

NAME

minput_config_command – Configure the key sequence of an input method command.

SYNOPSIS

```
int minput_config_command (MSymbol language, MSymbol name, MSymbol command, MPlist *keyseqlist)
```

DESCRIPTION

Configure the key sequence of an input method command. The **minput_config_command()** function assigns a list of key sequences **keyseqlist** to the command **command** of the input method specified by **language** and **name**.

If **keyseqlist** is a non-empty plist, it must be a list of key sequences, and each key sequence must be a plist of symbols.

If **keyseqlist** is an empty plist, any configuration and customization of the command are cancelled, and default key sequences become effective.

If **keyseqlist** is NULL, the configuration of the command is canceled, and the original key sequences (what saved in per-user customization file, or the default one) become effective.

In the latter two cases, **command** can be **Mnil** to make all the commands of the input method the target of the operation.

If **name** is **Mnil**, this function configures the key assignment of a global command, not that of a specific input method.

The configuration takes effect for input methods opened or re-opened later in the current session. In order to make the configuration take effect for the future session, it must be saved in a per-user customization file by the function **minput_save_config()**.

RETURN VALUE

If the operation was successful, this function returns 0, otherwise returns -1. The operation fails in these cases:

- **keyseqlist** is not in a valid form.
- **command** is not available for the input method.
- **language** and **name** do not specify an existing input method.

SEE ALSO

minput_get_commands(), **minput_save_config()**.

Example:

```
/* Add 'C-x u' to the 'start' command of Unicode input method. */
{
    MSymbol start_command = msymbol ('start');
    MSymbol unicode = msymbol ('unicode');
    MPlist *cmd, *plist, *key_seq_list, *key_seq;

    /* At first get the current key-sequence assignment. */
    cmd = minput_get_command (Mt, unicode, start_command);
    if (! cmd)
    {
        /* The input method does not have the command 'start'. Here
         should come some error handling code. */
    }
    /* Now CMD == ((start DESCRIPTION STATUS KEY-SEQUENCE ...) ...).
     Extract the part (KEY-SEQUENCE ...). */
    plist = mplist_next (mplist_next (mplist_next (mplist_value (cmd))));
    /* Copy it because we should not modify it directly. */
    key_seq_list = mplist_copy (plist);
```



```
key_seq = plist();
plist_add (key_seq, Msymbol, msymbol ('C-x'));
plist_add (key_seq, Msymbol, msymbol ('u'));
plist_add (key_seq_list, Mplist, key_seq);
m17n_object_unref (key_seq);

minput_config_command (Mt, unicode, start_command, key_seq_list);
m17n_object_unref (key_seq_list);
}
```

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>>.

