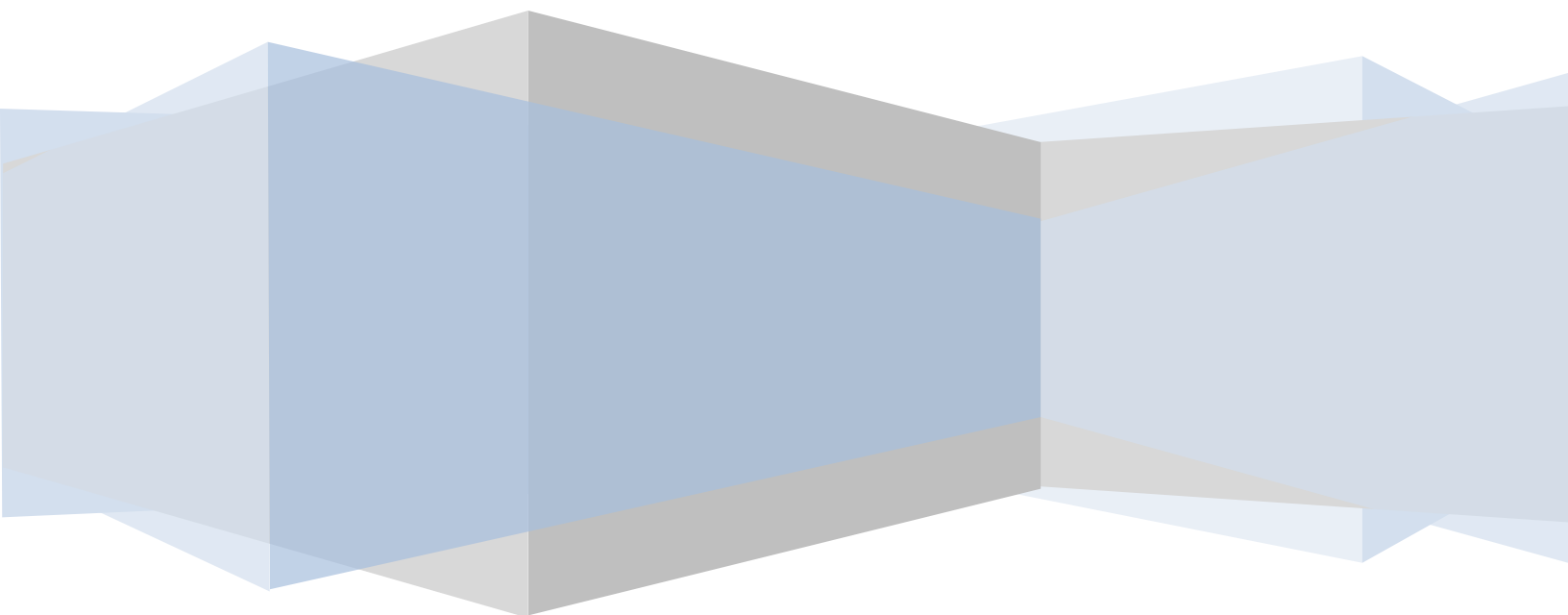


Perl On NetBeans

User Guide for version 0.6 (Beta)

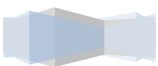


Contents

Scope.....	3
Features of Beta 6 release	3
Pre-requisites.....	3
Installation	4
Features	4
1. The Editor.....	4
1. Perl file support.....	4
2. Perl module support	5
3. Syntax highlighting.....	5
4. Brace matching	5
5. Export code as HTML	6
6. Toggle comments.....	7
7. Theme support.....	7
2. Perl Project.....	7
1. Create new project	7
2. Create project with existing sources.....	8
3. Open Project	9
4. Add new file to project	9
5. Add existing file to project.....	10
6. Project Copy/Move/Delete operations.....	10
7. Support for Perl-Dancer2 project	11
3. Check Files for Syntax Errors.....	12
4. Code execution	13
1. Execute files	13
2. Code execution with command line arguments	13
5. Source Code formatting.....	15
6. Source code analysis	16
7. Integrated PerlDOC support	17
8. Options Panel.....	18
1. Templates.....	18
2. General.....	18

3. Perl::Critic.....	19
4. Perl::Tidy	19
9. Integrated versioning system.....	20
Feedback	22
Web Links.....	22

Please note: Projects created with earlier versions (0.3 and below) of this software will not be recognized as valid projects. To import all projects previously created with this software, please create a “Project With Existing Sources” and import the project.



Scope

“Perl on NetBeans” is intended for Perl programmers who want the intuitiveness of a great editor with the ease of being able to execute the program without having to do it through a command Interpreter. Also, it has the capability to inspect the austerity of your Perl programs, and format the same in accordance to the best practices and conventions.

Features of Beta 6 release

1. Perl Editor
 1. Perl file support
 2. Perl module support
 3. Syntax highlighting
 4. Brace matching
 5. Export code as HTML
 6. Toggle comments
 7. Theme support
 8. Keyword assistance
2. Perl Project
 1. Create project with new sources
 2. Create project with existing sources
 3. Open existing project
 4. Add new file to project
 5. Add existing file to project
 6. Project Copy/Move/Delete operations
 7. Support for Perl-Dancer2 projects (Incubating)
3. Code Execution
 1. Code execution
 2. Code execution with command line arguments
4. Syntax checking
5. Source code formatting (using Perl::Tidy)
6. Source code analysis (using Perl::Critic)
7. Integrated Perl-Doc
8. Configuration using the Options Panel
 1. Templates
 2. General Options
 3. Perl Tidy
 4. Perl Critic
9. Integrated versioning system (SVN/Git/Hg)

Pre-requisites

This IDE requires the following to be installed on your system for the IDE to work properly:

3

1. NetBeans 13 or greater
2. Strawberry Perl 5.32.1.1 (Dancer2, Perl::Critic and Perl-Doc installed with your existing Perl installation)
3. Java 11 or greater

Installation

The installation of “Perl On NetBeans” can be summarized in the following steps:

1. Install NetBeans
2. Download and install the plugin from Release section of the GitHub repository

Features

The IDE currently just has the following features.

1. The Editor

1. Perl file support

Any Perl file can be opened in the editor and all available features would be available.

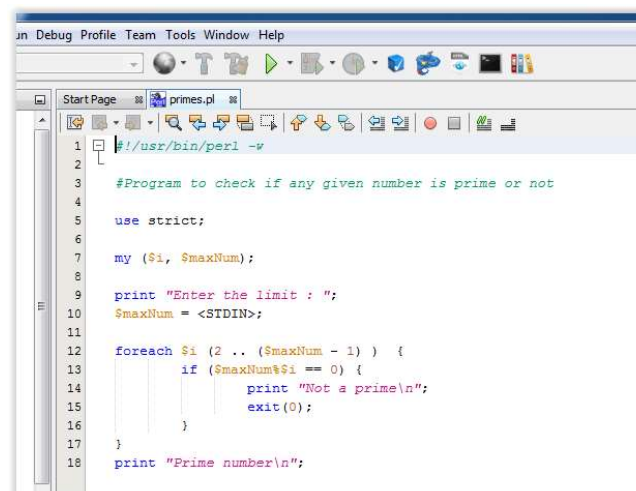


Figure 1 - A Perl (*.pl) file in the editor

2. Perl module support

Any Perl module file can be opened in the editor and all available features would be available.

