BluAge COBOL

The extension to **check COBOL syntax**, **compile your COBOL project** and **support BMS langage** powered by **[BluAge](https://www.bluage.com/" \o "https://www.bluage.com)**.

For **COBOL**, a **parser** is included and it is possible to do a **Compilation Backend and Frontend** exchanging data with a distant compiler.

For **BMS** (Basic Mapping Support), **autocompletion**, **parsing** and **highlighting** are available. Moreover, a **previewer** allows you to see your BMS map and to modify it on the fly.

How to use the extension

COBOL Compilation

* Open a project directory, this directory has to contain one or more COBOL programs and copybooks.
* In **Settings > Extensions > BluAge COBOL** , set the compilation server url.
* Select a cobol file (.cbl or .cob) in the editor.
* Open the Command line with **Ctrl + Shift + P**, type **BluAge COBOL Compilation** for a backend compilation and **BluAge COBOL Compilation (Frontend)** for a frontend compilation.
* If the compilation succeeds, a jar or a war is created in a bin directory in your project directory. Otherwise, compilation errors are shownin the problem view and on files concerned.

BMS Previewer

* Select a bms file with extension .bms in the editor.
* Click on the preview button on the right corner of the file editor or open the command line with **Ctrl + Shift + P**, type **BluAge BMS previewer** to preview the bms file. In the previewer you can iterate through the maps if the file is defining a mapset with multiple ones. The previewer is showing also a setting button to edit the theme style of your choice.

Settings

The settings allow you to :

* Parameter the compilation server url;
* Define the result directory name;
* Set up a timeout for the compiler call;
* Decide if the console is more or less verbose.

BluAgeConsole

The BluAge Console can be selected in the Ouput view list.

* Classic debug mode :

console.png

* Verbose debug mode :

verbose\_console.png