140
  Apple Galician language ID.
- #define TT_LANGUAGE_APPLE_AFRIKAANS 141
  Apple Aftikanns language ID.
- #define TT_LANGUAGE_APPLE_BRETON 142
  Apple Breton language ID.
- #define TT_LANGUAGE_APPLE_INUKTITUT 143
  Apple Inuktitut language ID.
- #define TT_LANGUAGE_APPLE_SCOTTISHGAELIC 144
  Apple Scottish language ID.
- #define TT_LANGUAGE_APPLE_MANXGAELIC 145

Apple Manx language ID.

- #define TT_LANGUAGE_APPLE_IRISHGAELICWITHDOT 146
  Apple Irish language ID.
- #define TT_LANGUAGE_APPLE_TONGAN 147
  Apple Tongan language ID.
- #define TT_LANGUAGE_APPLE_POLYTONICGREEK 148
  Apple Greek language ID.
- #define TT_LANGUAGE_APPLE_GREENLANDIC 149
  Apple Greeenlandic language ID.
- #define TT_LANGUAGE_APPLE_ROMANAZERBAIJANI 150
  Apple Azerbaijani language ID.
- #define **ST_SCRIPTVARIANT_MASK** 0x30000000
- #define **ST_TERRITORY_MASK** 0x0FFFC000
- #define **ST_LANGUAGE_MASK** 0x00003FFF
- #define **ST_SCRIPTVARIANT**(a) ((a) & ST_SCRIPTVARIANT_MASK)
- #define **ST_TERRITORY**(a) ((a) & ST_TERRITORY_MASK)
- #define **ST_LANGUAGE**(a) ((a) & ST_LANGUAGE_MASK)
- #define **ST_LOCALE**(s, t, l) ((s) | (t) | (l))
- #define **ST_LOCALE2**(t, l) ((t) | (l))
- #define **ST_SCRIPTVARIANT_DEFAULT** 0x00000000
- #define **ST_SCRIPTVARIANT_0** 0x00000000
- #define **ST_SCRIPTVARIANT_1** 0x10000000
- #define **ST_SCRIPTVARIANT_2** 0x20000000
- #define **ST_SCRIPTVARIANT_3** 0x30000000
- #define **ST_TERRITORY_DEFAULT** 0x00000000
- #define **ST_TERRITORY_NOTERRITORY** (0x3FFF << 14)
- #define **ST_TERRITORY_AFGHANISTAN** (1 << 14)
- #define **ST_TERRITORY_ALBANIA** (2 << 14)
- #define **ST_TERRITORY_ALGERIA** (3 << 14)
- #define **ST_TERRITORY_AMERICAN_SAMOA** (4 << 14)
- #define **ST_TERRITORY_ANDORRA** (5 << 14)
- #define **ST_TERRITORY_ANGOLA** (6 << 14)
- #define **ST_TERRITORY_ANGUILLA** (7 << 14)
- #define **ST_TERRITORY_ANTARCTICA** (8 << 14)
- #define **ST_TERRITORY_ANTIGUA_AND_BARBUDA** (9 << 14)
- #define **ST_TERRITORY_ARGENTINA** (10 << 14)
- #define **ST_TERRITORY_ARMENIA** (11 << 14)

- #define **ST_TERRITORY_ARUBA** (12 << 14)
- #define **ST_TERRITORY_AUSTRALIA** (13 << 14)
- #define **ST_TERRITORY_AUSTRIA** (14 << 14)
- #define **ST_TERRITORY_AZERBAIJAN** (15 << 14)
- #define **ST_TERRITORY_BAHAMAS** (16 << 14)
- #define **ST_TERRITORY_BAHRAIN** (17 << 14)
- #define **ST_TERRITORY_BANGLADESH** (18 << 14)
- #define **ST_TERRITORY_BARBADOS** (19 << 14)
- #define **ST_TERRITORY_BELARUS** (20 << 14)
- #define **ST_TERRITORY_BELGIUM** (21 << 14)
- #define **ST_TERRITORY_BELIZE** (22 << 14)
- #define **ST_TERRITORY_BENIN** (23 << 14)
- #define **ST_TERRITORY_BERMUDA** (24 << 14)
- #define **ST_TERRITORY_BHUTAN** (25 << 14)
- #define **ST_TERRITORY_BOLIVIA** (26 << 14)
- #define **ST_TERRITORY_BOSNIA_AND_HERZEGOWINA** (27 << 14)
- #define **ST_TERRITORY_BOTSWANA** (28 << 14)
- #define **ST_TERRITORY_BOUVET_ISLAND** (29 << 14)
- #define **ST_TERRITORY_BRAZIL** (30 << 14)
- #define **ST_TERRITORY_BRITISH_INDIAN_OCEAN_TERRITORY** (31 << 14)
- #define **ST_TERRITORY_BRITISH_VIRGIN_ISLANDS** (32 << 14)
- #define **ST_TERRITORY_BRUNEI_DARUSSALAM** (33 << 14)
- #define **ST_TERRITORY_BULGARIA** (34 << 14)
- #define **ST_TERRITORY_BURKINA_FASO** (35 << 14)
- #define **ST_TERRITORY_BURUNDI** (36 << 14)
- #define **ST_TERRITORY_CAMBODIA** (37 << 14)
- #define **ST_TERRITORY_CAMEROON** (38 << 14)
- #define **ST_TERRITORY_CANADA** (39 << 14)
- #define **ST_TERRITORY_CAPE_VERDE** (40 << 14)
- #define **ST_TERRITORY_CAYMAN_ISLANDS** (41 << 14)
- #define **ST_TERRITORY_CENTRAL_AFRICAN_REPUBLIC** (42 << 14)

- #define **ST_TERRITORY_CHAD** (43 << 14)
- #define **ST_TERRITORY_CHILE** (44 << 14)
- #define **ST_TERRITORY_CHINA** (45 << 14)
- #define **ST_TERRITORY_CHRISTMAS_ISLAND** (46 << 14)
- #define **ST_TERRITORY_COCOS_ISLANDS** (47 << 14)
- #define **ST_TERRITORY_COLOMBIA** (48 << 14)
- #define **ST_TERRITORY_COMOROS** (49 << 14)
- #define **ST_TERRITORY_CONGO** (50 << 14)
- #define **ST_TERRITORY_COOK_ISLANDS** (51 << 14)
- #define **ST_TERRITORY_COSTA_RICA** (52 << 14)

