# CHANGING A LIGHTBULB

## The Microsoft Way

*Written after reading the Windows NT Driver Manual for the 12th Time...*

Lightbulbs are an important and useful adjunct to your Windows NT System. Changing and modifying lightbulbs in use on the system, both by standard drivers and by applications, is controlled by the Lightbulb Handler module in your Windows NT system. The changing of lightbulbs in Windows NT is a dynamic process, carried out by the **Lightbulb Scheduler** and **Lightbulb Installer** modules (LGBSCHED.MOD and LGBINST.MOD) in your Windows NT directory. Control of these processes is initiated by a **Lightbulb Insertion Request (LIR)** or **Lightbulb Removal Request (LRR)**..Both of these are initiated by the user, passed to the Lightbulb Installer and proceed without further user input. The status of an LIR or LRR can be monitored with standard calls.

Prepare for the insertion process by creating a new LIR with **CreateLIR.** Pass your LIR a handle to your new lightbulb. (Note: if you are using non-standard lightbulbs: that is, those without Microsoft-registered handles, it will be necessary to create a handle of type **HLIGHTBULB** and assign it to your lightbulb. You can create a new handle with **CreateLightbulbHandle** or assign an existing handle with **AssignLightbulbHandle**. Remember that after your LIR has been completed with a non-Microsoft-compliant lightbulb, to remove the lightbulb handle with **RemoveLightbulbHandle**. DO NOT use **RemoveLightbulbHandle** with Microsoft-approved lightbulbs!)

Once the LIR is created, assign a socket to it. Sockets may be referenced by pointers or by handles of type **HSOCKET**. If using a pointer to a handle, If using a pointer to a socket, remember that its handle must be pre-assigned. **CreateSocketHandle** will perform this task. Normally it is not necessary to call **RemoveSocketHandle** until the socket is deleted. Socket assignment is done with **AssignSocketToLIR** or **AssignSocketHandleToLIR**.

Similarly, assign a lightbulb handle to the LIR using **AttachLightbulb**. It is possible to pass a handle of type **HSOCKET** to **AttachLightbulb**. Scheduling the resultant LIR can cause attachment of a socket to another socket, which can then lead to unforeseen results if **TurnOnLightbulb** is used improperly.

Next, assign a priority with **AssignLIRPriority**. Legal values are **PRIORITY\_NORMAL, PRIORITY\_DARK** and **PRIORITY\_I\_CANT\_SEE**. Note that if an LIR is scheduled with a priority of **PRIORITY\_I\_CANT\_SEE** on a socket which has already returned **ERROR\_SOCKET\_FULL**, an executed operator fault may result.

Now the completed LIR can be inserted in the Lightbulb Scheduler queue. Obtain the Lightbulb Scheduler queue as a file handle with **CreateFile** with an argument of "\\.LGBSCHED" and pass the address of your LIR to this queue with **ScheduleLightbulbRequest**. DO NOT ATTEMPT **TurnOnLightbulb** until your LIR has signalled completion! You can test for completion of your LIR in two ways. One is to assign a semaphore to your LIR, using **CreateSemaphore,** and testing this semaphore with **WaitForSiingleEvent** or **WaitForMultipleEvents**.

Another way of testing your LIR is to issue an **LIR Status Requestor** to the Lightbulb Scheduler. **CreateLIRStatusRequestor** will create an LIR Status Requestor and return its handle (of type **HREQUESTOR**.). You then assign the requestor to an existing LIR with **AssignLIRStatusRequestor**. Note that if the requestor is already assigned to a different LIR, it must first be de-assigned with **DeassignLIRStatusRequestor**. The completed requestor can then be submitted to the Lightbulb Scheduler with **ScheduleLightbulbRequest**, just as you previously assigned the LIR. Unlike the LIR, the LIR Status Requestor returns immediately with values **NO\_ERROR, ERROR\_PENDING**, or with an error code. (See the description of **ScheduleLightbulbRequest** for error codes.)

After use, be sure to remove the LIR Status Requestor and free its handle with R**emoveLIRStatusRequestor.** The LIR itself should also be removed with **RemoveLIR**.

Please note that when an LIR returns **ERROR\_SOCKET\_FULL**, it will be necessary to remove the lightbulb now present in that socket. For that purpose, you should issue a **Lightbulb Removal Request (LRR)** to the Lightbulb Scheduler. See **CreateLRR, AssignSocketToLRR, AssignLRRPriority, CreateLRRStatusRequestor, AssignLRRStatusRequestor,DeassignLRRStatusRequestor, RemoveLRRStatusRequestor, DetachLightbulb**, and **RemoveLRR** for further details.