



## USER MANUAL

---

## INDEX OF CONTENTS

<b>Index of Contents .....</b>	<b>2</b>
<b>1. System requisites.....</b>	<b>7</b>
<b>1.1. User .....</b>	<b>7</b>
<b>1.2. Developer .....</b>	<b>7</b>
<b>2. Menu bar .....</b>	<b>8</b>
<b>2.1. File Menu .....</b>	<b>8</b>
<b>2.1.1. New.....</b>	<b>8</b>
<b>2.1.2. Open .....</b>	<b>8</b>
<b>2.1.3. Open Recent Files.....</b>	<b>8</b>
<b>2.1.4. Open All Files.....</b>	<b>8</b>
<b>2.1.5. Close File .....</b>	<b>8</b>
<b>2.1.6. Close All Files .....</b>	<b>8</b>
<b>2.1.7. Save File.....</b>	<b>9</b>
<b>2.1.8. Save File As.....</b>	<b>9</b>
<b>2.1.9. Save All Files .....</b>	<b>9</b>
<b>2.1.10. Print File .....</b>	<b>9</b>
<b>2.1.11. Exit .....</b>	<b>9</b>
<b>2.2. Edit Menu .....</b>	<b>10</b>
<b>2.2.1. Undo .....</b>	<b>10</b>
<b>2.2.2. Redo.....</b>	<b>10</b>
<b>2.2.3. Copy.....</b>	<b>10</b>
<b>2.2.4. Paste .....</b>	<b>10</b>
<b>2.2.5. Cut .....</b>	<b>10</b>
<b>2.2.6. Select All .....</b>	<b>10</b>
<b>2.2.7. Go To Line .....</b>	<b>11</b>

2.2.8. Search.....	11
2.2.9. Replace.....	13
<b>2.3. Project Menu .....</b>	<b>14</b>
2.3.1. New Project.....	14
2.3.2. Open Project .....	15
2.3.3. Open Recent Projects.....	15
2.3.4. Close Project .....	15
2.3.5. Save Project.....	16
2.3.6. Save Project As .....	16
2.3.7. New Project File .....	16
2.3.8. Add All Opened Files .....	16
2.3.9. Add File.....	16
2.3.10. Remove File.....	16
2.3.11. Delete File .....	16
2.3.12. Add Folder .....	16
2.3.13. Remove Folder.....	17
2.3.14. Compile Project .....	18
2.3.14.1. Compilation based on “Extension” .....	18
2.3.14.2. Compilation based on “Marked Files For Compilation” .....	18
2.3.15. Execute Project.....	19
2.3.16. Set Compilable File.....	20
2.3.17. Unset Compilable File .....	20
2.3.18. Set Main File.....	20
2.3.19. Unset Main File .....	21
<b>2.4. View Menu.....</b>	<b>22</b>
2.4.1. Show Log.....	22

2.4.2. Project Browser .....	22
2.4.3. Console Window.....	22
<b>2.5. Configuration Menu .....</b>	<b>23</b>
2.5.1. Lexicon Configuration .....	23
2.5.1.1. New Lexicon.....	23
2.5.1.2. Document Lexicon.....	23
2.5.1.3. Modify Lexicon.....	23
2.5.1.3.1. Reserved Words Configuration.....	24
2.5.1.3.2. Delimiters Configuration .....	25
2.5.1.3.3. Remarks Configuration.....	26
2.5.1.4. Default Lexicons .....	28
2.5.2. Grammar Configuration.....	29
2.5.2.1. New Grammar .....	29
2.5.2.2. Load Grammar .....	31
2.5.2.3. Modify Grammar .....	31
2.5.2.4. Save Grammar .....	32
2.5.2.5. Save Grammar As.....	32
2.5.2.6. Configure Paths .....	32
2.5.3. Compiler .....	34
2.5.4. File Editor Configuration.....	35
2.5.4.1. File Editor Display Options Configuration.....	35
2.5.4.2. Automatic Indent.....	36
2.5.4.3. Line Wrapping.....	36
2.5.4.4. Maximum Line number to Send to Console .....	36
2.5.4.5. Send File Contents to Console .....	36
2.5.5. Console Configuration .....	37

2.5.5.1. Configure .....	37
2.5.5.2. Execute External Command .....	37
2.5.5.3. Console Display Configuration.....	38
2.5.5.4. Save Content into File .....	39
2.5.5.5. Document Lexicon.....	39
2.5.5.6. Buscar .....	39
2.5.5.7. Close Console.....	39
2.5.5.8. Reset Console .....	39
2.5.5.9. Clear Console Buffer .....	40
2.5.6. Language Configuration .....	41
2.5.7. Menu Configuration.....	42
2.5.7.1. New .....	42
2.5.7.2. Load .....	42
2.5.7.3. Modify .....	43
2.5.7.4. Save .....	43
2.5.7.5. Save As.....	43
2.5.8. Tool Bar Configuration.....	44
2.5.8.1. New .....	44
2.5.8.2. Load .....	45
2.5.8.3. Modify .....	46
2.5.8.4. Save .....	46
2.5.8.5. Save As.....	46
<b>2.6. Help Menu .....</b>	<b>47</b>
2.6.1. Show Help .....	47
2.6.2. About Us .....	47
<b>2.7. Accesibility Shorcuts .....</b>	<b>48</b>



## 1. SYSTEM REQUISITES

### 1.1. USER

ACIDE - A Configurable IDE does not require neither special system configurations nor special system requirements because the executable file is attached in its distribution, making the execution of ACIDE - A Configurable IDE easy and comfortable to the users.

The only mandatory requirement is the previous installation of the **Java Virtual Machine (JVM)**. The user will have to get the *JRE installation file* with **1.6 and later versions**, which is available in the following link:

<http://www.java.com/es/download/manual.jsp>.

Only with this easy and fast step the user will be able to run ACIDE - A Configurable IDE on his computer without problems. However, in order to fully enjoying all the feautures of the application such as ACIDE - A Configurable IDE *grammar configurations*, two extra tools will have to be also installed: **javac.exe** and **jar.exe**.

Those tools are available in the **Java Development Kit (JDK)** installation file, which is available in the following link:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

At last, in order to visualize the present document, it is mandatory for the users to have previously installed any software for **PDF files visualization**.

### 1.2. DEVELOPER

For developers, it is mandatory to have previously installed the **Java Development Kit (JDK)** with 1.6 and later versions and any software for the edition of the source code.

The source code has been fully edited with the **Eclipse IDE** tool which is available in:

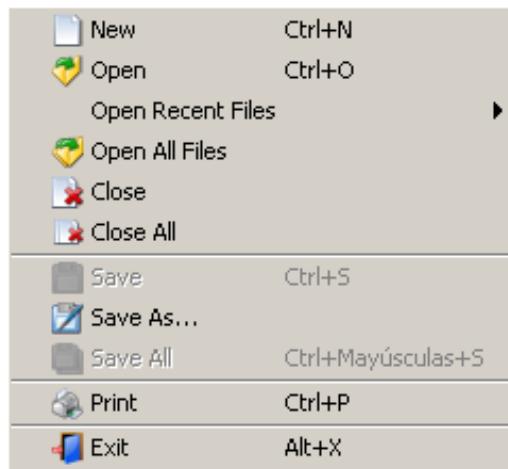
<http://www.eclipse.org/>

Furthermore, with the ACIDE - A Configurable source code distribution, the Eclipse *project file* is available. The developer has to import the project file into Eclipse and start the edition, fast and simple.

## 2. MENU BAR

### 2.1. FILE MENU

It contains the following menu items for the files management:



Next, all the previous menu items will be further explained:

#### 2.1.1. NEW

Creates a new empty file in the file editor.

#### 2.1.2. OPEN

Open a previously saved file into the file editor.

#### 2.1.3. OPEN RECENT FILES

Displays a list which contains all the files that have been previously opened in the file editor and the menu item to empty the list.

#### 2.1.4. OPEN ALL FILES

Open all the files associated to the current project in the file editor.

#### 2.1.5. CLOSE FILE

Close the active file in the file editor, asking to the user if he wants to save it if the file was previously modified.

#### 2.1.6. CLOSE ALL FILES

Close all the opened files in the file editor, asking to the user if he wants to save them if the files were previously modified.

### 2.1.7. SAVE FILE

Save the active file in the file editor at the same path that it was previously saved.

### 2.1.8. SAVE FILE AS

Save the active file in the file editor into a different path than it was saved before.

### 2.1.9. SAVE ALL FILES

Guarda todos los archivos abiertos en el editor de archivos.

### 2.1.10. PRINT FILE

Prints the active file in the file editor.

### 2.1.11. EXIT

Closes the application and if any changes have been encountered in the current project configuration displays the following dialog to the user:



Additionally, If any of the opened files in the file editor has been modified, it will ask to the user for saving them with the following dialog:



The user can abort the exit process in any time by cancelling any of the previous dialogs.

## 2.2. EDIT MENU

It contains the following menu items for the common file editor management:



Next, all the previous menu items will be further explained:

### 2.2.1. UNDO

Undo the changes in the file editor setting the focus on the file that is the owner of the change.

### 2.2.2. REDO

Redo the changes in the file editor setting the focus on the file that is the owner of the change.

### 2.2.3. COPY

Copy the selected text in the active file in the file editor and put it into the system clipboard.

### 2.2.4. PASTE

Paste the text stored in the system clipboard in the current position of the active file in the file editor.

### 2.2.5. CUT

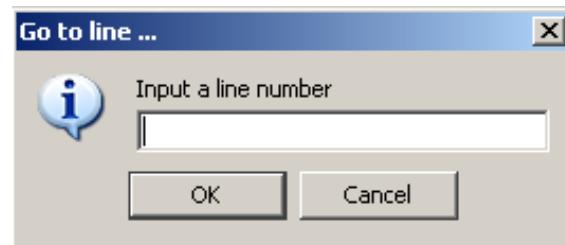
Cut the selected text in the active file in the file editor and put it into the system clipboard.

### 2.2.6. SELECT ALL

Selects all the content of the active file in the file editor.