- #define **ST_TERRITORY_COTE_DIVOIRE** (53 << 14)
- #define **ST_TERRITORY_CROATIA** (54 << 14)
- #define **ST_TERRITORY_CUBA** (55 << 14)
- #define **ST_TERRITORY_CYPRUS** (56 << 14)
- #define **ST_TERRITORY_CZECH_REPUBLIC** (57 << 14)
- #define **ST_TERRITORY_DENMARK** (58 << 14)
- #define **ST_TERRITORY_DJIBOUTI** (59 << 14)
- #define **ST_TERRITORY_DOMINICA** (60 << 14)
- #define **ST_TERRITORY_DOMINICAN_REPUBLIC** (61 << 14)
- #define **ST_TERRITORY_EAST_TIMOR** (62 << 14)
- #define **ST_TERRITORY_ECUADOR** (63 << 14)
- #define **ST_TERRITORY_EGYPT** (64 << 14)
- #define **ST_TERRITORY_EL_SALVADOR** (65 << 14)
- #define **ST_TERRITORY_EQUATORIAL_GUINEA** (66 << 14)
- #define **ST_TERRITORY_ERITREA** (67 << 14)
- #define **ST_TERRITORY_ESTONIA** (68 << 14)
- #define **ST_TERRITORY_ETHIOPIA** (69 << 14)
- #define **ST_TERRITORY_FALKLAND_ISLANDS** (70 << 14)
- #define **ST_TERRITORY_FAROE_ISLANDS** (71 << 14)
- #define **ST_TERRITORY_FIJI** (72 << 14)
- #define **ST_TERRITORY_FINLAND** (73 << 14)
- #define **ST_TERRITORY_FRANCE** (74 << 14)
- #define **ST_TERRITORY_FRENCH_GUIANA** (75 << 14)
- #define **ST_TERRITORY_FRENCH_POLYNESIA** (76 << 14)
- #define **ST_TERRITORY_FRENCH_SOUTHERN_TERRITORIES** (77 << 14)
- #define **ST_TERRITORY_GABON** (78 << 14)
- #define **ST_TERRITORY_GAMBIA** (79 << 14)
- #define **ST_TERRITORY_GEORGIA** (80 << 14)
- #define **ST_TERRITORY_GERMANY** (81 << 14)
- #define **ST_TERRITORY_GHANA** (82 << 14)
- #define **ST_TERRITORY_GIBRALTAR** (83 << 14)
- #define **ST_TERRITORY_GREECE** (84 << 14)
- #define **ST_TERRITORY_GREENLAND** (85 << 14)
- #define **ST_TERRITORY_GRENADA** (86 << 14)
- #define **ST_TERRITORY_GUADELOUPE** (87 << 14)
- #define **ST_TERRITORY_GUAM** (88 << 14)
- #define **ST_TERRITORY_GUATEMALA** (89 << 14)
- #define **ST_TERRITORY_GUINEA** (90 << 14)
- #define **ST_TERRITORY_GUINEA_BISSAU** (91 << 14)
- #define **ST_TERRITORY_GUYANA** (92 << 14)
- #define **ST_TERRITORY_HAITI** (93 << 14)

- #define **ST_TERRITORY_HEARD_AND_MC_DONALD_ISLANDS** (94 << 14)
- #define **ST_TERRITORY_HONDURAS** (95 << 14)
- #define **ST_TERRITORY_HONG_KONG** (96 << 14)
- #define **ST_TERRITORY_HUNGARY** (97 << 14)
- #define **ST_TERRITORY_ICELAND** (98 << 14)
- #define **ST_TERRITORY_INDIA** (99 << 14)
- #define **ST_TERRITORY_INDONESIA** (100 << 14)
- #define **ST_TERRITORY_IRAN** (101 << 14)
- #define **ST_TERRITORY_IRAQ** (102 << 14)
- #define **ST_TERRITORY_IRELAND** (103 << 14)
- #define **ST_TERRITORY_ISRAEL** (104 << 14)
- #define **ST_TERRITORY_ITALY** (105 << 14)
- #define **ST_TERRITORY_JAMAICA** (106 << 14)
- #define **ST_TERRITORY_JAPAN** (107 << 14)
- #define **ST_TERRITORY_JORDAN** (108 << 14)
- #define **ST_TERRITORY_KAZAKHSTAN** (109 << 14)
- #define **ST_TERRITORY_KENYA** (110 << 14)
- #define **ST_TERRITORY_KIRIBATI** (111 << 14)
- #define **ST_TERRITORY_KUWAIT** (112 << 14)
- #define **ST_TERRITORY_KYRGYZSTAN** (113 << 14)
- #define **ST_TERRITORY_LAOS** (114 << 14)
- #define **ST_TERRITORY_LATVIA** (115 << 14)
- #define **ST_TERRITORY_LEBANON** (116 << 14)
- #define **ST_TERRITORY_LESOTHO** (117 << 14)
- #define **ST_TERRITORY_LIBERIA** (118 << 14)
- #define **ST_TERRITORY_LIBYA** (119 << 14)
- #define **ST_TERRITORY_LIECHTENSTEIN** (120 << 14)
- #define **ST_TERRITORY_LITHUANIA** (121 << 14)
- #define **ST_TERRITORY_LUXEMBOURG** (122 << 14)
- #define **ST_TERRITORY_MACAU** (123 << 14)
- #define **ST_TERRITORY_MACEDONIA** (124 << 14)
- #define **ST_TERRITORY_MADAGASCAR** (125 << 14)
- #define **ST_TERRITORY_MALAWI** (126 << 14)
- #define **ST_TERRITORY_MALAYSIA** (127 << 14)
- #define **ST_TERRITORY_MALDIVES** (128 << 14)
- #define **ST_TERRITORY_MALI** (129 << 14)
- #define **ST_TERRITORY_MALTA** (130 << 14)
- #define **ST_TERRITORY_MARSHALL_ISLANDS** (131 << 14)
- #define **ST_TERRITORY_MARTINIQUE** (132 << 14)
- #define **ST_TERRITORY_MAURITANIA** (133 << 14)
- #define **ST_TERRITORY_MAURITIUS** (134 << 14)
- #define **ST_TERRITORY_MAYOTTE** (135 << 14)

