

Localizer

Massimiliano Origi

Copyright © (c)1996-97 Intuizione. All Rights Reserved.

COLLABORATORS

	<i>TITLE :</i> Localizer	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY	Massimiliano Origi	March 1, 2023
<i>SIGNATURE</i>		

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Localizer	1
1.1 Localizer Guide	1
1.2 Introduction	2
1.3 System Requirements	2
1.4 Installation	3
1.5 Main Window	3
1.6 Info Window	3
1.7 Edit Project	3
1.8 Load Project	4
1.9 Save Project	4
1.10 Scan Sources	5
1.11 Edit Strings	5
1.12 Translate Strings	7
1.13 Search Window	9
1.14 Preview Window	9
1.15 History	10
1.16 Credits	11
1.17 Legal stuff	11
1.18 Author info	11
1.19 Supported Escape Sequences	12
1.20 Index	12

Chapter 1

Localizer

1.1 Localizer Guide

Localizer 1.31

The Locale Development Tool.

Copyright © 1996-97 Massimiliano Origgi. All Rights Reserved.

```
~Introduction~~~~~
  What is it?

~System~Requirements~
  What you need

~Installation~~~~~
  Where do I put it?

~Main~Window~~~~~
  The Main Window

~Info~Window~~~~~
  The Information Window

~Edit~Project~~~~~
  Edit Project Settings

~Load~Project~~~~~
  Load New Project

~Save~Project~~~~~
  Save Current Project

~Scan~Sources~~~~~
  Scan Source Code

~Edit~Strings~~~~~
  Edit CD Files
```

```
~Translate~Strings~~~
    Edit CT Files

~Search~Window~~~~~
    The search engine

~Preview~Window~~~~~
    Quickly preview your strings

~History~~~~~
    A look into the past

~Credits~~~~~
    I wish to thank...

~Legal~stuff~~~~~
    Boring but necessary

~Author~Info~~~~~
    How to reach me

~Escape~sequences~~~
    Supported escape sequences
```

1.2 Introduction

Localizer has been created to help both programmers and translators.

Here is a brief list of what Localizer can do:

- scans source files, extracts locale IDs and builds a .cd file;
- merges old and new .cd and .ct files and hilights differences for fast ↔ editing;
- builds C source files to handle locale stuff;
- can add comments to each .cd ID, useful for translators;
- builds .catalog files;
- fast load/save of projects;
- bubble help for each gadget;
- quickly moves among new/changed entries for fast update of .cd and .ct files ↔ ;
- and even more...

This is just the beginning of Localizer a lot of even more advanced functions ↔ will~be added in the future!

1.3 System Requirements

Localizer needs:

- A 68020 or better
- AmigaOS 3.0 or better
- MUI 3.6 or better

1.4 Installation

Installing Localizer is really easy: just put its drawer wherever you want!

1.5 Main Window

This is the main window of Localizer, you can access all functions from here.

Localizer does not use any menu.

All gadgets have an associated bubble help with a brief description of their use.

All time consuming operations will be noticed with the usual busy pointer.

The window title bar will show the name of the currently loaded project or "---" ↔
if no

project is currently loaded or the project has no name.

1.6 Info Window

This window displays copyright notes about Localizer and MUI and
my~addresses

.

1.7 Edit Project

Here you can edit all the necessary options to work with Localizer .

Project name.

This name will be used as a prefix for .cd, .ct and .catalog files.

For example if you write 'test', Localizer will read and write test.cd,
test.ct,~test.catalog.

Scanner.

From version 1.24 Localizer offers 2 different source scanners.

The ID scanner is the same of the previous Localizer version; it scans the source ↔
code

to look for strings starting with the characters specified in the string gadget ↔
besides

the scanner cycle gadget.

Remember: ALL AND ONLY the locale IDs MUST use this prefix.

For example if you write 'MSG_', Localizer will get strings like MSG_OK,~
MSG_GADGET_CONTINUE, etc.

The Func scanner, instead, will look for fake functions whose name is specified ↔
in the

above mentioned string gadget.