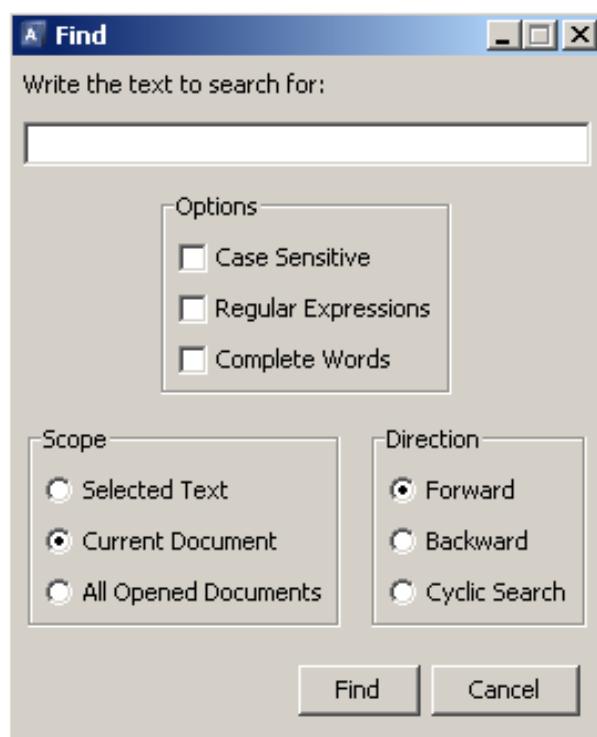
## 2.2.7. Go To Line

It displays a dialog in which the user will type down the number of the line where he wants to place the caret cursor in the active file in the file editor:



## 2.2.8. SEARCH

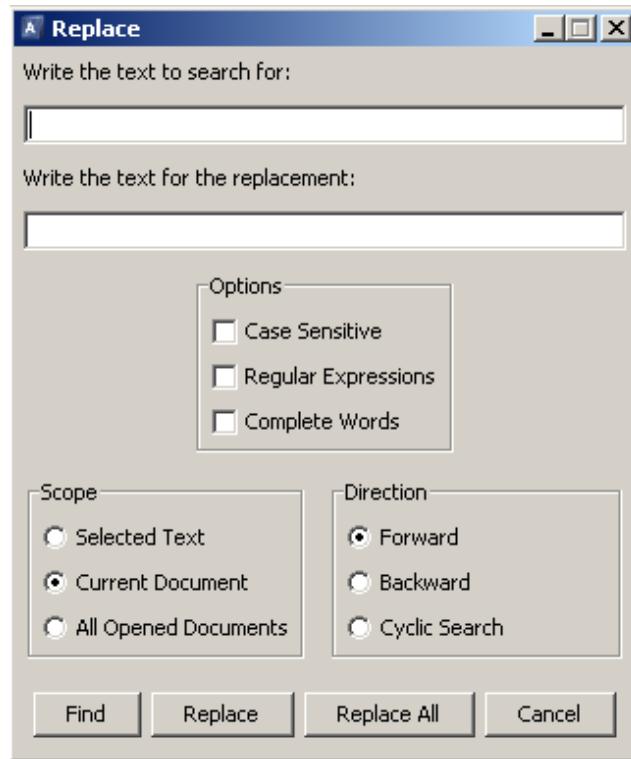
Shows the search text window of the file editor:



Then we proceed to describe each component of the window:

- **Text Box:** This is where you enter the search text.
- **Options:**
  - **Sensitive or Upper / Lower Case:** this option is used to search for strings without having or taking into account the Upper / Lowercase.
  - **Regular Expressions:** regular expression search associated with a search pattern.
  - **Whole words:** find whole words only.
- **Scope:**
  - **Selected text:** search within a selected text.
  - **Current document:** document-search in a certain position of the active file File Editor.
  - **All open documents:** searches all open files on the file editor.
- **Address:**
  - **Forward:** searches from the current cursor to the end of the file in the source file editor.
  - **Backward:** searches from the current cursor position to beginning of file in the file editor.
  - **Cyclic:** searches from the current cursor position to the end of the current file in the file editor, and start from the beginning to the starting position.

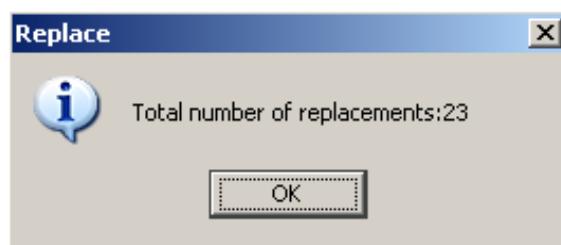
## 2.2.9. REPLACE



Displays the replace text window of the file editor:

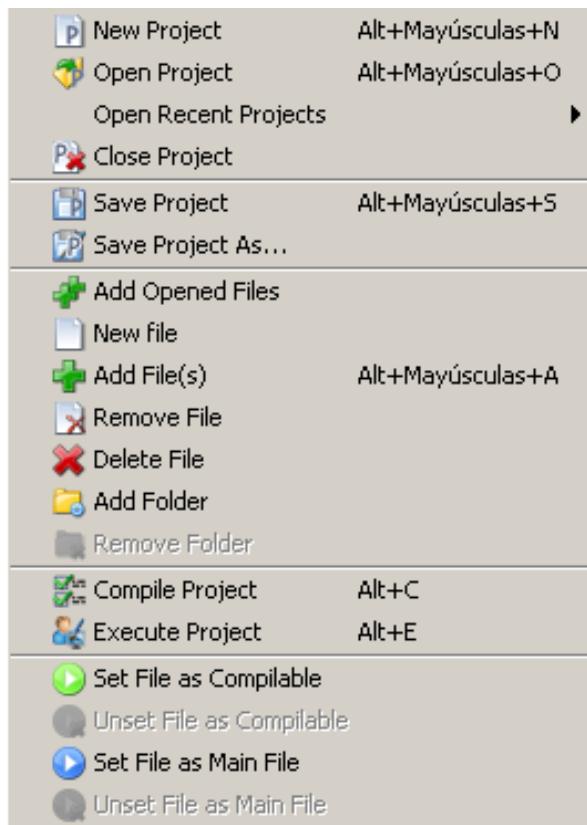
It offers the same options than the search window plus the **replace buttons** and the **text field** to select the text to use for the replacements.

When a general replacement is performed, it displays the following dialog to the user informing of the *number of replacements*:



## 2.3. PROJECT MENU

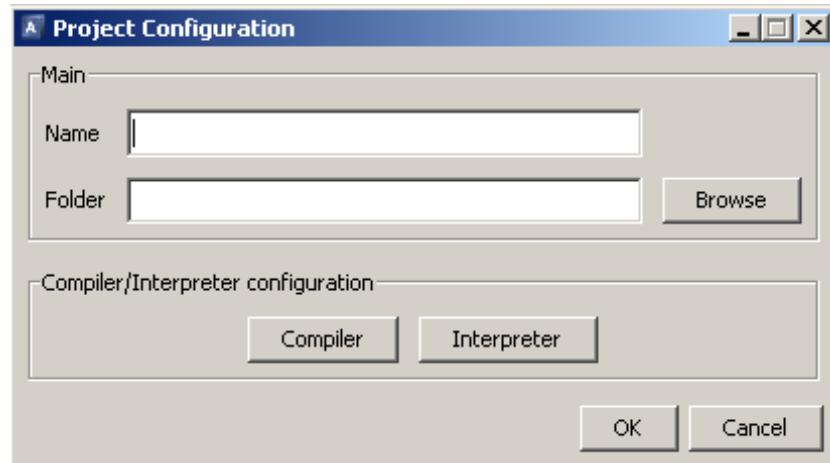
It contains the menu items required for the project configurations management:



Next, all the previous menu items will be further explained:

### 2.3.1. NEW PROJECT

Configures a new project displaying the following configuration window:



Next, the window options are further described:

- **Name:** indicates the project name.
- **Folder:** indicates the folder where the project file will be placed. If the project file already exists in the folder the application will give the chance to the user to overwrite it or not:



- **Compiler/Interpreter options**
  - **Compiler:** selects the compiler configuration for the new project.
  - **Interpreter:** selects the console panel configuration for the new project.

### 2.3.2. OPEN PROJECT

Open an existing project.

### 2.3.3. OPEN RECENT PROJECTS

Displays a list with the projects that have been already opened in the application plus the option to empty the list.

### 2.3.4. CLOSE PROJECT

Closes the current project and sets the default configuration.

### 2.3.5. SAVE PROJECT

Saves the current project configuration into its configuration file.

### 2.3.6. SAVE PROJECT AS

Saves the current project configuration into a different configuration file.

### 2.3.7. NEW PROJECT FILE

Creates a new empty file in the file editor and adds it to the current project configuration after asking to the user for its final destination.

### 2.3.8. ADD ALL OPENED FILES

Adds all the opened files in the file editor to the current project configuration, skipping all the files that already belong to the project.

### 2.3.9. ADD FILE

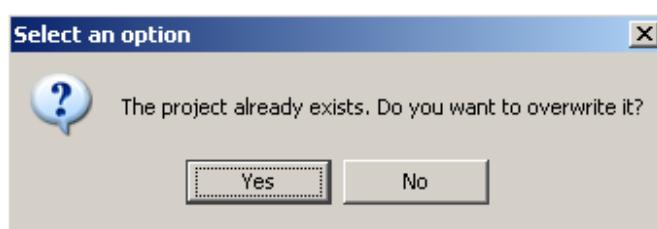
Adds the active file in the file editor to the project configuration.

### 2.3.10. REMOVE FILE

Removes the file from the project configuration but does not delete it from disk.

