**Multiplayer Assembly Package  
Tutorials  
Using cURL instead of TCPObjects**

This tutorial will teach you how to connect to a web page using the included cURL library instead of the TCPObject. I recommend using this method when you know large data amounts are going to be included as the TCPObject is highly unstable with large data values.

**Tutorial:**  
Since I have already included a class to work with the cURL library, this task will actually be relatively simple to accomplish.

This tutorial will pretty much explain how to use the tcpCurl.h/.cpp files correctly.

tcpCurl.h defines one function (other than the general MAP stuff), readURL(); which has the arguments location, and post data.

One could easily define a new TS function:

DefineEngineMethod(readURL, const char \*, (const char \*url, const char \*post),, “”) {

return->tcpBP(url, post).c\_str();

}

This function could then be used inside the engine to replace TCPObject connections to web pages.

For those of you who are un-familiar with the HTTP RFC, you need to know that without a HTTPObject, it can be a real pain in the rear to properly send a request to a web server. Thankfully for us, cURL makes this task really… really…REALLY… easy.

For example, say I have a php script with two post variables var1 and var2 to make it easy. With the above, I could call the following to get a reply:

readURL(“www.somesite.com/myPHP.php”, “var1=hello&var2=world”);

This would communicate to the PHP file sending var1 as hello and var2 as world. You don’t even need to send POST variables, omitting the second field will tell cURL to send it as a regular GET request which will give you the RAW TCP output from the requested page.

And that’s about all I have on using cURL with T3D. It’s easy to use and very powerful, and until the TCPObjects are fixed, this is a great replacement to them.