

Documentation

IRun.DLL Calling Syntax:

File based syntax:

In order to call IRun from an application include "irun.dll" and "sbdll.dll" in your DLL calling path. Call EXRTF2WEB function with the following parameters.

```
int EXRTF2WEB(  
char * source, // source file name  
char * destination, // destination file name  
int options, // an OR'ed list of options  
char * bgcolor, // HTML bgcolor  
char * title, // HTML title  
Int dpi) // WMF-GIF conversion DPI (dots Per inch)
```

for the options use the following enumeration.

```
typedef enum  
{  
    EXO_RESULTS= 0x0001,  
    EXO_INLINECSS= 0x0002,  
    EXO_WMF2GIF= 0x0004,  
    EXO_XML= 0x0008,  
    EXO_HTML= 0x0010,  
    EXO_MEMORY= 0x0020,  
    EXO_NOHEADER= 0x0040  
} exOptions ;
```

If *bgcolor* and *title* are NULL, the values are extracted from the source document. The *DPI* value should be greater than 30. *bgcolor* is the string representation of background color as in the HTML. This function creates image links in "_img_n.gif" format. Here "n" is the number of image in the referenced objects list. The function returns 0 if everything is OK. If error occurs the function returns negative values.

return	meaning
-1	Invalid parameter (s) (source or destination)
-2	Invalid parameter (dpi)
-3	Cannot read source file.
-4	Cannot write to destination file

-5	RTF syntax error
----	------------------

Buffer based syntax:

Following functions allow conversion on memory buffers

HGLOBAL EXBufferOpenSession ()

Opens a session for converting RTFs. Returns the session handle which is required for the following calls. Returns NULL if an error occurs.

```
int EXBufferConvert (
HGLOBAL sessionHandle, // session handle
unsigned char* inputBuffer, // input buffer
int inputBufferSize, // size of input buffer
int* outputBufferSize, // (out) output buffer size
int* referencedObjectCount, // (out) number of referenced objects
int options, // an OR'ed list of options
char* bgcolor, // HTML bgcolor
char* title, // HTML title
Int dpi) // WMF-GIF conversion DPI (dots Per inch)
```

Converts a single buffer and returns the size of output buffer and the number of referenced objects (images and other objects). The function returns 0 if everthing is OK. Others values are:

return	meaning
-1	Invalid session handle
-2	Input buffer null
-3	Input buffer size zero or negative
-4	RTF syntax error
-5	Invalid dpi

Use the same enumeration in "file based conversion" part for the options argument.

```
int EXBufferGetBody ( HGLOBAL sessionHandle, unsigned char* outputBuffer)
```

Copies the resulting buffer (HTML ve XML) into the outputBuffer. The output buffer must be allocated by the user with the value returned from the EXBufferConvert function. Returns 0 for "no error", -1 for "invalid handle", -2 for "no output buffer", -3 for "empty output buffer".

```
int EXBufferGetReferencedObjectSize ( HGLOBAL sessionHandle, int objectIndex,  
unsigned int* bufferSize)
```

Returns the size of the object at a specified index. objectIndex can have a value between 0 and number_of_objects – 1. The function sets the bufferSize value to the byte size of object and returns 0 in normal case. The function returns -1 for "invalid handle", -2 for "no referenced objects" and -3 for "invalid object index".

```
int EXBufferGetReferencedObject (HGLOBAL sessionHandle, int objectIndex,  
unsigned char* outputBuffer)
```

Copies the resulting object buffer (image or other) into the outputBuffer. The output buffer must be allocated by the user with the value returned from the EXBuffer GetReferencedObjectSize function. Returns 0 for "no error", -1 for "invalid handle", -2 for "no objects", -3 for "invalid index".

```
int EXBufferCloseSession ( HGLOBAL sessionHandle)
```

Closes the session and frees resources. It is users responsibility to free the buffers creates by the user. This function returns 0 in normal case, and -1 for "invalid handle".

