Solution Checker User Guid

# 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 XML 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:\\_E\_\SolutionTool\output\Debug>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: default.xml) 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:\\_E\_\SolutionTool\output\Debug>slntool.exe -r "F:\\_E\_\SolutionTool"

Begin checking F:\\_E\_\SolutionTool

\*\*\*File Structure Rule\*\*\*

--------------------------------------------------------------------------------

Directory Exists

.\src\SolutionTool\Modules\ManageRule\bin

.\src\SolutionTool\Modules\ManageTemplate\bin

--------------------------------------------------------------------------------

Directory Missing

.\deployment

--------------------------------------------------------------------------------

File Missing

.\output\debug\fileNotExists.txt

.\fileNotExists.txt

\*\*\*Output Path Rule\*\*\*

--------------------------------------------------------------------------------

Path OK

.\src\SolutionTool\slntool\slntool.csproj

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

--------------------------------------------------------------------------------

Path NG

.\src\SolutionTool\Modules\ManageRule\ManageRule.csproj

\*\*\*Code Analysis Rule\*\*\*

--------------------------------------------------------------------------------

Cannot find InspectCode app at [inspectcode.exe].

* Run the tool using a custom template

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

<?xml version="1.0"?>

<dir name=".">

  <dir name="src">

    <dir name="bin" ocurr="0" recursive="true" />

    <dir name="obj" ocurr="0" recursive="true" />

  </dir>

  <dir name="output">

    <dir name="debug />

    <dir name="release" />

  </dir>

  <dir name="doc" />

  <dir name="deployment" />

  <file name="Readme.md" />

  <file name=".gitignore" />

</dir>

1. Run the tool as below

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

* 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

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

* Manage Rule Module

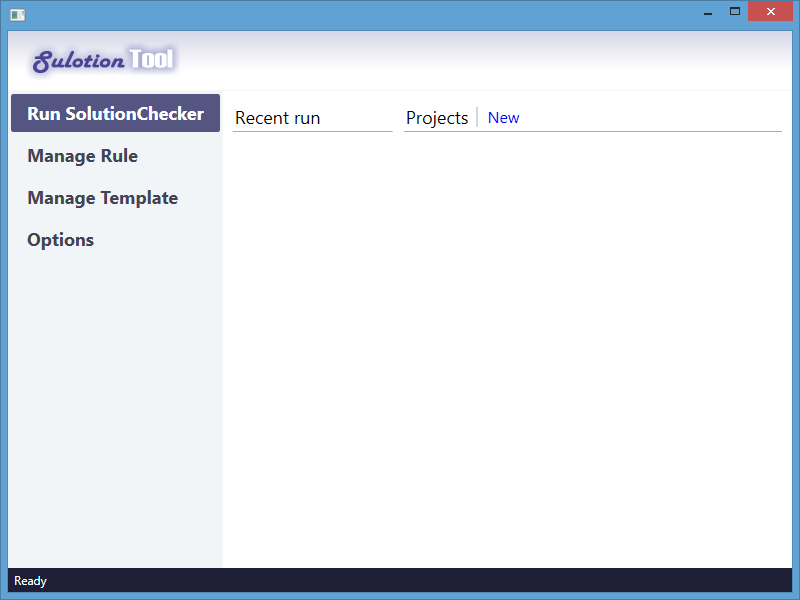
Users can view system built-in rules.

* Manage Template module

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

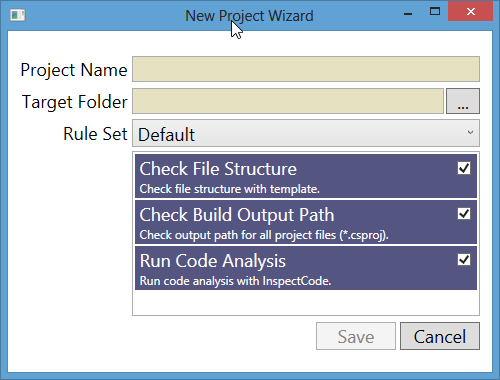
* Options module

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



## SolutionChecker Module

### Creating a New Project



### Running Checks

### Viewing Project Reports

## ManageTemplate Module

## ManageRule Module

## Options Moduel