PCFDR$ - Return Formatted Address

The PCFDR$ routine returns the formatted version of an address in the XML file created by PCODE$. To use this routine you will need a Postcode Anywhere account number and license key.

1. Invocation

To return a formatted address code:

```
CALL PCFDR$ USING FA
```

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

```
01    FA
  02  FAVERS       PIC 9(4) COMP     * BLOCK VERSION NUMBER
                                     * MUST BE 1
  02  FAACC        PIC X(10)         * Postcode Anywhere account number
  02  FALIC        PIC X(19)         * Postcode Anywhere license key
  02  FAINX        PIC 9(4) COMP     * Index number of entry required
  02  FAPRG        PIC 9 COMP        * 0 = No progress bar during file copy
                                     * 1 = Progress bar displayed during copy
                                     * of XML file from GX PC to server
  02  FASCODE      PIC 9(9) COMP     * HTTPS$ Return code
  02  FASTXT       PIC X(100)        * HTTPS$ Returned text
  02  FANAME       PIC X(80)         * Returned name
  02  FADEP        PIC X(80)         * Returned department name
  02  FAADDR OCCURS 5 PIC X(80)      * Returned address lines
  02  FATOWN       PIC X(20)         * Returned town
  02  FACOUNTY     PIC X(20)         * Returned county
  02  FAPCODE      PIC X(10)         * Returned post code
  02  FAERR        PIC 9(9) COMP     * Error code returned by Postcode Anywhere
  02  FAETXT       PIC X(100)        * Error text returned by Postcode Anywhere
```

2. STOP Codes and Exception Conditions

The following STOP codes may be generated by PCFDR$:

STOP code	Description
13625	PCFDR$ has been called by an application that is not running on GX.
13626	Invalid Control Block version.

The following EXIT codes may be returned by PCFDR$:

EXIT code	$$COND	Description
13608	08	Unable to allocate memory for temporary work buffer (HTTPO$).

13609	09	An error condition was returned by the open communications operation. The error code is available in the $$CRES System Variable.
13610	10	Unable to allocate memory for temporary work buffer (HTTPH$).
13611	11	The set HTTP server request header operation has suffered an exception. The error code is available in the $$CRES System Variable.
13612	12	Unable to allocate memory for temporary work buffer (HTTPS$).
13613	13	The send message to the HTTP servers has suffered an exception. The error code is available in the $$CRES System Variable.
13628	28	Unable to allocate memory for temporary work buffer.
13629	29	Unable to open the XML file produced by PCODE$.
13630	30	Unable to read the XML file produced by PCODE$.
13631	31	The index number supplied was invalid.
13632	32	The return code from the send message to HTTP servers is not 200 (OK). The send message result code has been returned in FASCODE and FASTXT. (See HTTPS$ documentation for full list)
13633	33	Unable to open the XML file produced on the server.
13634	34	Unable to allocate enough memory to read the XML file.
13635	27	Unable to read the XML file produced on the server.
13642	42	Postcode Anywhere has returned an error in the XML file. The error code is in FAERR and the error text in FAETXT.

7614	14	GXCOP$ has suffered an exception opening the input or output files. The Windows error code is available in the $$CRES System Variable.
7615	15	The GXCOP$ routine cannot find the input file.
7616	16	Unable to allocate temporary work buffer (GXCOP$).
7619	19	GXCOP$ has suffered an error opening the input file. The Windows error code is available in the $$CRES System Variable.
7620	20	GXCOP$ has suffered an error opening the output file. The Windows error code is available in the $$CRES System Variable.
7621	21	Error returned from GX Copy File function. The Windows error code is available in the $$CRES System Variable.
7622	22	GXCOP$ has suffered an unexpected error from GX. The Windows error code is available in the $$CRES System Variable.

3. Programming Notes

In order to use this subroutine you need to obtain an account number and license key from Postcode Anywhere at http://www.postcodeanywhere.co.uk/.

PCFDR$ is only available when running on GX. Any attempt to run PCFDR$ on a non-GX terminal will result in a STOP code.

PCFDR$ is only available on GSM SP-20, or later.

PCFDR$ processes the results returned by PCODE$ and must be used in conjunction with this routine.

The indexing for the FAINX index number passed begins at 1.

The PCFDR$ routine makes use of the HTTPO$, HTTPH$, HTTPS$ and HTTPC$ routines to obtain the postcode address file in XML format. The Postcode Anywhere method used is “Fetch_Address”. The XML file will be created in the “xml” sub-folder in the current GX folder on the GX PC and will be automatically copied to the “temp” sub-folder in the GSM folder on the server running the Global Client (e.g. GLOBAL.EXE). The name of the XML file is:

formatted_address_*cc_uuuuu_oooo*.xml

where *cc* is the computer-id, *uuuuu* the user number and *oooo* the operator-id of the user running the subroutine. Note this filename is different from the temporary XML file created by PCODE$.

There is further information held in the “formatted address” XML file that is not returned by PCFDR$. To extract this additional information you must access the file directly using the Open BDAM access method.

4. Examples

No examples available.

5. Copy-Books

None.

6. See Also

PCODE$	Postcode address file creation
PCADR$	Postcode address extraction
HTTPO$	Open HTTP session
HTTPC$	Close HTTP server session
HTTPH$	Set Request Header
HTTPS$	Send message and return status information and response text
GXCOP$	File copy routine