NAME

curl_ws_meta - meta data WebSocket information

SYNOPSIS

```
#include <curl/curl.h>

const struct curl_ws_frame *curl_ws_meta(CURL *curl);
```

DESCRIPTION

This function call is EXPERIMENTAL.

When the write callback (*CURLOPT_WRITEFUNCTION(3)*) is invoked on received WebSocket traffic, *curl_ws_meta(3)* can be called from within the callback to provide additional information about the current frame.

This function only works from within the callback, and only when receiving WebSocket data.

This function requires an easy handle as input argument for libcurl to know what transfer the question is about, but as there is no such pointer provided to the callback by libcurl itself, applications that want to use *curl_ws_meta(3)* need to pass it on to the callback on its own.

struct curl_ws_frame

```
struct curl_ws_frame {
  int age;
  int flags;
  curl_off_t offset;
  curl_off_t bytesleft;
};
```

age This field specify the age of this struct. It is always zero for now.

flags

This is a bitmask with individual bits set that describes the WebSocket data. See the list below.

offset

When this frame is a continuation of fragment data already delivered, this is the offset into the final fragment where this piece belongs.

bytesleft

If this is not a complete fragment, the *bytesleft* field informs about how many additional bytes are

expected to arrive before this fragment is complete.

FLAGS

CURLWS_TEXT
: The buffer contains text data. Note that this makes a difference to WebSocket but libcurl itself does not make any verification of the content or precautions that you actually receive valid UTF-8 content.

CURLWS_BINARY
: This is binary data.

CURLWS_CONT
: This is not the final fragment of the message, it implies that there is another fragment coming as part of the same message.

CURLWS_CLOSE
: This transfer is now closed.

CURLWS_PING
: This as an incoming ping message, that expects a pong response.

PROTOCOLS

This functionality affects ws only

EXAMPLE

```
/* we pass a pointer to this struct to the callback */
struct customdata {
  CURL *easy;
  void *ptr;
};

static size_t writecb(unsigned char *buffer,
                      size_t size, size_t nitems, void *p)
{
  struct customdata *c = (struct customdata *)p;
  const struct curl_ws_frame *m = curl_ws_meta(c->easy);

  printf("flags: %x\n", m->flags);
}
```

```
int main(void)
{
 CURL *curl = curl_easy_init();
 if(curl) {
   struct customdata custom;
   custom.easy = curl;
   custom.ptr = NULL;
   curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, writecb);
   curl_easy_setopt(curl, CURLOPT_WRITEDATA, &custom);

   curl_easy_perform(curl);

 }
}
```

AVAILABILITY

Added in curl 7.86.0

RETURN VALUE

This function returns a pointer to a *curl_ws_frame* struct with read-only information that is valid for this specific callback invocation. If it cannot return this information, or if the function is called in the wrong context, it returns NULL.

SEE ALSO

curl_easy_getinfo(3), **curl_easy_setopt**(3), **curl_ws_recv**(3), **curl_ws_send**(3), **libcurl-ws**(3)