

HTTPS\$ - Send Message to HTTP Server

The HTTPS\$ routine is used to send a message to an HTTP server, then wait for and accept the status information and the result text.

1. Invocation

To send a message to an HTTP server, code:

```
CALL HTTPS$ USING hs
```

where *hs* is a control block of the following format:

```
01  HS
   02  HSVERS          PIC 9(4) COMP      * BLOCK VERSION NUMBER
                                       * MUST BE 1
   02  HSHDL          PIC 9(4) COMP      * HTTP handle
   02  HSFILE         PIC PTR           * Pointer to XML filename
   02  HSCODE         PIC 9(9) COMP      * HTTP status code
   02  HSSTXT         PIC X(100)        * HTTP status text
   02  HSRFIL         PIC PTR           * Pointer to Response filename
```

2. STOP Codes and Exception Conditions

The following STOP codes may be generated by HTTPS\$:

STOP code	Description
13615	HTTPS\$ has been called by an application that is not running on GX.
25022	An attempt has been made to call HTTPS\$ on an incompatible version of GX. The version of GX must be V3.5x, or later
13616	Invalid Control Block version
13617	The total length of the GX command block has exceeded an internal limit.

The following EXIT codes may be returned by HTTPS\$:

EXIT code	\$\$COND	Description
13612	12	Unable to allocate memory for temporary work buffer

13613	13	The send message to HTTP server operation has suffered an exception. The error code has been returned in \$\$CRES. For GSM SP-35, and later, the precise GX exception code is returned in HSCODE (see below).
-------	----	---

3. Programming Notes

HTTPS\$ is only available when running on GX. Any attempt to use HTTPS\$ on a non-GX terminal will result in a STOP code. The version of GX must be V3.5x, or later. The version of GSM must be GSM SP-17, or later.

The session handle returned by a previous, successful call to HTTPO\$ must be passed, via HSHDL, to HTTPS\$.

The HSFILE and HSRFIL fields must point to zero-terminated strings that specify the name of the XML file to be sent and the name of the Response filename, respectively.

The HSFILE pointer may be set to point at a single byte of binary-zero if an XML message file is **not** required by the target URL. A NULL filename string is only supported by GX V4.0k, or later. Note that if an XML message file is not required by the target URL the HSFILE pointer must still be valid. That is, the correct coding technique is:

```

03      HSFILE      PIC PTR
        VALUE      BINZERO
...
77      BINZERO    PIC X
        VALUE      #00

```

rather than the move conventional:

```

03      HSFILE      PIC PTR
        VALUE      HIGH-VALUES

```

The HSCODE and HSSTXT will be returned by the subroutine in successful completion. The returned status code value, HSCODE, is a standard HTTP status code as described in the following table:

Value	Description
100	Continue
101	Switching protocols
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified
305	Use Proxy

307	Temporary Redirect
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Long
415	Unsupported Media Type
416	Requested Range Not Suitable
417	Expectation Failed
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	HTTP Version Not Supported

The XML filename specified by the HSFIL field must be present in the "GX XML folder". Before calling HTTP\$ this file must be copied to the "GX XML folder", using GXCOP\$, or created in the "GX XML folder", using the Open GX BDAM Access Method. Similarly, the response filename specified by the HSRFIL field will be created, by GX, in the "GX XML folder". After calling HTTP\$ this file must be copied from the "GX XML folder", using GXCOP\$, or opened in the "GX XML folder", using the Open GX BDAM Access Method. Note that the "%XML" string can be used in the GXCOP\$ sub-routine and the Open GX BDAM Access Method to specify the "GX XML folder".

If HTTPS\$ returns an exception 13, the Windows error code is returned in the \$\$CRES System Variable. However, the actual failing function that was executed by GX is not returned. For GSM SP-35, and later, this has been corrected – a GX failure code is returned in the HSCODE field:

Value of HSCODE	Meaning	\$\$CRES field valid
1	Block error (internal error)	No
2	Parameter error (internal error)	No
3	Logical error	No
16	HTTP open error	Yes
17	HTTP set request header error	Yes
18	DOM document load error	No
19	HTTP send error	Yes
20	Open file error	No

4. Examples

```
MOVE HTTPHDL TO HSHDL
MOVE "OUTPUT.XML" TO X-XML
* Zero-terminate this string
POINT HSFILE AT X-XML
*
MOVE "RESULTS.XML" TO X-RESU
* Zero-terminate this string
POINT HSRFIL AT X-RESU
*
CALL HTTPS$ USING HS
```

5. Copy-Books

None.

6. See Also

HTTPC\$	Close HTTP server session
HTTPH\$	Set Request Header
HTTPS\$	Open HTTP server session