# Text Input Source Services Reference

**Text & Fonts > Text Manipulation**

# Contents

# Text Input Source Services Reference

**Framework:**        Carbon/HIToolbox

**Declared in**        TextInputSources.h

## Overview

This document describes Text Input Source Services, a C programming interface in Carbon's HIToolbox framework.

Text input sources fall into three general categories:

- Keyboard input sources, including keyboard layouts, keyboard input methods and input modes

- Palette input sources, including the character palette, keyboard viewer, and private dictionary panels

- Ink input sources

Palette input sources and ink input source are categorized as non-keyboard input sources. However, palette input sources might still involve some keyboard interaction. Keyboard input methods can be mode-enabled (for instance, Kotoeri). Hence, they are potentially the parent of several input modes directly selectable in the user interface (for instance, hiragana, katakana, and romaji). With mode-enabled input methods, the parent input method is not directly selectable. Non-mode-enabled input methods are directly selectable.

Some input sources are provided and installed with Mac OS X. Other input sources can be installed as third party products themselves or as part of the installation of certain applications. International Preferences displays a list of all visible installed input sources. You can use International Preferences to enable or disable most input sources that are intended to be visible in the user interface. Setup Assistant also enables some input sources. A separate user interface provides a way to enable ink input sources.

You can use some of the functions in Text Input Source Services to enable or disable input sources programmatically in your application. One keyboard input source must be enabled. Only one ink input source may be enabled. Multiple instances of other input source types may be enabled.

Some enabled input sources are invisible but programmatically selectable, such as ink. Some are visible but not programmatically selectable, such as mode-savvy parent input methods (these must be visible so that International Preferences can display the parent input method for a group of input modes).

Input modes can only be changed from disabled to enabled if their parent input method is enabled. Input modes can only be selected if they and their parent input method are enabled.

Exactly one keyboard input source is selected at any time: the current keyboard input source. Selecting a new keyboard input source deselects the previous keyboard input source. Multiple palette input sources may be selected. For instance, there may be one or more character palettes and one or more keyboard

viewers selected in addition to the selected keyboard input source. Selecting or deselecting a palette (or ink) input source does not affect any other input source. Input methods that provide associated input palettes may programmatically deselect the palette when the input method is deselected, for example.

Text Input Source Source Services specifies modern, non-Script-Manager-based programming interfaces for operating on text input sources and performs the following functions:

■  Finds information about text input sources

■  Selects, enables, and disables text input sources

■  Receives notifications about relevant changes in text input sources.

# Functions by Task

## Getting the Type ID

TISInputSourceGetTypeID  (page 14)
>   Gets the `CFTypeID` of a text input source reference.

## Getting Information About Text Input Sources

TISGetInputSourceProperty  (page 13)
>   Gets the value of a specified property for a specified input source.

TISCreateInputSourceList  (page 11)
>   Creates a list of input sources that match specified properties.

## Getting Specific Input Sources

TISCopyCurrentKeyboardInputSource  (page 8)
>   Copies a text input source reference for the currently selected keyboard input source.

TISCopyCurrentKeyboardLayoutInputSource  (page 8)
>   Copies a text input source reference for the keyboard layout currently in use.

TISCopyCurrentASCIICapableKeyboardInputSource  (page 7)
>   Copies a text input source reference for the most recently used ASCII-capable keyboard input source.

TISCopyCurrentASCIICapableKeyboardLayoutInputSource  (page 8)
>   Copies a text input source reference for the most recently used ASCII-capable keyboard layout.

TISCopyInputSourceForLanguage  (page 9)
>   Copies a text input source reference for the input source that should be used to input the specified language.

TISCreateASCIICapableInputSourceList  (page 10)
>   Creates a list of ASCII-capable keyboard input sources.

