SolutionTool User Guid

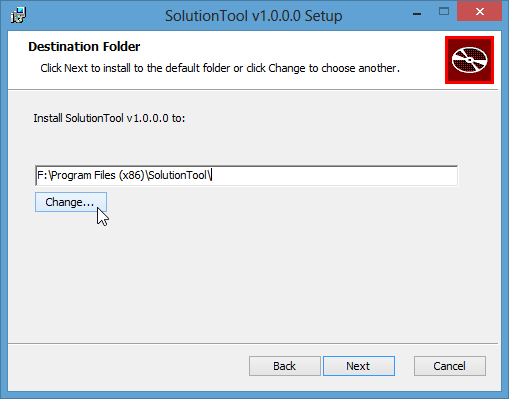
# Download and Install

## Download a Copy of SolutionTool from GitHub

<https://github.com/paytonli2013/SolutionTool/tree/master/Deployment>

## Install SolutionTool

Double click the SolutionToolSetup\_Release.msi and follow the steps. This installer will create a shortcut on your desktop once it’s successfully completed.



# slntool.exe

## This is a console app which is used to check if a code repository follows certain rules:

* File Structure Rule

If the repository is structured correctly via gitignore-like template;

* Build Output Rule

If the BuildOutput node value of each \*.csproj file in the reposotry is set to a correct value;

* Code Analysis Rule

If the \*.sln/\*.csproj in the repository follows rule by InspectCode tool;

## Usage of the console app:

* Running slntool.exe without any parameters will display the help screen;

F:\Program Files (x86)\SolutionTool>slntool.exe

slntool 1.0.0.0

Copyright c Hewlett-Packard Company 2013

ERROR(S):

-r/--repository required option is missing.

-r, --repository Required. The path of the repository.

-t, --template (Default: ./templates/default.txt) The xml template file

name for checking solution structure.

-b, --builds (Default: ./output/$(Configuration)/) The BuildOutput

path.

-i, --inspectcode The InspectCode executable path.

-l, --log The log file path.

--help Display this help screen.

* Run slntool.exe with the default template, which can be found at {slntool.exe\_dir}/Tempaltes/default.xml

F:\Program Files (x86)\SolutionTool>slntool.exe -r "F:\\_E\_\SolutionTool"

Begin checking F:\\_E\_\SolutionTool

#Default File Structure

Escape checking a comment pattern.

#Directory must exist

Escape checking a comment pattern.

src/

Pass

doc/

Pass

output/

Pass

output/debug/

Pass

output/release/

Pass

deployment/

Pass

#File must exist

Escape checking a comment pattern.

.gitignore

Pass

readme.md

Pass

#Directory must not exist

Escape checking a comment pattern.

!\*\*/bin/

F:\\_E\_\SolutionTool\src\SolutionTool\Modules\ManageRule\bin

F:\\_E\_\SolutionTool\src\SolutionTool\Modules\ManageTemplate\bin

..\..\..\output\$(Configuration)\ <--> ../../../output/$(Configuration)/

..\..\..\output\$(Configuration)\ <--> ../../../output/$(Configuration)/

Checking is done.

* Run the tool using a custom template

1. Create a new template file under {slntool.exe\_dir}/Templates/, say custom1.txt;

#Default File Structure

#Directory must exist

src/

doc/

output/

output/debug/

output/release/

deployment/

#File must exist

.gitignore

readme.md

#Directory must not exist

!\*\*/bin/

!\*\*/obj/

#File must not exist

1. Run the tool as below

F:\\_E\_\SolutionTool\output\Debug>slntool.exe -r "F:\\_E\_\SolutionTool" –t custom1.txt

* Run InspectCode from the tool

In order to run InspectCode, the executable path needs to be specified, say d:\jc-tools\InspectCode.exe. Then use below command:

F:\\_E\_\SolutionTool\output\Debug>slntool.exe -r "F:\\_E\_\SolutionTool" –i "d:\jc-tools\InspectCode.exe"

# SolutionTool.exe

Click SolutionTool.exe will display the main window, which is composed with 4 modules:

* Run SolutionChecker module

You can create new projects, run checks and view logs.