- #define **ST_TERRITORY_METROPOLITAN_FRANCE** (136 << 14)
- #define **ST_TERRITORY_MEXICO** (137 << 14)
- #define **ST_TERRITORY_MICRONESIA** (138 << 14)
- #define **ST_TERRITORY_MOLDOVA** (139 << 14)
- #define **ST_TERRITORY_MONACO** (140 << 14)
- #define **ST_TERRITORY_MONGOLIA** (141 << 14)
- #define **ST_TERRITORY_MONTSERRAT** (142 << 14)
- #define **ST_TERRITORY_MOROCCO** (143 << 14)
- #define **ST_TERRITORY_MOZAMBIQUE** (144 << 14)
- #define **ST_TERRITORY_MYANMAR** (145 << 14)
- #define **ST_TERRITORY_NAMIBIA** (146 << 14)
- #define **ST_TERRITORY_NAURU** (147 << 14)
- #define **ST_TERRITORY_NEPAL** (148 << 14)
- #define **ST_TERRITORY_NETHERLANDS** (149 << 14)
- #define **ST_TERRITORY_NETHERLANDS_ANTILLES** (150 << 14)
- #define **ST_TERRITORY_NEW_CALEDONIA** (151 << 14)
- #define **ST_TERRITORY_NEW_ZEALAND** (152 << 14)
- #define **ST_TERRITORY_NICARAGUA** (153 << 14)
- #define **ST_TERRITORY_NIGER** (154 << 14)
- #define **ST_TERRITORY_NIGERIA** (155 << 14)
- #define **ST_TERRITORY_NIUE** (156 << 14)
- #define **ST_TERRITORY_NORFOLK_ISLAND** (157 << 14)
- #define **ST_TERRITORY_NORTHERN_MARIANA_ISLANDS** (158 << 14)
- #define **ST_TERRITORY_NORTH_KOREA** (159 << 14)
- #define **ST_TERRITORY_NORWAY** (160 << 14)
- #define **ST_TERRITORY_OMAN** (161 << 14)
- #define **ST_TERRITORY_PAKISTAN** (162 << 14)
- #define **ST_TERRITORY_PALAU** (163 << 14)
- #define **ST_TERRITORY_PANAMA** (164 << 14)
- #define **ST_TERRITORY_PAPUA_NEW_GUINEA** (165 << 14)
- #define **ST_TERRITORY_PARAGUAY** (166 << 14)
- #define **ST_TERRITORY_PERU** (167 << 14)
- #define **ST_TERRITORY_PHILIPPINES** (168 << 14)
- #define **ST_TERRITORY_PITCAIRN** (169 << 14)
- #define **ST_TERRITORY_POLAND** (170 << 14)
- #define **ST_TERRITORY_PORTUGAL** (171 << 14)
- #define **ST_TERRITORY_PUERTO_RICO** (172 << 14)
- #define **ST_TERRITORY_QATAR** (173 << 14)
- #define **ST_TERRITORY_REUNION** (174 << 14)
- #define **ST_TERRITORY_ROMANIA** (175 << 14)
- #define **ST_TERRITORY_RUSSIAN_FEDERATION** (176 << 14)
- #define **ST_TERRITORY_RWANDA** (177 << 14)

- #define **ST_TERRITORY_SAINT_KITTS_AND_NEVIS** (178 << 14)
- #define **ST_TERRITORY_SAINT_LUCIA** (179 << 14)
- #define **ST_TERRITORY_SAINT_VINCENT_AND_THE_GRENADINES** (180 << 14)
- #define **ST_TERRITORY_SAMOA** (181 << 14)
- #define **ST_TERRITORY_SAN_MARINO** (182 << 14)
- #define **ST_TERRITORY_SAO_TOME_AND_PRINCIPE** (183 << 14)
- #define **ST_TERRITORY_SAUDI_ARABIA** (184 << 14)
- #define **ST_TERRITORY_SENEGAL** (185 << 14)
- #define **ST_TERRITORY_SEYCHELLES** (186 << 14)
- #define **ST_TERRITORY_SIERRA** LEONE (187 << 14)
- #define **ST_TERRITORY_SINGAPORE** (188 << 14)
- #define **ST_TERRITORY_SLOVAKIA** (189 << 14)
- #define **ST_TERRITORY_SLOVENIA** (190 << 14)
- #define **ST_TERRITORY_SOLOMON_ISLANDS** (191 << 14)
- #define **ST_TERRITORY_SOMALIA** (192 << 14)
- #define **ST_TERRITORY_SOUTH_AFRICA** (193 << 14)
- #define **ST_TERRITORY_SOUTH_KOREA** (194 << 14)
- #define **ST_TERRITORY_SPAIN** (195 << 14)
- #define **ST_TERRITORY_SRI_LANKA** (196 << 14)
- #define **ST_TERRITORY_ST_HELENA** (197 << 14)
- #define **ST_TERRITORY_ST_PIERRE_AND_MIQUELON** (198 << 14)
- #define **ST_TERRITORY_SUDAN** (199 << 14)
- #define **ST_TERRITORY_SURINAME** (200 << 14)
- #define **ST_TERRITORY_SVALBARD_AND_JAN_MAYEN_ISLANDS** (201 << 14)
- #define **ST_TERRITORY_SWAZILAND** (202 << 14)
- #define **ST_TERRITORY_SWEDEN** (203 << 14)
- #define **ST_TERRITORY_SWITZERLAND** (204 << 14)
- #define **ST_TERRITORY_SYRIA** (205 << 14)
- #define **ST_TERRITORY_TAIWAN** (206 << 14)
- #define **ST_TERRITORY_TAJIKISTAN** (207 << 14)
- #define **ST_TERRITORY_TANZANIA** (208 << 14)
- #define **ST_TERRITORY_THAILAND** (209 << 14)
- #define **ST_TERRITORY_TOGO** (210 << 14)
- #define **ST_TERRITORY_TOKELAU** (211 << 14)
- #define **ST_TERRITORY_TONGA** (212 << 14)
- #define **ST_TERRITORY_TRINIDAD_AND_TOBAGO** (213 << 14)
- #define **ST_TERRITORY_TUNISIA** (214 << 14)
- #define **ST_TERRITORY_TURKEY** (215 << 14)
- #define **ST_TERRITORY_TURKMENISTAN** (216 << 14)
- #define **ST_TERRITORY_TURKS_AND_CAICOS_ISLANDS** (217 << 14)

- #define **ST_TERRITORY_TUVALU** (218 << 14)
- #define **ST_TERRITORY_UGANDA** (219 << 14)
- #define **ST_TERRITORY_UNITED_KINGDOM** (220 << 14)
- #define **ST_TERRITORY_UKRAINE** (221 << 14)
- #define **ST_TERRITORY_UNITED_ARAB_EMIRATES** (222 << 14)
- #define **ST_TERRITORY_URUGUAY** (223 << 14)
- #define **ST_TERRITORY_UNITED_STATES** (224 << 14)
- #define **ST_TERRITORY_UNITED_STATES_MINOR_OUTLYING_ISLANDS** (225 << 14)
- #define **ST_TERRITORY_UNITED_STATES_VIRGIN_ISLANDS** (226 << 14)
- #define **ST_TERRITORY_UZBEKISTAN** (227 << 14)
- #define **ST_TERRITORY_VANUATU** (228 << 14)
- #define **ST_TERRITORY_VATICAN** (229 << 14)
- #define **ST_TERRITORY_VENEZUELA** (230 << 14)
- #define **ST_TERRITORY_VIETNAM** (231 << 14)
- #define **ST_TERRITORY_WALLIS_AND_FUTUNA_ISLANDS** (232 << 14)
- #define **ST_TERRITORY_WESTERN_SAHARA** (233 << 14)
- #define **ST_TERRITORY_YEMEN** (234 << 14)
- #define **ST_TERRITORY_YUGOSLAVIA** (235 << 14)
- #define **ST_TERRITORY_ZAIRE** (236 << 14)
- #define **ST_TERRITORY_ZAMBIA** (237 << 14)
- #define **ST_TERRITORY_ZIMBABWE** (238 << 14)
- #define **ST_TERRITORY_UK** ST_TERRITORY_UNITED_KINGDOM
- #define **ST_TERRITORY_US** ST_TERRITORY_UNITED_STATES
- #define **ST_LANGUAGE_DEFAULT** 0x00000000
- #define **ST_LANGUAGE_NOLANGUAGE** 0x3FFF
- #define **ST_LANGUAGE_ABKHAZIAN** 1
- #define **ST_LANGUAGE_AFAR** 2
- #define **ST_LANGUAGE_AFRIKAANS** 3
- #define **ST_LANGUAGE_ALBANIAN** 4
- #define **ST_LANGUAGE_AMHARIC** 5
- #define **ST_LANGUAGE_ARABIC** 6
- #define **ST_LANGUAGE_ARMENIAN** 7
- #define **ST_LANGUAGE_ASSAMESE** 8
- #define **ST_LANGUAGE_AYMARA** 9
- #define **ST_LANGUAGE_AZERBAIJANI** 10
- #define **ST_LANGUAGE_BASHKIR** 11
- #define **ST_LANGUAGE_BASQUE** 12
- #define **ST_LANGUAGE_BENGALI** 13
- #define **ST_LANGUAGE_BHUTANI** 14
- #define **ST_LANGUAGE_BIHARI** 15

