

## **Call Lock**

## Table of Contents

1	Introduction.....	3
2	Important information .....	3
3	Call Lock.....	3

## 1 Introduction

The Call Lock functionality ensures that two (or more) softphones do not use the same Jabra device at the same time - it does so by controlling access to the call control functionality in the SDK library.

## 2 Important information

Please note that Call Lock is not an absolute lock; it is a semaphore to assist softphones using the same device without interfering with each other. It restricts access to the SDK library, not the device itself.

If you have two softphones that use the same device interference may still occur if only one of the softphones implements the Call Lock functionality.

Please also note that if you have multiple Jabra devices connected to the same computer, you will not be able to have two concurrent call-control sessions.

Call Lock will restrict access to the library, not just the active device, hence two different softphones using two different Jabra devices cannot perform call-control at the same time if they both implement the Call Lock functionality.

## 3 Call Lock

You acquire a 'Call Lock' before you perform any form of call control. The Call Lock informs other applications that use the SDK to "keep hands-off". The lock is kept for the duration of the call, once the call is finished, you release the lock hereby allowing other softphones to use the device.

By acquiring a lock beforehand softphones can co-exist without interfering with each other - if acquiring the lock fails another softphone is currently using the device.

Call Lock consists of three API functions:

- `Jabra_GetLock()`
- `Jabra_ReleaseLock()`
- `Jabra_IsLocked()`

Use '`Jabra_GetLock`' to acquire the lock, use '`Jabra_ReleaseLock`' to release the lock.

'`Jabra_IsLocked`' reads the current state of the lock within your application. This can be used to check if your application already has obtained the lock; using '`Jabra_IsLocked`' does not affect performance and has been provided as a convenient way of checking this without using '`Jabra_GetLock`'.

Depending on the situation, the lock might already be acquired by your application, hence an internal check for the lock should be the first step in acquiring the lock. Executing multiple features in quick succession is an example of when the lock might already be acquired.

Note that '`Jabra_IsLocked`' does not check if another softphone has obtained a lock on the device. You must use '`Jabra_GetLock`' to see if you can obtain the lock or not.

The current Call Lock functionality is backward compatible with the one introduced in the first generation of Jabra SDK libraries (C# only).