## Manipulating Input Sources

## Registering an Input Source

# Functions

### TISCopyCurrentASCIICapableKeyboardInputSource

Copies a text input source reference for the most recently used ASCII-capable keyboard input source.

```
TISInputSourceRef TISCopyCurrentASCIICapableKeyboardInputSource (
   void
);
```

**Return Value**
A text input source reference for the most recently used ASCII-capable keyboard input source.

**Discussion**
If no ASCII-capable keyboard input source has been used yet, the function returns the default ASCII-capable keyboard layout chosen by Setup Assistant.

**Availability**
Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**
TISCopyCurrentKeyboardInputSource  (page 8)
TISCopyCurrentASCIICapableKeyboardLayoutInputSource  (page 8)

**Declared In**
TextInputSources.h

### TISCopyCurrentASCIICapableKeyboardLayoutInputSource

Copies a text input source reference for the most recently used ASCII-capable keyboard layout.

```
TISInputSourceRef TISCopyCurrentASCIICapableKeyboardLayoutInputSource (
    void
);
```

**Return Value**

A text input source reference for the most recently used ASCII-capable keyboard layout.

**Discussion**

If no ASCII-capable keyboard input source has been used, the function returns the default ASCII-capable keyboard layout chosen by Setup Assistant.

This function is used by input methods to get the keyboard layout used for key translation if no specific keyboard layout override exists. The TISCopyCurrentASCIICapableKeyboardInputSource (page 7) can return input sources that are not keyboard layouts.

**Availability**

Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**

TISCopyCurrentASCIICapableKeyboardInputSource (page 7)

**Declared In**

TextInputSources.h

### TISCopyCurrentKeyboardInputSource

Copies a text input source reference for the currently selected keyboard input source.

```
TISInputSourceRef TISCopyCurrentKeyboardInputSource (
    void
);
```

**Return Value**

A text input source reference to the currently selected keyboard input source.

**Availability**

Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**

TISCopyCurrentKeyboardLayoutInputSource (page 8)

**Declared In**

TextInputSources.h

### TISCopyCurrentKeyboardLayoutInputSource

Copies a text input source reference for the keyboard layout currently in use.

```
TISInputSourceRef TISCopyCurrentKeyboardLayoutInputSource (
   void
);
```

**Return Value**
A text input source reference to the keyboard layout currently in use.

**Discussion**
If the currently selected keyboard input source is a keyboard layout, the text input source reference refers to that layout. If the currently selected keyboard input source is an input method or mode, the text input source reference refers to the keyboard layout used by that input method or mode.

**Availability**
Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**
TISCopyCurrentKeyboardInputSource (page 8)

**Declared In**
TextInputSources.h

## TISCopyInputMethodKeyboardLayoutOverride

Copies a text input source reference for the currently selected input method's keyboard layout override.

```
TISInputSourceRef TISCopyInputMethodKeyboardLayoutOverride (
   void
);
```

**Return Value**
If the current keyboard input source is an input method or mode that has a keyboard layout override, a text input source reference for that keyboard layout is returned. Otherwise, the function returns NULL.

**Availability**
Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**
TISSetInputMethodKeyboardLayoutOverride (page 16)

**Declared In**
TextInputSources.h

## TISCopyInputSourceForLanguage

Copies a text input source reference for the input source that should be used to input the specified language.

```
TISInputSourceRef TISCopyInputSourceForLanguage (
   CFStringRef language
);
```

**Parameters**

*language*

> A `BCP 47` language code (in the same form returned by the Core Foundation function`CFLocaleCreateCanonicalLanguageIdentifierFromString function`) that represents the language an input source should be returned for.

**Return Value**

An enabled input source that can input the specified language. If more than one such input source exists and at least one has been used, the most recently used input source is chosen. If none has been used, one is chosen based on the intended languages of the input sources. If no enabled input source exists that can input the specified language, the function returns `NULL`.

**Discussion**

If a text field is expected to have input in a particular language, an application can call `TISCopyInputSourceForLanguage` and then `TISSelectInputSource` (page 15) to select an input source appropriate for that language. This function is a replacement for the deprecated KeyScript programming interface's capability to select the default input source associated with a particular script code.