- #define **ST_LANGUAGE_BISLAMA** 16
- #define **ST_LANGUAGE_BRETON** 17
- #define **ST_LANGUAGE_BULGARIAN** 18
- #define **ST_LANGUAGE_BURMESE** 19
- #define **ST_LANGUAGE_BYELORUSSIAN** 20
- #define **ST_LANGUAGE_CAMBODIAN** 21
- #define **ST_LANGUAGE_CATALAN** 22
- #define **ST_LANGUAGE_CHINESE** 23
- #define **ST_LANGUAGE_CORSICAN** 24
- #define **ST_LANGUAGE_CROATIAN** 25
- #define **ST_LANGUAGE_CZECH** 26
- #define **ST_LANGUAGE_DANISH** 27
- #define **ST_LANGUAGE_DUTCH** 28
- #define **ST_LANGUAGE_ENGLISH** 29
- #define **ST_LANGUAGE_ESPERANTO** 30
- #define **ST_LANGUAGE_ESTONIAN** 31
- #define **ST_LANGUAGE_FAEROESE** 32
- #define **ST_LANGUAGE_FIJI** 33
- #define **ST_LANGUAGE_FINNISH** 34
- #define **ST_LANGUAGE_FRENCH** 35
- #define **ST_LANGUAGE_FRISIAN** 36
- #define **ST_LANGUAGE_GALICIAN** 37
- #define **ST_LANGUAGE_GEORGIAN** 38
- #define **ST_LANGUAGE_GERMAN** 39
- #define **ST_LANGUAGE_GREEK** 40
- #define **ST_LANGUAGE_GREENLANDIC** 41
- #define **ST_LANGUAGE_GUARANI** 42
- #define **ST_LANGUAGE_GUJARATI** 43
- #define **ST_LANGUAGE_HAUSA** 44
- #define **ST_LANGUAGE_HEBREW** 45
- #define **ST_LANGUAGE_HINDI** 46
- #define **ST_LANGUAGE_HUNGARIAN** 47
- #define **ST_LANGUAGE_ICELANDIC** 48
- #define **ST_LANGUAGE_INDONESIAN** 49
- #define **ST_LANGUAGE_INTERLINGUA** 50
- #define **ST_LANGUAGE_INTERLINGUE** 51
- #define **ST_LANGUAGE_INUKTITUT** 52
- #define **ST_LANGUAGE_INUPIAK** 53
- #define **ST_LANGUAGE_IRISH** 54
- #define **ST_LANGUAGE_ITALIAN** 55
- #define **ST_LANGUAGE_JAPANESE** 56
- #define **ST_LANGUAGE_JAVANESE** 57
- #define **ST_LANGUAGE_KANNADA** 58

- #define **ST_LANGUAGE_KASHMIRI** 59
- #define **ST_LANGUAGE_KAZAKH** 60
- #define **ST_LANGUAGE_KINYARWANDA** 61
- #define **ST_LANGUAGE_KIRGHIZ** 62
- #define **ST_LANGUAGE_KIRUNDI** 63
- #define **ST_LANGUAGE_KOREAN** 64
- #define **ST_LANGUAGE_KURDISH** 65
- #define **ST_LANGUAGE_LAOTHIAN** 66
- #define **ST_LANGUAGE_LATIN** 67
- #define **ST_LANGUAGE_LATVIAN** 68
- #define **ST_LANGUAGE_LINGALA** 69
- #define **ST_LANGUAGE_LITHUANIAN** 70
- #define **ST_LANGUAGE_MACEDONIAN** 71
- #define **ST_LANGUAGE_MALAGASY** 72
- #define **ST_LANGUAGE_MALAY** 73
- #define **ST_LANGUAGE_MALAYALAM** 74
- #define **ST_LANGUAGE_MALTESE** 75
- #define **ST_LANGUAGE_MAORI** 76
- #define **ST_LANGUAGE_MARATHI** 77
- #define **ST_LANGUAGE_MOLDAVIAN** 78
- #define **ST_LANGUAGE_MONGOLIAN** 79
- #define **ST_LANGUAGE_NAURU** 80
- #define **ST_LANGUAGE_NEPALI** 81
- #define **ST_LANGUAGE_NORWEGIAN** 82
- #define **ST_LANGUAGE_OCCITAN** 83
- #define **ST_LANGUAGE_ORIYA** 84
- #define **ST_LANGUAGE_OROMO** 85
- #define **ST_LANGUAGE_PASHTO** 86
- #define **ST_LANGUAGE_PERSIAN** 87
- #define **ST_LANGUAGE_POLISH** 88
- #define **ST_LANGUAGE_PORTUGUESE** 89
- #define **ST_LANGUAGE_PUNJABI** 90
- #define **ST_LANGUAGE_QUECHUA** 91
- #define **ST_LANGUAGE_RHAETO_ROMANCE** 92
- #define **ST_LANGUAGE_ROMANIAN** 93
- #define **ST_LANGUAGE_RUSSIAN** 94
- #define **ST_LANGUAGE_SAMOAN** 95
- #define **ST_LANGUAGE_SANGRO** 96
- #define **ST_LANGUAGE_SANSKRIT** 97
- #define **ST_LANGUAGE_SCOTS_GAELIC** 98
- #define **ST_LANGUAGE_SERBIAN** 99
- #define **ST_LANGUAGE_SERBO_CROATIAN** 100
- #define **ST_LANGUAGE_SESOTHO** 101