### 2.3.11. DELETE FILE

Removes the file from both project configuration and disk previous user's confirmation:



### 2.3.12. ADD FOLDER

Adds a new folder to the project in the selected level at the explorer tree, and checks that it does not exist another folder with the same name before adding it:



If already exists another folder with the same name at the same level at the explorer tree displays the following message and it does not add it:



### 2.3.13. REMOVE FOLDER

Removes the folder from the project configuration previous user's confirmation:



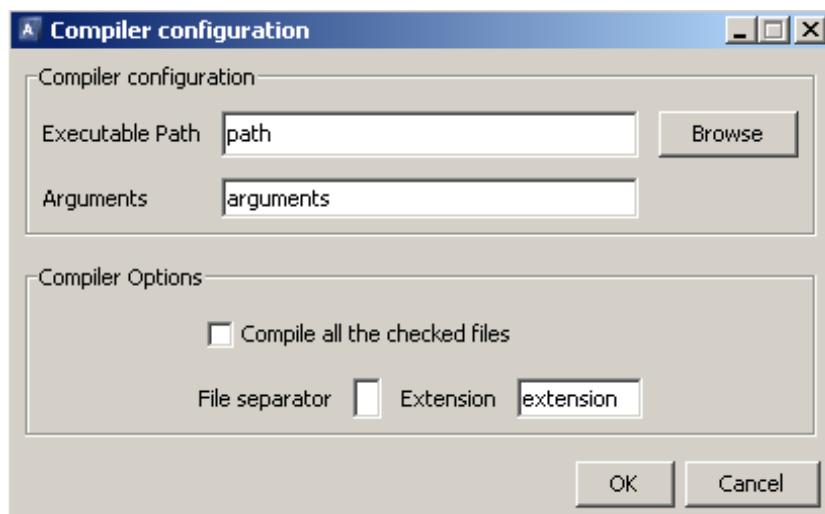
## 2.3.14. COMPILE PROJECT

The project is compiled with the selected parameters in the compiler configuration window that will be further detailed in the following chapters of the present document. Next, we illustrate its usage with two examples:

### 2.3.14.1. COMPILEMENT BASED ON “EXTENSION”

The process has the following steps:

1. First, in the compiler configuration window the user selects the extension of the files that he wants to compile:

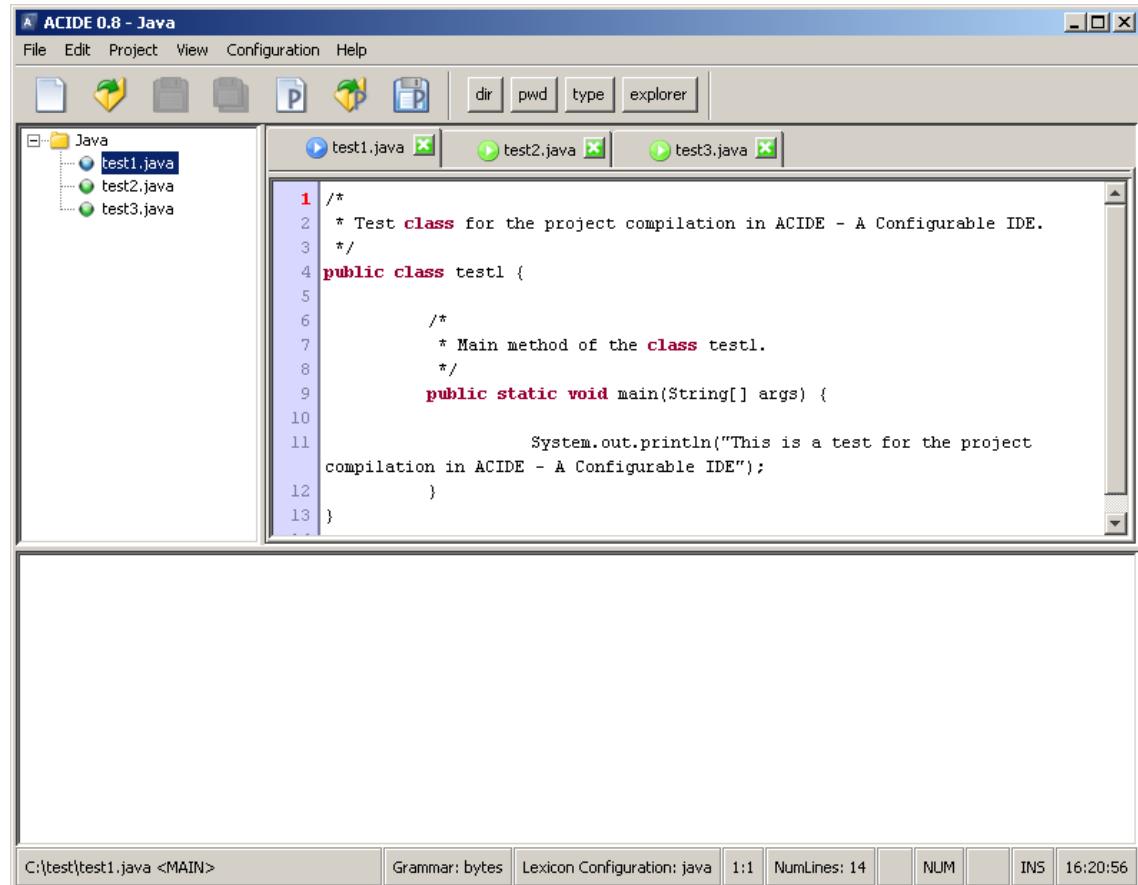


Finally, the project is compiled using the *Menu/Project/Compile* menu item option.

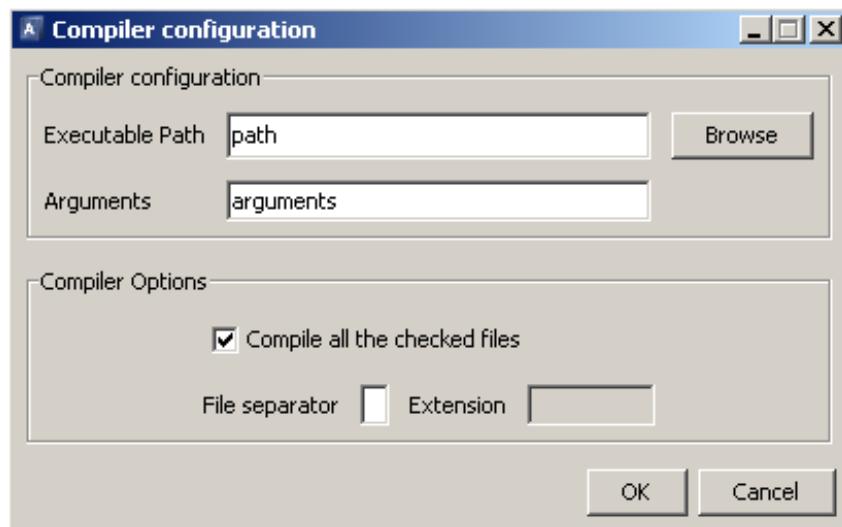
### 2.3.14.2. COMPILEMENT BASED ON “MARKED FILES FOR COMPILATION”

The process has the following steps:

1. First, the user marks all the files that he wants to compile in the file editor or in the explorer tree using the correspondent options for the purpose:



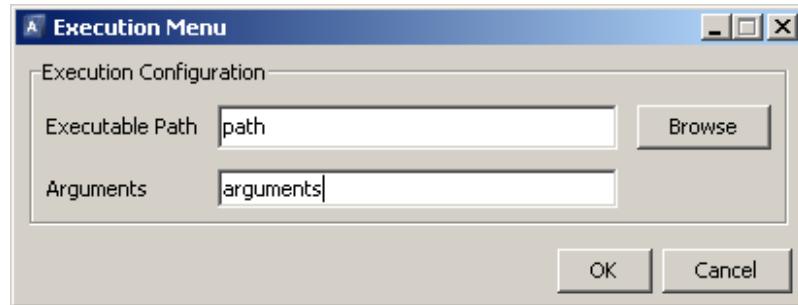
2. Next, the user configures the compiler options in the compiler configuration window as follows:



Finally, he selects the *Menu/Project/Compile* menu item option.

### 2.3.15. EXECUTE PROJECT

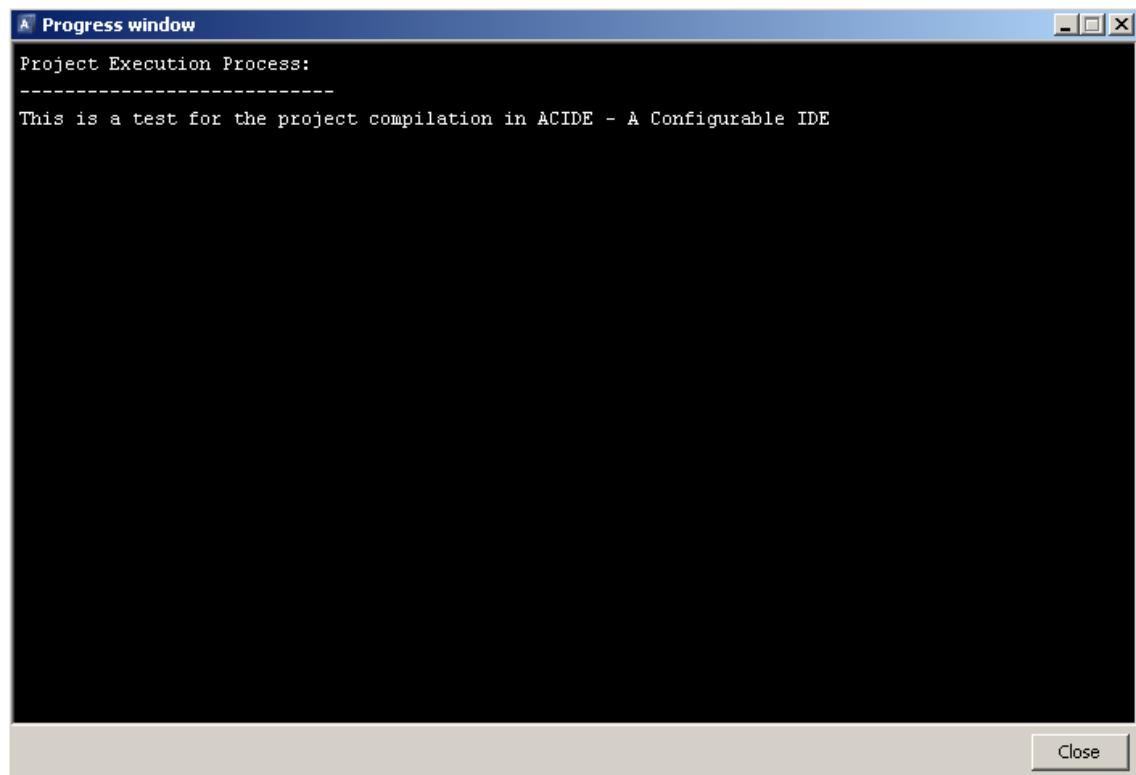
It displays the following configuration window:



Next, we further detail all the window components:

- **Executable path:** path of the selected executable.
- **Executable Arguments:** arguments for the selected executable.