**Availability**

Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**

`TISSelectInputSource` (page 15)

**Declared In**

`TextInputSources.h`


## TISCreateASCIICapableInputSourceList

Creates a list of ASCII-capable keyboard input sources.

```
CFArrayRef TISCreateASCIICapableInputSourceList (
   void
);
```

**Return Value**

A list of text input source references.

**Discussion**

This list represents a snapshot of ASCII-capable keyboard input sources enabled when the call was made. Successive calls to `TISCreateASCIICapableInputSourceList` may return different results because, for example, in between the calls the user may enable or disable an input source in the International Preferences pane. When a keyboard input source is enabled or disabled, by the user or programmatically, the distributed CF notification `kTISNotifyEnabledKeyboardInputSourcesChanged` (page 22) is posted.

**Availability**

Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**

`TISCopyCurrentASCIICapableKeyboardInputSource` (page 7)

**Declared In**
TextInputSources.h

## TISCreateInputSourceList

Creates a list of input sources that match specified properties.

```
CFArrayRef TISCreateInputSourceList (
    CFDictionaryRef properties,
    Boolean includeAllInstalled
);
```

**Parameters**

*properties*
> A dictionary of property keys and corresponding values to filter the input source list. May be NULL, in which case no filtering is performed.

*includeAllInstalled*
> Typically, set to false so that only enabled input sources are included; set to true to include all installed input sources that match the filter .

**Return Value**
A CFArrayRef for a list of text input source references that match the specified properties.

**Discussion**
This list represents a snapshot of input sources that matched the properties specified when you made the call. If you want to include input sources that are installed but not currently enabled, set the *includeAllInstalled* parameter to true. Typically, you do this to obtain a text input source reference for a newly installed input source. In this case, the properties parameter would include very specific criteria limiting the matching input sources.

> **Important:** Calling this function with the *includeAllInstalled* parameter set to true can have significant memory impact on the calling application if the properties parameter is NULL (match all) or if it specifies criteria that might match many installed input sources. This setting might force caching of data for all matching input sources and result in an allocation of up to 120K.
>
> If you are calling TISCreateInputSourceList to find a specific input source or sources from among the sources included in the list, first call TISCreateInputSourceList with *includeAllInstalled* set to false and check whether the returned array includes the desired input source(s). If this is not the case, call TISCreateInputSourceList again with the *includeAllInstalled* parameter set to true.

**Availability**
Available in Mac OS X v10.5 and later in Carbon.framework.

**Declared In**
TextInputSources.h

## TISDeselectInputSource

Deselects the specified input source.

```
OSStatus TISDeselectInputSource (
   TISInputSourceRef inputSource
);
```

**Parameters**

*inputSource*

    The input source you want to deselect.

**Return Value**

A result code. It returns `paramErr` if the input source is not deselectable. Otherwise, it returns `noErr`.

**Discussion**

`TISDeselectInputSource` is for use with palette or ink input sources only. It has no effect on other input sources. When palette input sources are disabled, the palette disappears. Ink input sources are usually deselected and disabled at the same time.

**Availability**

Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**

`TISSelectInputSource` (page 15)

**Declared In**

`TextInputSources.h`

## TISDisableInputSource

Disables the specified input source.

```
OSStatus TISDisableInputSource (
   TISInputSourceRef inputSource
);
```

**Parameters**

*inputSource*

    The input source you want to disable.

**Return Value**

A result code. The function returns `paramErr` if the input source cannot be disabled. Otherwise, it returns `noErr`.

**Discussion**

`TISDisableInputSource` is primarily intended for input methods, or for applications that supply their own input sources (for example, applications that provide keyboard layouts or palette input methods, and keyboard input methods that provide their own keyboard layouts and input modes). It makes the specified input source unavailable for selection and removes it from the user interface.

