

XREC as Library (DLL) Function (version 3.1 or later)

Compiled by David Watts

11/13/07

XREC is now provided as a DLL library function, the advantages to users are:

- no writing of images to the hard disk and therefore quicker execution time
- makes implementation easier in programming language of choice
- time of termination of XREC's processing is known exactly, no polling of the activity monitor is required (i.e. No checking if the Xrec_check.txt file contains a value of 1).

The entry point to the dll function is declared as follows:

```
__declspec(dllexport) long BW7B_wrapper(long num_images, long width, long height,  
                                         long border_bottom, long border_top,  
                                         long border_left, long border_right,  
                                         float* phi_array,  
                                         unsigned char** image_arr,  
                                         long loop_pos,  
                                         long prealignment_flag,  
                                         long set_cross_flag,  
                                         long *res_phi,  
                                         long *res_radius,  
                                         long *res_x,  
                                         long *res_y,  
                                         long *res_precentring,  
                                         long *reliability)
```

An example Microsoft Visual Studio C++ 2008 project is included in the download tar file and demonstrates the use of the DLL. The Xrec32_WinMain.cpp file gives examples of default values for many of the parameters. The project links to image reading libraries that will be need to be substituted for equivalent libraries on the users system. The XREC 3.0 manual (downloadable from the XREC web site) gives an explanation of the meaning of the input and output parameters in the context of crystal centering using a goniometer.

The dll can be loaded can called dynamically as in the except from the code file given below:

[outside the main function]

```
typedef long (*funcPtr)(long num_images, long width, long height,  
                        long border_bottom, long border_top,  
                        long border_left, long border_right,  
                        float* phi_array,  
                        unsigned char** image_arr,  
                        long loop_pos,  
                        long prealignment_flag,  
                        long set_cross_flag,  
                        long *res_phi,  
                        long *res_radius,  
                        long *res_x,  
                        long *res_y,  
                        long *res_precentring,  
                        long *reliability);
```

[within main function]

```
HINSTANCE hDLL = LoadLibraryA(dll_name);    // dll_name is the pathname of the dll file
```

```
funcPtr XREC_EntryFunction;
```

```
if( hDLL == NULL )
```

```
{  
    printf("Unable to load/find dll\n");  
    return -1;
```

```
}  
else
```

```
{  
    XREC_EntryFunction = (funcPtr)GetProcAddress(hDLL, "BW7B_wrapper");
```

```
    // call dll function
```

```
    err = XREC_EntryFunction((long) num_images, (long) width, (long) height,  
                             (long) border[0], (long) border[1],  
                             (long) border[2], (long) border[3],  
                             angle_array,  
                             image_ptr_array,  
                             (long) loop_pos,  
                             (long) prealignment_flag,  
                             (long) set_cross_flag,  
                             &res_phi,  
                             &res_radius,  
                             &res_x,  
                             &res_y,  
                             &precentring,  
                             &reliability);
```

```
    FreeLibrary(hDLL);
```

```
}
```

The XREC dll can also be called from Labview and Matlab environments. For example in matlab the dll can be loaded and called as

Static and dynamic libraries are also available for Apple Mac (iMac) and Linux 32 and 64 bit operating systems. The libraries are called libXREC.a and libXREC.la and in the case of the mac libXREC.dylib. these libraries reside in the hidden .libs directory. The function call and arguments are identical to the Windows DLL example. The declaration of the function is simply declared and called as follows

[outside the main function]

```
#ifdef __cplusplus
extern "C"
{
#endif
```

```
long BW7B_wrapper(long num_images, long width, long height,
                  long border_bottom, long border_top,
                  long border_left, long border_right,
                  float* phi_array,
                  unsigned char** image_arr,
                  long loop_pos,
                  long prealignment_flag,
                  long set_cross_flag,
                  long *res_phi,
                  long *res_radius,
                  long *res_x,
                  long *res_y,
                  long *res_precentring,
                  long *reliability);
```

```
#ifdef __cplusplus
}
#endif
```

[within main function]

```
err = BW7B_wrapper((long) num_images, (long) width, (long) height,
                  (long) border[0], (long) border[1],
                  (long) border[2], (long) border[3],
                  angle,
                  image_ptr,
                  (long) loop_pos,
                  (long) prealignment_flag,
                  (long) set_cross_flag,
                  &res_phi,
                  &res_radius,
                  &res_x,
                  &res_y,
                  &precentring,
                  &reliability);
```