

NAME

curl_mime_filedata – set a mime part's body data from a file contents

SYNOPSIS

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

```
CURLcode curl_mime_filedata(curl_mimepart *part,  
                           const char *filename);
```

DESCRIPTION

curl_mime_filedata(3) sets a mime part's body content from the named file's contents. This is an alternative to *curl_mime_data(3)* for setting data to a mime part.

part is the part's to assign contents to.

filename points to the null-terminated file's path name. The pointer can be NULL to detach the previous part contents settings. Filename storage can be safely be reused after this call.

As a side effect, the part's remote filename is set to the base name of the given *filename* if it is a valid named file. This can be undone or overridden by a subsequent call to *curl_mime_filename(3)*.

The contents of the file is read during the file transfer in a streaming manner to allow huge files to get transferred without using much memory. It therefore requires that the file is kept intact during the entire request.

If the file size cannot be determined before actually reading it (such as for a character device or named pipe), the whole mime structure containing the part is transferred using chunks by HTTP but is rejected by IMAP.

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

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 data from this file */
        curl_mime_filedata(part, "image.png");

        /* set name */
        curl_mime_name(part, "data");
    }
}
```

AVAILABILITY

Added in curl 7.56.0

RETURN VALUE

This function returns a CURLcode indicating success or error.

CURLE_OK (0) means everything was OK, non-zero means an error occurred, see *libcurl-errors(3)*. If *CURLOPT_ERRORBUFFER(3)* was set with *curl_easy_setopt(3)* there can be an error message stored in the error buffer when non-zero is returned.

CURLE_READ_ERROR is only an indication that the file is not yet readable: it can be safely ignored at this time, but the file must be made readable before the pertaining easy handle is performed.

SEE ALSO

curl_mime_addpart(3), curl_mime_data(3), curl_mime_filename(3), curl_mime_name(3)