* Manage Rule Module

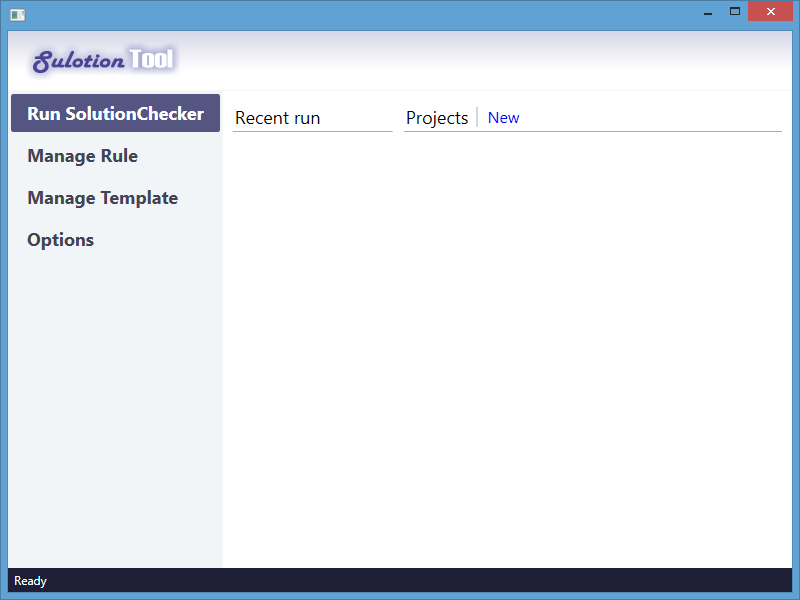
You can view system built-in rules.

* Manage Template module

You can define file structure in a gitignore-like template file.

* Options module

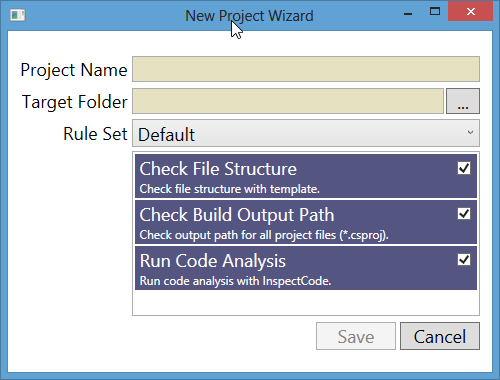
You can specify system settings, e.g., the executable path of InspectCode.



