
Subject: : AmigaOS4

Topic: : Porting to AmigaOS4 thread

Re: Porting to AmigaOS4 thread

Author: : Raziell

Date: : 2020/11/20 9:05:55

URL:

@Capehill

Could you help me to

1 - get rid of the warnings

2 - to use utf8

3 - polish (or rather fix) the code

please

especially utf8 is way beyond me

Thank you

These are the warnings and the utf8 error

```
C++ backends/dialogs/amigaos/amigaos-dialogs.o
```

```
backends/dialogs/amigaos/amigaos-dialogs.cpp: In member function
```

```
'virtual Common::DialogManager::DialogResult AmigaOSDialogManager::showFileBrowser(const Common::U32String&, Common::FSNode&, bool)':
```

```
backends/dialogs/amigaos/amigaos-dialogs.cpp:44:18: warning: declaration of 'AslBase' shadows a global declaration [-Wshadow]
```

```
struct Library *AslBase = IExec->OpenLibrary("asl.library", 50);
```

```
^~~~~~
```

```
In file included from backends/dialogs/amigaos/amigaos-dialogs.cpp:38:
```

```
/SDK/include/include_h/proto/asl.h:30:27: note: shadowed declaration is here
```

```
extern struct Library * AslBase;
```

```
^~~~~~
```

```
backends/dialogs/amigaos/amigaos-dialogs.cpp:45:19: warning: declaration of 'IAsl' shadows a global declaration [-Wshadow]
```

```
struct AslIFace *IAsl = NULL;
```

```
^~~~
```

```
In file included from backends/dialogs/amigaos/amigaos-dialogs.cpp:38:
```

```
/SDK/include/include_h/proto/asl.h:47:27: note: shadowed declaration is here
```

```
extern struct AslIFace *IAsl;
```

```
^~~~
```

```
backends/dialogs/amigaos/amigaos-dialogs.cpp:65:56: warning: cast from type 'const value_type*' {aka 'const char*'} to type 'char*' casts away qualifiers [-Wcast-qual]
```

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

```
AR    backends/libbackends.a
RANLIB backends/libbackends.a
C++   base/version.o
AR    base/libbase.a
RANLIB base/libbase.a
LINK  scummvm
backends/libbackends.a(amigaos-dialogs.o): In function `
Common::SharedPtrDeletionInternal::~~SharedPtrDeletionInternal()':
./common/ptr.h:43: undefined reference to `AmigaOSDialogManager::utf8ToLocal(char*)'
gmake: *** [scummvm] Error 1
```

and here is amigaos-dialogs.cpp

```
/* ScummVM - Graphic Adventure Engine
 *
 * ScummVM is the legal property of its developers, whose names
 * are too numerous to list here. Please refer to the COPYRIGHT
 * file distributed with this source distribution.
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License
 * as published by the Free Software Foundation; either version 2
 * of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
 */

#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/exec.h>
```

```
#include <proto/dos.h>
#include <proto/asl.h>
```

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

```
    char pathBuffer[MAXPATHLEN];
```

```
    struct Library *AslBase = IExec->OpenLibrary("asl.library", 50);
    struct AslIFace *IAsl = NULL;
```

```
    DialogResult result = kDialogCancel;
    Common::String utf8Title = title.encode();
```

```
    IAsl = (struct AslIFace*)IExec->GetInterface(AslBase, "main", 1, NULL);
```

```
    if (AslBase) {
```

```
        struct FileRequester *fr = NULL;
```

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

```
        fr = (struct FileRequester *)IAsl->AllocAslRequestTags(ASL_FileRequest, TAG_DONE);
```

```
        if (!fr)
            return result;
```

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

```
        if (IAsl->AslRequestTags(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);
        IAsl->FreeAslRequest((APTR)fr);
        IExec->CloseLibrary(AslBase);
    }
```

```
    return result;
```

}

#endif