

Dorkalize

COLLABORATORS

	<i>TITLE :</i> Dorkalize		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 10, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Dorkalize	1
1.1	Dorkalize	1
1.2	What's this?	2
1.3	Why this?	2
1.4	Disclaimer	2
1.5	Requirements	2
1.6	Usage	2
1.7	Using CatComp with Dorkalize	3
1.8	Main window	4
1.9	String window	5
1.10	Filtered string window	6
1.11	Filter window	6
1.12	Option window	7
1.13	Menus	7
1.14	Bugs and limitations	8
1.15	History	8
1.16	Distribuzione	9
1.17	Contacts	9

Chapter 1

Dorkalize

1.1 Dorkalize

DORKALIZE 0.3

by

Francesco Borghese

Giacomo Di Giacomo

What's this?

Why this?

Disclaimer

Requirements

Usage

Generating catalogs

Windows

Main window

String window

Filtered string window

Filter window

Setting options

Menus

Bugs and limitations

History

Distribution

Contacts

1.2 What's this?

Dorkalize is a utility that takes C source files and adds support for AmigaOS localization by replacing the strings contained in them with calls to the appropriate OS functions, and generates the catalog descriptor files used by CatComp to create catalogs. This allows you to program without cluttering your sources with calls which make them much less readable.

This program was started to be a replacement for the buggy parsing function of Commodore's "Localize" program, which is used to localize C source programs. At least for us, it has never worked. It has then turned into a powerful utility featuring advanced string filtering and automatic patching of source files.

We will add built-in editing and translation of strings, recovery of translated strings from older catalog descriptors, and spawning of CatComp.

1.3 Why this?

Because when we tried to localize our Italian lotto program AmiSuperLotto we discovered, much to our disappointment, that Localize did not work correctly. Specifically, the line numbers to patch were incorrectly reported. In addition to this, we would like to develop a program to suit our needs (and maybe other programmers') and allow for a quick translation of programs.

1.4 Disclaimer

No warranty is given that Dorkalize will perform the expected task. We take no responsibility for any kind of damage that Dorkalize can cause to its users.

1.5 Requirements

Dorkalize requires any Amiga with at least OS 2.1 and MUI 3.0 or better, the MUI custom class NList.mcc, and the Commodore localization tool Localize (you can find Localize in the 3.1 Native Developer Kit).

1.6 Usage

When you start Dorkalize, you will be presented with the main window . The first time you start Dorkalize you should set up all the options , in particular you should tell Dorkalize where Localize is found. This is required to generate the patched source files and the catalog descriptor file.

When you have set the global options, you should set the options for your current project. Select the "Project options" tab in the main window to set them up. You should specify the descriptor file name and the string output file name.

You can also choose a filter file from the "Filters" tab to automatically exclude some strings from the localization process. The filter window allows you to edit filter files. A general-purpose filter file, `dorkafilter.dflt`, containing patterns for strings which you usually don't want to localize, is supplied with Dorkalize.

Now you can choose the files you want to localize. After you have done so, press the button labeled "Dorkalize" to parse the files.

Dorkalize will open the string window , which contains the strings which survived the filtering process. To go on with the localization process, remove the strings you don't want to translate double-clicking on them or selecting them and then pressing the "Remove" button. You will find the strings you removed in the filtered string window , which contains a list similar to the one above. Double-click on any entry in this list to move it back to the initial list; alternatively you can select some entries and move them back with the "Add back" button.

At this point, you can patch your sources and generate the catalog descriptor by pushing the "Patch sources" button. If you want to generate the patch file without actually patching the sources, press the "Create patch file" button.

See

Generating catalogs
to learn how to use the catalog descriptor
to generate catalogs with CatComp.

1.7 Using CatComp with Dorkalize

You can use CatComp to generate a blank translation file from a catalog descriptor with:

```
CatComp <catalog descriptor.cd> CTFILE=<blank translation file.ct>
```

You can refer to CatComp's guide for a full explanation of how to edit this to get a translation file. Finally, use

```
CatComp <catalog descriptor.cd> <translation file.ct>  
  CATALOG=<catalog file.catalog> CFILE=<header file.h>
```

to generate all the files you need.

Note that you need to include the generated header file in your source and define the constant CATCOMP_ARRAY before compiling (at least this is the method we use).

1.8 Main window

The main window contains a register with three tabs.

