

GXDEL\$ - Delete Windows File on GX

The GXDEL\$ routine can be used to delete a file on the PC that is running the current GX session.

1. Invocation

To delete a file on the PC that is running GX code:

```
CALL GXDEL$ USING fd
```

where *fd* is a GX Direct Access Method FD of the following structure:

```
01    FILLER REDEFINES FDGX
02    FILLER          PIC X(100)      * Start of FD is unused
02    ODFNAM          PIC X(99)       * Zero terminated filename string
02    ODZERO          PIC X           * String end-stop MUST BE LOW-VALUES
02    FILLER          PIC X(14)       * Rest of FD is unused
```

2. STOP Codes and Exception Conditions

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

STOP code	Description
7910	GXDEL\$ has been called by an application that is not running on GX.
7911	The ODZERO byte in the FD does not contain LOW-VALUE (binary-zero)
7912	An internal error has occurred (the file name was longer than 8Kb)
7913	An attempt was made to call GXDEL\$ on an incompatible version of GX. The version of GX must be V2.6, or later.
7914	A file name was not specified.

The following exception conditions may be returned by GXDEL\$:

EXIT code	\$\$COND	Description
7903	3	An internal error has occurred (GXDEL\$ was unable to allocate an additional temporary data page)

7904	4	The replaceable file parameter is not present in the GX.INI file.
7905	5	The Windows DeleteFile operation failed. The Windows 32-bit error code is available in the \$\$CRES System Variable.
7906	6	An internal error has occurred (GXDEL\$ was unable to allocate an initial temporary data page)

3. Programming Notes

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

If GXDEL\$ returns an Exception 4 the 32-bit Windows error code is available in \$\$CRES. The FMESS\$ [fmess\\$.doc](#) sub-routine may be used to convert the error code to a verbose error message.

The file name parameter can specify either:

- an absolute filename (on the filing system of the PC that is running GX);
- an absolute UNC filename on a shared directory (as "seen" from the PC that is running GX);
- a file on, or under, the current directory on the PC that is running GX;
- a file on, or under, the Windows directory on the PC that is running GX;
- a file on, or under, the LocalTempDirectory specified in the GX.INI file;
- a file on, or under, the folder "PFPrintImage", relative to the current GX folder;
- a file on, or under, the folder "XML", relative to the current GX folder;
- a file on, or under, a free-format directory keyword specified in the GX.INI file;
- a file on, or under, a free-format directory keyword specified as a Windows environment variable;

For example assuming that the "current directory" for GX is "C:\GX"; the "Windows directory" is "C:\Windows"; the GX.INI file contains the following setting in the [general] section:

```
LocalTempDirectory=C:\TEMPORARY
```

and the following settings in the [folders] section:

```
DTEMP1=D:\TEMPORARY
DTEMP2=F:\TEMPORARY
```

And the following Windows environment variable is defined:

DTEMP3=G:\TEMPORARY

then the following *filename* parameters will be mapped to the following pathnames on the PC running GX:

Filename field in FD	Filename on PC running GX.EXE
C:\WINDOWS\EXAMPLE.TXT	C:\WINDOWS\EXAMPLE.TXT
\\SERVER1\SHARE\EXAMPLE.TXT	\\SERVER1\SHARE\EXAMPLE.TXT
EXAMPLE.TXT	C:\GX\EXAMPLE.TXT
.\EXAMPLE.TXT	C:\GX\EXAMPLE.TXT
SUBDIR\EXAMPLE.TXT	C:\GX\SUBDIR\EXAMPLE.TXT
%TEMP\EXAMPLE.HTML	C:\TEMPORARY\EXAMPLE.HTML
%WINDOWS\EXAMPLE.XML	C:\WINDOWS\EXAMPLES.XML
%DTEMP1\EXAMPLE.TXT	D:\TEMPORARY\EXAMPLE.TXT
%DTEMP2\EXAMPLE.TXT	F:\TEMPORARY\EXAMPLE.TXT
%DTEMP3\EXAMPLE.TXT	G:\TEMPORARY\EXAMPLE.TXT
C:\%DTEMP3\EXAMPLE.TXT	The results of including a variable parameter in the middle of a filename string are undefined.

If a %variable exists both as a Windows environment variable **and** as an option in the [folders] section of the GX.INI the GX.INI file setting will be used.

If the file specified by the filename string does not exist an exception condition 4 with a \$\$CRES value of 2 will be returned by GXDEL\$.

The %TEMP and %WINDOWS strings are effectively reserved words and cannot be used to define either Windows environments variables or options in the [folders] section of the GX.INI file. The %TEMP string is mapped to the value of the LocalTempDirectory setting in the [general] section of the GX.INI file. If the LocalTempDirectory setting is not defined, the %TEMP string is mapped to the Windows temp directory. The %WINDOWS string is always mapped directly to the Windows folder.

The %XML and %TCPIIMAGES strings are also reserved words and cannot be used to define either Windows environments variables or options in the [folders] section of the GX.INI file. Both these special strings are reserved for internal use only. The %XML string is always mapped directly to the folder "XML", relative to the current GX folder. The %TCPIIMAGES string is always mapped directly to the folder "PFPrintImage", relative to the current GX folder;

The ODZERO field **MUST** always be set to LOW-VALUES. A STOP 7911 will occur if ODZERO does not contain LOW-VALUES. This mandatory byte of binary-zero is required to ensure that the filename string in ODFNAM is guaranteed to be followed by a byte of #00. In addition, the filename string itself (in ODFNAM) **MUST** be terminated with a byte of LOW-VALUES to allow correct operation of the Access Method.

For example:

```
DATA DIVISION
77  X-NAME          PIC X(?)
   VALUE "EXAMPLE.TXT"
PROCEDURE DIVISION
   MOVE LOW-VALUES TO ODZERO          * ENSURE ODZERO IS #00
   MOVE X-NAME TO ODFNAME             * ESTABLISH FILENAME
```

will establish a Windows filename containing the 11 characters "EXAMPLE.TXT" **followed by 88 bytes of SPACE's**. One of the many correct techniques is shown below:

```
DATA DIVISION
01  X-NAME
   02  FILLER          PIC X(?)
      VALUE "EXAMPLE.TXT"
   02  FILLER          PIC X
      VALUE LOW-VALUES
PROCEDURE DIVISION
   MOVE LOW-VALUES TO ODZERO          * ENSURE ODZERO IS #00
   MOVE X-NAME TO ODFNAME             * ESTABLISH FILENAME
```

Note that the ODZERO field a redefinition of the FD and cannot be initialised in the DATA DIVISION (i.e. the MOVE statement in the PROCEDURE DIVISION is always required).

4. Examples

[EXAMPLES REQUIRED]

5. Copy-Books

None.

6. See Also

GXDEX\$ Delete Windows file on GX (passing filename as a zero-terminated string)
STRML\$ Move Zero-Terminated string