If you specify 'STRING' as the fake function name, the scanner will look at functions with this syntax: STRING(<locale ID>,<default string>). So you can have functions like: STRING(MSG_OK,"OK"). Thanks to this syntax Localizer can completely build the .cd file from your code, both the IDs and the default strings! You don't have to write the strings by hand in the .cd file anymore, everything is inside your source, saving a lot of time and boring work!

Locale source dir.

In this directory Localizer will save the C source code it generates. Localizer generates source code for both type of scanners.

Source files.

Here you can specify the source file Localizer will scan.

The list supports multiselection and you can sort the entries with drag & drop. The Add button opens a standard asl requester where you can select one or more source-files.

The Del button removes all selected files from the list.

The Clear button will remove all files from the list.

Catalogs dir.

This is the directory which contains all the .cd, .ct and .catalog files in the~standard locale structure.

That is, if you write 'RAM:' and the project name is 'test', Localizer will search,~for example, RAM:test.cd, RAM:italiano/test.ct or RAM:italiano/test. catalog.

The catalogs directory contains the .cd file and the directories named after their~language. These directories contain the .ct and .catalog files.

1.8 Load Project

Here you can load a new project.

The list displays all the currently available projects.

You can select a project in the list or write its name in the string gadget below

Press OK to load the project or Cancel to abort the operation.

From Localizer 1.24 there is also a Del gadget besides the string gadget. It will be

enabled only when the string gadget contains the name of an existing project file.

The close window gadget acts just like the Cancel button.

1.9 Save Project

Here you can save the project you are currently working on.

The list displays all the currently available projects.

You can select a project in the list or write its name in the string gadget below

If a project with the selected name already exists, it will be overwritten.

From Localizer 1.24 there is also a Del gadget besides the string gadget. It will be enabled only when the string gadget contains the name of an existing project file. Press OK to save the project or Cancel to abort the operation. The close window gadget acts just like the Cancel button.

1.10 Scan Sources

As soon as you press the Scan Sources button in the main window, Localizer will start scanning your source files. A small window will display the status of the operation: number of scanned files and total number of files. You can't currently abort this operation, sorry. The scanner function generates a file with the .newcd extension in your catalogs directory. This file will be read when you use the Edit Strings function.

1.11 Edit Strings

Here you can finally edit your .cd file. Localizer will look at both a .newcd file and a .cd file. If only a .newcd file exists, it will load it and sort entries alphabetically. If only a .cd file exists, it will simply load it. If both a .newcd and a .cd file exist, it will merge them. If an entry is present only in the .newcd file, it will be marked as NEW. If an entry is present only in the .cd file, it will be removed. If an entry is present in both files, it will be marked as OK.

Author.

Enter here the author name.

Language.

Select the language for this .cd file.

Either enter it in the string gadget or select one from the popup list.

Version.

Enter the version number of this .cd file.

IDs list.

This list displays all available IDs.

It's divided in 4 columns: they contain the ID name, the id-number, the status of this ID (see below) and the text of this ID.

There are 3 possible status codes:

- new(the ID has just been added after the last scan),
- Empty(no text),
- OK(self explanatory :-).

Text String.

In the string below the list you can enter the text of the currently selected ID.

When this gadget is active, you can move through the entries of the list with the ←
~up~
and down cursor keys.

You can enter
escape~sequences
here.

Min Length.

Here you can select the min length that the translated text (in the corresponding
.ct~file) must have.

You can enable and disable this option with the checkbox gadget on the left.

Max Length.

Here you can select the max length that the translated text (in the corresponding
.ct~file) must have.

You can enable and disable this option with the checkbox gadget on the left.

ID Value.

Here you can choose the numeric value associated with this ID.

If the checkbox is disabled a default value will be chosen by Localizer.

Next.

When you press this gadget Localizer will jump to the next New or Empty entry if ←
any,
unless the current entry is still New or Empty.

Prev.

When you press this gadget Localizer will jump to the Previous New or Empty entry ←
if
any, unless the current entry is still New or Empty.

Search.

When you press this gadget Localizer will open the
search~window

.

Preview.

When you press this gadget Localizer will open the
preview~window
(if not already open)
and display a preview of the current entry.

Comment String.

Here you can enter a comment string for the current ID. It's useful when you ←
have~to~
send the .cd file to translators.

Save CD.

It saves the .cd file over the old one.

Build Source.