The "Source files" tab contains the list of the source files to process. You can add files entering their name in the string gadget below or selecting them with the associated popup gadget (which supports multiple selection).

The "Project options" tab contains:

- A text gadget displaying the command line used to invoke Localize. You cannot alter its contents directly, but only acting upon the various options.
- A checkmark labelled "Merge catalogs". When selected it activates the MERGECATALOG option of Localize, which means that you will get only one catalog descriptor file for all of your source files. It is selected by default and should stay so unless you have any special requirements (e.g. the source files refer to more than one executable file). When selected, the descriptor file name is decided by the contents of the "Descriptor name" gadget.
- A string gadget and associated popup labelled "Descriptor name". If you selected the "Merge catalogs" option, you must insert here the path to the catalog descriptor (.cd) file you want to create.
- A checkmark labelled "Localize strings in defines". When selected, the strings found inside #define directives are by default localized, otherwise they are not. You can individually change this attribute for each one of them in the
 string window
 .
- A string gadget and associated popup labelled "String output file". You must insert here the path to the patch file created by Dorkalize as an input to Localize.

The "Filters" file contains a string gadget and associated popup labelled "Filter file". This gadget contains the path to the file containing the filters used to rule off strings while scanning the source files. You can view and edit its contents by pressing the "Show filters" button, which will

open the

```
filter window
.
```

Under the register file is located the "Dorkalize" button, which starts the parsing of the files and opens the

```
string window
. This is the only moment in
```

which filtering is applied.

1.9 String window

This window contains a listview holding the strings that Dorkalize ←→ found in the selected source files. Each line refers to a string. The listview has seven columns:

The "Source" column contains the source file from which the string was extracted.

The "Pos" column contains a character which is d or D if the string was contained in a #define, S if the string was contained in the body of a function, and E otherwise. Strings marked with D or E will not be localized. Dorkalize will set the attribute for strings in #defines depending upon the status of the "Localize strings in defines" checkmark in the option window

Double-clicking on this column will toggle between the d and D attributes.

The "Line", "Start" and "End" columns contain the line of the source file, the initial column and the final column where the string is found.

The "Msg #" column contains the number that Dorkalize assigns to each string. Identical strings in different lines have the same number.

The "Text" column contains the text of the message.

Below the listview are six buttons. The "Remove" button will remove the selected strings from the list and put them into the filtered strings list. The "Add to filter" button adds the current string to the filter list, without removing it from the string list. You have to press again the Dorkalize button to have the new filters take effect. The "Show filtered" button opens the

```
filtered string window
. The "Show filters" button opens the
filter window
.
```

The "Create patch file" button will create a patch file containing all and only the strings in the list. The "Patch sources" button will create the same patch file and then start Localize to generate the localized C sources and the catalog descriptor. See the description of the project options in the

```
main window
to learn how to configure the process. The localized C sources
```


will be put into the "Localized-source" subdirectory of the directory containing the original source file.

Double-click on any column of a line (except the "Pos" column for 'd' or 'D' strings) to remove the string from the list and insert it into the filtered strings list (see later). Double-click on the "Pos" column of a 'd' or 'D' string to toggle its status between 'd' and 'D'.

1.10 Filtered string window

This window contains a listview holding the strings that were filtered off the source files or removed later by the user from the list in the string window.

The "Source" column contains the source file from which the string was extracted.

The "Pos" column contains a character which is d or D if the string was contained in a #define, S if the string was contained in the body of a function, and E otherwise.

The "Line", "Start" and "End" columns contain the line of the source file, the initial column and the final column where the string is found.

The "Msg #" column contains the number that Dorkalize assigns to each string. Identical strings in different lines have the same number.

The "Text" column contains the text of the message.

Below the listview are two buttons. The "Add back" button will remove the selected strings from the list and put them back into the strings list in the

string window. The "Add to filter" button adds the current string to the filter list, without removing it from the filtered string list. You have to press again the Dorkalize button to have the new filters take effect.

Double-click on any column of a line to remove the string from the list and insert it back into the strings list (see string window).

1.11 Filter window

This window contains a listview holding the filters for your current project.

The listview has three columns. The "Filter" column contains the filters. The "Type" column controls whether the filter is a stopping or a passing one. The "Status" column controls whether the filter is active; if it contains

"INACTIVE" the filter is ignored.

