NAME

curl_mime_encoder - set a mime part’s encoder and content transfer encoding

SYNOPSIS

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

CURLcode curl_mime_encoder(curl_mimepart *part, const char *encoding);
```

DESCRIPTION

curl_mime_encoder() requests a mime part’s content to be encoded before being transmitted.

part is the part’s handle to assign an encoder. *encoding* is a pointer to a null-terminated encoding scheme. It may be set to NULL to disable an encoder previously attached to the part. The encoding scheme storage may safely be reused after this function returns.

Setting a part’s encoder multiple times is valid: only the value set by the last call is retained.

Upon multipart rendering, the part’s content is encoded according to the pertaining scheme and a corresponding *"Content-Transfer-Encoding"* header is added to the part.

Supported encoding schemes are:

"*binary*": the data is left unchanged, the header is added.

"*8bit*": header added, no data change.

"*7bit*": the data is unchanged, but is each byte is checked to be a 7-bit value; if not, a read error occurs.

"*base64*": Data is converted to base64 encoding, then split in CRLF-terminated lines of at most 76 characters.

"*quoted-printable*": data is encoded in quoted printable lines of at most 76 characters. Since the resulting size of the final data cannot be determined prior to reading the original data, it is left as unknown, causing chunked transfer in HTTP. For the same reason, this encoder may not be used with IMAP. This encoder targets text data that is mostly ASCII and should not be used with other types of data.

If the original data is already encoded in such a scheme, a custom *Content-Transfer-Encoding* header should be added with *curl_mime_headers(3)* instead of setting a part encoder.

Encoding should not be applied to multiparts, thus the use of this function on a part with content set with *curl_mime_subparts(3)* is strongly discouraged.

PROTOCOLS

This functionality affects http, imap and smtp

EXAMPLE

```
int main(void)
{
  curl_mime *mime;
  curl_mimepart *part;

  CURL *curl = curl_easy_init();
  if(curl) {
    /* create a mime handle */
    mime = curl_mime_init(curl);

    /* add a part */
    part = curl_mime_addpart(mime);

    /* send a file */
    curl_mime_filedata(part, "image.png");

    /* encode file data in base64 for transfer */
    curl_mime_encoder(part, "base64");
  }
}
```

AVAILABILITY

Added in curl 7.56.0

RETURN VALUE

CURLE_OK or a CURL error code upon failure.

SEE ALSO

curl_mime_addpart(3), **curl_mime_headers**(3), **curl_mime_subparts**(3)