- #define **ST_LANGUAGE_SETSWANA** 102
- #define **ST_LANGUAGE_SHONA** 103
- #define **ST_LANGUAGE_SINDHI** 104
- #define **ST_LANGUAGE_SINHALESE** 105
- #define **ST_LANGUAGE_SISWATI** 106
- #define **ST_LANGUAGE_SLOVAK** 107
- #define **ST_LANGUAGE_SLOVENIAN** 108
- #define **ST_LANGUAGE_SOMALI** 109
- #define **ST_LANGUAGE_SPANISH** 110
- #define **ST_LANGUAGE_SUNDANESE** 111
- #define **ST_LANGUAGE_SWAHILI** 112
- #define **ST_LANGUAGE_SWEDISH** 113
- #define **ST_LANGUAGE_TAGALOG** 114
- #define **ST_LANGUAGE_TAJIK** 115
- #define **ST_LANGUAGE_TAMIL** 116
- #define **ST_LANGUAGE_TATAR** 117
- #define **ST_LANGUAGE_TELUGU** 118
- #define **ST_LANGUAGE_THAI** 119
- #define **ST_LANGUAGE_TIBETAN** 120
- #define **ST_LANGUAGE_TIGRINYA** 121
- #define **ST_LANGUAGE_TONGA** 122
- #define **ST_LANGUAGE_TSONGA** 123
- #define **ST_LANGUAGE_TURKISH** 124
- #define **ST_LANGUAGE_TURKMEN** 125
- #define **ST_LANGUAGE_TWI** 126
- #define **ST_LANGUAGE_UIGHUR** 127
- #define **ST_LANGUAGE_UKRAINIAN** 128
- #define **ST_LANGUAGE_URDU** 129
- #define **ST_LANGUAGE_UZBEK** 130
- #define **ST_LANGUAGE_VIETNAMESE** 131
- #define **ST_LANGUAGE_VOLAPUK** 132
- #define **ST_LANGUAGE_WELSH** 133
- #define **ST_LANGUAGE_WOLOF** 134
- #define **ST_LANGUAGE_XHOSA** 135
- #define **ST_LANGUAGE_YIDDISH** 136
- #define **ST_LANGUAGE_YORUBA** 137
- #define **ST_LANGUAGE_ZHUANG** 138
- #define **ST_LANGUAGE_ZULU** 139
- #define **ST_SCRIPT_ARABIC** 0x61726162
- #define **ST_SCRIPT_ARMENIAN** 0x61726D6E
- #define **ST_SCRIPT_BENGALI** 0x62656E67
- #define **ST_SCRIPT_BOPOMOFO** 0x626F706F
- #define **ST_SCRIPT_BRAILLE** 0x62726169

- #define **ST_SCRIPT_CANADIAN_SYLLABICS** 0x63616E73
- #define **ST_SCRIPT_CHEROKEE** 0x63686572
- #define **ST_SCRIPT_COPTIC** 0x71616163
- #define **ST_SCRIPT_CYRILLIC** 0x6379726C
- #define **ST_SCRIPT_DESERET** 0x64737274
- #define **ST_SCRIPT_DEVANAGARI** 0x64657661
- #define **ST_SCRIPT_ETHIOPIC** 0x65746869
- #define **ST_SCRIPT_GEORGIAN** 0x67656F72
- #define **ST_SCRIPT_GOTHIC** 0x676F7468
- #define **ST_SCRIPT_GREEK** 0x6772656B
- #define **ST_SCRIPT_GUJARATI** 0x67756A72
- #define **ST_SCRIPT_GURMUKHI** 0x67757275
- #define **ST_SCRIPT_HAN** 0x68616E69
- #define **ST_SCRIPT_HANGUL** 0x68616E67
- #define **ST_SCRIPT_HEBREW** 0x68656272
- #define **ST_SCRIPT_HIRAGANA** 0x68697261
- #define **ST_SCRIPT_JAMO** 0x6A616D6F
- #define **ST_SCRIPT_KANNADA** 0x6B6E6461
- #define **ST_SCRIPT_KATAKANA** 0x6B617461
- #define **ST_SCRIPT_KHMER** 0x6B686D72
- #define **ST_SCRIPT_LAO** 0x6C616F6F
- #define **ST_SCRIPT_LATIN** 0x6C61746E
- #define **ST_SCRIPT_MALAYALAM** 0x6D6C796D
- #define **ST_SCRIPT_MONGOLIAN** 0x6D6F6E67
- #define **ST_SCRIPT_MYANMAR** 0x6D796D72
- #define **ST_SCRIPT_OGHAM** 0x6F67616D
- #define **ST_SCRIPT_OLDITALIC** 0x6974616C
- #define **ST_SCRIPT_ORIYA** 0x6F727961
- #define **ST_SCRIPT_RUNIC** 0x72756E72
- #define **ST_SCRIPT_SINHALA** 0x73696E68
- #define **ST_SCRIPT_SYRIAC** 0x73797263
- #define **ST_SCRIPT_TAMIL** 0x74616D6C
- #define **ST_SCRIPT_TELUGU** 0x74656C75
- #define **ST_SCRIPT_THAANA** 0x74686161
- #define **ST_SCRIPT_THAI** 0x74686169
- #define **ST_SCRIPT_TIBETAN** 0x74696274
- #define **ST_SCRIPT_CANADIAN_ABORIGINAL** 0x63616E73
- #define **ST_SCRIPT_YI** 0x79696969
- #define **ST_SCRIPT_ZYYY** 0x7A797979
- #define **ST_SCRIPT_QAAI** 0x71616169
- #define **ST_SCRIPT_NOSCRIPT** 0x00000000

Typedefs

- typedef unsigned char byte
  Definition of a byte.
- typedef unsigned char uint8
  Definition of a uint8 (8 bit unsigned int).
- typedef signed char int8
  Definition of an int8 (8 bit signed int).
- typedef unsigned short int uint16
  Definition of a uint16 (16 bit unsigned int).
- typedef short int int16
  Definition of an int16 (16 bit signed int).
- typedef unsigned int uint32
  Definition of a uint32 (32 bit unsigned int).
- typedef int int32
  Definition of an int32 (32 bit signed int).
- typedef uint32 ucs4
  Definition of a ucs4 encoded character.
- typedef uint16 utf16
  Definition of a utf16 encoded character.
- typedef byte utf8
  Definition of a utf8 encoded character.
- typedef TPoint * TPointPtr
  Pointer to TPoint structure.
- typedef void * Pointer
  A generic pointer.
- typedef STOpaqueTypeEnv * STTypeEnv
  Type environment object.
- typedef STOpaqueText * STText
  Text object.

- typedef STOpaqueGlyphVector * STGlyphVector
  Glyph vector object.
- typedef STOpaqueLine * STLine
  Line object.
- typedef STOpaqueStyle * STStyle
  Style object.
- typedef uint32 STObject
  Generic drop-in object ID.
- typedef uint32 STScaler
  Scaler ID.
- typedef uint32 STLayoutEngine
  Layout Engine ID.
- typedef void * **STPointer**
- typedef uint32 **STHandle**
- typedef uint32 STTag
  untypes 4 character tag.
- typedef uint32 STFont
  font ID.
- typedef uint32 STFontFamily
  font family ID.
- typedef uint32 SCScaler
  scaler ID.
- typedef uint32 STRGBAColor
  32 bit RGBA color.
- typedef int STDirection
  Text direction.
- typedef int STJustification
  Text justification.

