

# Developing Applications With The C/C++ API

In order to create a program that uses the API, you must `#include jcam_dll.h` in your source code and link to the `jcam_dll.lib` file in your project. In Visual C++, this is accomplished by going to:

## Project > Properties > Linker > Input > Additional Dependencies

From there, enter the location of the `jcam_dll.lib` file. We suggest that you put it in your project directory.

## Static Linking

If you statically link the API functions into your own application, you won't need to distribute `jcam_dll.dll` with your application. However, you do need to do five things:

- Link `jcam_dll.lib` with your project.
- `#define JCAM_STATIC_LIB`
- `#include jcam_dll.h`
- Use the `joescan` namespace.
- Call `jsInitialize()` before calling any other Joescan API functions.

```
#define JCAM_STATIC_LIB
#include "jcam_dll.h"
using namespace joescan;

main(int argc, char ** argv)
{
    if(jsInitialize() == FALSE)
        exit(0);

    //Your code goes here...
}
```

## Dynamic Linking

If you dynamically link the API functions to your own application, you will need to distribute `jcam_dll.dll` with your application. `jsInitialize()` will be automatically called by the Windows dynamic library loader. That leaves you with five things you must do:

- Link `jcam_dll.lib` with your project.
- Distribute `jcam_dll.dll` with your application.
- `#define JCAM_DYNAMIC_LIB`
- `#include jcam_dll.h`
- Use the `joescan` namespace.

You should put the `jcam_dll.dll` file in your project's Debug and Release directories so your program runs when you test it. When you distribute your application, the `jcam_dll.dll` file must be in the search path for the program, probably the install directory.

```
#define JCAM_DYNAMIC_LIB
#include "jcam_dll.h"
using namespace joescan;
```

All the symbolic constants used in this manual are defined in the file `jcam_dll.h`.

## The joescan Namespace

If you're using C++, you can either use the `joescan` namespace as above and in the examples, or you can fully qualify its members. For example:

```
joescan::JCONNECTION jc = joescan::jsOpenConnection("192.168.1.105");
```