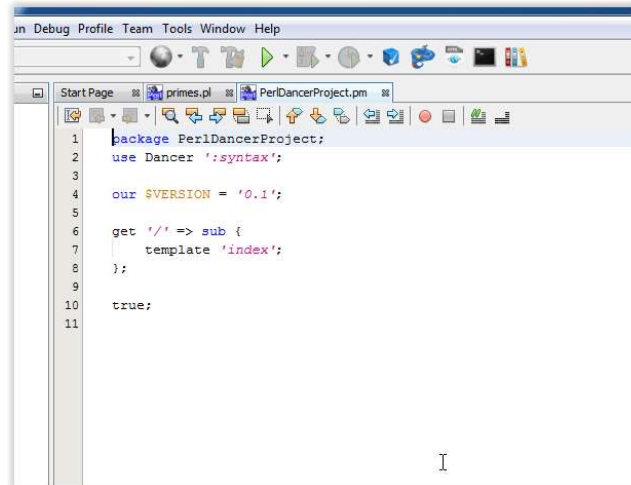


Figure 2 - A Perl module (*.pm) file in the editor

3. Syntax highlighting

As shown above, Perl files and modules are displayed with syntax highlighting in the editor. You also get the option of customizing the highlighted colors to your preference. Go to Tools -> Options -> Fonts and Colors. From the dropdown, select Perl and then customize as much as you want.

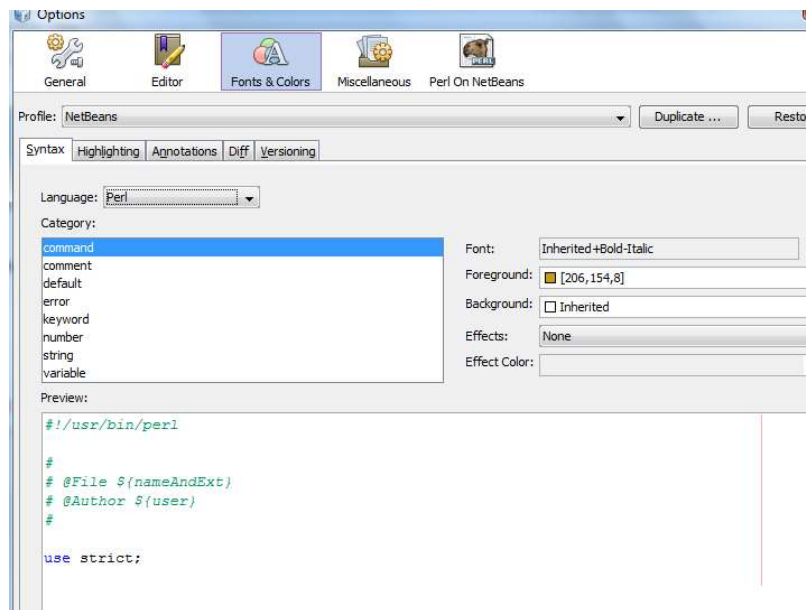


Figure 3 - Syntax highlighting options from the Options panel

4. Brace matching

This feature lets you understand the brace completion. If you create a brace mismatch, it would be highlighted in “RED” in the editor. Else, clicking on a brace will highlight the corresponding “opening/closing” brace.

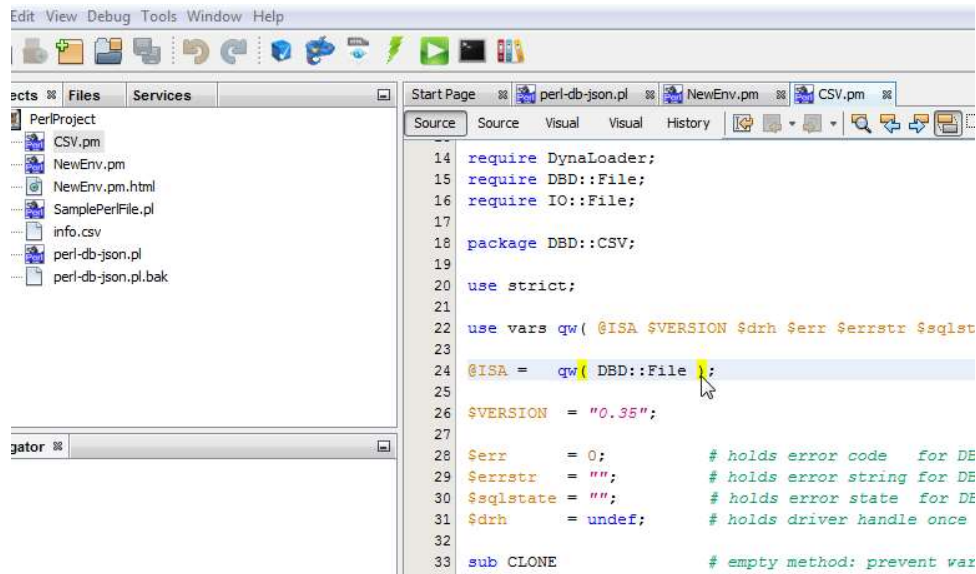


Figure 4 - Brace matching in action

5. Export code as HTML

You can export Perl code to HTML format, using the bundled Perl::Tidy module. To do that, just open any file in the editor and click on the generate HTML button. The output HTML can be configured from the Options panel.

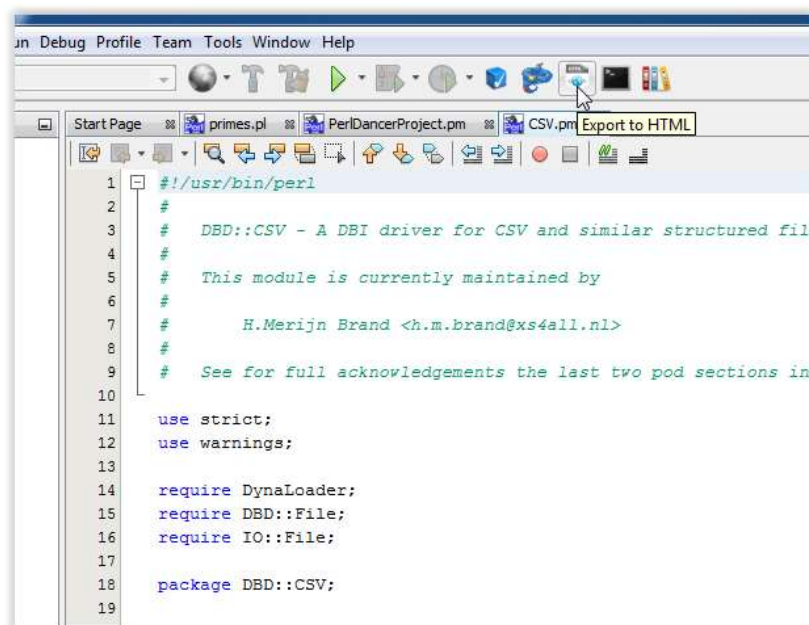


Figure 5 - The toolbar option to export code as HTML

6. Toggle comments

Select any block of the code and press Ctrl + / to comment or un-comment the block or the following toolbar option to comment/un-comment selected blocks of code.

