Files in this folder:

Important Note: FWRITEVAX provides a common interface for writing data to files in VAXD and VAXG formats using FWRITEVAXD and FWRITEVAXG respectively. However, both FWRITEVAXD and FWRITEVAXG can also be used independently of FWRITEVAX. These functions differ in their signatures and the supported number of arguments, as discussed below.

1. fwriteVAX.m: Call this function from your program to write to a binary file using VAX format. For example:

```
fid = fopen('vaxFileName', 'w', 'ieee-le');    % Using 'ieee-le' is required.
num = rand(3,3);
vaxfNumCount = fwriteVAX(fid, num, 'single');          % VAXF
vaxdNumCount = fwriteVAX(fid, num, 'double', 'vaxd');   % VAXG is the other supported format
fclose(fid);
```

** Execute 'help fwriteVAX' for more information.

2. fwriteVAXD.m Call this function from your program to write to a binary file using VAXD format. For example:

```
fid = fopen('vaxFileName', 'w', 'ieee-le');   % Using 'ieee-le' is required.
num = rand(3,3);
vaxfNumCount = fwriteVAXD(fid, num, 'single');          % VAXF
vaxdNumCount = fwriteVAXD(fid, num, 'double');
fclose(fid);
```

** Execute 'help fwriteVAXD' for more information.

3. fwriteVAXG.m Call this function from your program to write to a binary file using VAXG format. For example:

```
fid = fopen('vaxFileName', 'w', 'ieee-le');    % Using ‘ieee-le’ is required.

num = rand(3,3);

vaxfNumCount = fwriteVAXG(fid, num, ‘single’);          % VAXF

vaxdNumCount = fwriteVAXG(fid, num, ‘double’);

fclose(fid);
```

** Execute ‘help fwriteVAXG’ for more information.

4. VAXF_to_uint32le.m: To be called internally from fwriteVAX, fwriteVAXD and fwriteVAXG.

5. VAXD_to_uint64le.m: To be called internally from fwriteVAX, fwriteVAXD and fwriteVAXG.

6. VAXG_to_uint64le.m: To be called internally from fwriteVAX, fwriteVAXD and fwriteVAXG.

7. readme.pdf: This file.