**Availability**

Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**

`TISEnableInputSource` (page 13)

**Declared In**

`TextInputSources.h`

## TISEnableInputSource

Enables the specified input source.

```
OSStatus TISEnableInputSource (
   TISInputSourceRef inputSource
);
```

**Parameters**

*inputSource*

> The input source you want to enable.

**Return Value**

A result code. The function returns `paramErr` if the input source cannot be enabled. Otherwise, it returns `noErr`.

**Discussion**

`TISEnableInputSource` is primarily for input methods or for applications that supply their own input sources (for example, applications that provide keyboard layouts or palette input methods, and keyboard input methods that provide their own keyboard layouts and input modes). It makes the specified input source available in the user interface for selection.

For `TISEnableInputSource` to succeed, the input source must be capable of being enabled (that is, `kTISPropertyInputSourceIsEnableCapable` (page 18) is set to `true`). Furthermore, if the input source is an input mode, its parent must already be enabled for the mode to become enabled.

**Availability**

Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**

`TISDisableInputSource` (page 12)

**Declared In**

`TextInputSources.h`

## TISGetInputSourceProperty

Gets the value of a specified property for a specified input source.

```
void* TISGetInputSourceProperty (
   TISInputSourceRef inputSource,
   CFStringRef propertyKey
);
```

**Parameters**

*inputSource*

> The text input source whose property value is requested.

*propertyKey*

> The property key constant specifying the desired property value.

**Return Value**

A pointer type appropriate for the value object associated with the property key. The specific pointer type is specified for each key. Typically, it is a CF type reference of some sort, but in one case it is an icon reference. The function might return `NULL` if the specified property is missing or invalid for the specified input source. You should not release the objects referred to by the pointer unless you have first retained them.

**Availability**

Available in Mac OS X v10.5 and later in Carbon.framework.

**Declared In**

`TextInputSources.h`

## TISInputSourceGetTypeID

Gets the `CFTypeID` of a text input source reference.

```
CFTypeID TISInputSourceGetTypeID (
    void
);
```

**Return Value**

The `CFTypeID` of a text input source reference, for comparison with the result of the Core Foundation's `CFGetTypeID` function.

**Availability**

Available in Mac OS X v10.5 and later in Carbon.framework.

**Declared In**

`TextInputSources.h`

## TISRegisterInputSource

Registers any new input sources in a file or bundle so a text input source reference can be obtained immediately for each new input source.

```
OSStatus TISRegisterInputSource (
    CFURLRef location
);
```

**Parameters**

*location*

> The location of the input sources in a file or bundle.

**Return Value**

A result code. The function returns `paramErr` if location is invalid or the input sources in the specified location cannot be registered. Otherwise, it returns `noErr`.

**Discussion**

This function enables an installer for an input method bundle or a keyboard layout file or bundle to notify the system to register these new input sources. The system can then locate the specified file or bundle and perform any necessary cache rebuilds so that the installer can immediately call `TISCreateInputSourceList` (page 11) with appropriate properties (for example, a bundle ID or input source ID) to get text input source references for one or more of the newly registered input sources.

You can only use this function to register the following:

■ Keyboard layout files or bundles in the `/Library/Keyboard Layouts/` or `~/Library/Keyboard Layouts/` directory (available to all users or to the current user, respectively). Such keyboard layouts, once enabled, are selectable.

■ Input method bundles in the new `Library/Input Methods/` or `~/Library/Input Methods/` directories (available to all users or to the current user, respectively).

> **Note:** Input method bundles can include private non-selectable keyboard layouts for use with `TISSetInputMethodKeyboardLayoutOverride` (page 16). These are registered automatically when the input method is registered, and do not need to be separately registered. They are not registered using `TISRegisterInputSource`.

*Security*: Any code that calls `TISRegisterInputSource` is part of an application or service that has already been validated in some way (for instance, by the user).

**Availability**
Available in Mac OS X v10.5 and later in Carbon.framework.

**Declared In**
`TextInputSources.h`

## TISSelectInputSource

Selects the specified input source.

```
OSStatus TISSelectInputSource (
    TISInputSourceRef inputSource
);
```

**Parameters**
*inputSource*
> The input source you desire to select.

**Return Value**
A result code. The function returns `paramErr` if the input source is not selectable. Otherwise, it returns `noErr`.