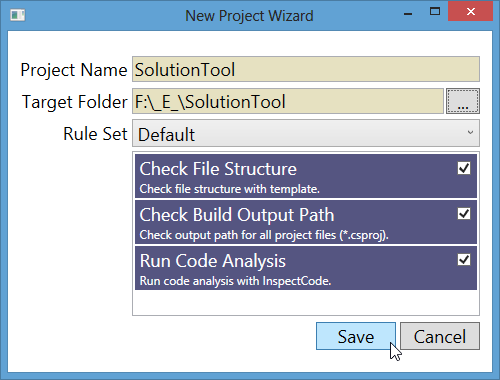
## SolutionChecker Module

### Creating a New Project

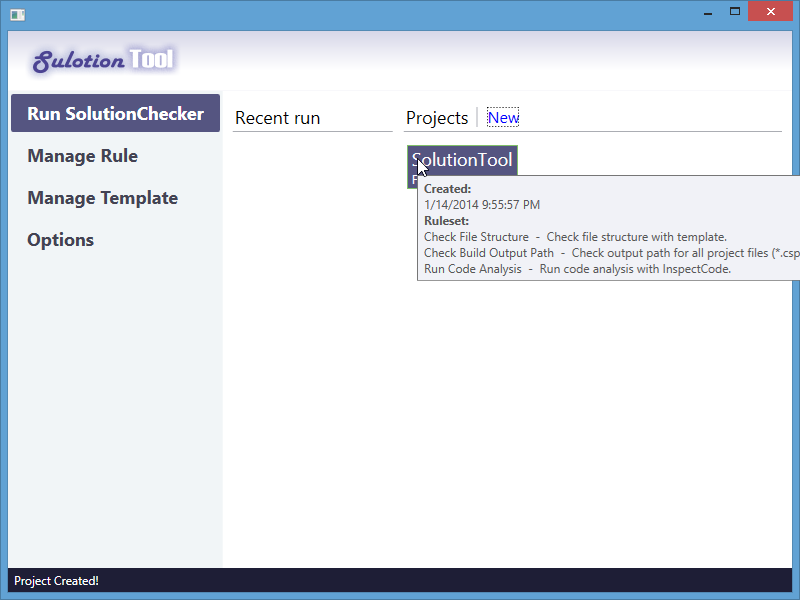
Click the New button on the "Run SolutionChecker" view will bring up the "New Project Wizard” dialog.



Select a "Target Folder" and select the rules you want to apply to the project, and finally click the "Save" button.

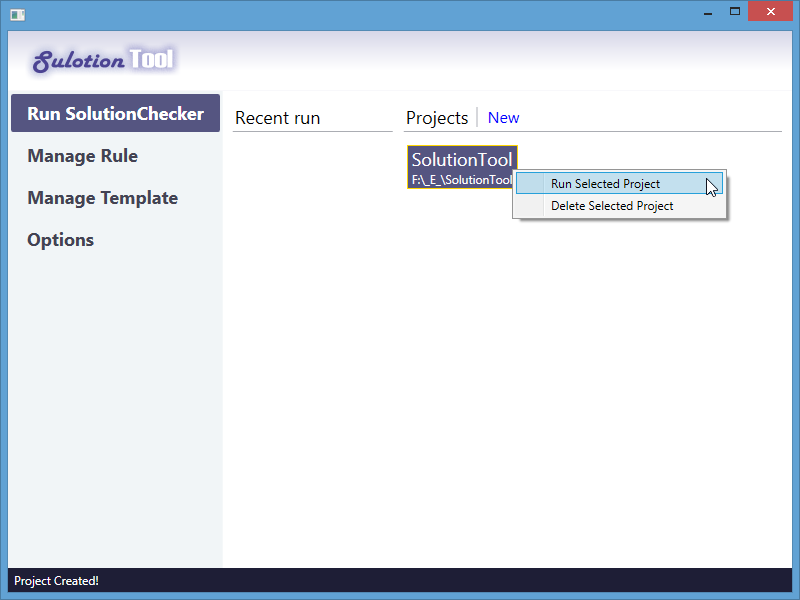


The newly created project will display on the SolutionChecker view, a tooltip will show while the mouse hovers on the item.

