

Subject: : AmigaOS4

Topic: : Porting to AmigaOS4 thread

Re: Porting to AmigaOS4 thread

Author: : Raziel

Date: : 2020/11/22 15:02:33

URL:

@Capehill

Sorry for being a noob.

These are both .cpp/.h

```
#ifndef BACKEND_AMIGAOS_DIALOGS_H
#define BACKEND_AMIGAOS_DIALOGS_H

#if defined(__amigaos4__) && defined(USE_SYSDIALOGS)

#include "common/fs.h"
#include "common/dialogs.h"

class AmigaOSDialogManager : public Common::DialogManager {
public:
    virtual DialogResult showFileBrowser(const Common::U32String &title, Common::FSNode &choice,
bool isDirBrowser);

private:
    const char *utf8ToLocal(const char *in);
};

#endif

#endif // BACKEND_AMIGAOS_DIALOGS_H

#define FORBIDDEN_SYMBOL_EXCEPTION_FILE
#define FORBIDDEN_SYMBOL_EXCEPTION_unistd_h
#define FORBIDDEN_SYMBOL_EXCEPTION_time_h
#define FORBIDDEN_SYMBOL_EXCEPTION_strdup

#include "common/scummsys.h"

#if defined(__amigaos4__) && defined(USE_SYSDIALOGS)
```

```

#include "backends/dialogs/amigaos/amigaos-dialogs.h"

#include "common/config-manager.h"
#include "common/encoding.h"

#include <proto/asl.h>
#include <proto/codesets.h>
#include <proto/dos.h>
#include <proto/exec.h>

#include <cstring>
#include <cstdlib>

struct CodesetsIFace *ICodesets = nullptr;
struct Library *CodesetsBase = nullptr;

const char *AmigaOSDialogManager::utf8ToLocal(const char *in) {

    if (!in) {
        return strdup("");
    }

    CodesetsBase = IExec->OpenLibrary("codesets.library", 6);

    if (CodesetsBase) {

        ICodesets = (CodesetsIFace *)IExec->GetInterface(CodesetsBase, "main", 1L, nullptr);

        struct codeset *srcmib = ICodesets->CodesetsFind("ISO-8859-1", CSA_FallbackToDefault, FALSE,
TAG_DONE);
        struct codeset *dstmib = ICodesets->CodesetsFind("UTF-8", CSA_FallbackToDefault, FALSE,
TAG_DONE);

        if (dstmib != nullptr) {
            ULONG dstlen = ICodesets->CodesetsStrLen(in, TAG_END);

            char *out = (char *)malloc(dstlen + 1);
            if (out) {
                if ((out=ICodesets->CodesetsConvertStr(CSA_SourceCodeset, srcmib, CSA_DestCodeset, dstmib,
CSA_Source, in, TAG_DONE)) != nullptr) {
                    return out;
                }
            }
            free(out);
        }
        IExec->DropInterface((struct Interface *)ICodesets);
        ICodesets = nullptr;
        IExec->CloseLibrary(CodesetsBase);
        CodesetsBase = nullptr;
    }
}

```

```

return strdup(in);
}

struct AsIFace *IAI;
struct Library *AsIBase;

Common::DialogManager::DialogResult AmigaOSDialogManager::showFileBrowser(const Common::U32String
&title, Common::FSNode &choice, bool isDirBrowser) {

    char pathBuffer[MAXPATHLEN];

    Common::String utf8Title = title.encode();

    DialogResult result = kDialogCancel;

    AsIBase = IExec->OpenLibrary(AsIName, 50);
    if (AsIBase) {

        IAI = (struct AsIFace*)IExec->GetInterface(AsIBase, "main", 1, nullptr);

        struct FileRequester *fr = nullptr;

        if (ConfMan.hasKey("browser_lastpath")) {
            strncpy(pathBuffer, ConfMan.get("browser_lastpath").c_str(), sizeof(pathBuffer)-1);
        }

        fr = (struct FileRequester *)IAI->AllocAsIRequestTags(ASL_FileRequest, TAG_DONE);

        if (!fr)
            return result;

        const char *newTitle = utf8ToLocal((const char *)utf8Title.c_str());

        if (IAI->AsIRequestTags(fr, ASLFR_TitleText, newTitle, ASLFR_RejectIcons, TRUE, ASLFR_InitialDrawer
, pathBuffer, ASLFR_DrawersOnly, (isDirBrowser ? TRUE : FALSE), TAG_DONE)) {

            if (strlen(fr->fr_Drawer) < sizeof(pathBuffer)) {
                strncpy(pathBuffer, fr->fr_Drawer, sizeof(pathBuffer));
                if (!isDirBrowser) {
                    IDOS->AddPart(pathBuffer, fr->fr_File, sizeof(pathBuffer));
                }
                choice = Common::FSNode(pathBuffer);
                ConfMan.set("browser_lastpath", pathBuffer);
                result = kDialogOk;
            }
            free(newTitle);
            IAI->FreeAsIRequest((APTR)fr);
        }
        IExec->DropInterface((struct Interface*)IAI);
        IAI = nullptr;
    }
}

```

```
    |Exec->CloseLibrary(AslBase);
    AslBase = nullptr;
}

return result;
}

#endif
```

I get this now on compiling

```
C++   backends/dialogs/amigaos/amigaos-dialogs.o
backends/dialogs/amigaos/amigaos-dialogs.cpp: In member function
'virtual Common::DialogManager::DialogResult AmigaOSDialogManager::showFileBrowser(const Common::U3
2String&, Common::FSNode&, bool)':
backends/dialogs/amigaos/amigaos-dialogs.cpp:123:9: error: invalid conversion from 'const void*' to 'void*' [-
fpermissive]
    free(newTitle);
      ^~~~~~
In file included from /SDK/newlib/include/stdio.h:29,
                 from ./common/scummsys.h:118,
                 from backends/dialogs/amigaos/amigaos-dialogs.cpp:28:
/SDK/newlib/include/stdlib.h:86:20: note:   initializing argument 1 of 'void free(void*)'
_VOID _EXFUN(free,(_PTR));
      ^~~~
gmake: *** [backends/dialogs/amigaos/amigaos-dialogs.o] Error 1
```