It generates 2 C source files to handle the .cd file: Locale.c and Locale.h.
They will be saved in the source dir you have selected in the

Edit~Project~window

.

There are 3 functions to open the catalog, close it and get a string.

Cancel.

Just cancels all changes and closes the window.

The close gadget does the same function.

If any string has been modified, it will put up a requester asking for confirmation before closing the window.

1.12 Translate Strings

Here you can select the language of the .ct file you want to edit or create.

Select Language window.

The list displays the languages a .ct file exists for.

You can select a language in the list, type its name in the string gadget or select it from the popup list.

If you press Cancel or the close window button, the operation is cancelled.

If you press OK, the Translate window opens.

It's not too much different from the Edit Strings window.

Translate strings window.

Localizer will look at both a .cd file and a .ct file.

If only a .cd file exists, it will simply load it.

If only a .ct file exists, it will simply load it, if the .ct file has not been generated by Localizer, it won't contain enough information to make it usable, so you

really need the .cd file. If, instead, the .ct file has been generated by Localizer you

won't need the .cd file anymore, because the .ct file contains all necessary informations.

If both a .cd and a .ct file exist, it will merge them.

If an entry is present only in the .cd file, it will be marked as NEW.

If an entry is present only in the .ct file, it will be removed.

If an entry is present in both files, it will be marked as OK if the default string is the same in both files, as Changed if it is different.

Author.

Enter here the author name. it will be saved in the .catalog file.

Language.

Displays the language of this .ct file.

Version.

Enter the version and revision number of this .ct file.

IDs list.

This list displays all available IDs.

It's divided in 4 columns: they contain the ID name, the id number, the status of this ID (see below) and the translated text of this ID.

There are 5 possible status codes:

- new (the ID~has~just~been~added~after~the~last~scan),
- Empty (no text),
- Changed (the original text in~the~.cd~file~has ben changed),
- WrongSize (the string lenght is not within the min/max values),
- OK (self explanatory :-).

Text String.

In the list below the original text of this ID is displayed.

Translation String.

In the string gadget below you can enter the translation for the current ID.

You can enter

- escape~sequences
- here.

Min Size.

Min size allowed for current entry.

Current Size.

Current size of selected entry.

Max Size.

Max size allowed for current entry.

Next.

Goes to the next New, Changed, Empty or WrongSize entry if any, unless the ←
current
entry is not OK.

Prev.

Goes to the previous New, Changed, Empty or WrongSize entry if any, unless the ←
current
entry is not OK.

Search.

When you press this gadget Localizer will open the
search~window

.

Preview.

When you press this gadget Localizer will open the
preview~window
(if not already open)
and display a preview of the current entry.

Comment String.

Displays the comment for the current ID.

Save CT.

It saves the .ct file over the old one.

Build Catalog.

It generates the .catalog file.

Cancel.

Just cancels all changes and closes the window.

The close gadget does the same function.
If any string has been modified, it will put up a requester asking for confirmation before closing the window.

1.13 Search Window

The search window is available both when you edit cd and ct files and it's fully asynchronous.

The search window in ct mode will offer you a cycle gadget at the top, so you can decide if you want to perform the search among the original strings or the translated ones.

In the string gadget you can enter the string you want to search; the search engine fully supports ALL standard AmigaOS wildcards (just like AmigaDOS programs). Due to this usefull feature, if you want to search a substring inside your ones, you have to surround it with the '#?' sequence. For example to search for 'prog' you have to type '#?prog#?'.

With the case cycle gadget you can perform a case sensitive or insensitive search

The 3 bottom gadgets finally perform the search operation:
the 'All' button searches from the first string in the list;
the 'Next' button searches forward from the currently selected entry in the list;
the 'Prev' button searches backward from the currently selected entry in the list

You can close and reopen the window at any time and it will preserve its contents

1.14 Preview Window

The preview window is available both when you edit cd and ct files and it's fully asynchronous.

The preview window in ct mode will offer you a cycle gadget at the top, so you can decide if you want to preview the original strings or the translated ones. You can quickly switch between a preview of the original string and of its translation with a simple click on this button!

The text area below will display the parsed string (as saved in the catalog file) of the currently selected entry.

To change the entry, simply select a new one in the list and press the preview button

again.