**Discussion**
When the input source is a selectable keyboard, the specified input source becomes the new current keyboard input source, and the previous input source is deselected. When the input source is a palette, that palette is displayed and made available for input. Ink input sources are typically enabled and selected at the same time. When you call `TISSelectInputSource` for a palette or ink input source, there is no effect on other input sources. When you call `TISSelectInputSource` for an already selected input source, there is, similarly, no effect.

For `TISSelectInputSource` to succeed, the input source must be selectable (that is, `kTISPropertyInputSourceIsSelectCapable` (page 18) is set to `true`) and the input source must be enabled (that is, `kTISPropertyInputSourceIsEnabled` (page 18) is set to `true`). Furthermore, if the input source is an input mode, its parent must be enabled for it to be selected.

**Availability**
Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**
`TISDeselectInputSource` (page 11)

**Declared In**
`TextInputSources.h`

## TISSetInputMethodKeyboardLayoutOverride

Sets the keyboard layout override for an input method or mode.

```
OSStatus TISSetInputMethodKeyboardLayoutOverride (
   TISInputSourceRef keyboardLayout
);
```

**Parameters**

*keyboardLayout*

>   `TISInputSourceRef` for the keyboard layout that should be used until the current input method is deactivated (if it should be something other than the most recently used ASCII-capable keyboard layout).

**Return Value**

A result code. The function returns `paramErr` if the current keyboard input source is not an input method or mode or if *keyboardLayout* does not designate a keyboard layout. Otherwise, it returns `noErr`.

**Discussion**

When an input method or mode is the selected input source, the most recently used ASCII-capable keyboard layout to translate key events is used. This keyboard layout is also the one that appears in Keyboard Viewer. An input source for this keyboard layout is returned by the function `TISCopyCurrentASCIICapableKeyboardLayoutInputSource` (page 8). If a different keyboard layout should be used for a particular input method or mode, the activated input method or mode should call `TISSetInputMethodKeyboardLayoutOverride` to specify the desired keyboard layout.

For example, when a Kotoeri user selects kana layout for kana input, Kotoeri should call `TISSetInputMethodKeyboardLayoutOverride` to set the kana keyboard as the override for the appropriate input modes.

The keyboard layout set in this way is used for the final stage of key translation in the Window Server, the connection, or application-specific key translation.

The override setting is lost when the input method that set it is deactivated.

The keyboard layout used for overriding need not be enabled or explicitly selectable. It can be a non-selectable layout included in an input method bundle and automatically registered.

The default behavior is new with Mac OS X 10.5, and eliminates the necessity that input methods have a user interface for setting the ASCII- capable keyboard for phonetic input based on Latin characters.

**Availability**

Available in Mac OS X v10.5 and later in Carbon.framework.

**See Also**

`TISCopyInputMethodKeyboardLayoutOverride` (page 9)

**Declared In**

`TextInputSources.h`

# Data Types

### TISInputSourceRef

Defines a pointer to an opaque CF object that unambiguously (within a single process) represents a text input source.

```
typedef struct __TISInputSource*  TISInputSourceRef;
```