- typedef double STFlushFactor

  Line flush factor.

- typedef int STPosition

  Location of character position within text.

- typedef int STCount

  Number of items.

- typedef int STCharCount

  Number of characters.

- typedef int STSize

  Size ??? not used??

- typedef int STBoolean

  Boolean representation.

- typedef uint32 STMask

  Bitmask.

- typedef uint16 STGlyph

  STSF Glyph ID.

- typedef uint32 STStyleMask

  STStyleBitMask This bitmask is ORed together to indicate which values to change for the following functions: STStyleSetFont STStyleSetScaler STStyleSetEffects STStyleCompare STStyleClear STStyleResetAttributes STStyleCopyAttributes STStyleOverwriteAttributes STStyleUnderwriteAttributes STStyleSetOptions.

- typedef uint32 STGraphicsMask

  STGraphicsMask This bitmask is ORed together to indicate which values or colors to change for the following functions: STGraphicsNew STGraphicsSetColors.

- typedef uint32 STTextMask

  STTextMask This bitmask is ORed together to indicate which values to change for the following functions: STTextSetControls STTextCopyAttributes STTextResetAttributes.

- typedef uint32 STStyleEffects

  STStyleEffects This bitmask indicates the style effects to use for the length of the style run.

- typedef uint32 STFontLocationsMask

This bitmask is used to specify which locations to search for fonts.

- typedef uint32 STStrikeThrough

  STStrikeThrough Use this bitmask to set the type of strike through to use for the style run.

- typedef uint32 STUnderline

  STUnderline Use this bitmask to set the type of underline to use for the style run.

- typedef uint32 STOutputMode

  STOutputMode an OR of one of the ST_OM_ constants and one or more ST_OF_-flags The bitmask is used in the following functions: STGraphicsNew STGraphicsSet-Device.

- typedef uint32 STHintingMode

  STHintingMode This value is used to inform the scaler what type of hinting to apply to the font.

- typedef uint32 STSbitsMode

  STSbitsMode This value is used to inform the scaler whether to use embedded bitmaps or not.

- typedef STFontMetrics * STFontMetricsPtr

  Pointer to STFontMetrics structure.

- typedef STLineMetrics * STLineMetricsPtr

  Pointer to STLineMetrics structure.

- typedef STExtLineMetrics * STExtLineMetricsPtr

  Pointer to STExtLineMetrics structure.

- typedef STNameTag * STNameTagPtr

  Pointer to STNameTag structure.

- typedef STNameTagsRec * STNameTagsRecPtr

  Pointer to STNameTag structure.

- typedef uint32 STFontFeatureTag

  Font features.

- typedef STCaret * STCaretPtr

  Pointer to STCaret structure.

- typedef STTrapezoid * STTrapezoidPtr
  Pointer to STTrapezoid structure.
- typedef STRectangle * STRectanglePtr
  Pointer to STRectangle structure.
- typedef STPoint * STPointPtr
  Pointer to STPoint structure.
- typedef STMatrix * STMatrixPtr
  Pointer to STMatrix structure.
- typedef STLookupTable * STLookupTablePtr
  Pointer to STLookupTable structure.
- typedef double STBaselines [32]
  Array of 32 baseline values.
- typedef uint32 STBaselineFlag
  Defines standard baselines.
- typedef uint32 STBounds
  Defines bounds type.
- typedef uint32 STFontFallbackPolicy
  Defines font fallback policy.
- typedef int STCaretMovement
  Defines caret movement type.
- typedef int STCaretDirection
  Defines caret direction type.
- typedef int STTextChanged
  Text change notification.
- typedef int STStyleComparison
  Style comparison result.
- typedef uint32 STLayoutOptions
  Style layout options for ICU.

- typedef int STFontType
  Font types for ST API.
- typedef uint32 STScalerFlags
  STScalerFlags - returned by STScalerGetInfo().
- typedef uint32 **STLayoutEngineFlags**
- typedef unsigned int STFontServerFontType
  Font type.
- typedef unsigned int STFontServerFontMask
  Font type mask.
- typedef uint32 STFontInfoFlags
  Font Info Flags These refer to the capabilities of the font.
- typedef int STRenderingMode
  Rendering Mode.
- typedef STBitBltRec * STBitBltPtr
  Pointer to STBitBltRec.
- typedef int STDeviceType
  Type of output device (Raster or Vector).
- typedef stdevice STDeviceStruct
  STDevice structure.
- typedef STDeviceStruct * STDevice
  Pointer to STDeviceStruct structure.
- typedef void(* BitBltF)(struct STRasterDeviceStruct *dev, STBitBltPtr bbdata, byte *source)
  Raster callback function.
- typedef void(* HighlightF)(struct STRasterDeviceStruct *dev, STTrapezoid *trap, int depth, STRenderingMode mode)
  Raster highlight callback function.
- typedef void(* UnderlineF)(struct STRasterDeviceStruct *dev, STPoint p1, STPoint p2, int highlight, int depth, STUnderline uleffects, STRenderingMode mode)
  Raster underline callback function.

- typedef void(* StrikeThroughF)(struct STRasterDeviceStruct *dev, STPoint p1, STPoint p2, int highlight, int depth, STStrikeThrough steffects, STRenderingMode mode)

  Raster strike through callback function.

- typedef void(* CopyOutlineF)(struct STVectorDeviceStruct *dev, STPath path)

  Outline callback function.

- typedef STRasterDeviceStruct * STRasterDevice

  Pointer to STRasterDeviceStruct structure.

- typedef STVectorDeviceStruct * STVectorDevice

  Pointer to STVectorDeviceStruct.

- typedef stgraphics STGraphicsStruct

  STGraphics structure.

- typedef STGraphicsStruct * STGraphics

  Pointer to STGraphicsStruct structure.

- typedef uint32 STLocale

  *ST_SCRIPT_VARIANT_** | *ST_TERRITORY_** | *ST_LANGUAGE_**.

- typedef uint32 STScriptVariant

  *ST_SCRIPTVARIANT_**.

- typedef uint32 STLanguage

  *ST_LANGUAGE_**.

- typedef uint32 STTerritory

  *ST_TERRITORY_**.

- typedef uint32 STScript

  *ST_SCRIPT_**.

- typedef uint16 STFontWeightClass

  Weight of font as described by the font.

- typedef uint16 STFontWidthClass

  Width of font as described by the font.

- typedef uint16 STFontStyle

  STFontStyle denotes the value that the font has embedded for its italicity.

Enumerations

- enum STPathElement { fBezierCurve, fBSpline, fLineSeg }

  Type of path element for vector font output.