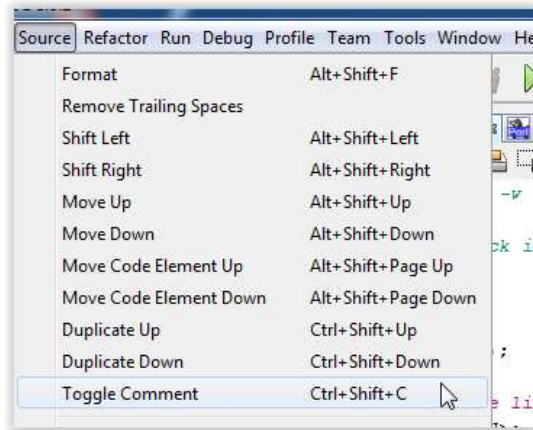


Figure 6 - Menu option to comment selected block

7. Theme support

Perl files will be displayed in any theme that you select for your IDE.

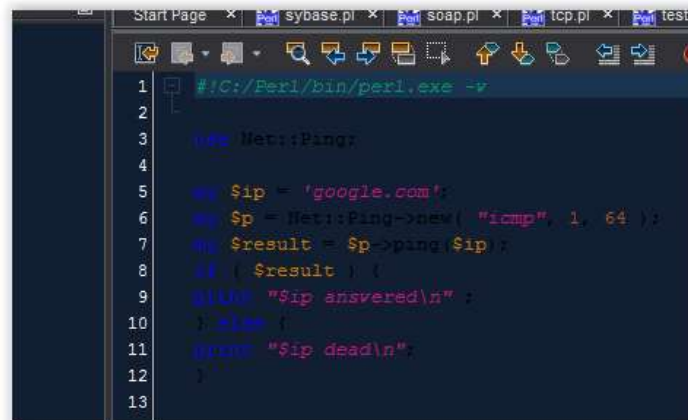


Figure 7 - Theme support for Perl files

2. Perl Project

A Perl project can be defined as a collection of a set of Perl files that can be logically grouped together. The basic operations on a Perl Project have been outlined below.

1. Create new project

To create a new project, click on the “**New Project**” icon on the toolbar or select “**File -> New project**”. The following screen is displayed:

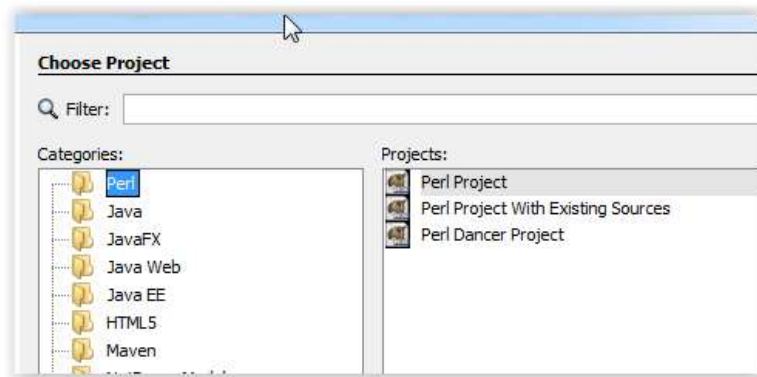


Figure 8 : Creating New PERL Project

Select “**Perl**” in the category and “**Perl Project**” under “**Projects**”. The subsequent screen will ask for the location where you want the project to be saved and will create a folder by that name with the project sources. The IDE will also open the project for further activities.

2. Create project with existing sources

Assuming a scenario where you have a few files that you want to create a project with and then use them. Here you can create a project with existing files using the newly available project type. Please note that all the selected files are imported to the project location and all modification would be done to the copy inside the project location.

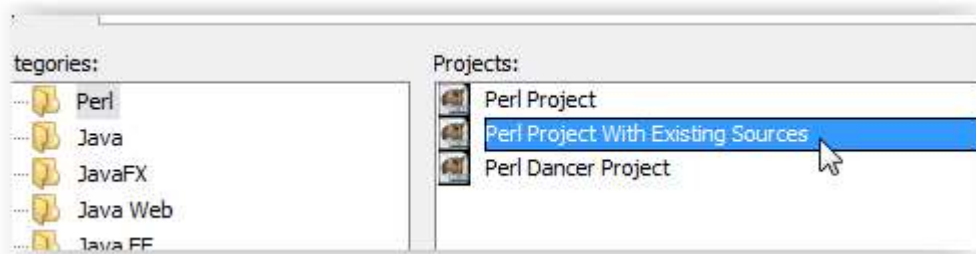


Figure 9 - Creating a project with exiting sources

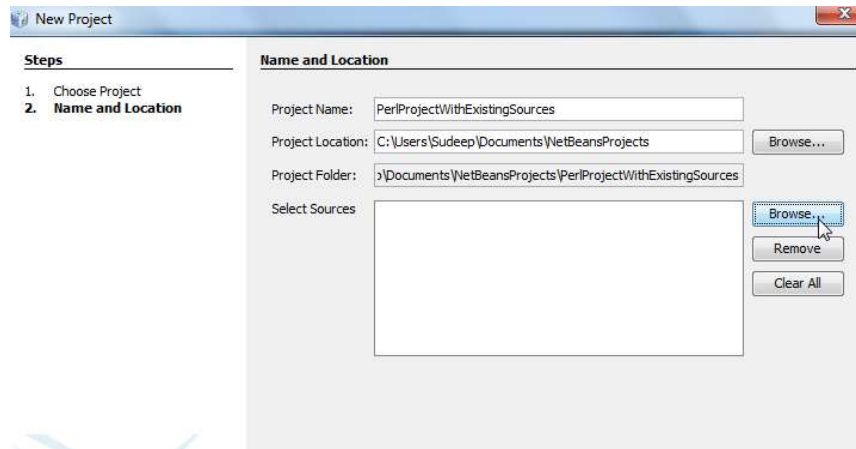


Figure 10 - Selecting the files for the project

3. Open Project

To open an already existing project, select the **“Open Project”** icon in the toolbar or go to **“File -> Open Project”**. This opens up the project selection dialog box and enable you to select an existing Perl Project to be opened. Perl project have been assigned a different icon to distinguish them from other kinds of projects, as shown below.

