

# HostID values

*The information on this page refers to LM-X v5.5 and newer, which added support for 32-bit Linux ARM. If you are using an older version of LM-X, refer to [documentation for earlier versions](#).*

For maximum flexibility in your licensing options, you can define the HostID values for your protected application within the license rather than inside the protected application. This enables you to license your application differently for different users, without requiring your application to be recompiled.

**Note:** The maximum length of a HostID value is 256 bytes.

You can use the `lmxendutil` utility or the LM-X End-user Configuration Tool to print out the valid HostIDs for a particular machine.

The following HostIDs are supported in the license:

HostID Type	Description	Platform Availability
LMX_HOSTID_ETHERNET	Network card HostID	All
LMX_HOSTID_USERNAME	Username HostID	All
LMX_HOSTID_HOSTNAME	Hostname HostID	All
LMX_HOSTID_IPADDRESS	IP address HostID	All
LMX_HOSTID_CUSTOM	Custom HostID	All
LMX_HOSTID_DONGLE_HASPHL	HaspHL Dongle HostID	Windows (x86 and x64) Linux (x86, x64 and arm64) Mac OS
LMX_HOSTID_HARDDISK	HostID of physical harddisk	Windows (x86 and x64; <i>not</i> available for use with MinGW compiler)
LMX_HOSTID_LONG	System-specific HostID	Mac OS
LMX_HOSTID_BIOS	Bios HostID	Windows (x86 and x64; <i>not</i> available for use with MinGW compiler)
LMX_HOSTID_WIN_PRODUCT_ID	Windows Product ID	Windows (x86 and x64)
LMX_HOSTID_AWS_INSTANCE_ID	Amazon EC2 Instance ID	All
LMX_HOSTID_GCE_ID	Google Compute Engine ID	All
LMX_HOSTID_AZURE_ID	Azure ID	All
LMX_HOSTID_RPI_SN	Raspberry Pi serial number	Linux arm32 and arm64

For help on deciding which HostID or combination of HostIDs fits your needs, see [Determining which HostID to use](#). For complete information about the operating system versions that are supported by LM-X, see [Supported platforms](#).

## LMX\_HOSTID\_ETHERNET

For license generation, LM-X can handle different Ethernet HostID formats in the license template based on the operating system representation of the address when using `ipconfig` (Windows) or `ifconfig` (Unix); for example, AABCCDDEEFF, AA:BB:CC:DD:EE:FF or AA-BB-CC-DD-EE-FF.

## LMX\_HOSTID\_DONGLE\_HASPHL

LMX\_HOSTID\_DONGLE\_HASPHL is a HostID representing physical dongles. You may use 3rd-party dongles, which require some custom programming, or purchase dongles from X-Formation (provided by Aladdin) that work out of the box, without need for further customization. While dongles provide the best possible security of the aforementioned HostIDs, they come with additional distribution overhead.

To use dongles on Windows, you should instruct your end users to plug in the dongle and let Windows find the device driver automatically using Windows Update. For other platforms, you can [download the appropriate device driver from our website](#).

When LMX\_HOSTID\_DONGLE\_HASPHL is used with a license server, background checking is done to ensure that users don't remove the dongle from the server machine when serving licenses. Doing so will cause the license server to stop functioning after a short time.

When using LMX\_HOSTID\_DONGLE\_HASPHL with local standalone licenses, the client application should ensure that the dongle is not removed during client runtime.

You can check the dongle by calling [LMX\\_Heartbeat](#) on a separate thread continuously, or by ensuring that LMX\_HOSTID\_DONGLE\_HASPHL HostID is in use and comparing the HostID value against the last known good value, as demonstrated in the following example.

```

int i;
LMX_FEATURE_INFO FI;
LMX_HOSTID LmxHostid[LMX_MAX_HOSTIDS];
int nHostids;
LMX_GetFeatureInfo(LmxHandle, "my_app", &FI);
/* Go through each hostid used for this particular feature */
for (i = 0; i < FI.nClientLicenseHostids; i++)
{
    /* See if the hostid is a dongle hostid */
    if (FI.ClientLicenseHostid[i].nHostidType == LMX_HOSTID_DONGLE_HASPHL)
    {
        /* See if hostid function reports the same value as we used when checking out the license */
        if (LMX_Hostid(LmxHandle, LMX_HOSTID_DONGLE_HASPHL, LmxHostid, &nHostids) != LMX_SUCCESS)
            return BAD_DONGLE;
        if (nHostids != 1)
            return BAD_DONGLE;
        /* Compare if the hostid at checkout time matches the hostid at present time */
        if (strcmp(FI.ClientLicenseHostid[0].szValue, LmxHostid[0].szValue) != 0)
            return BAD_DONGLE;
    }
}

```

This check should preferably be done every few minutes to ensure that users do not move the dongle to other workstations and overuse standalone licenses.

See [LMX\\_Hostid](#) for further information on LMX\_Hostid.