**Discussion**
A text input source reference cannot be shared across processes.

**Availability**
Available in Mac OS X v10.5 and later in Carbon.framework.

**Declared In**
```
TextInputSources.h
```

# Constants

## Property Key Constants

CF string references that tell TISGetInputSourceProperty (page 13) what type of property to return; the returned properties are different specified types such as CFBooleanRef, CFArrayRef, and so forth. (Also used with TISCreateInputSourceList (page 11).

```
const CFStringRef kTISPropertyInputSourceCategory
const CFStringRef kTISPropertyInputSourceType
const CFStringRef kTISPropertyInputSourceIsASCIICapable
const CFStringRef kTISPropertyInputSourceIsEnableCapable
const CFStringRef kTISPropertyInputSourceIsSelectCapable
const CFStringRef kTISPropertyInputSourceIsEnabled
const CFStringRef kTISPropertyInputSourceIsSelected
const CFStringRef kTISPropertyInputSourceID
const CFStringRef kTISPropertyBundleID
const CFStringRef kTISPropertyInputModeID
const CFStringRef kTISPropertyLocalizedName
const CFStringRef kTISPropertyInputSourceLanguages
const CFStringRef kTISPropertyUnicodeKeyLayoutData
const CFStringRef kTISPropertyIconRef
const CFStringRef kTISPropertyIconImageURL
```

**Constants**
```
kTISPropertyInputSourceCategory
```
> This property key designates a property value, which is a `CFStringRef` indicating the category of an input source.
>
> Possible values are specified by the following property value constants: kTISCategoryKeyboardInputSource (page 21), kTISCategoryPaletteInputSource (page 21), and kTISCategoryInkInputSource (page 21).

`kTISPropertyInputSourceType`
> This property key designates a property value, which is a `CFStringRef` indicating the specific type of an input source.
>
> Possible values are specified by the following property value constants:
> `kTISTypeKeyboardLayout` (page 21), `kTISTypeKeyboardInputMethodWithoutModes` (page 21), `kTISTypeKeyboardInputMethodModeEnabled` (page 21), `kTISTypeKeyboardInputMode` (page 22), `kTISTypeCharacterPalette` (page 22), `kTISTypeKeyboardViewer` (page 22), and `kTISTypeInk` (page 22).

`kTISPropertyInputSourceIsASCIICapable`
> This property key designates a property value, which is a `CFBooleanRef` indicating whether the input source is intended to be capable of ASCII input.

`kTISPropertyInputSourceIsEnableCapable`
> This property key designates a property value which is a `CFBooleanRef` indicating whether the input source can ever be programmatically enabled using `TISEnableInputSource` (page 13).
>
> You can enable most input sources programmatically at any time. `kTISPropertyInputSourceIsEnableCapable` is set to `true` for these input sources.
>
> Some input sources can never be programmatically enabled. These are mainly input method private keyboard layouts used by the input method via the function `TISSetInputMethodKeyboardLayoutOverride` (page 16). You cannot directly enable these layouts, nor use them as keyboard layout input sources. `kTISPropertyInputSourceIsEnableCapable` is set to `false` for these.
>
> Some input sources can be programmatically enabled only under the correct conditions. These are mainly input modes, which can be changed from disabled to enabled only if their parent input method is enabled. However, they can already be in the enabled state, but not currently selectable if their parent input method is disabled. `kTISPropertyInputSourceIsEnableCapable` is `true` for these.

`kTISPropertyInputSourceIsSelectCapable`
> This property key designates a property value, which is a `CFBooleanRef` indicating whether the input source can ever be programmatically selected using `TISSelectInputSource` (page 15).
>
> This static property of an input source does not depend on any current state. For input sources that can be programmatically selected if they are enabled, `kTISPropertyInputSourceIsSelectCapable` is set to `true`.
>
> For input sources that can never be programmatically selected even if they are enabled, `kTISPropertyInputSourceIsSelectCapable` is set to `false`. Such sources are mainly input methods that have modes (parent input methods); only their modes can be selected.
>
> For input sources that are enabled and can only be programmatically selected under the correct conditions, `kTISPropertyInputSourceIsSelectCapable` is set to `true`. Such input sources are mainly input modes, which can only be selected if both they and their parent input method are enabled.
>
> For input sources that can never be enabled or selected, `kTISPropertyInputSourceIsSelectCapable` is set to `false`.

`kTISPropertyInputSourceIsEnabled`
> This property key designates a property value, which is a `CFBooleanRef` indicating whether the input source is currently enabled.

`kTISPropertyInputSourceIsSelected`
> This property key designates a property value, which is a `CFBooleanRef` indicating whether the input source is currently selected.

