

NAME

minput_get_variable – Get information about input method variable(s).

SYNOPSIS

Mplist* minput_get_variable (**M**Symbol *language*, **M**Symbol *name*, **M**Symbol *variable*)

DESCRIPTION

Get information about input method variable(s). The **minput_get_variable()** function returns information about variable **variable** of the input method specified by **language** and **name**. An input method variable controls behavior of an input method.

There are two kinds of variables, global and local. A global variable has a global definition, and the description and the value may be inherited by a local variable. Each input method defines a local variable which has local value. It may also declare a local variable that inherits definition of a global variable of the same name.

If **language** is **Mt** and **name** is **Mnil**, information about a global variable is returned. Otherwise information about a local variable is returned.

If **variable** is **Mnil**, information about all variables is returned.

The return value is a *well-formed* plist (**Property List**) of this format:

((NAME DESCRIPTION STATUS VALUE [VALID-VALUE ...]) ...)

NAME is a symbol representing the variable name.

DESCRIPTION is an **M**-text describing the variable, or **Mnil** if the variable has no description.

STATUS is a symbol representing how the value is decided. The value is **Mnil** (the default value), **Mcustomized** (the value is customized by per-user customization file), or **Mconfigured** (the value is set by the call of **minput_config_variable()**). For a local variable only, it may also be **Minherited** (the value is inherited from the corresponding global variable).

VALUE is the initial value of the variable. If the key of this element is **Mt**, the variable has no initial value. Otherwise, the key is **Minteger**, **Msymbol**, or **Mtext** and the value is of the corresponding type.

VALID-VALUES (if any) specify which values the variable can have. They have the same type (i.e. having the same key) as **VALUE** except for the case that **VALUE** is an integer. In that case, **VALID-VALUE** may be a plist of two integers specifying the range of possible values.

If there no **VALID-VALUE**, the variable can have any value as long as the type is the same as **VALUE**.

If **variable** is not **Mnil**, the first element of the returned plist contains the information about **variable**.

RETURN VALUE

If the requested information was found, a pointer to a non-empty plist is returned. As the plist is kept in the library, the caller must not modify nor free it.

Otherwise (the specified input method or the specified variable does not exist), **NULL** is returned.

COPYRIGHT

Copyright (C) 2001 Information-technology Promotion Agency (IPA)

Copyright (C) 2001–2011 National Institute of Advanced Industrial Science and Technology (AIST)

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License <<http://www.gnu.org/licenses/fdl.html>>.

