

## RENLINK.DLL (documentation v1.2)

RENLINK is a WIN32 dynamic link library, used to run *Renovator* (or a test stub) from any external system that is capable of linking to it – which includes any system capable of linking to the standard DLLs in the Windows API.

The DLL exports a single entry:

```
long RUNRENO(const char *)
```

i.e. a function that takes a string argument and returns a long (4 byte) signed number.

The parameter is a pointer to a zero-terminated string of ASCII characters. This string specifies the pathname, and parameters, of the *Renovator* executable to be run. Unless it refers to the current directory of the invoking process, the path specified should be a full pathname. If the pathname contains spaces it should be quoted in the normal way. Parameters should be separated from the pathname by at least one space character.

*Renovator* itself expects the parameter passed to be the name (typically the full pathname) of a case file, conventionally with a `.sci` extension. This file must exist already at the specified location in the filesystem. A minimal case file (such as may be used when first performing a test of resources) consists of the following line:

```
application=renovate
```

Such a file may exist permanently in the filesystem, or it may be extracted as required from a text field of a database table. In the latter case the invoking application will update the contents of the file into the table when the call to RUNRENO returns, then delete the file from the filesystem. In this situation the file is typically named after the case reference or other unique identifier specific to the case record, though it may have a generated temporary name.

The case file may be seeded with additional information, as known already by the invoking application. Items of information provided this way are not asked of the user by *Renovator*. For example, the following file contents:

```
application=renovate
grant_type=2
relevant_persons_mt=1
relevant_persons_is=0
cost_of_works=8003.12
grant_limit_band=1
case_reference=123456789
case_date=19/12/2003
case_clientname=Mervyn Jones
```

specifies a Disabled-facility Grant case (1 is a Renovation Grant case), with a single relevant person to be means-tested (i.e. not receiving IS/JSA/GPC) and no relevant persons receiving IS/JSA/GPC, with a cost-of-works figure as specified. The grant limit is that appropriate for England (2 is Wales) and the other settings relate to basic applicant information. All these items are optional – if not supplied the user will be prompted for the information by *Renovator*.

The return value from RUNRENO is either non-negative, or a negative error sentinel. If non-negative, the value returned is the **net contribution**, in pence, as calculated by *Renovator* (or returned by a test stub). This may be zero, of course.

A negative return value signifies an error, as follows:

-1	RenLink could not create its window. This return will be preceded by an alert message, and should not occur unless system resources are very low.
-2	RenLink could not run the pathname passed as the parameter. This return will be preceded by an alert diagnostic of the specific cause of failure.
-3	RenLink detected the WM_QUERYENDSESSION message that Windows sends to top-level windows when the user initiates a shut-down, or attempts to log out the Windows session.
-4	The program run by RenLink (i.e. <i>Renovator</i> , or a test stub) terminated unexpectedly with a non-zero process exit code.
-100	<i>Renovator</i> detected that no case file pathname was specified.
-n	Other negative values are reserved for future use.

## Sample C++ program invoking RENLINK

```
/*
 * Name:      LinkTest
 * Purpose:   Test Renovator invocation via RENLINK.DLL
 * File:      linktest.cpp
 * System:    WIN32
 */

#define WIN32_LEAN_AND_MEAN
#include <windows.h>
#include <stdlib.h>
#include <string.h>

static void
message(const char *s)
{
    MessageBox(NULL, s, "LinkTest", MB_OK | MB_TASKMODAL | MB_ICONINFORMATION);
}

int WINAPI
WinMain(HINSTANCE ghInst, HINSTANCE, LPSTR lpszCmdLine, int)
{
    // Link dynamically
    HINSTANCE hlib = LoadLibrary("RENLINK.DLL");

    if (hlib)
    {
        longrc = 0;

        typedef long (WINAPI *FUNCTYPE)(const char *);
        FUNCTYPE func = (FUNCTYPE)GetProcAddress(hlib, "RUNRENO");

        if (func)
        {
            char command[512];
            strcpy(command, "RENOVATE.EXE "); // In current directory
            // Pass any parameters on to Renovator stub
            if (lpszCmdLine)
                strcat(command, lpszCmdLine);

            message("About to invoke RenLink:RUNRENO");
        }
    }
}
```

```

rc = (*func) (command);

char temp[80];
if (rc<0)
wsprintf(temp, "Error return, code %ld", rc);
else
wsprintf(temp, "Return from Renovator was %ld pence", rc);

message(temp);
}
else
message("Failed to invoke RUNRENO!");

FreeLibrary(hlib);
}
else
message("Failed to link to RENLINK.DLL!");

return 0;
}

// Eof

```

## Sample Visual Basic fragment invoking RENLINK

```

' Declare the function
Private Declare Function RUNRENO Lib "renlink" (ByVallpString As String) _
As Long
' Invoke it, passing a case filename
Dim NetContributionPenceAs Long
NetContributionPence = RUNRENO("C:\FERRET\RENO211\RENOVATE.EXE ACASE.SCI")
' Check return, and act accordingly
If NetContributionPence<0 Then
' Error return
Else
' Normal return
End If

```

*David Palmer*  
*Ferret Information Systems*