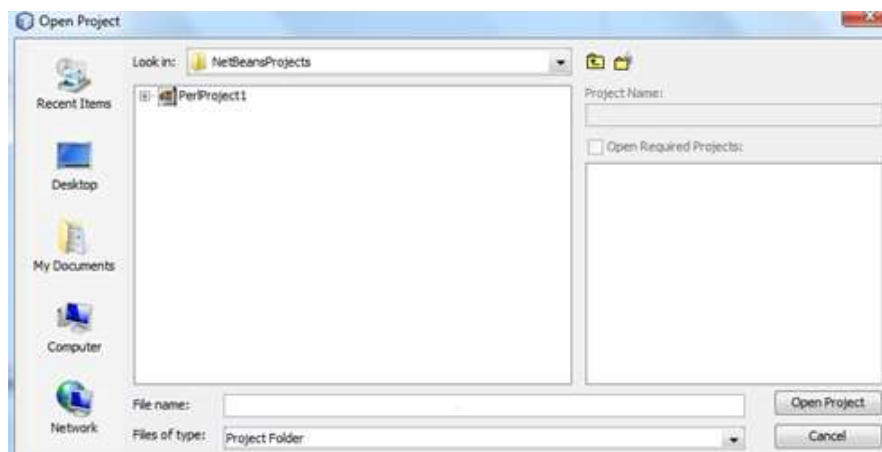


Figure 11 : Opening an existing PERL Project

4. Add new file to project

A new file can be added to any project by Right Click on the project node and then by selecting **“New ->Other”** as shown below.

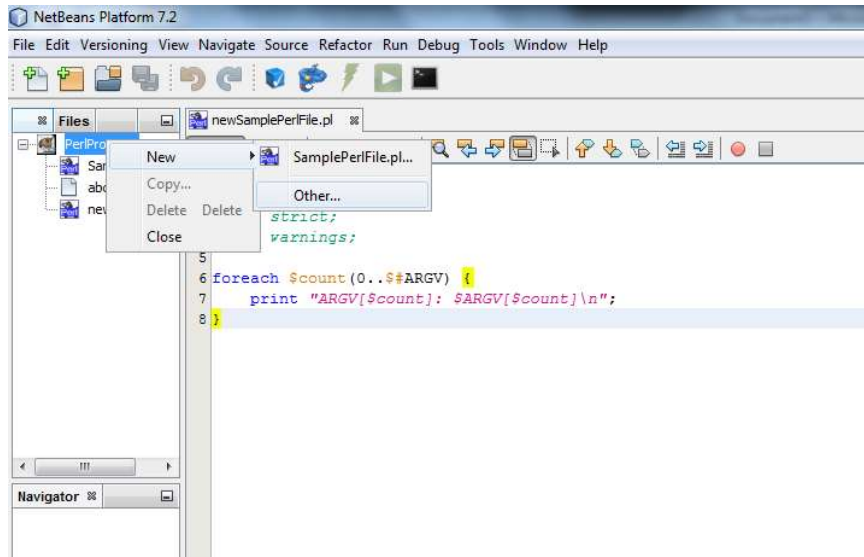


Figure 12 : Adding a new PERL file to project

This opens-up the subsequent window for selecting the file type and locations. Select the “Perl” category and then select “SamplePerlFile.pl” as the file type.

The subsequent windows will ask for the location where the files should be saved. This should be ideally left on the IDE to decide as it selects the code folder automatically.

5. Add existing file to project

To add an existing file to a project, select the project node and then click on the “Add existing file” icon as shown below. This will present you with another dialog window where you can select the files. Once the files are selected, they would be copied over to your project and be available under the project for all purposes.

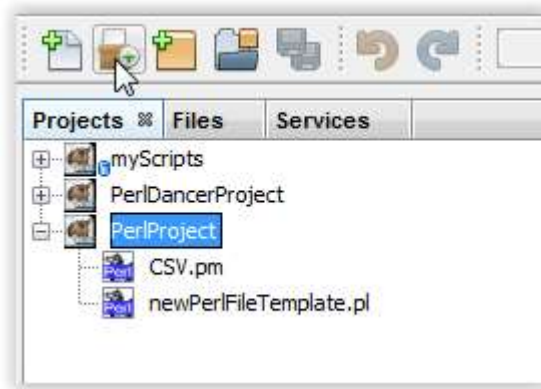


Figure 13 : Toolbar icon to add existing file to a project

6. Project Copy/Move/Delete operations

The new version enables project level operations to copy/move/delete projects. This is an adaptation of the native project level operations on the Netbeans platform.

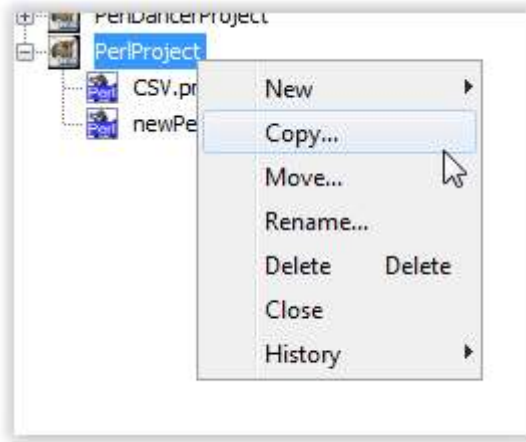


Figure 14 - Project level operations

7. Support for Perl-Dancer2 project

Before creating a Dancer project, the Dancer binaries need to be linked to the IDE. This needs to be done from the **Options -> Perl On Netbeans -> General** tab.

To create a Dancer project, please ensure that you have the proper modules installed. And select **New Project -> Perl Dancer Project**

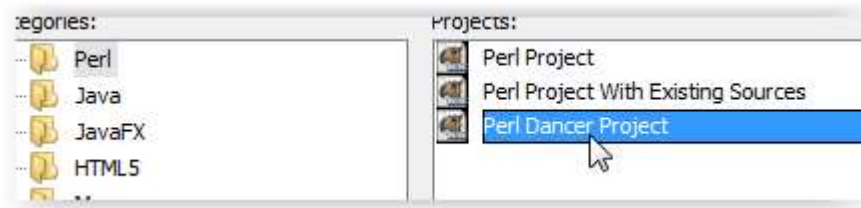


Figure 15 - Perl Dancer Project

The module is used to create a template application and then you can open bin/app.psgi and run it

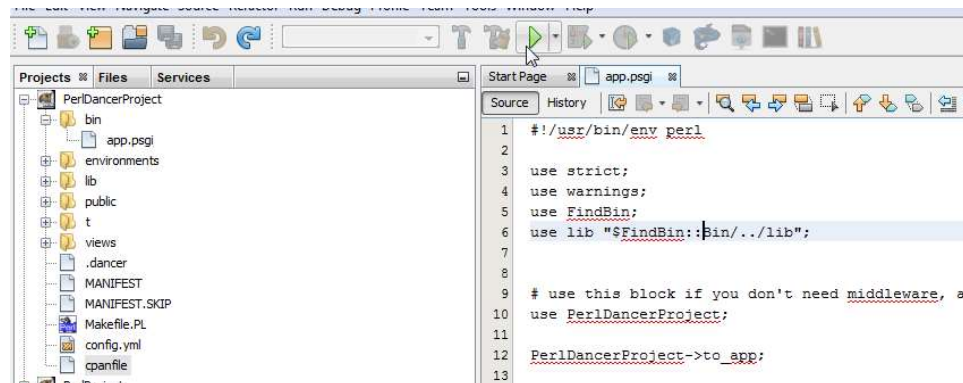


Figure 16 - Dancer main file

Running any other file within a "Perl Dancer" project will give the following message.