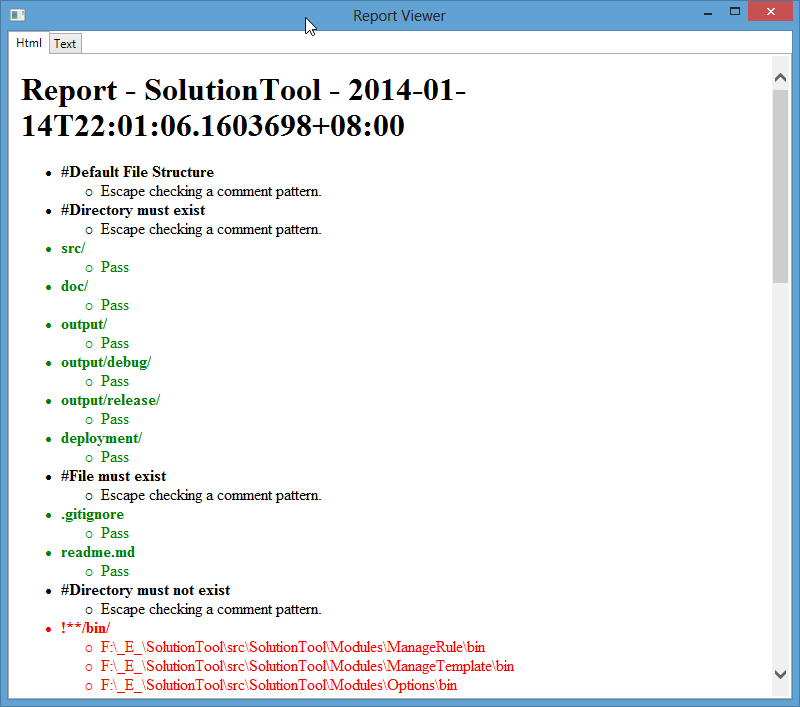


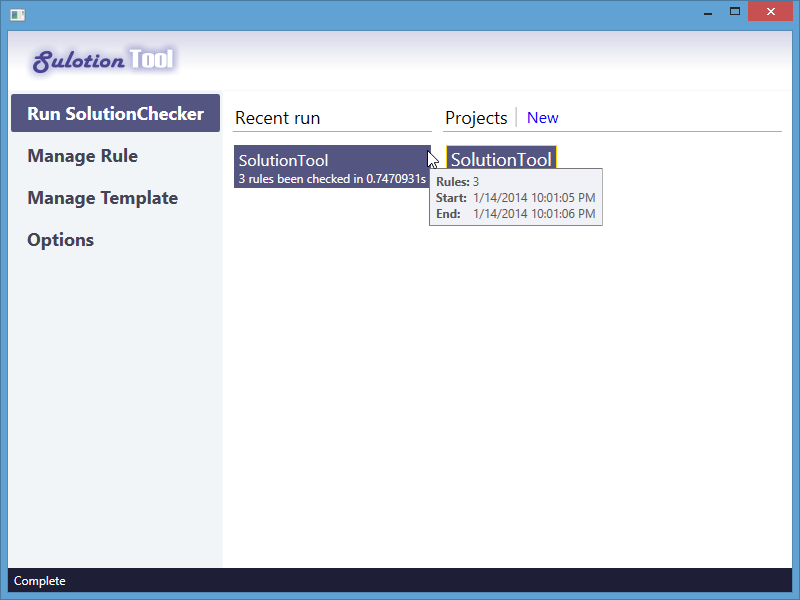
### Running Checks

Right click on the project, and click "Run Selected Project" will trigger the checker does its jobs.



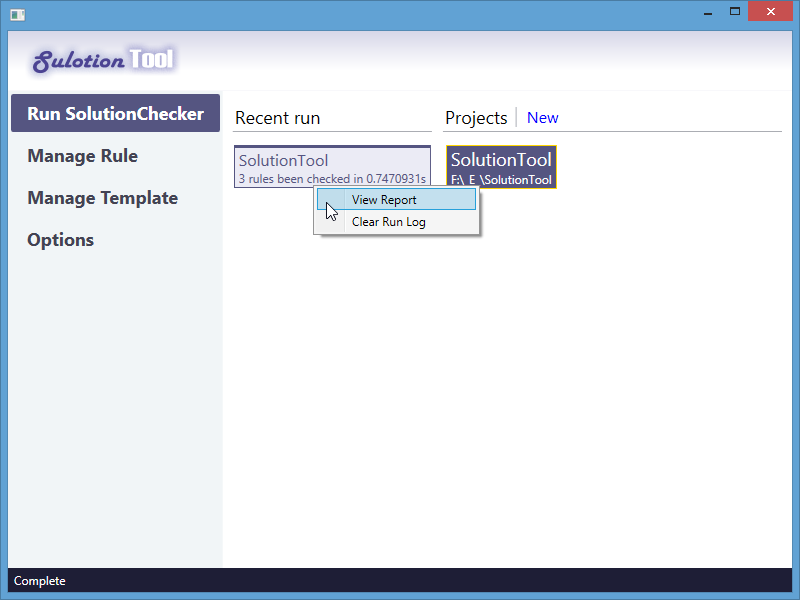
Once the checking is done, a run log will be inserted at the "Recent Run" list, and checking report shows as soon as the checking is done.





