

DebugStringView App

Introduction

It is an application that allows you to monitor the debug output on your local system. It's free and will show us all the strings that are displayed [using the Windows OutputDebugString API](#).

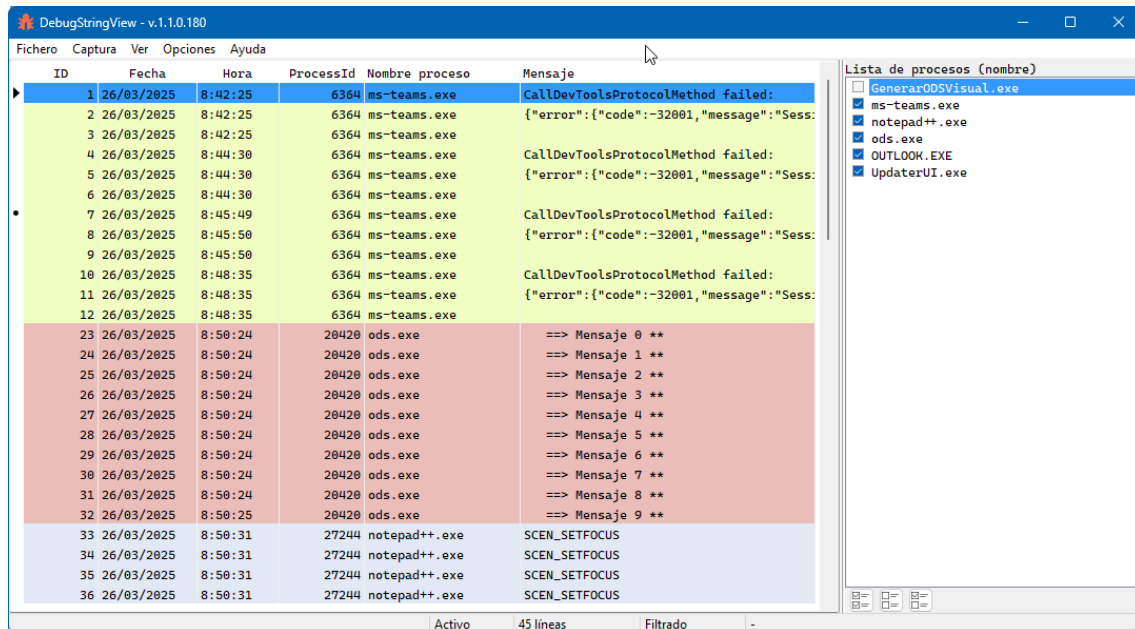
Index

DebugStringView App	1
Introduction	1
Index.....	1
Installation and use	2
Characteristics	2
Menu Options	2
File menu.....	3
Capture menu	3
View Menu	3
Options menu.....	4
Multilanguage	4
How to generate a new translation	4
Steps to generate a new translation	4



Installation and use

The application does not require installation, you simply need to unzip the file to your desired location and run it. Once started, message capture will begin.



Characteristics

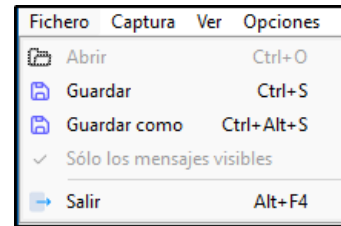
- It allows you to save the content of the capture in files with extension (*.dsvlog).
- Fast filtering of messages by processes (uses the name of the process), including N processes with different ProcessID, but the same name.
- Possibility to color the output of messages by process (better and clearer visualization for the user).
- Hide and show columns based on what interests you (more information or more space)
- Auto-advance as messages appear so that the latest messages sent to the console are always displayed.
- "Process List" panel that shows us the processes (ProcessName – process name) that are sending messages to the console.
- Allows quick change of the font used in the display with the most common monospaced fonts. Also the option to access the properties screen and change the rest of the features.
- The application is multi-language and easily expandable with more language files without the need for changes to the application (dynamic language loading).
- Check for updates available.

Menu Options

The app menu has the following options.

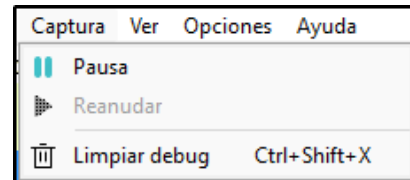
File menu

- **Open:** Not active at this time; It will allow you to open a file saved by the same application (not available in the initial version)
- **Save:** It allows us to save the messages on the screen in a file.
- **Save As:** Allows us to save the messages on the screen in a new file by defining the new name.