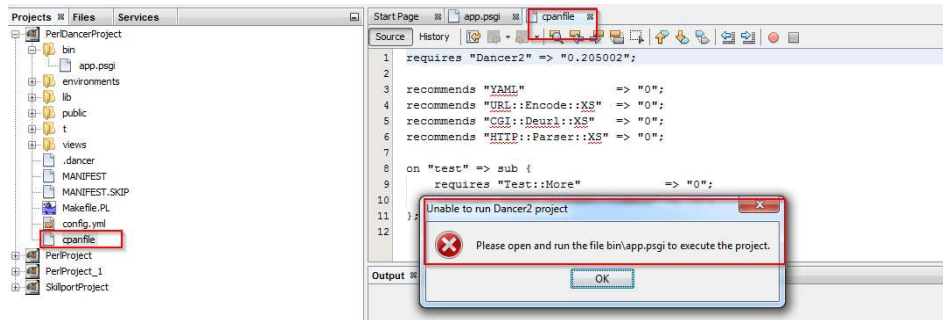


Figure 17 - Running a non-app file in a Dancer2 project

Running the bin/app.psgi file creates an instance of the internal webserver and launches the default page.

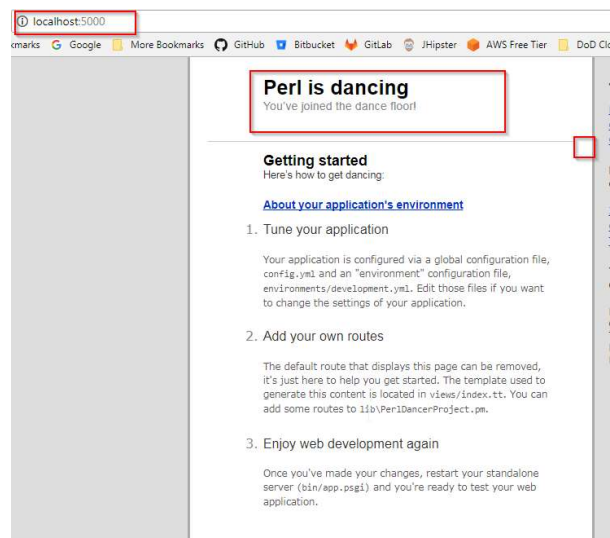


Figure 18 - Application in action

From here you can use your experience to tweak the application as you want.

Note: Please remember to shut-down the program before you want to start it again. Else the port would be in-use and the listener cannot be installed on the default port.

3. Check Files for Syntax Errors

This feature enables you to check the syntax of the file that is currently selected in the editor. To execute any specific file that is currently selected in the editor, click on the “Run -> Compile File” button as shown below. This operation can be accessed using the “F9 key”. This checking happens based on the version of the Perl compiler that is currently installed on your system.

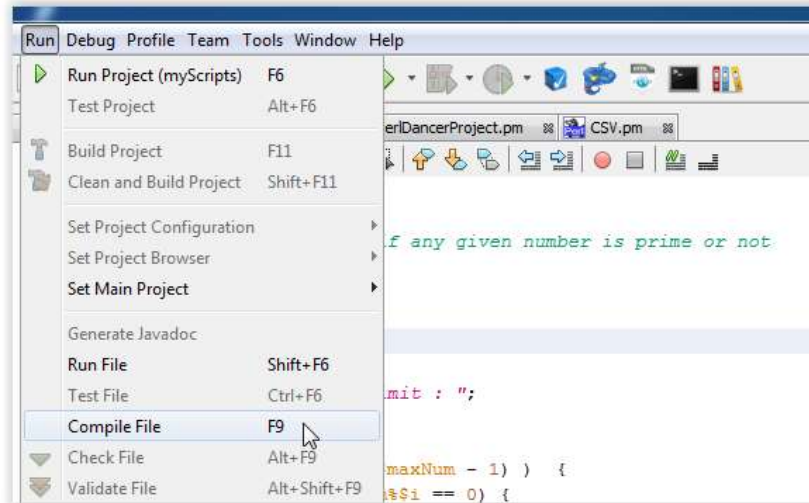


Figure 19 : Syntax checking results

4. Code execution

1. Execute files

This feature enables you to execute the file that is currently selected in the editor. To execute any specific select it in the editor and click on the “**Execute**” button on the toolbar as shown below. This execution happens based on the version of the Perl compiler that is currently installed on your system or selected from the options panel (*described below*).

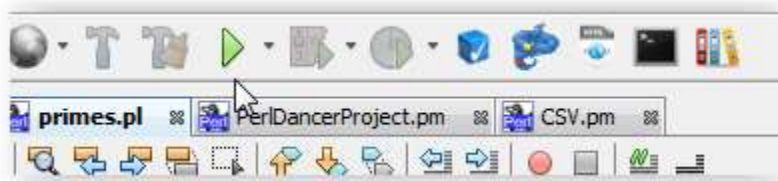


Figure 20 : Toolbar icon to execute selected PERL file

2. Code execution with command line arguments

The latest release allows one to supply command-line argument(s). To execute a file with command line options, select the file in the editor and click the “**Execute (with command line options)**” button on the toolbar as shown below. On using this feature, a dialog box pops up, in turn facilitating the input of command-line parameters to be used in one’s program.



Figure 21 : Toolbar icon to execute selected PERL file with command line arguments