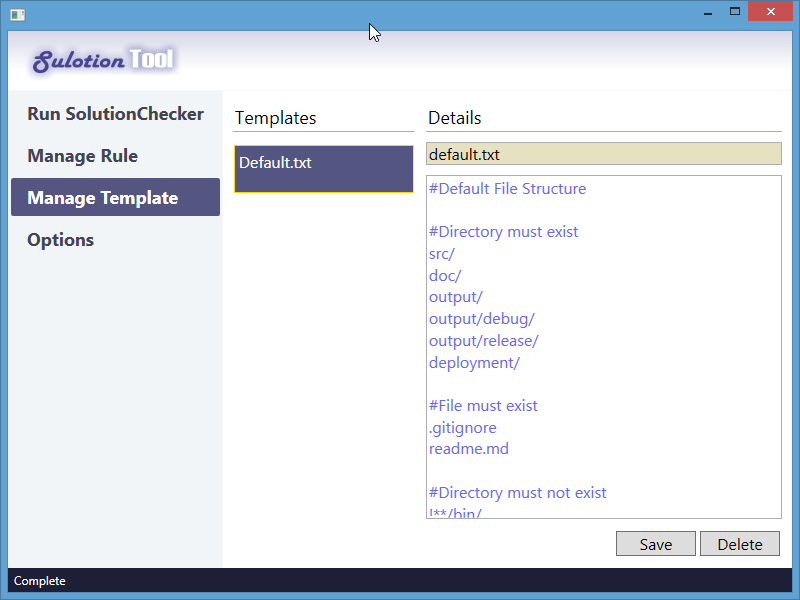
### Viewing Project Reports

Right click on the log item, select "View Report" will display the checking results again.

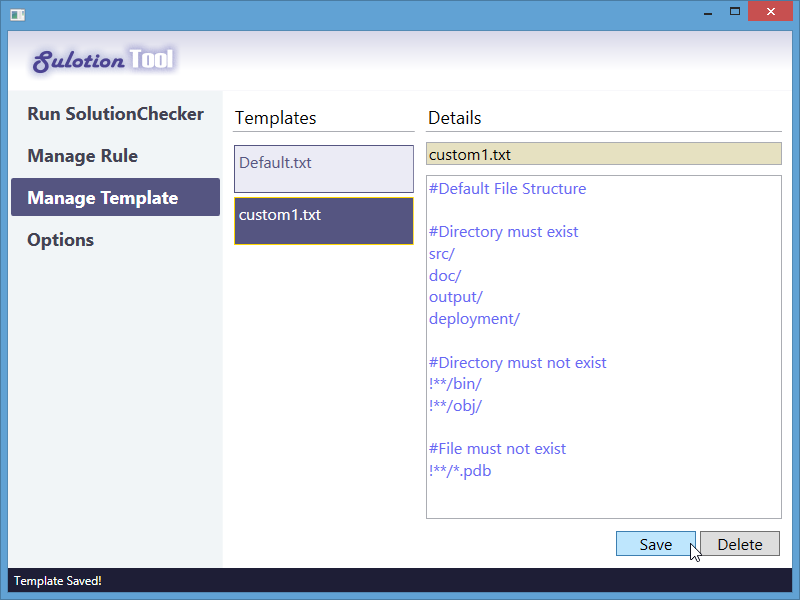


## ManageTemplate Module

You can view application built-in template on the view, and create new templates base on the default template.

