LMX_Checkout

The LMX_Checkout function will checkout one or more licenses for a specific feature.

Prototype

```
LMX_STATUS LMX_Checkout
(
   LMX_HANDLE LmxHandle,
   const char *szFeatureName,
   int nVerMajor,
   int nVerMinor,
   int nCount
);
```

Parameters

LmxHandle
[in/out] LM-X handle.

szFeatureName
[in] Feature name.

nVerMajor
[in] Major version number, in the range 0-9999.

nVerMinor
[in] Minor version number, in the range 0-9999.

nCount
[in] Number of licenses to checkout.

This value can be one of the following:

An integer in the range 1-2147483647	Lets you set the count to the specified number.
LMX_LOGICAL_CPU_COUNT	Lets you implement processor-based licensing by setting the count to the number of logical CPUs.
LMX_PHYSICAL_CPU_COUNT	Lets you implement processor-based licensing by setting the count to the number of physical CPUs.

Return values

On success, this function returns the status code LMX_SUCCESS.

If softlimit licensing is enabled and the license count has exceeded the specified softlimit licenses, this function can return the status code LMX_SOFTLIMIT, which also indicates success.

On failure, this function returns an error code in the format described in Return codes. Note that the error code returned reflects only the last license file tested or license server contacted. (See the execution order in Remarks, below.)

To get a complete error description, use the API function LMX_GetErrorMessage().

Remarks

See Search paths in the *LM-X End Users Guide* for more information about how the LMX_Checkout method finds the license.

For local license files, the count value is ignored and checking will depend only on HostID, version, etc.

By default, the version number requested in the function call can be lower than the one available in the license file.

This function will attempt to retrieve the feature from either a license server or a local license file. If the feature is available on a license server, all other feature requests will go to the same license server. LM-X will establish the license server connection transparently to the client application. The execution order is as follows:

1. Try to checkout feature from Borrow licenses store.
2. Try to checkout feature from string.
3. Try to checkout feature from local path.
4. If network licenses are enabled:
    a. Try to checkout feature from license server.
    b. In case of an error other than LMX_NOT_ENOUGH_LICENSES, try to checkout feature from Grace licenses store.
5. Try to checkout feature from Trial licenses store.

It is recommended that all licenses come from the same source (that is, the same local license file or the same license server) or as few sources as possible for simplicity and to make licensing most straightforward for users. However, it is technically possible to take licenses from multiple license files and multiple license servers at the same time. See License server for more information.

It is also recommended that you do as few checkouts as possible to fulfill requests. Preferably, checkouts should map directly to the number of licenses you sell to your customer. This helps ensure that customers understand the products they have bought and the licenses associated with them.

When License queuing is enabled, LMX_QUEUE will be returned (see Return codes) if a license checkout is not successful, after which you should call LMX_Checkout again to see if a license is available. The queue request is made simultaneously to all the license servers to which the client connects. When one of the servers satisfies the client request, the queue request is removed from all of the servers.

For information on how to increase license checkout performance, see Optimizing license checkout speed.

Example

The following example illustrates the process of checking out a license for feature "f2", version 1.0.

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

LMX_HANDLE h;

int main()
{
  exit_on_error(LMX_Init(&h));
  exit_on_error(LMX_Checkout(h, "f2", 1, 0, 1));
  return 0;
}
```