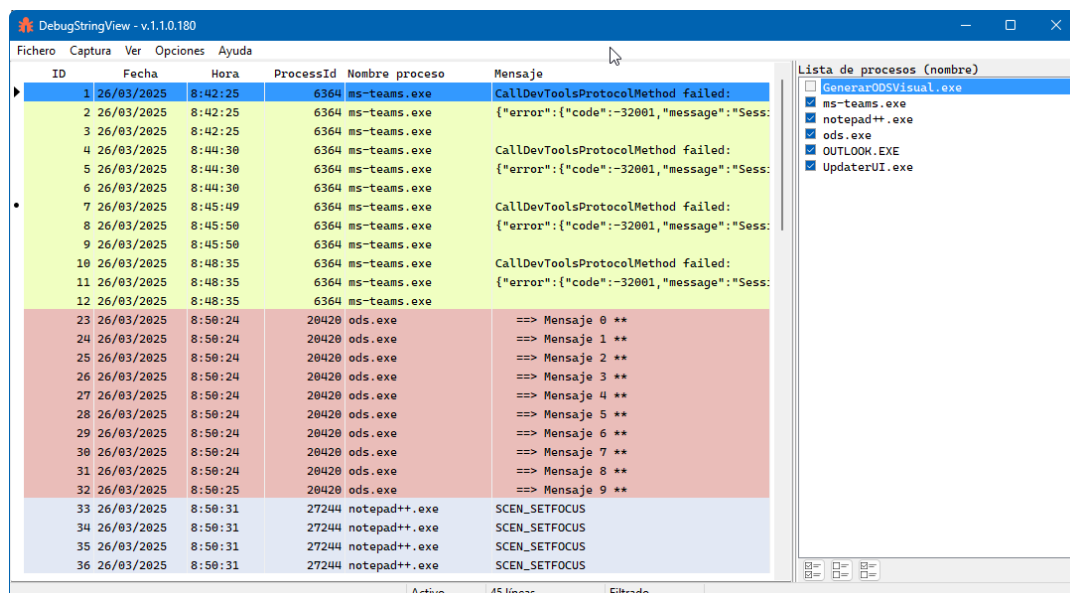
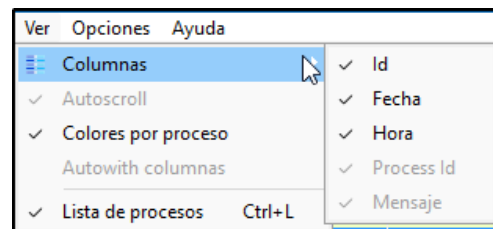
Capture menu

- **Pause:** Allows us to stop capturing messages temporarily.
- **Resume:** Allows the application to resume debug message capture.
- **Clean debug:** Deletes all messages captured by the app so far



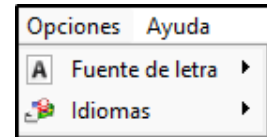
View Menu

- **Columns:** Allows you to view and hide some columns in the process message display (disabled columns cannot be hidden)
- **Autoscroll:** Automatically the view of messages advances when new ones appear.
- **Colors by Process:** Automatically assign each process that appears in the app a different color to improve on-screen display
- **Process list:** It allows you to view the list of processes that are displaying messages and also the list itself allows **you to filter** messages according to existing processes.



Options menu

Font fonts: Allows you to quickly change the font in which messages are displayed. It also has an option to access the rest of the properties of the font we are using.



Languages: It allows us to switch between the different languages configured in the app.

Multilanguage

The application works with multiple languages, which also load dynamically. More languages can be easily added (and without recompiling) and the application will recognize them and incorporate them into the menu automatically once it is run.

The application searches the "Languages" field for language files.

How to generate a new translation

To create a new translation you must use the free translation editor/generator of **dkLang** (translation package used in this application).

You can find information about "DKLang Translation Editor" here:

<https://yktoo.com/en/software/dklang-translation-editor/>

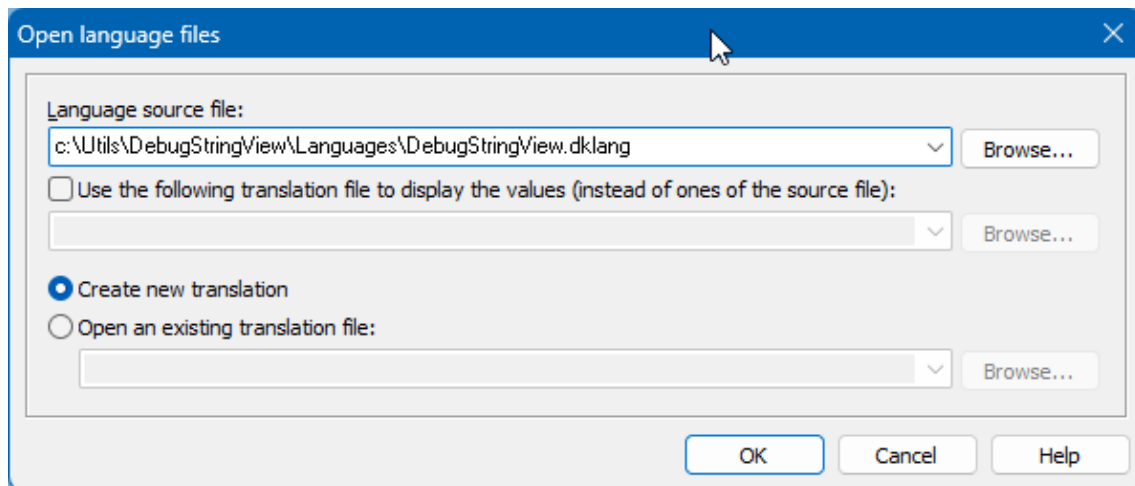
[And you can download it from here.](#)

To generate a new translation we only need the file with the original strings **<DebugStringView.dklang>** which is located in the **<Languages> directory** of the application.

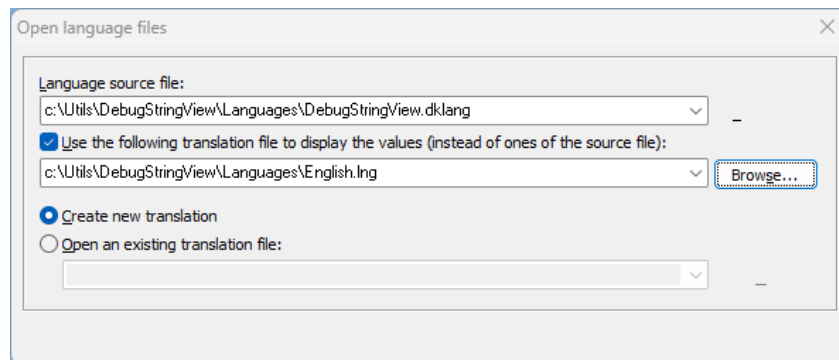
Steps to generate a new translation

STEP 1: select the file with the original strings **<DebugStringView.dklang>**; Additionally, if it is more convenient, you can choose an existing language file in the application as the original file to be translated.

Option 1: If we want to generate a new translation (Italian - Italy) from the original translation file **< DebugStringView.dklang >** in Spanish:

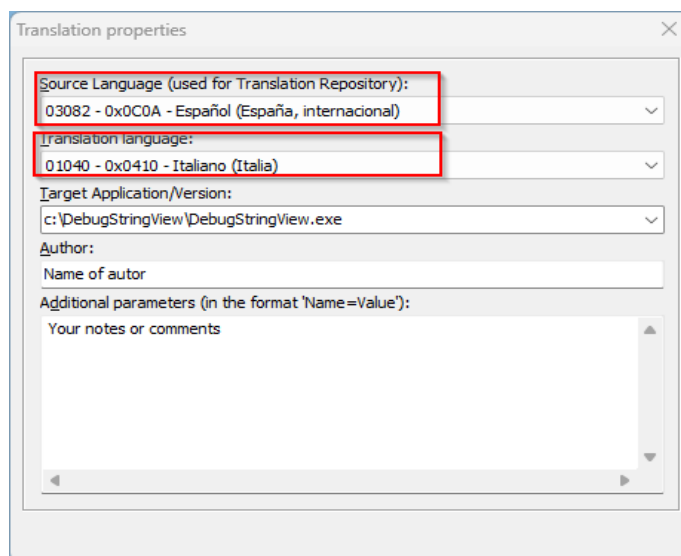


Option 2: We want to generate a new translation (Italian - Italy) using an existing translation (e.g. in English).