Examples :

Here is a "C" example:

```
/* proc definition */  
typedef int __stdcall (* EXRTF2WEB)  
    (char* source,  
     char* dest,  
     int options,  
     char* title,  
     char* bgcolor,  
     int dpi);
```

```

typedef HGLOBAL __stdcall __export (* EXBufferOpenSession)();
typedef int __stdcall __export (* EXBufferConvert)( HGLOBAL sessionHandle,
                                                    unsigned char* inputBuffer,
                                                    int inputBufferSize,
                                                    int* outputBufferSize,
                                                    int* referencedObjectCount,
                                                    int options,
                                                    char* bgcolor,
                                                    char* title,
                                                    int dpi);

typedef int __stdcall __export (* EXBufferCloseSession)(HGLOBAL sessionHandle);

typedef int __stdcall __export (* EXBufferGetBody)(HGLOBAL sessionHandle,
                                                    unsigned char* outputBuffer);

typedef int __stdcall __export (* EXBufferGetReferencedObjectSize)(HGLOBAL sessionHandle,
                                                                    int objectIndex,
                                                                    unsigned int* bufferSize);

typedef int __stdcall __export (* EXBufferGetReferencedObject)(HGLOBAL sessionHandle,
                                                                int objectIndex,
                                                                unsigned char* outputBuffer);
void
Convert( void )
{
    HINSTANCE inst;
    EXRTF2WEB proc;
    int ret;

    inst=LoadLibrary((LPCTSTR)"irun.dll");
    if(inst)
    {
        proc=(EXRTF2WEB)GetProcAddress(inst,(LPCTSTR)"EXRTF2WEB");
        if(proc)
            ret=(*proc)( "test.rtf", "test.html" , 0x0f, NULL, NULL, 96);
        FreeLibrary(inst);
    }
}

void
BufferConvert(void)
{
    int ret=0;
    int options=0x15;
    int dpi=96;
    FILE* fp;
    unsigned char* ib;
    unsigned char* ob=NULL;
    int referencedObjectCount=0;
    int isize,osize=0;
    int i;
    HINSTANCE inst;
    EXBufferOpenSession pOpenSession;
    EXBufferConvert pConvert;
    EXBufferGetBody pGetBody;
    EXBufferGetReferencedObjectSize pGetReferencedObjectSize;
    EXBufferGetReferencedObject pGetReferencedObject;
    EXBufferCloseSession pCloseSession;
    HGLOBAL sessionHandle;

    options=options|EXO_MEMORY;

    // read the file to buffer
    fp = fopen("d:\\temp\\IRun\\Examples\\Example.rtf", "r");
    fseek( fp, 0L, SEEK_END );
    isize = ftell( fp );

```

```

fseek(fp,0L,SEEK_SET);
ib=(unsigned char*)malloc(isize);
fread(ib,1, isize,fp);
fclose(fp);

// call the converter
inst=LoadLibrary((LPCTSTR)"irun.dll");
if(inst)
{
    pOpenSession=(EXBufferOpenSession)GetProcAddress(inst,
        (LPCTSTR)"EXBufferOpenSession");
    pConvert=(EXBufferConvert)GetProcAddress(inst,
        (LPCTSTR)"EXBufferConvert");
    pGetBody=(EXBufferGetBody)GetProcAddress(inst,
        (LPCTSTR)"EXBufferGetBody");
    pGetReferencedObjectSize=(EXBufferGetReferencedObjectSize)GetProcAddress(inst,
        (LPCTSTR)"EXBufferGetReferencedObjectSize");
    pGetReferencedObject=(EXBufferGetReferencedObject)GetProcAddress(inst,
        (LPCTSTR)"EXBufferGetReferencedObject");
    pCloseSession=(EXBufferCloseSession)GetProcAddress(inst,
        (LPCTSTR)"EXBufferCloseSession");
    if(!pOpenSession || !pConvert || !pGetBody || !pGetReferencedObjectSize
        || !pGetReferencedObject || !pCloseSession) return;

    //open a new session
    sessionHandle = (*pOpenSession)();
    if(!sessionHandle) return;

    //now convert the RTF to HTML
    ret=(*pConvert)(sessionHandle,ib, isize,&osize,&referencedObjectCount,
        options,NULL,NULL,dpi);
    if(ret==0)
    {
        //Allocate space for the output with the osize returned from Convert
        ob = (unsigned char*) malloc(osize);
        //Copy the result to my output buffer
        ret = (*pGetBody)(sessionHandle,ob);
        // write output to file
        fp = fopen("d:\\temp\\IRun\\Examples\\ExampleNew.htm","w");
        fwrite(ob,1,osize,fp);
        fclose(fp);
        //clean-up
        free(ob);
        ret = (*pCloseSession)(sessionHandle);
    }
}
free(ib);
FreeLibrary(inst);
}

```

Last modified on 2017-03-14 by admin