Each filter is an AmigaDOS standard pattern. Filtering works like this: Dorkalize checks every strings it finds in the source files against each active filter in sequence, top to bottom of the list. When it finds a filter matching the string, if the filter is a stopping one the string will not be added to the list of strings to be localized, otherwise it will be added; in any case Dorkalize will go on with the next string in the source files. Matching is case-sensitive. If the string does not match any filter it will be added to the list.

From above derives that the filters appear in the list by descending priority, top to bottom. This means that, if you want for example to exclude all the strings starting with "dork", except "dorkalize", you have to insert "dorkalize" as a passing filter and, below it, "dork#?" as a stopping filter. You can refer as an example to the supplied file, dorkafilter.dflt, which contains some filters you should always keep active.

On the right of the listview are two buttons that you can use to move the active filter up and down the list. You can also move selected filters using drag-n-drop. Double-clicking on the Filter column removes a filter from the list, while double-clicking on the Type and Status column will toggle the contents of the clicked field. The active filter can be edited in the string gadget below the listview; the changes are entered by pressing the Enter key.

In the bottom part of the window are seven buttons. "New" inserts a new filter. "Toggle type" toggles the type of the selected filters between STOP and PASS. "Toggle status" toggles the status of the selected filters between ACTIVE and INACTIVE. "Remove" removes the selected filters. "Clear" empties the list. "Save" saves the list into the file specified in the "Filters" tab in the

main window

. "Save as" saves the list into a user-specified file, which then is also set as the current filter file.

1.12 Option window

The options window is opened from the "Set options..." menu item in the "Options" menu. It contains:

- A string gadget with associated ASL popup gadget marked "Localize executable:". This must contain the location of the Localize executable.
- A set of radio buttons and a string gadget with associated popup labelled "Startup project". From here you can select if the next time you start Dorkalize the startup project will be none, the last project you saved, or a default project specified in the string gadget.
- Two buttons labelled "Use" and "Save", whose function is straightforward.

1.13 Menus

Dorkalize has two menus: the Project and the Options menu.

The Project menu contains the following items:

- New project (shortcut RightAmiga + N): Clears all the current project files and filter file name and resets the project options to defaults.
- Load project (shortcut RightAmiga + L): Loads the project files, project options and filter file from a user-specified project file.
- Save project (shortcut RightAmiga + S): Saves the project files, project options and filter file to the same file they were written last time. If this is the first time you save the project, you will be prompted for a file name.
- Save project as: Saves the project files, project options and filter file to a user-specified project file.
- About: Displays information about Dorkalize.
- About MUI: Displays information about MUI.
- Quit (shortcut RightAmiga + Q): Exits Dorkalize.

The Options menu contains only the "Set options" item, which opens the global option window

.

1.14 Bugs and limitations

- Filtering cannot be safely applied to strings containing AmigaDos special characters as '#', '?', '%', '|'. When needed, those strings must be removed by hand.
- Strings containing the '|' character are truncated when the descriptor file is generated. This is more a limitation of Localize rather than Dorkalize. Those strings must be corrected by hand in the descriptor file until the patch function of Localize will be replaced by an internal and, hopefully, bug-free one.
- The up/down buttons in the filter windows don't work correctly if the active filter name is splitted in two or more rows. That is, I believe, a NList bug. You can bypass this problem dragging the filter up and down the list or changing the width of the filter name column.

1.15 History

0.3	14/07/00	Added projects. Added filtering. Added confirmation requesters before quitting or saving.
-----	----------	---

- Reorganized the GUI.
Removed some bugs in parsing functions.
- 0.2 12/05/00 Added selection of strings to localize.
Added menus.
Added spawning of Localize.
Added support for strings in #defines and outside functions.
- 0.1 26/08/99 First release. Replaces Localize parse function.

1.16 Distribuzione

This program is freeware. You can use and spread it any way you like. Please email us if you use it and you think there is something which does not work or could be improved.

1.17 Contacts

You can contact the authors at the following addresses:

FRANCESCO BORGHESE
VIA G. SPINEDI 39
00015 MONTEROTONDO (ROMA)
ITALY

email: fraborg@tiscalinet.it

ICQ UIN: 66104693

<http://members.tripod.com/BorgheseF/>

Feel free to send comments and suggestions. Money and gifts will not be rejected with disgust.
