This document discusses the included VSIPL User functions. These functions are not part of the VSIPL specification, and are included to help with writing example code, or to replace VSIPL functions not included with the library. They should run with any VSIPL core profile.

The functions below read from stream a1 into view a2. The data file is an ASCII file with an index and a value, so that for a

real vector each entry in the file is	i	number		
complex vector each entry in the file is	i	real	imag	
real matrix	i	j	number	
complex matrix	i	j	real	imag

The routine reads the file until it is empty. There is no error checking, no value not indexed will be replaced, and any index used more than once will write over the previous value.

```
void VU_mreadf_f(
   FILE* a1,
   vsip_mview_f* a2);
```

```
void VU_cmreadf_f(
   FILE* a1,
   vsip_cmview_f* a2);
```

```
void VU_vreadf_f(
   FILE* a1,
   vsip_vview_f* a2);
```

```
void VU_cvreadf_f(
   FILE* a1,
   vsip_cvview_f* a2);
```

Argument a1 File pointer

Argument a2 view to input data to.

The view print functions below print to standard output a matrix or vector. The format is suitable for pasting into Matlab. This will print any size matrix or vector, so use it with caution. It is designed for small views to allow for outputting test data.

```
void VU_vprintm_f(
   char* a1,
   vsip_vview_f* a2);
```

```
void VU_cvprintm_f(
   char* a1,
   vsip_cvview_f* a2);
```

```
void VU_mprintm_f(
   char* a1,
   vsip_mview_f* a2);
```

```
void VU_cmprintm_f(
   char* a1,
   vsip_mview_f* a2);
```

Argument a1 This is a format as in “6.4” which would fit in a print statement as “%6.4”.

Argument a2 view to be printed

EXAMPLE

```
include<stdio.h>
#include<vsip.h>
#include<VU.h>
int main()
{
   FILE *fptr = fopen("tcv.data","r");
   vsip_cvview_f *M = vsip_cvcreate_f(6,0);
   VU_cvreadf_f(fptr,M);
   VU_cvprintm_f("6.4",M);
   return 0;
}
```