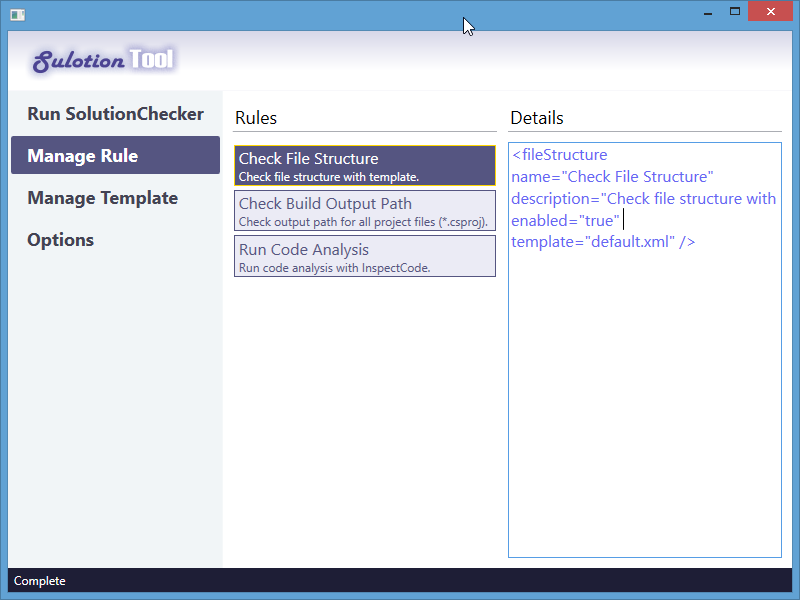


Simply adding/removing git patterns, give it a new name, and click "Save" button.



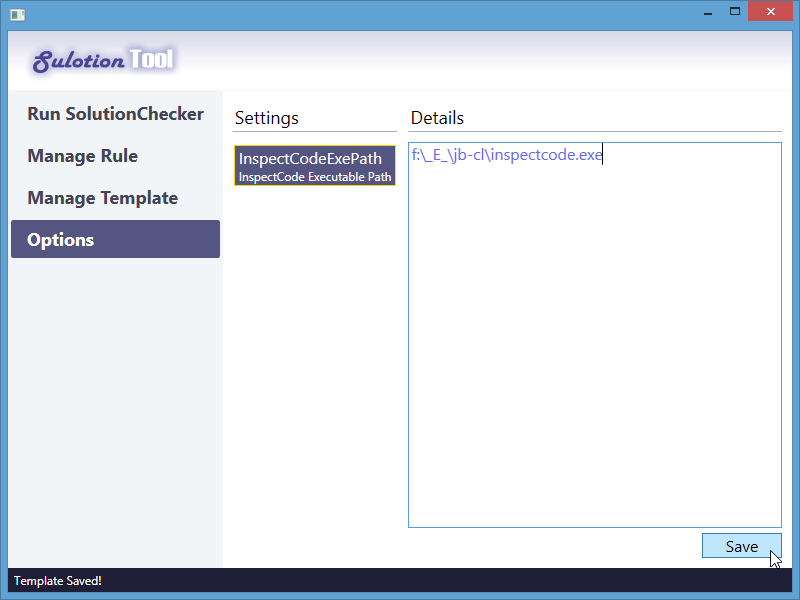
## ManageRule Module

You can change view application built-in rules on the view, but changing to rule settings must be done manully to {app\_root\_dir}\Rules\default.xml.



## Options Moduel

Currently SolutionTool can execute InspectCode.exe to do code analysis, but explicit path to the executable must be configured on this view. Fill in the correct path, and click "Save".



You don’t need to specify the path each time running the app, since it will be saved in the app configuration file.

Please be aware that projects created before the path is saved, will not run InspectCode to do code analysis, but new ones do. Because rules is kept in the project’s XML right when it’s created.