The result of the execution is displayed in the following progress window:



### 2.3.16. SET COMPILABLE FILE

Set the active file in the file editor as compilable.

### 2.3.17. UNSET COMPILABLE FILE

Unset the active file in the file editor as compilable.

### 2.3.18. SET MAIN FILE

Set the active file in the file editor as main.

### 2.3.19. UNSET MAIN FILE

Unset the active file in the file editor as main.

## 2.4. VIEW MENU

If contains the menu items for the displaying management of the visible parts of the application and the log visualization:



Next, all the previous menu items will be further explained:

### 2.4.1. SHOW LOG

Shows the application log in the file editor.

### 2.4.2. PROJECT BROWSER

Hides or shows the explorer panel at the left side of the main window of the application.

### 2.4.3. CONSOLE WINDOW

Hides or shows the console panel at the bottom side of the main window of the application.

## 2.5. CONFIGURATION MENU

It contains all the menu item options for the configuration management of all the modules of the application:

### 2.5.1. LEXICON CONFIGURATION

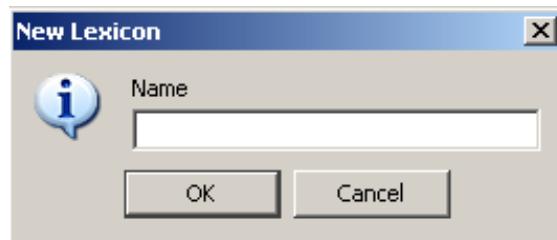
It contains all the menu item options for the lexicon configuration management of the application:



Next, all the previously mentioned options are further explained:

#### 2.5.1.1. NEW LEXICON

Creates a new lexicon configuration with the name that the user types down in the following window applying it to the active file in the file editor:



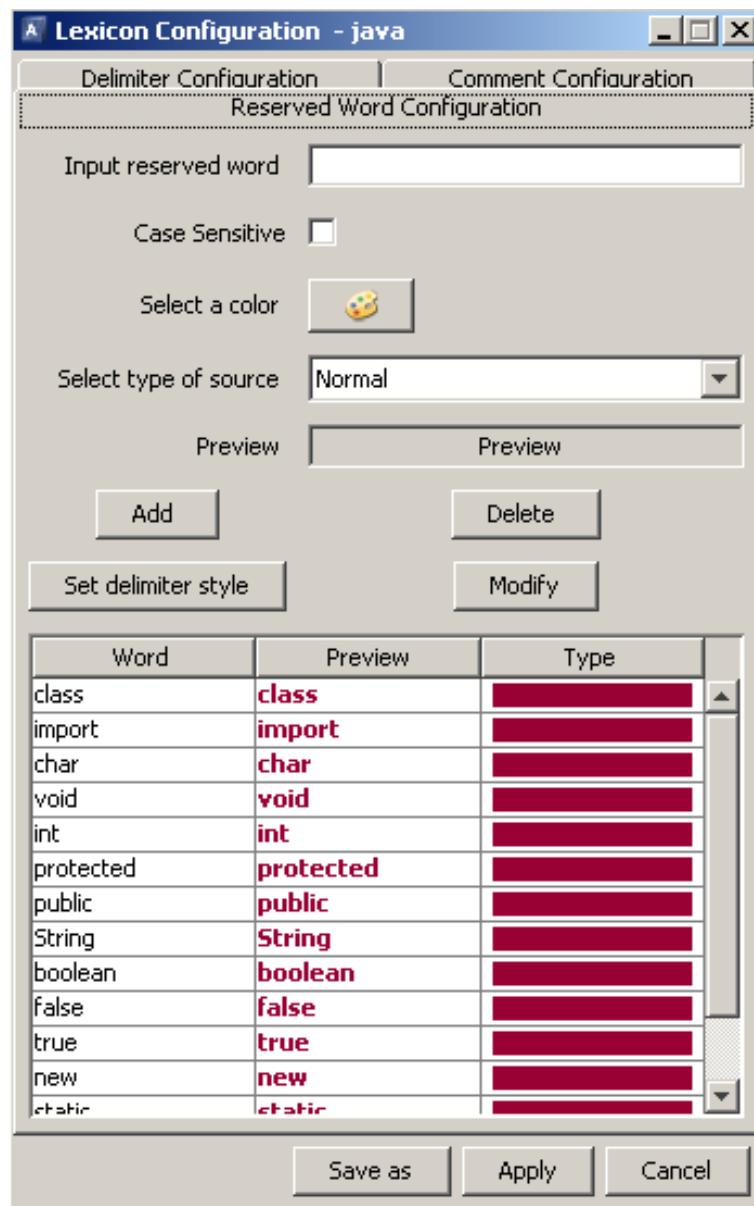
#### 2.5.1.2. DOCUMENT LEXICON

Loads the lexicon configuration file with **XML** extension in the active file in the file editor.

#### 2.5.1.3. MODIFY LEXICON

Open the lexicon configuration window that contains the following tabs:

### 2.5.1.3.1. RESERVED WORDS CONFIGURATION



Next, we further describe each one of its components as follows:

- **Add:** adds a new table reserved word entry.
- **Delete:** removes a table reserved word entry.
- **Modify:** modifies a table reserved word entry.
- **Set delimiter style:** the delimiter list now is also taken as reserved words.
- **Table:** contains the list with the reserved words groups by types and colors. *Note:* it is not allowed to modify the table entries directly on the

table itself and the changes will not be applied until the **modify button** is pressed down.

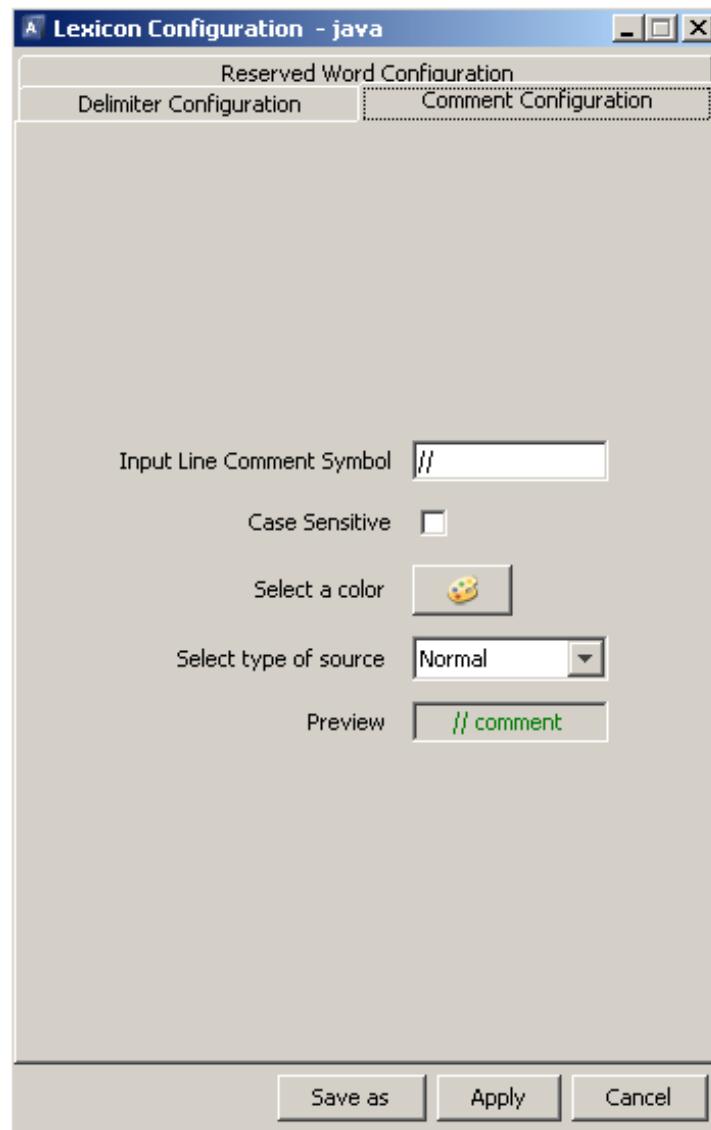
#### 2.5.1.3.2. DELIMITERS CONFIGURATION



Next, all its components are further detailed:

- **Input new delimiter text field:** the user inputs the name of the new delimiter.
- **Add button:** adds the input delimiter in the text field to the table.
- **Delete button:** removes selected delimiter from the table.
- **Table:** contains the delimiter list, and it is possible to modify it directly on it.

### 2.5.1.3.3. REMARKS CONFIGURATION



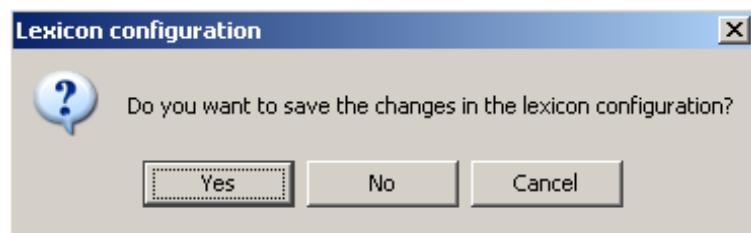
Next, we further detail all its components:

- **Comment symbol text field:** for input the remark symbol.
- **Case sensitive check box:** for specify if the remark is case sensitive or not.
- **Color selection button:** for the color selection of the remarks.
- **Font style combo box:** for the font style selection.
- **Preview text field:** shows a preview of the remarks.