You can close and reopen the window at any time and it will preserve its contents ↔

1.15 History

Version 1.0

- First release

Version 1.1 to 1.23

- Internal versions

Version 1.24

- Removed a couple of bugs which caused a lot of generated .catalog files to be unreadable.
- New layout of the main window.
- The project name is now displayed in the main window title bar.
- Cycle chain now works in the edit ct window.
- Revised some parts of this doc.
- No more problems when saving .ct files.
- Revised several parts of the program.
- Does not quit anymore when the iconify gadget gets pressed.
- Added cancel button in the Edit Project window.
- Handles correctly strings starting with the '#' or ';' characters.
- Added del button in the load/save windows.
- Removed a couple of bugs in the cd and ct parsers.
- Added several error requesters.
- Added a new source scanner.
- You can now generate source code or the .catalog file even if not all ↔ entries are ok.

Version 1.25 to 1.28

- Internal versions

Version 1.29

- Added context sensitive AmigaGuide help.
- Performs some optimizations on the generated catalog files, they will be a ↔ bit smaller often.
- Faster cd and ct loading in most cases.
- Added search window.
- Added preview window.

Version 1.30

- Now saves comment lines in .cd and .ct files only when present.
- Fixed a bug when loading comments from .cd and .ct files.
- Puts up a requester before closing the .cd or .ct edit windows if the ↔ strings have been modified.
- Checks the format of the version string before generating a .catalog file.
- NEW home page!

Version 1.31

- Now if you save the cd or ct file after modifying it and then close the window, Localizer won't put up the 'strings have been modified' requester anymore.
- Fixed conversion of hexadecimal escape sequences (\xHH) in both the preview window and the catalog generator.

1.16 Credits

I want to thank:

Ole Friis, Paolo Menichetti, Laurent Kempè for their ideas and for beta testing.

Krister Larsson, Peter Eriksson and Rune Jacobsen for bug reports.

Oliver Wagner for his interesting suggestions.

Stefan Stuntz for his great MUI package.

Dietmar Eilert for Gilded.

And all the ones I forgot to mention!

1.17 Legal stuff

Localizer is Copyright (c) 1996-97 Massimiliano Origgi and it's Giftware.

You can freely distribute it as long as all of its files are included in their original form without additions, deletions or modifications of any kind, and only a

nominal fee is charged for its distribution to cover media and expedition costs.

This software is provided as is without warranty of any kind either expressed or implied.

By using Localizer you agree to accept the entire risk as to the quality and performance of the program.

Of course it was extensively tested before it was released...

Localizer is GiftWare. It means that if you like it and use it to localize your applications or translate catalogs, you should send me a gift which will act as a payment for the usefulness of Localizer .

This way you will support further development of Localizer!

Localizer will be ported to pOS as soon as possible.

I hope you'll enjoy it and find it useful!

1.18 Author info

Send comments, bug reports, complaints and ideas to:

origgi@pointest.com (Massimiliano Origgi)

Watch my home page for bug fixes and updates at:

<http://amigaworld.com/intui/>

1.19 Supported Escape Sequences

Both catalog description and catalog translation strings can contain the following escape sequences:

<code>\b</code>	Backspace (Ascii 8)
<code>\c</code>	Control Sequence Introducer (Ascii 155)
<code>\e</code>	Escape (Ascii 27)
<code>\f</code>	Form Feed (Ascii 12)
<code>\g</code>	Display beep (Ascii 7)
<code>\n</code>	Line Feed, newline (Ascii 10)
<code>\r</code>	Carriage Return (Ascii 13)
<code>\t</code>	Tab (Ascii 9)
<code>\v</code>	Vertical tab (Ascii 11)
<code>\</code>	The backslash itself
<code>\xHH</code>	A character expressed in hex digits.
<code>\OOO</code>	A character expressed in octal digits.

1.20 Index

Index of Localizer Guide

Load~Project

Author~info
Credits
Edit~Project
Edit~Strings
Escape~Sequences
History
Info~Window
Introduction
Legal~stuff~and~Author~info
Localizer~Guide
Main~Window
Preview~Window
Save~Project
Scan~Sources
Search~Window
System~Requirements
Translate~Strings