`kTISPropertyInputSourceID`

This property key designates a property value, which is a `CFStringRef` for the unique reverse DNS name associated with the input source.

1. For keyboard input methods and for palette or ink input sources, typically, the bundle ID, for instance, `com.apple.Kotoeri`.

2. For keyboard input modes, typically, the bundle ID of the parent input method plus a suffix that uniquely identifies the input mode, for instance, `com.apple.Kotoeri.Katakana`. It is not the generic input mode name used across input methods, for instance, `com.apple.inputmethod.Japanese.Katakana`.

3. For keyboard layouts, a new identification mechanism typically structured as `com.company.keyboardlayout.name`, for instance, `com.apple.keyboardlayout.US`.

`kTISPropertyBundleID`

This property key designates a property value, which is a `CFStringRef` for the reverse DNS BundleID associated with the input source.

`kTISPropertyInputModeID`

This property key designates a property value, which is a `CFStringRef` for a reverse DNS string that identifies a particular usage class for input modes.

For example, `com.apple.inputmethod.Japanese.Katakana` identifies a standard Katakana-input usage class that may be associated with input modes from several different input methods.

You can attach this input mode to a `TSMDocument` using the `TSMSetDocumentProperty` function with the tag `kTSMDocumentInputModePropertyTag`, to control the input mode usage class that should be used with that TSM document.

`kTISPropertyLocalizedName`

This property key designates a property value, which is a `CFStringRef` for the input source's localized name as intended for user interface use.

Uses the best match (determined by `CFBundle`) between the localization used by the caller and the available localizations of the input source name. In some cases, this might be an unlocalized name.

`kTISPropertyInputSourceLanguages`

This property key designates a property value, which is a `CFArrayRef` for an array of `CFStringRefs`, where each string is the language code for a language that can be input using the input source. You can use this constant with the `TISGetInputSourceProperty` (page 13) function, but you may not use it in the filter dictionary passed to the `TISCreateInputSourceList` (page 11) function.

Languages codes are in the same `BCP 47` form returned by the `CFLocaleCreateCanonicalLanguageIdentifierFromString` function. The first language code in the array is the language the input source is intended for. If no such language exists (for example, for the Unicode Hex Input keyboard layout), the first language code is an empty string.

kTISPropertyUnicodeKeyLayoutData

This property key designates a property value, which is a `CFDataRef` that refers to the `'uchr'` keyboard layout data for a keyboard layout input source.

The `'uchr'` data is in native-endian order. If the input source is not a keyboard layout, or is a keyboard layout for which only `'KCHR data'` is available, the value is `NULL`.

> **Note:** You may not use this key (and its corresponding value) used in the filter dictionary passed to the `TISCreateInputSourceList` (page 11) function.

kTISPropertyIconRef

This property key designates a property value, which is an `IconRef` value for the input source icon.

Icon references are the typical icon format for keyboard layouts and input methods. If an icon reference is not available for the specified input source, the value is `NULL`.

> **Note:** You may not use this key (and its corresponding value) used in the filter dictionary passed to the `TISCreateInputSourceList` (page 11) function.

kTISPropertyIconImageURL

This property key designates a property value, which is a `CFURLRef` indicating the file containing the image (typically TIFF) to be used as the input source icon.

If an image file URL is not available for the specified input source, the value is `NULL`. Note that other image formats (for example, JPEG, PNG) may also be used in the future.

> **Note:** You may not use this key (and its corresponding value) used in the filter dictionary passed to the `TISCreateInputSourceList` (page 11) function.

**Discussion**

Input sources may have additional properties, and some input sources do not have values for some of the properties described in this section. To obtain the property value for a particular input source, use the `TISGetInputSourceProperty` (page 13) function. To use a set of specific property key-value pairs as a filter when creating a list of input sources, call the `TISCreateInputSourceList` (page 11) function.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`TextInputSources.h`

## Input Source Category Property Values

Describe input source categories associated with various property keys.

```
const CFStringRef kTISCategoryKeyboardInputSource
const CFStringRef kTISCategoryPaletteInputSource
const CFStringRef kTISCategoryInkInputSource
```

**Constants**

kTISCategoryKeyboardInputSource

>One input source category value associated with the property key
>kTISPropertyInputSourceCategory (page 17).