The lexicon configuration window has in the bottom side the following buttons:

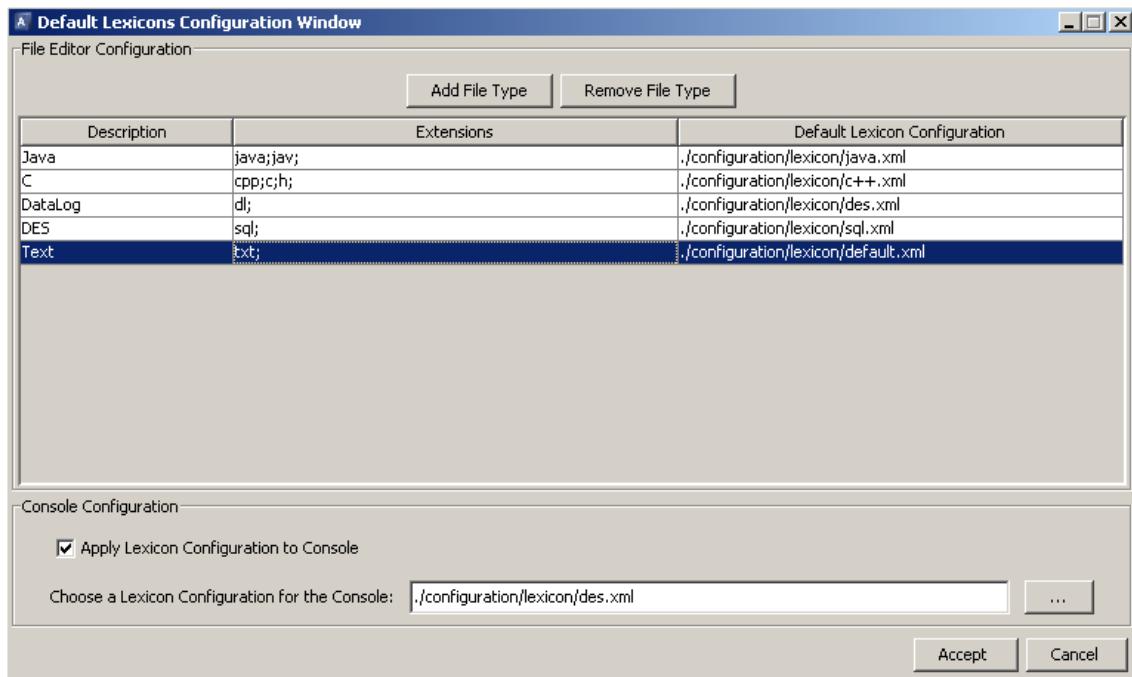
- **Save as:** saves the current lexicon configuration in other path with **XML** extension.
- **Apply:** applies the changes to all the opened files with the current lexicon configuration in the file editor and saves the changes in the configuration file with **XML** extension.
- **Cancel:** closes the lexicon configuration window without applying the changes.

Finally, if there are any changes in the current configuration in the previously described panels and the user closes the window with the close button or the ESC key, the following dialog will be displayed:



#### 2.5.1.4. DEFAULT LEXICONS

Shows the default lexicons configuration window:



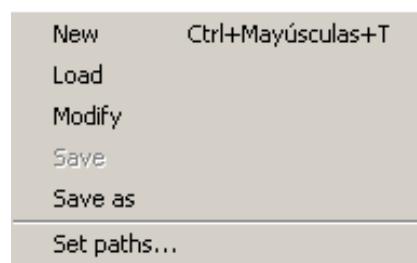
Next, we explain each one of its components:

- **File editor configuration:** contains the elements for the default lexicon configurations management in the file editor:
  - **Add file type:** adds a new default lexicon configuration to the table.
  - **Remove file type:** removes a default lexicon configuration from the table.
  - **Table:** contains the following columns:
    - **Description.**
    - **Extensions:** extension list separated by ";". *Note:* the format ".txt" is not a valid extension.
    - **Default Lexicon Configuration.**
- **Console configuration:** contains the elements for the default lexicon configurations management in the console panel:

- **Apply lexicon configuration to the console:** indicates if the default lexicon configuration has to be applied or not to the console panel.
- **Console Lexicon configuration.**

### 2.5.2. GRAMMAR CONFIGURATION

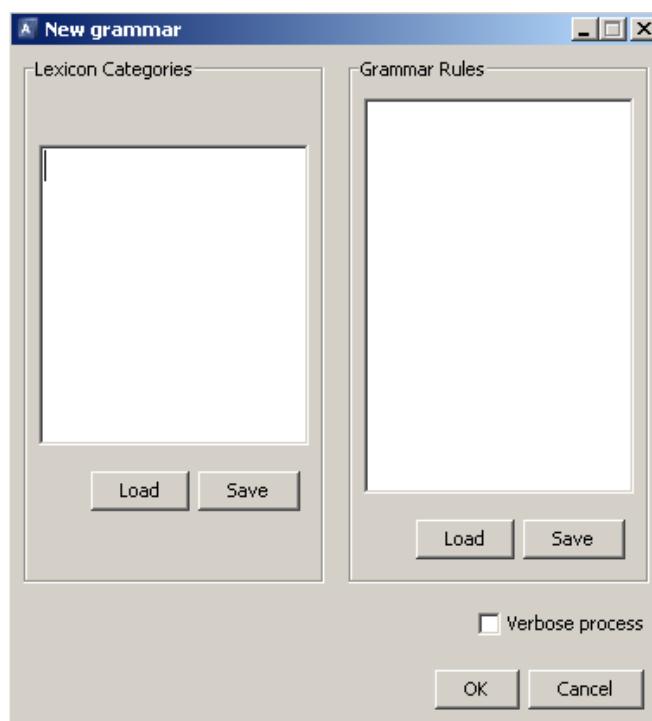
It contains the menu item options for the grammar configuration management:



Next, we further explain each one of the previous menu item options:

#### 2.5.2.1. NEW GRAMMAR

Creates new grammar configurations from lexicon categories and grammar rules with EBNF format in the following configuration window:



The window has the following components:

- **Lexicon categories panel:**

- **Lexicon categories text area:** shows the content of the lexicon categories plain text file with **TXT** extension.
- **Load button:** loads the content of the lexicon categories plain text file with **TXT** extension into the lexicon categories text area.
- **Save button:** saves the content of the lexicon categories text area into a plain text file with **TXT** extension.
- **Grammar rules panel:**
  - **Cuadro de texto de reglas de la gramática:** shows the content of the grammar rules plain text file with **TXT** extension.
  - **Load button:** loads the content of the grammar rules plain text file with **TXT** extension into the grammar rules text area.
  - **Save button:** saves the content of the grammar rules text area into a plain text file with **TXT** extension.
- **Verbose process check box:** indicates if the grammar generation process is verbose in a progress window or not.
- **Accept button:** initializes the grammar creation process.
- **Cancel button:** closes the window without applying the changes.

In the moment that the new grammar is created, it is not saved until the user selects the save menu option. In the case that the user closes the application without saving it, the last grammar configuration will be loaded.

If the user selects to verbose the grammar creation process, the following window will be displayed:

A Progress window

```
Grammar file generation process:
-----
Executing ANTLR...
"C:\Archivos de programa\Java\jdk1.6.0_21\bin\java.exe" -cp ./lib/antlr.jar antlr.Tool grammar.g
ANTLR execution task completed successfully!
Executing generated files by ANTLR modification...
Generated files by ANTLR modification successfully!
Compiling generated files by ANTLR...
"C:\Archivos de programa\Java\jdk1.6.0_21\bin\javac.exe" -cp .;c:\classes *.java -d .
Compilation of generated files by ANTLR task completed successfully!
Reallocating generated files by ANTLR...
Reallocation of generated files by ANTLR task completed successfully!
Generating the .jar file...
Generation of .jar file task completed successfully!
Deleting generated files by ANTLR...
Deletion of generated files by ANTLR task completed successfully!
Reallocating the .jar file into the configuration folder...
Reallocation of the .jar file into the configuration folder task completed successfully!
```

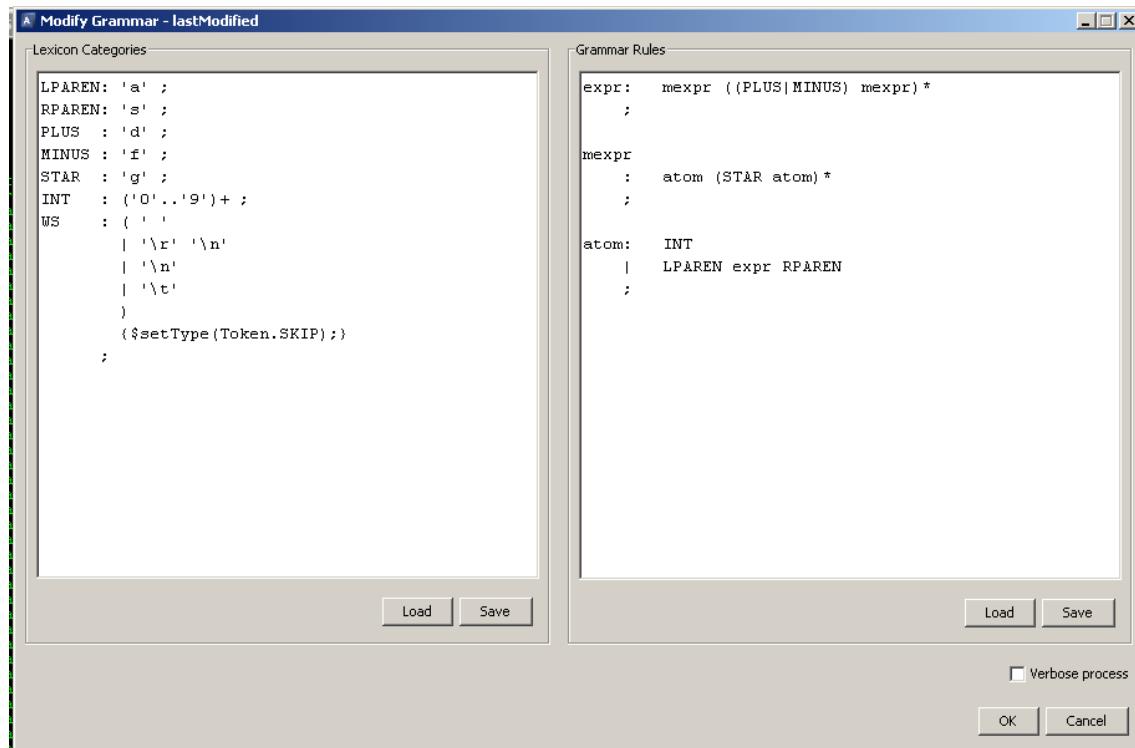
Close

### 2.5.2.2. LOAD GRAMMAR

Loads a grammar configuration with **JAR** extension.