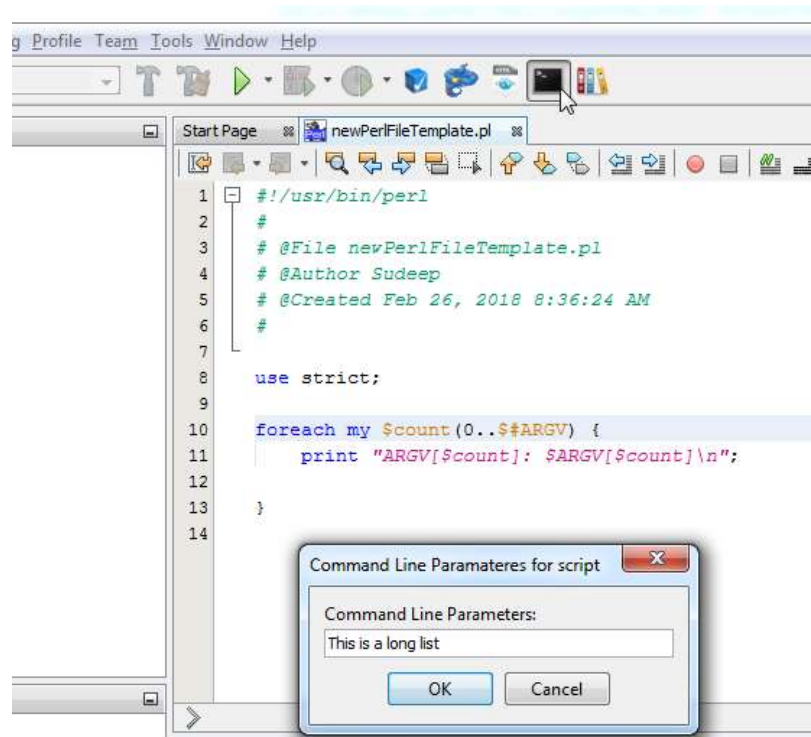


Figure 22 : Providing the command line arguments

Subsequent Execution of the script

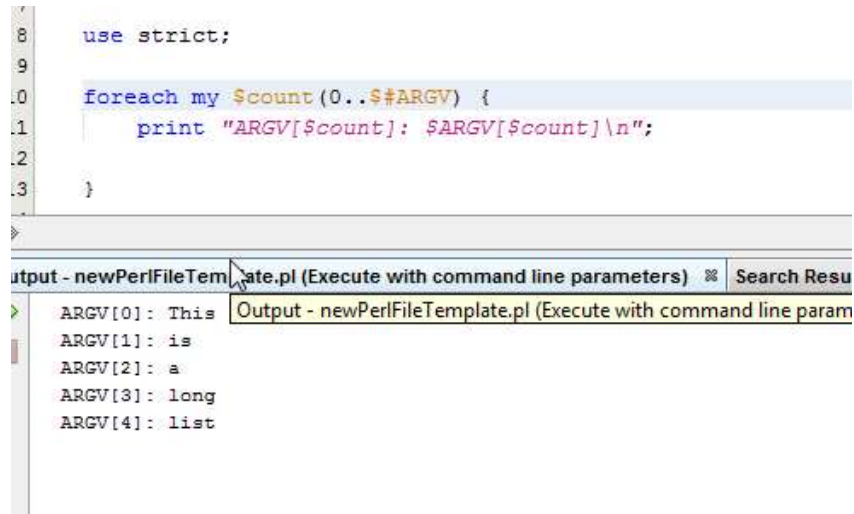


Figure 23 : Results of execution with the command line arguments

5. Source Code formatting

This feature enables you to perform source re-formatting as per **Perl::Tidy**. To perform analysis, select the file in the editor and click on the code formatter button on the toolbar as shown below. This will run the source code formatter against the selected source.



Figure 24 : Toolbar icon to format selected PERL file

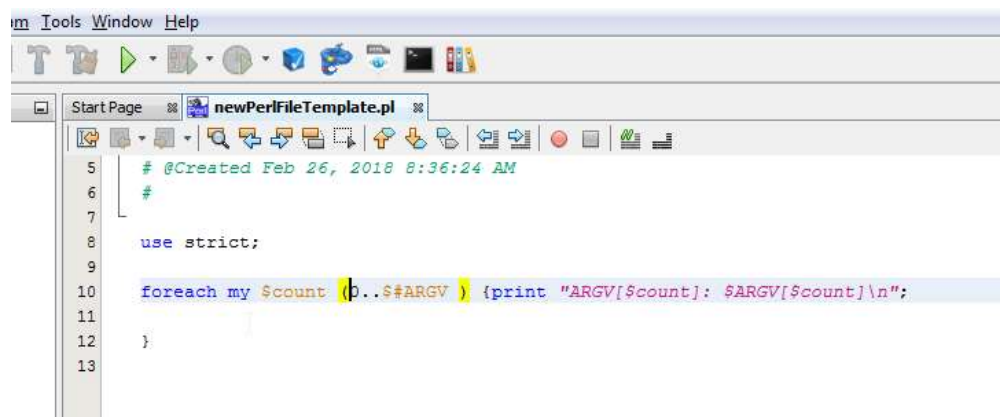


Figure 25 : Code formatting (Before)

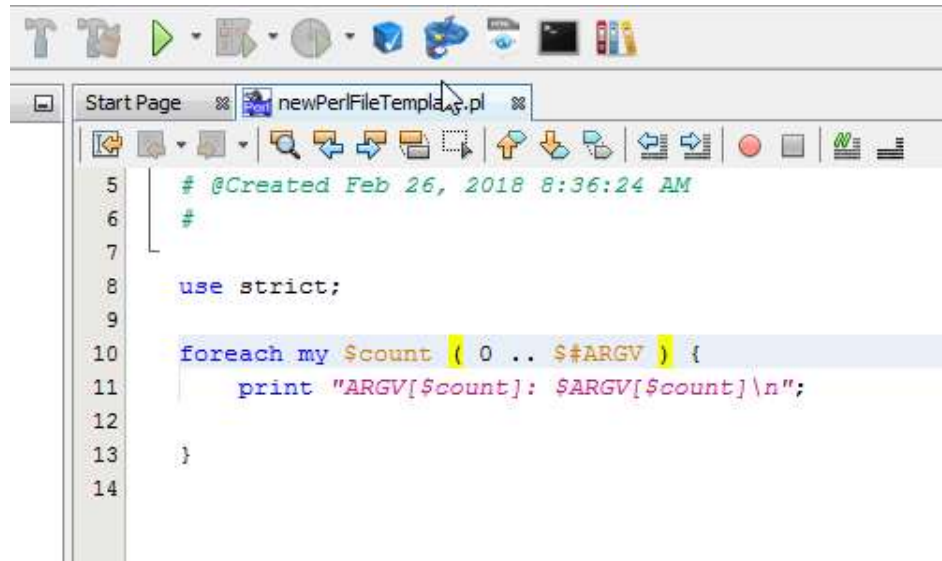


Figure 26 : Code Formatting (After)

6. Source code analysis

This feature enables you to perform source analysis as per **Perl::Critic**. To perform analysis, select the file in the editor and click on the code analyzer button on the toolbar as shown below. This will run the source code analyzer against the selected source. You'll need **Perl::Critic** installed.



Figure 27 : Toolbar icon to analyze source code of selected PERL file

The output of the analysis will be provided in the Output window, as shown in the screenshot below:

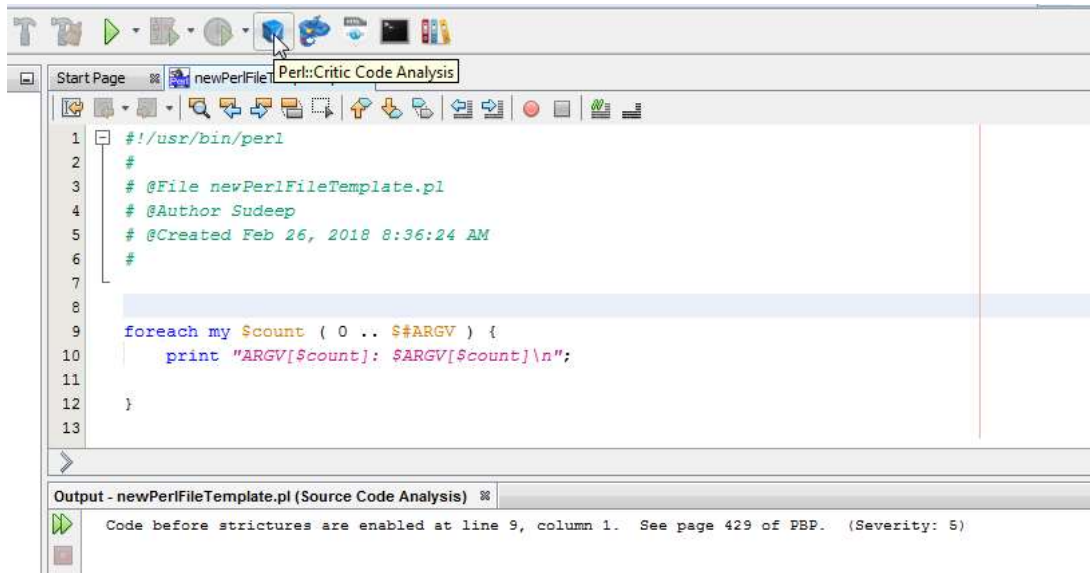


Figure 28 : Source code analysis results

7. Integrated PerlDOC support

PerlDOC support has been also made available. If you want more information about any keyword, just select it in your editor and press on the “PerlDOC” button, and it will display the documentation in the output window. Similarly, if you just press the button, a pop-up window will ask you for the help topic and display it in the output window. You’ll need **perl-doc** installed for this.

