

String Constants: GUI Internationalization

Version 8.12.900

April 24, 2024

This library provides the facility for multiple languages in DrRacket's GUI.

1 Using String Constants

```
(require string-constants)      package: string-constants-lib
```

```
(string-constant name)
```

This form returns the string constant named *name* for this-language.

```
(string-constants name)
```

This form returns a list of string constants, one for each language that DrRacket's GUI supports.

```
(string-constant-in-current-language? name)
```

Produces *#t* if *key* has been translated for this-language.

```
(dynamic-string-constant name) → string?  
  name : string-constant?
```

This, like `string-constant`, returns the string constant named *name*, but without any compile-time checking on the argument.

```
(dynamic-string-constants name) → (listof string?)  
  name : string-constant?
```

This, like `string-constants`, returns the string constants matching *name*, but without any compile-time checking on the argument.

```
(string-constant? v) → boolean?  
  v : any/c
```

Returns *#t* if *v* is a symbol naming a known string constant.

```
(dynamic-string-constant-in-current-language? key) → boolean?  
  key : string-constant?
```

Like `string-constant-in-current-language?`, but without compile-time checking.

```
(string-constant-language? v) → boolean?  
  v : any/c
```

Determines if *v* is a symbol naming a language that is supported by the string constants library. The symbol `'english` is guaranteed to return *#t*.

```
(call-with-current-language sc-language
                             thunk) → any
sc-language : string-constant-langauge?
thunk : (-> any)
```

Calls *thunk*; in the dynamic extent of the call, all uses of `string-constant` and `dynamic-string-constant` will use string constants from *sc-language*.

This function is designed to facilitate testing of libraries that use string constants, so they can work regardless of the user's language setting.

```
(this-language)
```

This form returns the name of the current language as a symbol.

```
(all-languages)
```

This form returns a list of symbols (in the same order as those returned from `string-constants`) naming each language.

```
(set-language-pref lang) → void?
lang : string?
```

Sets the language for the next run of DrRacket to *lang*, which must be a symbol returned from `all-languages`. Does not affect the running DrRacket.

2 Adding String Constants

To add string constants to DrRacket, see the file "private/english-string-constants.rkt" and the other string constants files in the "private" directory. (Some string constants files that have a less permissive license are also available on the pkg server in the string-constants-lib-lgpl pkg.)

Each file has the same format. They are each modules in the `string-constants/private/string-constant-lang` language. The body of each module is a finite mapping table that gives the mapping from the symbolic name of a string constant to its translation in the appropriate language. Multiple string constants that appear together are implicitly concatenated.

The "english-string-constants" is considered the master file; string constants will be set there and translated into each of the other language files. In addition, the "english-string-constants.rkt" file should contain hints about the context of the strings whose symbol name might not be clear.

To add a new set of string constants, pull requests on the string constants repository are welcome, but adding a definition of `string-constants-info` to a value that satisfies the contract

```
(listof (list/c symbol? regexp? regexp?
              (and/c module-path? (not/c path-string?))))
```

to a collection-level "info.rkt" file allows string constants to be added via other packages. The first element of each list must be the name of the language, the second and third are regular expressions that are used to match the result of `system-language+country` to determine if the string constant set should be used (by default). The final element is an absolute module path to the file containing the string constants.

3 Language Environment Variables

- `PLTSTRINGCONSTANTS`
- `STRINGCONSTANTS`

If either of these environment variables are set, DrRacket shows you, during startup, which string constants are not yet defined for each language.

You can also specify which languages you are interested in. If either environment variable is bound to a symbol (as interpreted by `read`) you see only the corresponding language's messages. If either one is bound to a list of symbols (again, as interpreted by `read`) you see the messages for all the languages in the list. If either is bound to anything else, you see all of the languages.

The `PLTSTRINGCONSTANTS` environment variable takes precedence over the `STRINGCONSTANTS` environment variable.

The `PLTSTRINGCONSTANTSLANG` controls the language choice, overriding the default saved in the preferences file. If it is not set to one of the languages in the result of `all-languages`, it is ignored.