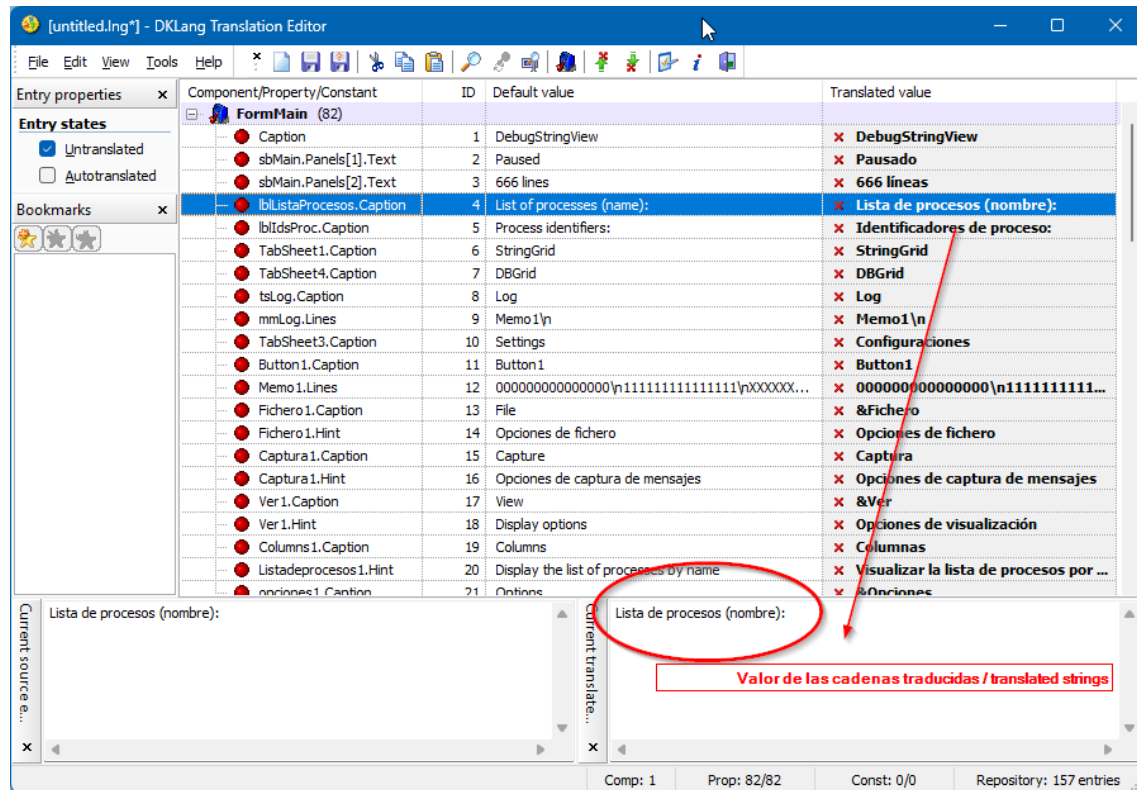


STEP 2: choose the language of the new translation (in this example Italian) and fill in the translation properties.

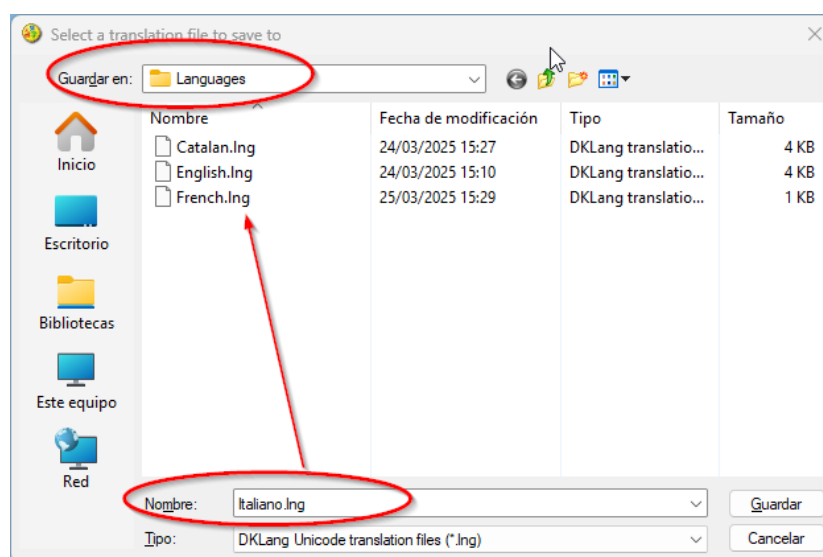
- **Source Language:** Must be 03082 – Spanish (Spain, International)
- **Translation Language:** Language of translation; in our case (Italian)
- **Other properties:** They are filled in to the user's liking.



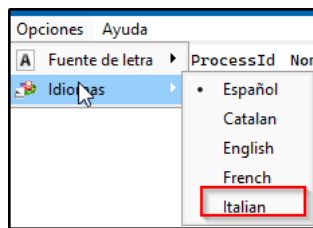
STEP 3: Perform the translations using the editor. The chains on the left are the originals, those on the right are the new translations.



STEP 4: Save the translation file in the <Languages> folder of the application. The name must be that of the new language and the extension must be .lng



When you enter the application again, the new translation will appear in the **Languages menu**.



IMPORTANT NOTE: If you generate a new translation (in whatever form) respect the syntax of the strings when you find %s (keep it in the same place in the translated string).