### 2.5.2.3. MODIFY GRAMMAR

Displays the same grammar configuration window than the **New Grammar** menu item option but it contains the lexicon categories and grammar rules text areas filled with their file contents:



#### 2.5.2.4. SAVE GRAMMAR

Saves the current grammar configuration into a file with **JAR** extension.

#### 2.5.2.5. SAVE GRAMMAR AS

Saves the current grammar configuration into a file with **JAR** extension with a different path.

#### 2.5.2.6. CONFIGURE PATHS

For the creation, modification and grammar configurations handling it is mandatory to define the required tools paths as it was mentioned in the first chapter of the present document.

It displays the following window:

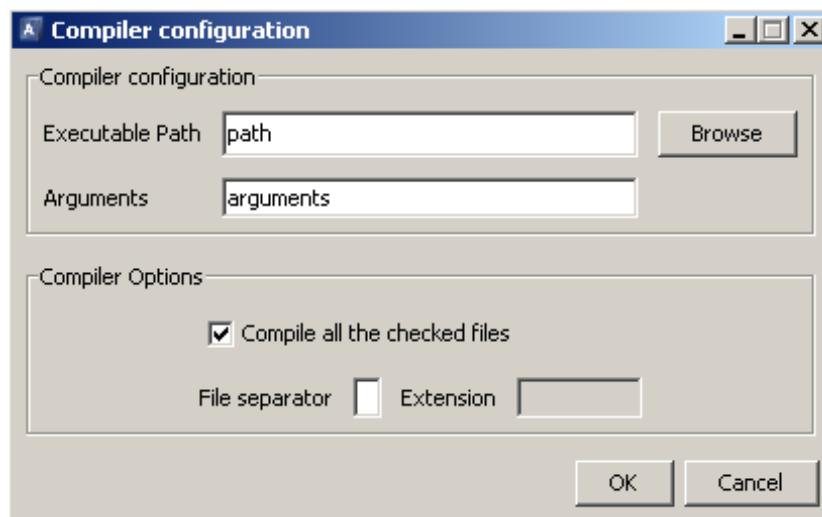


In each one of the text fields the user will select the path of each one of the required tools. The window also contains the following components:

- **Check box:** if it is selected the application will use the path selected in the text field that corresponds; in the case the it is disabled the application will use its Operative System CLASSPATH.
- **Explorer buttons:** open a dialog window for the files selection.

### 2.5.3. COMPILER

The following window will be displayed:

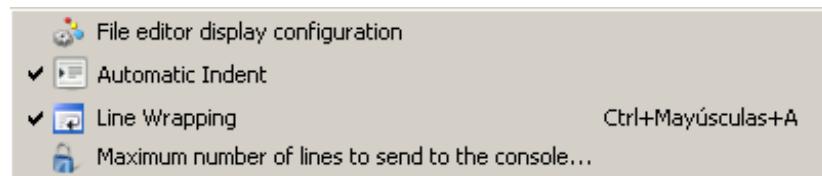


The window has the following components:

- **Compiler configuration panel:**
  - **Executable path:** path that contains the compiler executable file.
  - **Compiler arguments:** arguments for the compiler.
- **Compiler options panel:**
  - **Compile all the checked files:** indicates if all the compilable files have to be compiled or not.
  - **File separator:** file separator to separate each one of the files to compile.
  - **Extension:** file extension of the files to compile.
- **Accept button:** apply the changes.
- **Botón cancelar:** close the window and do not apply the changes.

## 2.5.4. FILE EDITOR CONFIGURATION

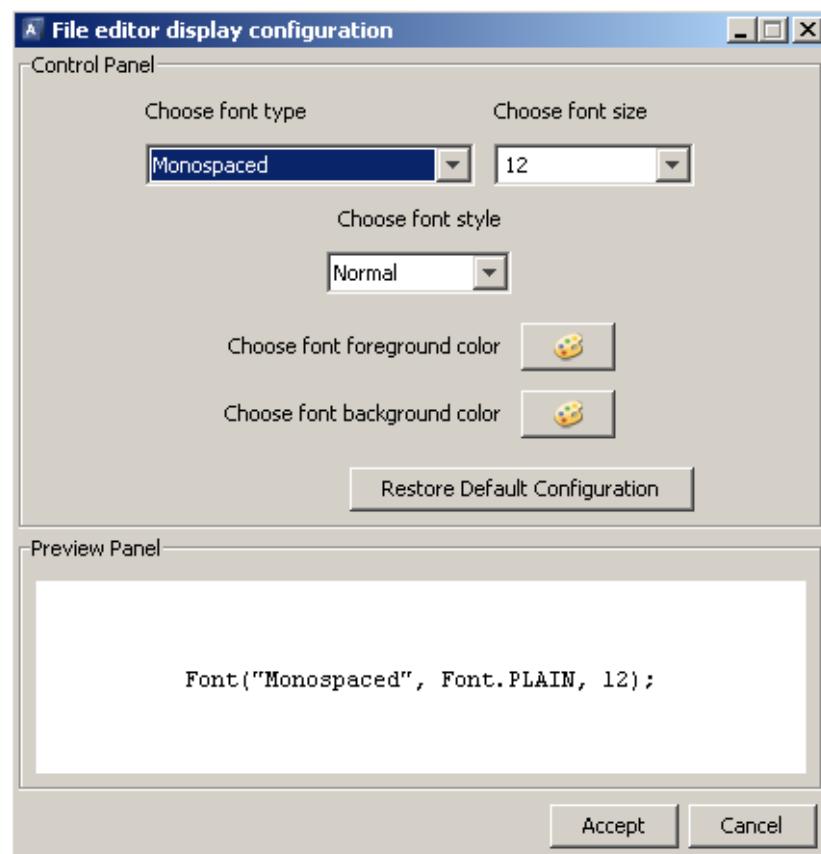
It contains the menu item options for the file editor configuration management:



Next, we further detail each one of the previous menu item options:

### 2.5.4.1. FILE EDITOR DISPLAY OPTIONS CONFIGURATION

Displays the following configuration window:



In the configuration window, the user can configure the following parameters:

- **Font type.**
- **Font size.**
- **Font style.**
- **Foreground color.**
- **Background color.**

- **Restore default values:** applies the default configuration: "Monospaced" font , plain, size of 12, black with white background.

#### 2.5.4.2. AUTOMATIC INDENT

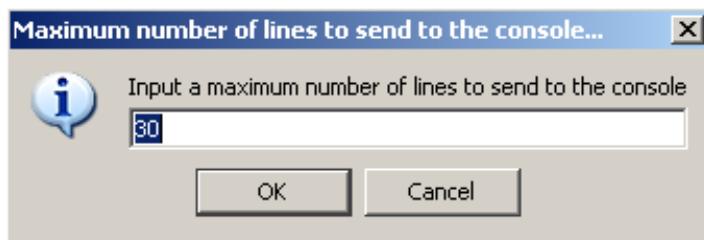
Enables or disables the automatic indent in the file editor.

#### 2.5.4.3. LINE WRAPPING

Enables or disables the line wrapping in the file editor.

#### 2.5.4.4. MAXIMUM LINE NUMBER TO SEND TO CONSOLE

Asks to the user for the maximum number of lines to send to the console panel from the active file in the file editor:



#### 2.5.4.5. SEND FILE CONTENTS TO CONSOLE

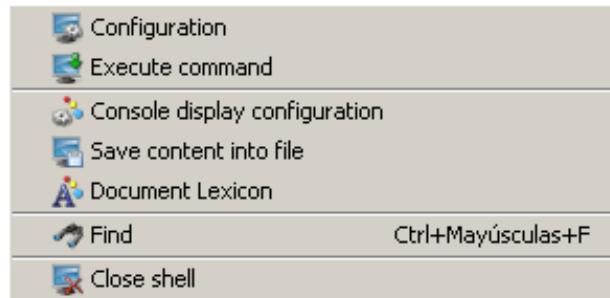
The option is available from the *popup menu* in the file editor. If the number of lines of the current file is bigger than the maximum number of lines defined by the previous option it will display the following confirmation message:



If the current number of lines is not bigger than the specified maximum number of lines, it simply sends the active file content to the console panel adding each sent line as a separate command in the console panel command record.