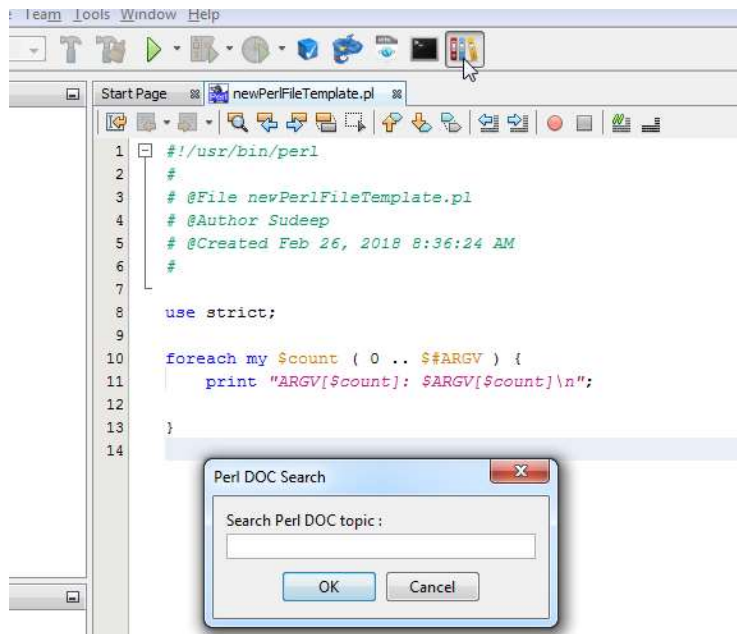


Figure 29 - Toolbar icon to display documentation

8. Options Panel

The Options panel contains a section called “Perl OnNetBeans” for our usage. This tab contains 3 sub-tabs that are utilized for various options. Also template support has been added for programming efficiency.

1. Templates

You can specify any code template that you use regularly as a shortcut here. For example “p” can translate to “print”. So, if you add a template here, and then go to your editor with a Perl file open, type “p” and press “tab”, and voila, this would get translated in the editor as “print”. You can add as many templates as you want.

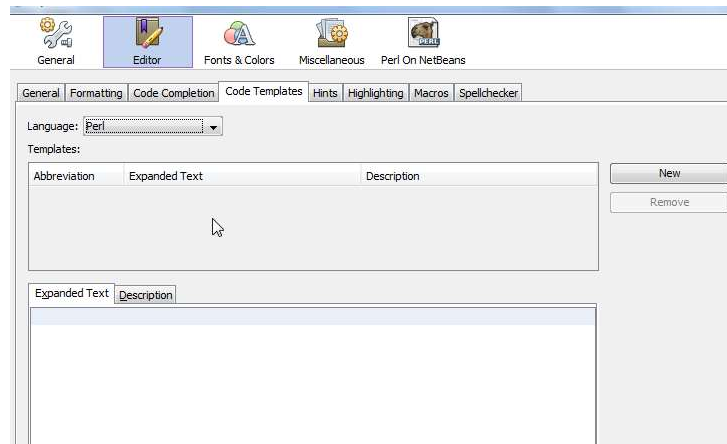


Figure 30 : Code template configuration panel

2. General

This tab contains the option to over-ride the default PERL installed on your machine. Let’s say that you have PERL 5.14 installed on your machine as the default but you want to use PERL 5.16 available at a non-default location, you can point the perl.exe binary using this tab and your program will be operated on by the executable specified here. If you do not specify anything here, the default PERL will be used.

You can also specify any custom library file location here. These locations would be used as an addition to your @INC in the script.

Also, the Dancer2 executables need to be pointed to at this location. The Dancer2 Path should be the folder where the Dancer2 and Plackup scripts are installed. This is not required to be populated if Dancer2 is installed with the default Perl installation at the default location.

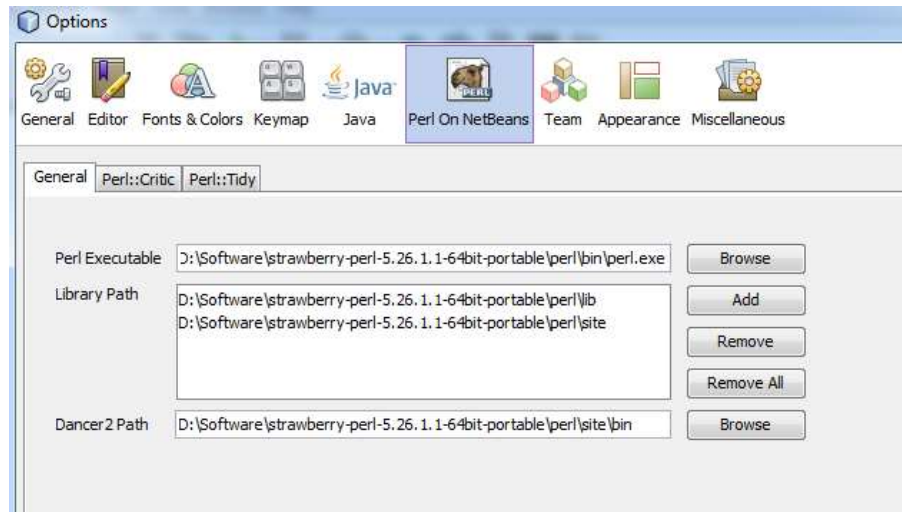


Figure 31 : PERL binary configuration panel

3. Perl::Critic

This tab can be used to customize the code analysis options. For a complete meaning of the options here, please refer to the URL <http://linux.die.net/man/1/perlritic>.

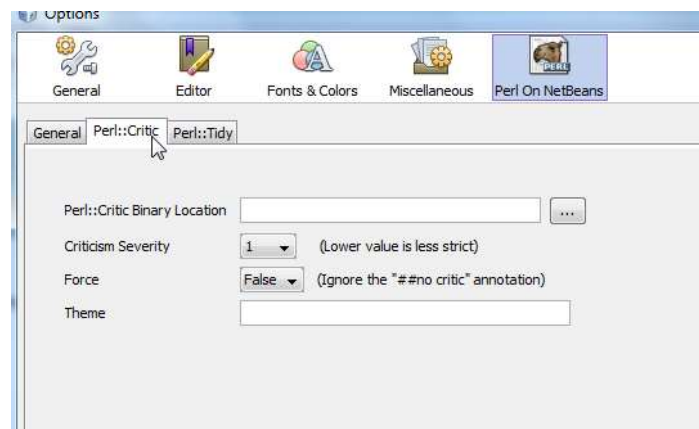


Figure 32 : PERL::Critic (Code Analyzer) configuration panel

4. Perl::Tidy

This tab can be used to customize the code formatting options. For a complete meaning of the options here, please refer to the URL <http://perltidy.sourceforge.net/perltidy.html>.