- enum STStatus { ST_NO_ERROR = 0, ST_OK = ST_NO_ERROR, ST_MEMORY, ST_NULL_OR_EMPTY_STRING, ST_NULL_POINTER, ST_BAD_FONTPATH, ST_FONTENUMERATOR_FAILED, ST_BAD_SCALERPATH, ST_SCALERENUMERATOR_FAILED, ST_FONTMANAGER_FAILED, ST_BAD_FONTID, ST_NO_SCALER, ST_BUFSIZE, ST_BITDEPTH, ST_PADDING, ST_SERVER_ERROR, ST_FONT_NOT_FOUND, ST_SCALER_NOT_FOUND, ST_LAYOUTENGINE_NOT_FOUND, ST_TEXT_RANGE, ST_RENDER_FLAGS, ST_BAD_TYPEENV, ST_BAD_FONTSIZE, ST_UNKNOWN_LANGUAGE, ST_UNKNOWN_ENCODING, ST_NAME_NOT_FOUND, ST_BAD_FONTINSTANCE, ST_BAD_FAMILYID, ST_OBJECT_NOT_FOUND, ST_RANGE, ST_FILEIO, ST_NOT_IMPLEMENTED, ST_INTERNAL_ERROR, ST_CONN_ERROR, ST_CONN_SEND_ERROR, ST_CONN_RECEIVE_ERROR, ST_CONN_TIMEOUT_ERROR, ST_MAX_ERROR }

6.1.1 Detailed Description

STSF Type definitions.

Author:
Alexander Gelfenbain <adg@sun.com> , Hideki Hiura <hiura@sun.com>

Version:
0.5

6.1.2 Define Documentation

6.1.2.1 #define ST_BASELINE_LOWCENTERED 3

Similar to STCenteredBaseline but with the glyphs lowered.

Used to align Roman glyphs within ideographic fonts with Roman glyphs in Roman fonts

6.1.2.2 #define ST_BASELINE_MATH 4

The baseline for setting mathematics.

Centered on symbols such as the minus sign

6.1.2.3 #define ST_CD_NEXT 3

Move caret to the next movement value.

Takes bidi into account

6.1.2.4 #define ST_CD_PREVIOUS 2

Move caret to the previous movement value.

Takes bidi into account

6.1.2.5 #define ST_SCALER_DUMMY 0x444D4D59

'DMMY' Dummy scaler for testing.

Sun Microsystems, Inc. maintains an open registry of tags. Submit new tags by sending an email message to stsf-registry@sun.com

6.1.3 Typedef Documentation

6.1.3.1 typedef void(* BitBltF)(struct STRasterDeviceStruct *dev, STBitBltPtr bbdata, byte *source)

Raster callback function.

Parameters:

dev the "this" pointer

tx target X coordinate in device space

ty target Y coordinate in device space

source pointer to the (0,0) point in the source bitmap

depth number of bytes per pixel in the source bitmap

width width of the source bitmap

height height of the source bitmap

rowbytes number of bytes in one row of the source bitmap

sx X offset into the source buffer

sy Y offset into the source buffer

6.1.3.2 typedef void(* CopyOutlineF)(struct STVectorDeviceStruct *dev, STPath path)

Outline callback function.

Parameters:

dev the "this" pointer

path an outline path in device space

6.1.3.3 typedef void(* HighlightF)(struct STRasterDeviceStruct *dev, STTrapezoid *trap, int depth, STRenderingMode mode)

Raster highlight callback function.

Parameters:

dev the "this" pointer

trap trapezoid bounding highlighted region

depth depth of drawable

mode rendering mode (LCD, B&W, etc)

6.1.3.4 typedef uint32 STFontFeatureTag

Font features.

Values are specified in FontFeatureTags enum

6.1.3.5 typedef uint32 STFontLocationsMask

This bitmask is used to specify which locations to search for fonts.

Use the following functions to set this value: STTypeEnvSetLocations

The bitmask is returned to indicate which locations are presently being searched by the following functions: STTypeEnvGetLocations

6.1.3.6 typedef uint16 STFontStyle

STFontStyle denotes the value that the font has embedded for its italicity.

This is a value stored in the font and is not associated with the name of the font. The name may say oblique, or included, but if the value specifies italic, then the STFontStyle will be italic.

6.1.3.7 typedef uint16 STGlyph

STSF Glyph ID.

Since all existing fonts use 16 bit glyph IDs, do not waste memory on 32 glyph IDs

6.1.3.8 typedef uint32 STGraphicsMask

STGraphicsMask This bitmask is ORed together to indicate which values or colors to change for the following functions: STGraphicsNew STGraphicsSetColors.

A bitmask is returned by the following functions to indicate values that were previously set: STGraphicsGetColors

6.1.3.9 typedef uint32 STHintingMode

STHintingMode This value is used to inform the scaler what type of hinting to apply to the font.

Note that the scaler must support the requested hinting mode or no change will occur. The value is used in the following functions: STStyleSetScaler

It is returned to indicate the hinting mode used by the following functions: STStyleGetScaler

6.1.3.10 typedef uint32 STOutputMode

STOutputMode an OR of one of the ST_OM_ constants and one or more ST_OF_-flags The bitmask is used in the following functions: STGraphicsNew STGraphicsSetDevice.

It is returned to indicate the output mode and any formatting options by the following functions: STGraphicsGetDevice

6.1.3.11 typedef void(* StrikeThroughF)(struct STRasterDeviceStruct *dev, STPoint p1, STPoint p2, int highlight, int depth, STStrikeThrough steffects, STRenderingMode mode)

Raster strike through callback function.

Parameters:

dev the ”this” pointer

p1 coordinate value to start underline

p2 coordinate value to end underline

depth depth of drawable

seffects type of strikethrough to draw

mode rendering mode (LCD, B&W, etc)

6.1.3.12 typedef uint32 STSbitsMode

STSbitsMode This value is used to inform the scaler whether to use embedded bitmaps or not.

Note that the scaler must support embedded bitmaps. The value is used in the following functions: STStyleSetScaler

It is returned to indicate the hinting mode used by the following functions: STStyleGetScaler

6.1.3.13 typedef uint32 STStrikeThrough

STStrikeThrough Use this bitmask to set the type of strike through to use for the style run.

Note that ST_STRIKETHROUGH_THICK must be ORed with either of the other two values to be valid and ORing ST_STRIKETHROUGH_SINGLE and ST_STRIKETHROUGH_DOUBLE together is not valid. The bitmask is used in the following functions: STStyleSetEffects

It is returned to indicate the type of strike through by the following functions: STStyleGetEffects

6.1.3.14 typedef uint32 STStyleEffects

STStyleEffects This bitmask indicates the style effects to use for the length of the style run.