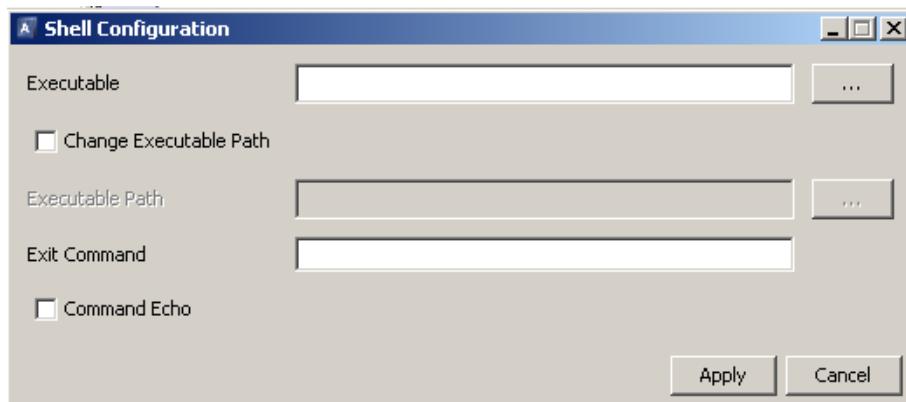
## 2.5.5. CONSOLE CONFIGURATION

It contains the menu item options for the console panel configuration management:



### 2.5.5.1. CONFIGURE

Configures the shell configurations that are loaded in the console panel:

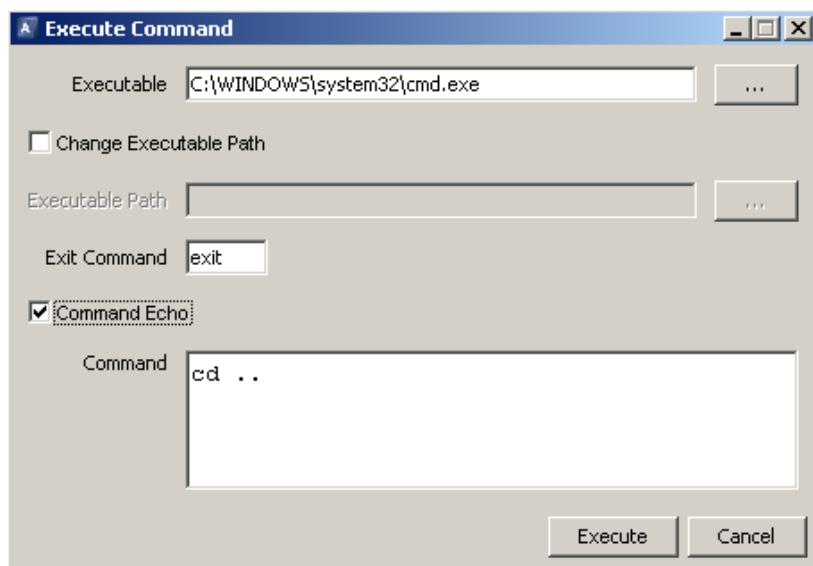


It contains the following components:

- **Executable:** executable file path.
- **Change executable path:** it is used for specifying a different folder where the executable file is placed.
- **Executable path:** executable file folder.
- **Exit command:** exit command for closing the data stream.
- **Echo del comando:** indicates if the commands typed in the console panel have to be displayed or not.

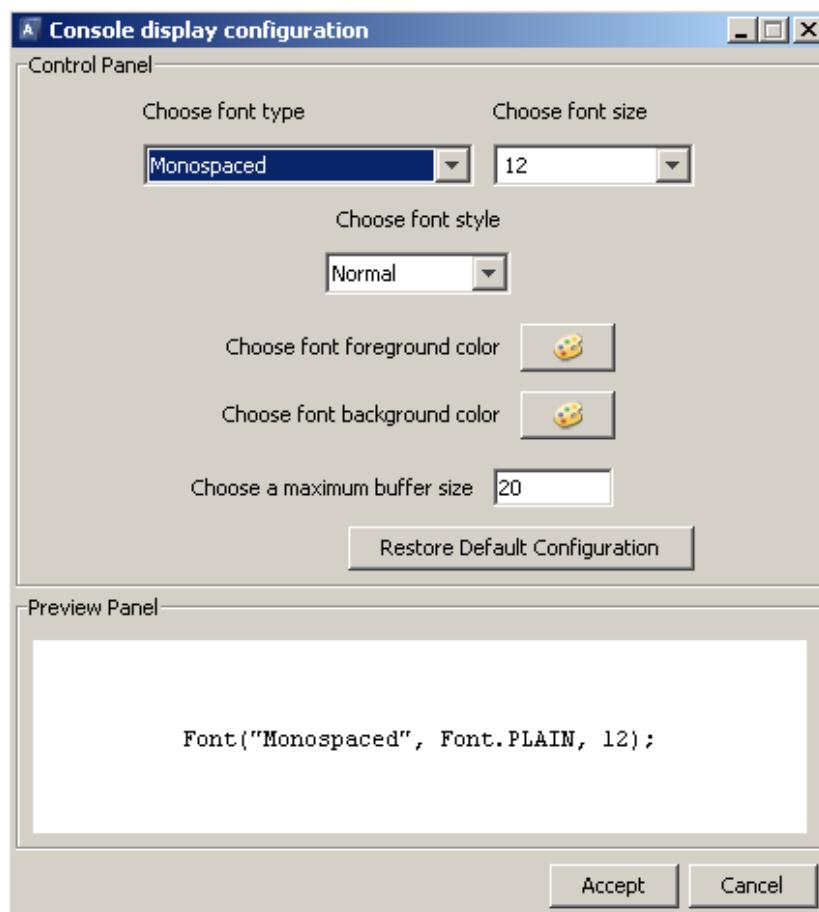
### 2.5.5.2. EXECUTE EXTERNAL COMMAND

Executes a command into a shell and displays the result in a separate window that looks like:



### 2.5.5.3. CONSOLE DISPLAY CONFIGURATION

Displays the following configuration window:



The user can select:

- **Font type.**

- **Font size.**
- **Font style.**
- **Font color.**
- **Background color.**
- **Maximum buffer size:** specifies the maximum number of lines that are displayed in the console panel.
- **Restore default configuration:** applies the default configuration for the console panel: "Monospaced" font, plain, size of 12, black with white background.

#### 2.5.5.4. **SAVE CONTENT INTO FILE**

Saves the console content into a file.

#### 2.5.5.5. **DOCUMENT LEXICON**

Loads a lexicon configuration with **XML** extension into the console panel.

#### 2.5.5.6. **BUSCAR**

Displays the search text window for the console panel:



#### 2.5.5.7. **CLOSE CONSOLE**

Closes the active shell in the console panel.

#### 2.5.5.8. **RESET CONSOLE**

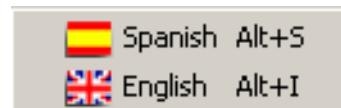
Only available in the *popup menu* of the console panel. Resets the active shell in the console panel.

### 2.5.5.9. CLEAR CONSOLE BUFFER

Only available in the *popup menu* of the console panel. Clears the console panel content and leaves only the last line of the previous buffer content.

## 2.5.6. LANGUAGE CONFIGURATION

Shows the available language list of the application:



In this case, the user can choose only between *English* or *Spanish*.

## 2.5.7. MENU CONFIGURATION

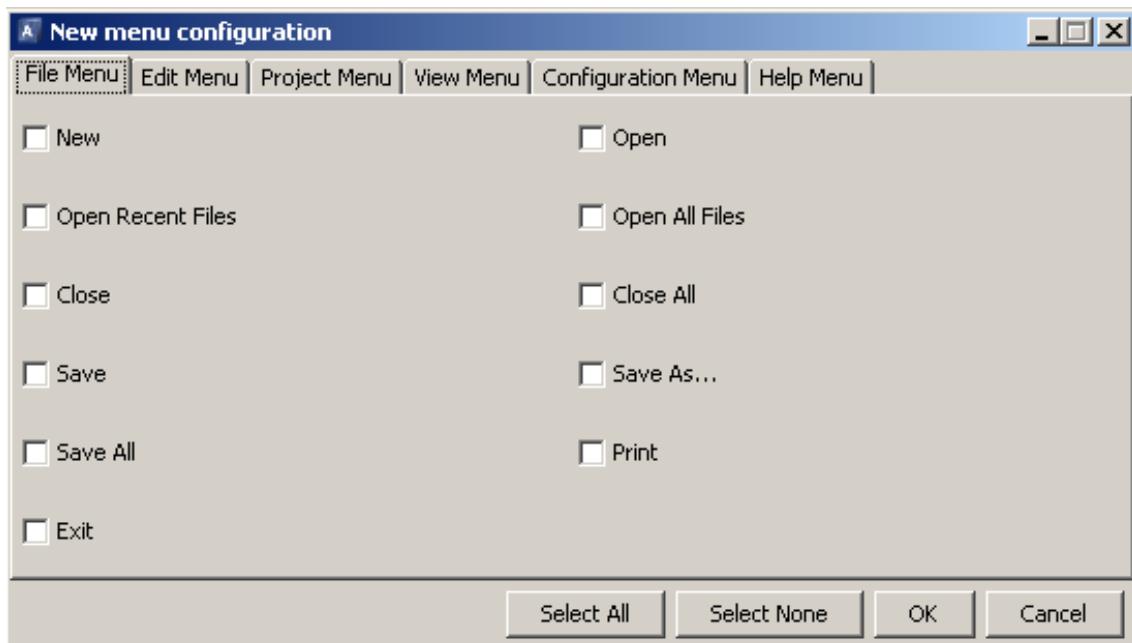
It contains the menu item options for the menu configuration management:



Next, we further describe each one of the previous menu item options:

### 2.5.7.1. NEW

Displays the following configuration window:



The user can select all the menu item options that he wants to see at the menu bar, selecting each option one by one or using the **select all** and **select none** buttons.

The new configuration will not be saved until the user selects the **save menu** item option. In case that he closes the application without saving, the previous configuration will be restored and the current one will be lost.

