LMX_Hostid

The information on this page refers to LM-X v5.3.1 and newer, which added support for Raspberry Pi HostIDs. If you are using an older version of LM-X, refer to documentation for earlier versions.

The LMX_Hostid function retrieves the HostID values from the computer system.

Prototype

```
LMX_STATUS LMX_Hostid
(
   LMX_HANDLE LmxHandle,
   LMX_HOSTID_TYPE eHostidType,
   LMX_HOSTID *lpHostid,
   int *npHostids
);
```

Parameters

LmxHandle
[in/out] LM-X handle.

eHostidType
[in] Value that specifies the HostID type to be retrieved.

Possible values are:

HostID Type	Description
LMX_HOSTID_ETHERNET	Network card HostID
LMX_HOSTID_USERNAME	Username HostID
LMX_HOSTID_HOSTNAME	Hostname HostID
LMX_HOSTID_IPADDRESS	IP address HostID
LMX_HOSTID_CUSTOM	Custom HostID
LMX_HOSTID_DONGLE_HASPHL	HaspHL Dongle HostID
LMX_HOSTID_HARDDISK	HostID of physical harddisk
LMX_HOSTID_LONG	System-specific HostID
LMX_HOSTID_BIOS	Bios HostID
LMX_HOSTID_WIN_PRODUCT_ID	Windows product ID
LMX_HOSTID_AWS_INSTANCE_ID	Amazon EC2 Instance ID
LMX_HOSTID_GCE_ID	Google Compute Engine ID
LMX_HOSTID_AZURE_ID	Microsoft Azure ID
LMX_HOSTID_RPI_SN	Raspberry Pi serial number
LMX_HOSTID_ALL	All HostIDs

lpHostid
[out] Pointer to array of LMX_HOSTID structures. See lmx.h for a description of LMX_HOSTID. The array must be of size LMX_MAX_HOSTIDS.

npHostids
[out] Pointer to a variable that will hold the number of HostIDs of a specific type retrieved. If no HostIDs are available of the type requested, this variable will be set to zero.

Return values

On success, this function returns the status code LMX_SUCCESS.

On failure, this function returns an error code in the format described in Return codes.

Remarks

To make use of custom HostIDs, you must set a callback function using LMX_SetOption with the flag LMX_OPT_CUSTOM_HOSTID_FUNCTION.

Example

You can use the following code to retrieve the HostIDs that are currently in use and list information contained in LMX_HOSTID.

```
#include <lmx.h>
#include <stdio.h>

LMX_HANDLE h;

int main()
{
  LMX_HOSTID hostID[LMX_MAX_HOSTIDS];
  int nbHosts, i;

  exit_on_error(LMX_Init(&h));
  exit_on_error(LMX_Hostid(h, LMX_HOSTID_ALL, hostID, &nbHosts));
  printf("HostIDs found: %d\n", nbHosts);

  for(i = 0; i < nbHosts; ++i)
  {
    printf("Host Type: %d\n", hostID[i].eHostidType);
    printf("Description: %s\n", hostID[i].szDescription);
    printf("Value: %s\n\n", hostID[i].szValue);
  }

  return 0;
}
```