

# How to Use LuaSockets from AutoPlay Media Studio 6.0

## Introduction

AutoPlay Media Studio 6.0 now has the ability to make use of the built-in loadlib function which enables you to use many standard Lua libraries that are available.

**Disclaimer:** LuaSocket is a third-party Lua extension library and is not supported in any way by Indigo Rose Corporation. This article is for educational purposes only. Use LuaSocket completely at your own risk.

## What is LuaSocket?

From the LuaSocket Web site:

“LuaSocket is a Lua extension library that is composed by two parts: a C core that provides support for the TCP and UDP transport layers, and a set of Lua modules that add support for functionality commonly needed by applications that deal with the Internet.”

Basically, the LuaSocket library allows you to do both low-level and high-level TCP and UDP communication.

## Download and Install LuaSocket

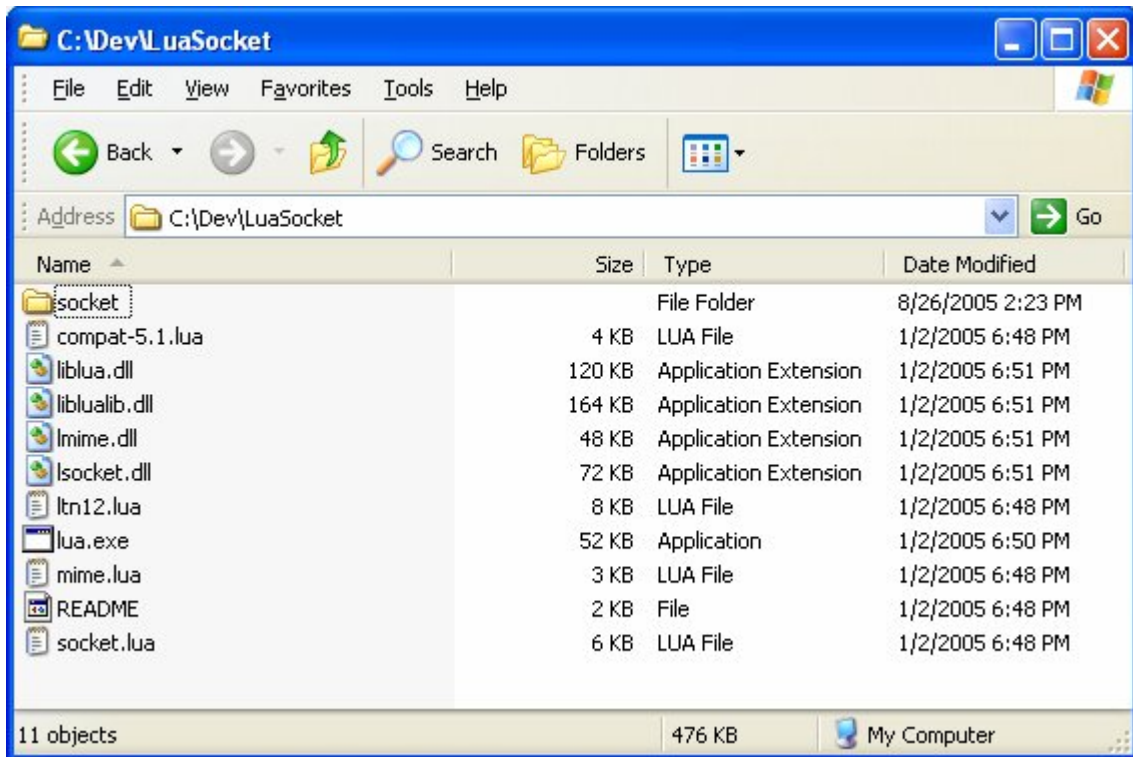
The first step is to download and install the LuaSocket library.

Go to:

<http://www.cs.princeton.edu/~diego/professional/luasocket/>

Then go to the Download page and download the “PC Win32 binaries”. Don’t download the source code (unless you are ambitious).

Once downloaded, unzip the package to a folder on your hard drive. The directory structure should look something like this:



## Add LuaSockets Files to Your Project

Now start a new, blank project in AutoPlay Media Studio 6.0 or open an existing project that you want to use LuaSocket in.

Next, open the Project Browse pane (View > Panes > Project Browser). In the Project Browser pane, open the “Scripts” folder. Drag and drop the following files from your LuaSocket folder into your “Scripts” folder:

Sockets (the whole folder with its contents)

compat-5.1.lua

liblua.dll

liblualib.dll

lmime.dll

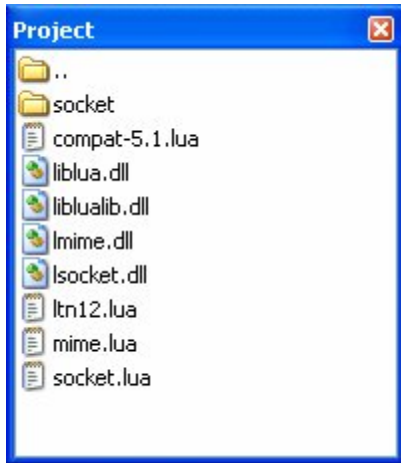
lsocket.dll

ltn12.lua

mime.lua

socket.lua

Your Project Browser should now look like this:



## Initialize LuaSocket

In your project, go to your Global Functions (Project > Global Functions) window and enter the following:

```
-- Load compat for Lua 5.1 package proposal compatibility
require ("compat-5.1.lua");
package.cpath = _SourceFolder.."\\AutoPlay\\Scripts\\?.dll;"

-- Set the working folder before loading the library
-- so that LuaSocket can find all appropriate DLLs
local strOldWorkingFolder = Folder.GetCurrent();
Folder.SetCurrent(_SourceFolder.."\\AutoPlay\\Scripts");
socket = require("socket");
Folder.SetCurrent(strOldWorkingFolder);
```

## Use LuaSocket

Now LuaSocket should be initialized and ready to use. To test it, make a button on the page and set its On Click event to be:

```
Dialog.Message("SocketVersion",socket.VERSION);
```

Preview the project and click the button. You should see a dialog that shows the version of LuaSocket that you are using.

As another example, you can easily get the IP address of a named address using the script:

```
Dialog.Message("IP Address",socket.dns.toip("www.google.com"));
```

There is a whole lot that you can do with LuaSocket. For a full reference see:

<http://www.cs.princeton.edu/~diego/professional/luasocket/reference.html>

## **Conclusion**

Hopefully this article contains enough details to help you get started using the LuaSocket library in your AutoPlay Media Studio 6.0 projects. Be sure to share any useful or interesting LuaSocket projects that you make using this information on our forums.