LMX_ServerFunction

LMX_ServerFunction sends a message to the license server for processing for a checked out feature. The license server will invoke the specified custom callback function and return a response.

Prototype

```
LMX_STATUS LMX_ServerFunction
(
   LMX_HANDLE LmxHandle,
   const char *szFeatureName,
   char *szMessage
);
```

Parameters

LmxHandle
[in/out] LM-X handle.

szFeatureName
[in] Feature name.

szMessage
[in/out] The message to send and the response.

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

This function will send a message to the server from which a feature was checked out.

The length of the message sent to the server and the response can be up to LMX_MAX_LONG_STRING_LENGTH bytes.

In order for this function to succeed, the license server must have a function callback implemented. See lmx_server_conf.c for details.

Example

The following example shows a function that could be registered in lmx_server_conf.c:

```
/* The function parameters should be overwritten by the response back to the client. */
void LMX_CALLBACK ServerFunc(char *szMessage)
{
  char *szTest = "this is a simple response";
  LmxLogprintf("server function called: %s", szMessage);

  /* Use the client message in some way ...*/
  ...
  ...
  ...
  /* Prepare response */
  strcpy(szMessage, szTest);
}
```

The client can then invoke the function given above as follows:

```
char szMyMessage[LMX_MAX_LONG_STRING_LENGTH+1];
char *szString = "This is my simple query";
int nLength;
LMX_STATUS LmxStat;
/* Specify a message to send (here we use a zero-terminated string, but any data could be used */
strcpy(szMyMessage, szString);
LmxStat = LMX_ServerFunction(handle, "f1", szMyMessage);
/* Now the query has been replaced with a response */
/* Print out the response since we know the server sent us a string */
printf("%s\n", szMyMessage);
```