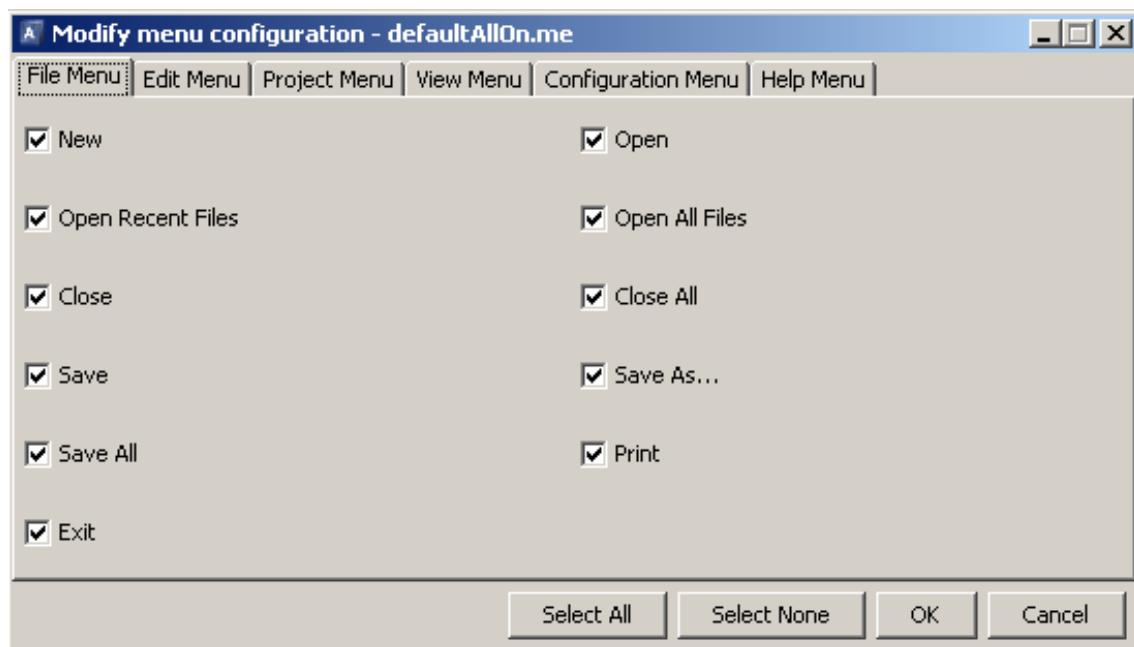
The menu configuration files has **menuConfig** extensions.

### 2.5.7.2. LOAD

Loads a menu configuration with **menuConfig** extension.

### 2.5.7.3. MODIFY

Al seleccionar ésta opción se muestra la siguiente ventana de configuración, similar a la de nueva creación de una configuración pero con las opciones correspondientes marcadas y con el nombre de la configuración en el título de la ventana:



The modified configuration will not be saved until the user selects the **save menu** item option. In case that he closes the application without saving, the previous configuration will be restored and the current one will be lost.

### 2.5.7.4. SAVE

Saves the current menu configuration into a menu configuration file with **menuConfig** extension.

### 2.5.7.5. SAVE AS

Saves the current menu configuration into a menu configuration file with **menuConfig** extension with a different path.

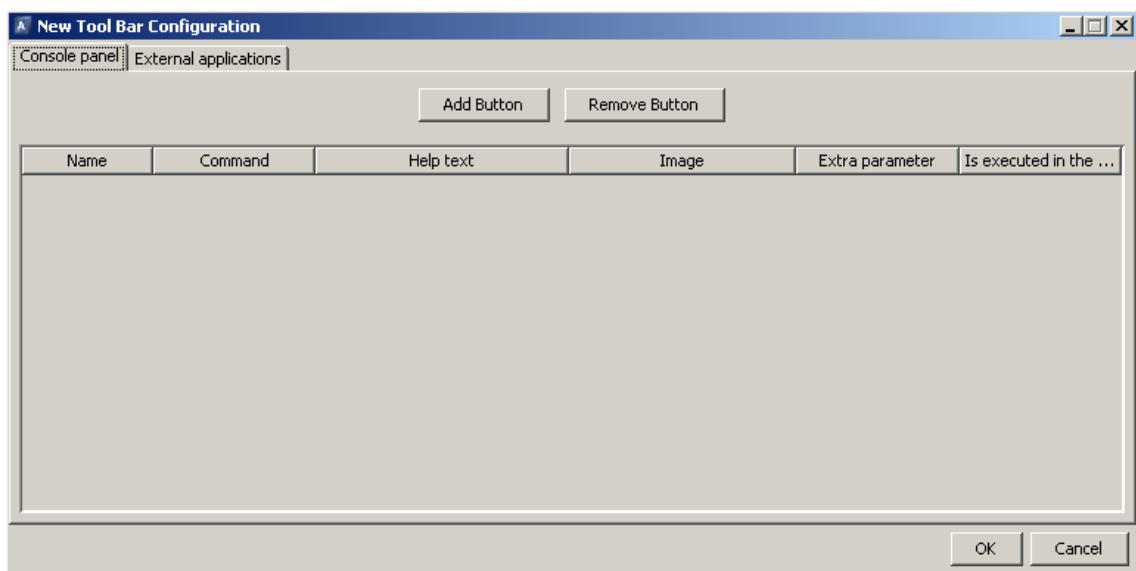
## 2.5.8. TOOL BAR CONFIGURATION

It contains the menu item options for the tool bar configuration management:



### 2.5.8.1. NEW

It displays the following configuration window:



The window has two different panels:

- **Console Panel:** defines the commands related to the console panel tool bar that are executed in the console panel.
- **External Applications Panel:** defines the commands related to the external applications tool bar that are executed externally the application.

In each one of the panels, the user can do the following operations:

- **Add button:** adds a new command to the command list in the table.
- **Remove button:** removes the selected command from the command list.

- **Direct edition on the tables:** the user can modify the commands by editing directly on the table. However, the changes will not be applied until the focus changes or the user presses down the **ENTER** key.

In the *console panel* tab the table contains the following parameters:

- **Name:** text to display in the button. If this field is empty the application will assign it a number as name by default.
- **Command:** command itself.
- **Help text:** hint text of the button.
- **Image:** image for the button which can be selected by the option available in the *popup menu* of the column.
- **Extra parameter:** shows a combo box with the following options: *None*, *Text*, *File*, *Directory*. Each one of the previous options will ask the user for the selected type with different dialog windows.
- **Is executed in the OS shell:** indicates if the command is executed in the Operative System shell or in the loaded shell in the console panel.

In the *external applications panel* tab the table contains the following parameters:

- **Name:** text to display in the button. If this field is empty the application will assign it a number as name by default.
- **Executable path:** executable path of the command to execute.
- **Help text:** hint text of the button.
- **Image:** image for the button which can be selected by the option available in the *popup menu* of the column.

The new configuration will not be saved until the user selects the **save tool bar** item option. In case that he closes the application without saving, the previous configuration will be restored and the current one will be lost.

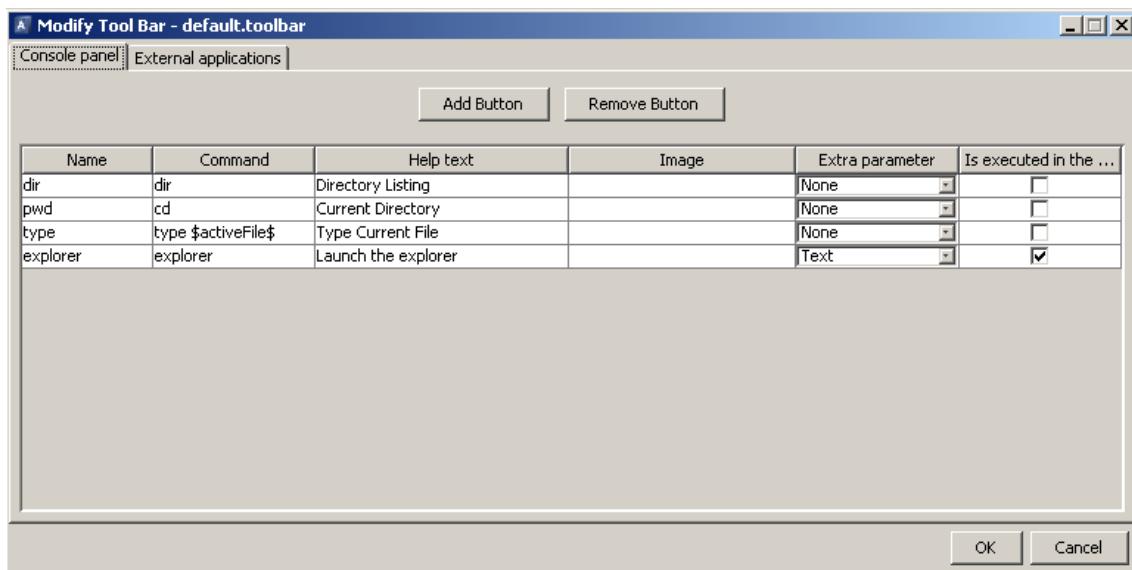
The files tool bar files have **toolbarConfig** extension.

### 2.5.8.2. LOAD

Loads a tool bar configuration with **toolbarConfig** extension.

### 2.5.8.3. MODIFY

It displays the following configuration window:



It contains the same options than the configuration window displayed by the *menu/configuration/menu/new*.

In this case, the window displays the current tool bar configuration loaded in the tables and also with a different window title which contains the name of the current configuration to modify.

The modified configuration will not be saved until the user selects the **save tool bar** item option. In case that he closes the application without saving, the previous configuration will be restored and the current one will be lost.

### 2.5.8.4. SAVE

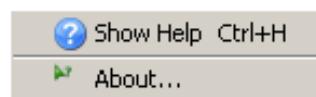
Saves the current tool bar configuration into a tool bar configuration file with **toolbarConfig** extension.

### 2.5.8.5. SAVE AS

Saves the current tool bar configuration into a tool bar configuration file with **toolbarConfig** extension and with a different path.

## 2.6. HELP MENU

Contains the following menu items:



Next, the previous menu options are further explained:

### 2.6.1. SHOW HELP

Links directly to the present document.

### 2.6.2. ABOUT US

Displays the following window with some extra information about the application:



## 2.7. ACCESIBILITY SHORCUTS

The application offers some accessibility shorcuts to wrapper common user actions such as:

- **F3+Selected text:** performs the *forward* text search in the file editor.
- **F3+Shift+Selected text:** performs the *backward* text search in the file editor.
- **Control+Up arrow/Down arrow:** performs the vertical scroll line by line in the file editor and console panel.
- **Mouse wheel:** performs the vertical scroll line by line in the file editor and console panel.
- **Control+Mouse wheel:** performs the zoom effect for the font size in the file editor and console panel.