>This category includes keyboard layouts, keyboard input methods (both with modes and without), and keyboard input modes. At least one input source in this category is installed. Of all input sources in this category, exactly one is selected; selecting a new one deselects the previous one.

kTISCategoryPaletteInputSource

>One input source category value associated with the property key
>kTISPropertyInputSourceCategory (page 17).

>This category includes character palettes and keyboard viewers. You can select 0 or more categories.

kTISCategoryInkInputSource

>One input source category value associated with the property key
>kTISPropertyInputSourceCategory (page 17).

>You can install or select 0 or 1 ink input source.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

TextInputSources.h

## Input Source Type Property Values

Input source type values associated with various property keys.

```
const CFStringRef kTISTypeKeyboardLayout
const CFStringRef kTISTypeKeyboardInputMethodWithoutModes
const CFStringRef kTISTypeKeyboardInputMethodModeEnabled
const CFStringRef kTISTypeKeyboardInputMode
const CFStringRef kTISTypeCharacterPalette
const CFStringRef kTISTypeKeyboardViewer
const CFStringRef kTISTypeInk
```

**Constants**

kTISTypeKeyboardLayout

>One input source type value associated with the property key kTISPropertyInputSourceType (page 18).

>This type belongs to the category kTISCategoryKeyboardInputSource (page 21).

kTISTypeKeyboardInputMethodWithoutModes

>One input source type value associated with the property key kTISPropertyInputSourceType (page 18).

>This type belongs to the category kTISCategoryKeyboardInputSource (page 21).

kTISTypeKeyboardInputMethodModeEnabled

>One input source type value associated with the property key kTISPropertyInputSourceType (page 18).

>This type belongs to the category kTISCategoryKeyboardInputSource (page 21).

`kTISTypeKeyboardInputMode`

One input source type value associated with the property key `kTISPropertyInputSourceType` (page 18).

This type belongs to the category `kTISCategoryKeyboardInputSource` (page 21).

`kTISTypeCharacterPalette`

One input source type value associated with the property key `kTISPropertyInputSourceType` (page 18).

This type belongs to the category `kTISCategoryPaletteInputSource` (page 21).

`kTISTypeKeyboardViewer`

One input source type value associated with the property key `kTISPropertyInputSourceType` (page 18).

This type belongs to the category `kTISCategoryPaletteInputSource` (page 21).

`kTISTypeInk`

One input source type value associated with the property key `kTISPropertyInputSourceType` (page 18).

This type belongs to the category `kTISCategoryInkInputSource` (page 21). Even though it is the only type in that category, a type is provided so that clients who don't need category information can just check input source type.

**Availability**
Available in Mac OS X v10.5 and later.

**Declared In**
`TextInputSources.h`

## Notifications

CF distributed notifications for changes to input sources.

```
const CFStringRef kTISNotifySelectedKeyboardInputSourceChanged
const CFStringRef kTISNotifyEnabledKeyboardInputSourcesChanged
```

**Constants**
`kTISNotifySelectedKeyboardInputSourceChanged`

The name of the CF distributed notification for a change to the selected keyboard input source.

`kTISNotifyEnabledKeyboardInputSourcesChanged`

The name of the CF distributed notification for a change to the set of enabled keyboard input sources.

**Availability**
Available in Mac OS X v10.5 and later.

**Declared In**
`TextInputSources.h`

# Document Revision History

This table describes the changes to *Text Input Source Services Reference*.

| Date | Notes |
|------|-------|
| 2007-06-07 | New document that describes the API for operating on text input sources, such as keyboards, character palettes, and ink input. |

# Index

**25**