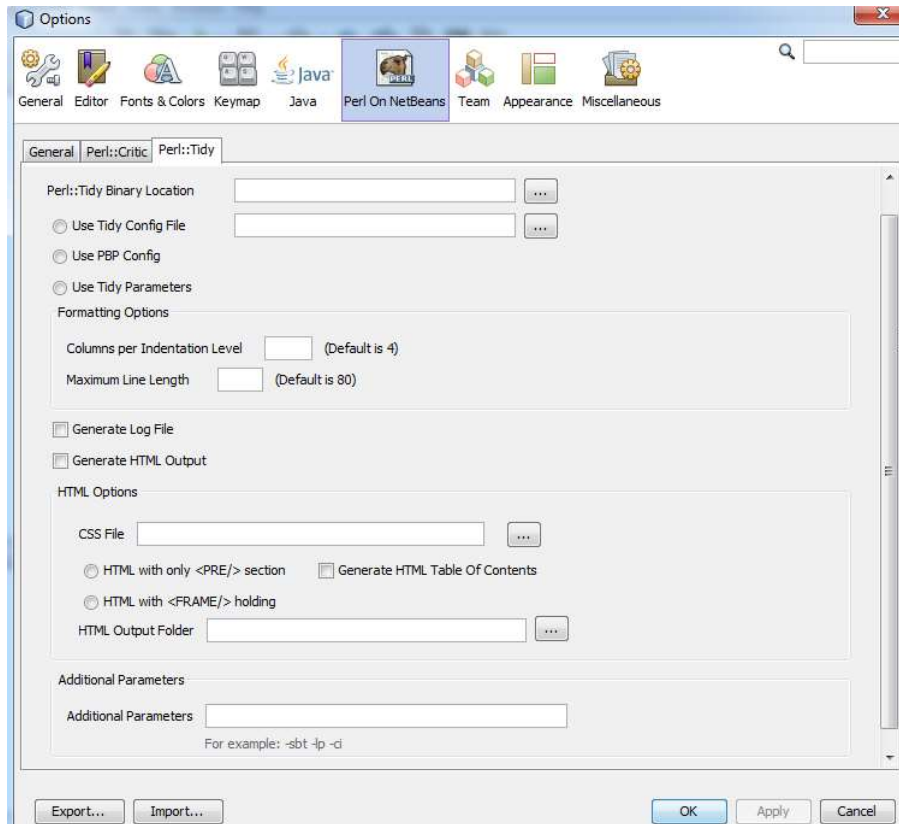


Figure 33 : PERL::Tidy (Code Formatting) configuration panel

We encountered a feature of “Export to HTML” and it can be configured from this panel. You can specify the output folder of the generated HTML (project folder by default), specify if you require a “Table Of Contents”, specify any CSS file, and also FRAME and PRE tags in the output HTML.

9. Integrated versioning system

The new version also includes integrated support for major code versioning systems including Subversion, Git and Mercurial. The appropriate context menu options are displayed based on whether the code is connected with a SVN/Git/Hg repository.

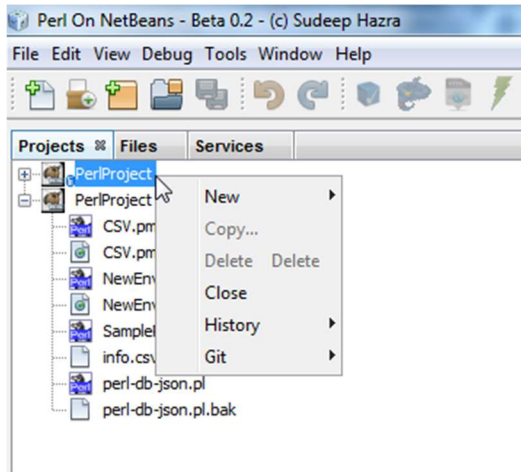


Figure 34 : A Git repo Project

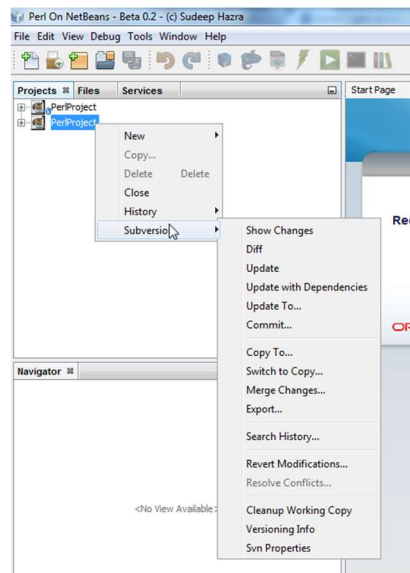


Figure 35 : A SVN repo Project

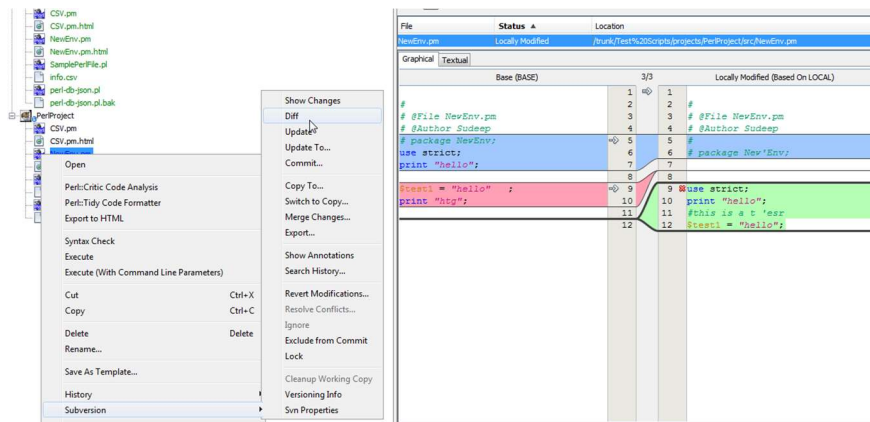


Figure 36 : A Subversion file diff

Feedback

Since this is a community driven initiative, we would request everyone to log-on to the Facebook space for any comments/suggestions or follow us on Twitter for staying up-to-date with the latest developments. Also for any issues, please do log a ticket with GitHub on the project space as mentioned in the “Web Links” section below.

Web Links

1. Facebook page - <http://www.facebook.com/PerlOnNetBeans>
2. Twitter - @perlonnetbeans
3. Issue Tracker – <https://github.com/sudeephazra/perl-on-netbeans/issues>
4. NetBeans download - <https://netbeans.org/downloads/>
5. Perl download - <https://strawberryperl.com/>
6. Zulu JDK download - <https://www.azul.com/downloads/>

