Programming with cURLpp

Jean-Philippe Barrette-LaPierre

January 25, 2006

1 About this Document

2.1 Compiling the Program

Your compiler needs to know where cURLpp's and libcurl's headers are located. Therefore you must set your compiler's include path to point to the directory where you installed them. The 'curlpp-config'[1] tool can be used to get this information:

```
# curlpp-config --cflags
```

If(ag)114458Td[(directo)1(ry)-308(where)-308(y)27(o)1(u)-308(insta)f5

```
easyhandle.perform();
```

The `cURLpp::Easy::perform()` will connect to the remote site, do the

```
cURLpp::Options::Url myUrl;
handle.getOpt(myUrl);
std::cout << myUrl.getValue() << std::endl;
```

```
// don’t use the instance further in your code.
```

5.2.1 The bool type

```
          CURLOPT_READFUNCTION > ReadFunction;
        typedef cURLpp::OptionTrait< void *,
          CURLOPT_READDATA > ReadData;
        typedef cURLpp::OptionTrait< cURL::curl_progress_callback,
          CURLOPT_PROGRESSFUNCTION > ProgressFunction;
        typedef cURLpp::OptionTrait< void *,
          CURLOPT_PROGRESSDATA > ProgressData;
        typedef cURLpp::OptionTrait< cURL::curl_write_callback,
          CURLOPT_HEADERFUNCTION > HeaderFunction;
        typedef cURLpp::OptionTrait< void *,
          CURLOPT_HEADERDATA > HeaderData;
        typedef cURLpp::OptionTrait< cURL::curl_debug_callback,
          CURLOPT_DEBUGFUNCTION > DebugFunction;
        typedef cURLpp::OptionTrait< void *,
          CURLOPT_DEBUGDATA > DebugData;
#ifdef CURLOPT_SSL_CTX_FUNCTION
```

5.3.6 HTTP options

```
      typedef cURLpp::OptionTrait< bool,
        cURL::CURLOPT_AUTOREFERER > AutoReferer;
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_ENCODING > Encoding;
      typedef cURLpp::OptionTrait< bool,
        cURL::CURLOPT_FOLLOWLOCATION > FollowLocation;
#ifdef cURL::CURLOPT_UNRESTRICTED_AUTH
      typedef cURLpp::OptionTrait< bool,
        cURL::CURLOPT_UNRESTRICTED_AUTH > UnrestrictedAuth;
#endif
      typedef cURLpp::OptionTrait< long,
        cURL::CURLOPT_MAXREDIRS > MaxRedirs;
#ifndef cURL::CURLOPT_UPLOAD
      typedef cURLpp::OptionTrait< bool,
        cURL::CURLOPT_PUT > Put;
#else
      typedef cURLpp::OptionTrait< bool,
        cURL::CURLOPT_UPLOAD > Put;
#endif
      typedef cURLpp::OptionTrait< bool,
        cURL::CURLOPT_POST > Post;
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_POSTFIELDS > PostFields;
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_POSTFIELDSIZE > PostFieldSize;
#ifdef cURL::CURLOPT_POSTFIELDSIZE_LARGE
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_POSTFIELDSIZE_LARGE > PostFieldSizeLarge;
#endif
      typedef cURLpp::OptionTrait< struct cURLpp::cURL::HttpPost *,
        cURL::CURLOPT_HTTPPOST > HttpPost;
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_REFERER > Referer;
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_USERAGENT > UserAgent;
      typedef cURLpp::OptionTrait< std::list< std::string >,
```

```
        cURL::CURLOPT_FTP_RESPONSE_TIMEOUT > FtpResponseTimeout;
#endif
#ifdef cURL::CURLOPT_FTP_SSL
      typedef cURLpp::OptionTrait< cURLpp::curl_ftpssl,
        cURL::CURLOPT_FTP_SSL > FtpSsl;
#endif
#ifdef cURL::CURLOPT_FTP_AUTH
      typedef cURLpp::OptionTrait< cURLpp::curl_ftpauth,
        cURL::CURLOPT_FTP_AUTH > FtpAuth;
#endif
```

5.3.8 Protocol options

```
      typedef cURLpp::OptionTrait< bool,
        cURL::CURLOPT_TRANSFERTEXT > TransferText;
      typedef cURLpp::OptionTrait< bool,
        cURL::CURLOPT_CRLF > Crlf;
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_RANGE > Range;
      typedef cURLpp::OptionTrait< long,
        cURL::CURLOPT_RESUME_FROM > ResumeFrom;
#ifdef cURL::CURLOPT_RESUME_FROM_LARGE
      typedef cURLpp::OptionTrait< curl_off_t,
        cURL::CURLOPT_RESUME_FROM_LARGE > ResumeFromLarge;
#endif
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_CUSTOMREQUEST > CustomRequest;
      typedef cURLpp::OptionTrait< bool,
        cURL::CURLOPT_FILETIME > FileTime;
      typedef cURLpp::OptionTrait< bool,
        cURL::CURLOPT_NOBODY > NoBody;
      typedef cURLpp::OptionTrait< long,
        cURL::CURLOPT_INFILESIZE > InfileSize;
```

```
      typedef cURLpp::OptionTrait< long,
        cURL::CURLOPT_IPRESOLVE > IpResolve;
#endif
```

5.3.10 SSL and security options

```
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_SSLCERT > SslCert;
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_SSLCERTTYPE > SslCertType;
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_SSLCERTPASSWD > SslCertPasswd;
      typedef cURLpp::OptionTrait< std::string,
        cURL::CURLOPT_SSLKEY > SslKey;
      typedef cURLpp::OptionTrait< std::string,
```