It is ORed together and used in the following functions: STStyleSetEffects

It is returned to indicate which fields are set by the following functions: STStyleGetEffects

6.1.3.15 typedef uint32 STStyleMask

STStyleBitMask This bitmask is ORed together to indicate which values to change for the following functions: STStyleSetFont STStyleSetScaler STStyleSetEffects STStyle-Compare STStyleClear STStyleResetAttributes STStyleCopyAttributes STStyle-OverwriteAttributes STStyleUnderwriteAttributes STStyleSetOptions.

A bitmask is returned by the following functions to indicate values that were previously set: STStyleGetFont STStyleGetScaler STStyleGetOptions STStyleGetEffects

6.1.3.16 typedef uint32 STTag

untypes 4 character tag.

Objects of this class should be instantiated by users of ST API. STDevice children - STRasterDevice and STVectorDevice define callback functions that are called by STClientLibrary as it renders the line of text.

In addition to the callback function STDevice communicates important characteristics of an output device, such as its depth and resolution to STClientLibrary.

STRasterDevice and STVectorDevice include a transformation matrics that defines an affine transform from user-space coordinates to device-space coordinates. It can be accessed directly or through STGraphics methods.

6.1.3.17 typedef uint32 STTextMask

STTextMask This bitmask is ORed together to indicate which values to change for the following functions: STTextSetControls STTextCopyAttributes STTextReset-Attributes.

A bitmask is returned by the following functions to indicate values that were previously set: STTextGetControls

6.1.3.18 typedef uint32 STUnderline

STUnderline Use this bitmask to set the type of underline to use for the style run.

Note that ST_UNDERLINE_DOUBLE and ST_UNDERLINE_THICK must be ORed with one of the underline types to be valid, and ORing two or more underline types is not valid. Having two underlines present in the style run is allowed. To perform this, create two underline values, and use the ST_COMBINE_UNDERLINE macro to create the two underlines. (This might be used by a word processing application where the text is underlined, but also misspelled.) The bitmask is used in the following functions: STStyleSetEffects

It is returned to indicate the type of underline by the following functions: STStyleGet-Effects

6.1.3.19 typedef void(* UnderlineF)(struct STRasterDeviceStruct *dev, STPoint p1, STPoint p2, int highlight, int depth, STUnderline uleffects, STRenderingMode mode)

Raster underline callback function.

Parameters:

dev the "this" pointer

p1 coordinate value to start underline

p2 coordinate value to end underline

depth depth of drawable

uleffects type of underline to draw

mode rendering mode (LCD, B&W, etc)

6.1.4 Enumeration Type Documentation

6.1.4.1 enum STPathElement

Type of path element for vector font output.

Enumeration values:

fBezierCurve path element is a bezier curve.

fBSpline path element is a BSpline curve.

fLineSeg path element is a line segment.

```
739             {
740     fBezierCurve,
741     fBSpline,
742     fLineSeg
743 } STPathElement;
```

6.1.4.2 enum STStatus

Enumeration values:

ST_NO_ERROR no error.

ST_OK ditto.

ST_MEMORY memory allocation error.

ST_NULL_OR_EMPTY_STRING unexpected NULL or empty argument to a function.

ST_NULL_POINTER unexpected NULL pointer.

ST_BAD_FONTPATH invalid font path.

ST_FONTENUMERATOR_FAILED error in font enumerator.

ST_BAD_SCALERPATH invalid scaler path.

ST_SCALERENUMERATOR_FAILED error in scaler enumerator.

ST_FONTMANAGER_FAILED error in font manager.

ST_BAD_FONTID font ID is not valid.

ST_NO_SCALER no scaler is available for the font.

ST_BUFSIZE insufficient size of a buffer.

ST_BITDEPTH incorrect bit depth of a raster device.

ST_PADDING incorrect bitmap padding.

ST_SERVER_ERROR error communicating with the STFontServer.

ST_FONT_NOT_FOUND font with the specified name does not exist.

ST_SCALER_NOT_FOUND scaler with the specified scaler ID does not exist.

ST_LAYOUTENGINE_NOT_FOUND layout engine with the specified ID does not exits.

ST_TEXT_RANGE invalid range of text of the Layout object.

ST_RENDER_FLAGS invalid rendering flags.

ST_BAD_TYPEENV corrupted type environment.

ST_BAD_FONTSIZE font size is not valid.

ST_UNKNOWN_LANGUAGE unknown language.

ST_UNKNOWN_ENCODING unknown encoding.

ST_NAME_NOT_FOUND name was not found in the font.

ST_BAD_FONTINSTANCE incorrect font instance.

ST_BAD_FAMILYID fontfamily ID is not valid.

ST_OBJECT_NOT_FOUND drop-in object with the specified tag is not available.

ST_RANGE index is out of range.

ST_FILEIO error reading or writing files.

ST_NOT_IMPLEMENTED feature not implemented.

ST_INTERNAL_ERROR internal error.

ST_CONN_ERROR Client/Server connection error.

ST_CONN_SEND_ERROR Client/Server connection send error.

ST_CONN_RECEIVE_ERROR Client/Server connection receive error.

ST_CONN_TIMEOUT_ERROR Client/Server connection timeout error.

ST_MAX_ERROR total number of error codes defined, no function returns this code.

```
1623             {
1624     ST_NO_ERROR = 0,
1625     ST_OK = ST_NO_ERROR,
1626     ST_MEMORY,
1627     ST_NULL_OR_EMPTY_STRING,
1628     ST_NULL_POINTER,
1629     ST_BAD_FONTPATH,
1630     ST_FONTENUMERATOR_FAILED,
1631     ST_BAD_SCALERPATH,
1632     ST_SCALERENUMERATOR_FAILED,
1633     ST_FONTMANAGER_FAILED,
1634     ST_BAD_FONTID,
1635     ST_NO_SCALER,
1636     ST_BUFSIZE,
1637     ST_BITDEPTH,
1638     ST_PADDING,
1639     ST_SERVER_ERROR,
1640     ST_FONT_NOT_FOUND,
1641     ST_SCALER_NOT_FOUND,
1642     ST_LAYOUTENGINE_NOT_FOUND,
1643     ST_TEXT_RANGE,
1644     ST_RENDER_FLAGS,
1645     ST_BAD_TYPEENV,
1646     ST_BAD_FONTSIZE,
1647     ST_UNKNOWN_LANGUAGE,
1648     ST_UNKNOWN_ENCODING,
1649     ST_NAME_NOT_FOUND,
1650     ST_BAD_FONTINSTANCE,
1651     ST_BAD_FAMILYID,
1652     ST_OBJECT_NOT_FOUND,
1653     ST_RANGE,
1654     ST_FILEIO,
1655     ST_NOT_IMPLEMENTED,
1656     ST_INTERNAL_ERROR,
1657     ST_CONN_ERROR,
1658     ST_CONN_SEND_ERROR,
1659     ST_CONN_RECEIVE_ERROR,
1660     ST_CONN_TIMEOUT_ERROR,
1661     ST_MAX_ERROR
1662 } STStatus;
```

Index