Use coverity to scan the Misra C rules

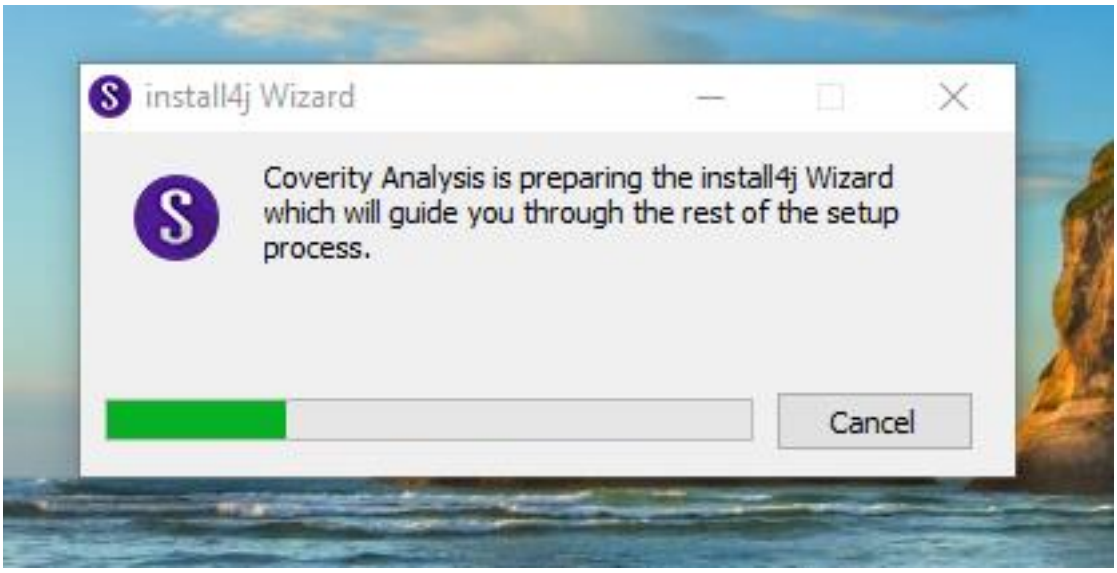
1. How to install Coverity Analysis

Download the Coverity **Analysis** installer and file by visiting the Download Coverity link (Only access from the Fsoft network): Download the setup file corresponding to your system and the license.dat file for the activation license.

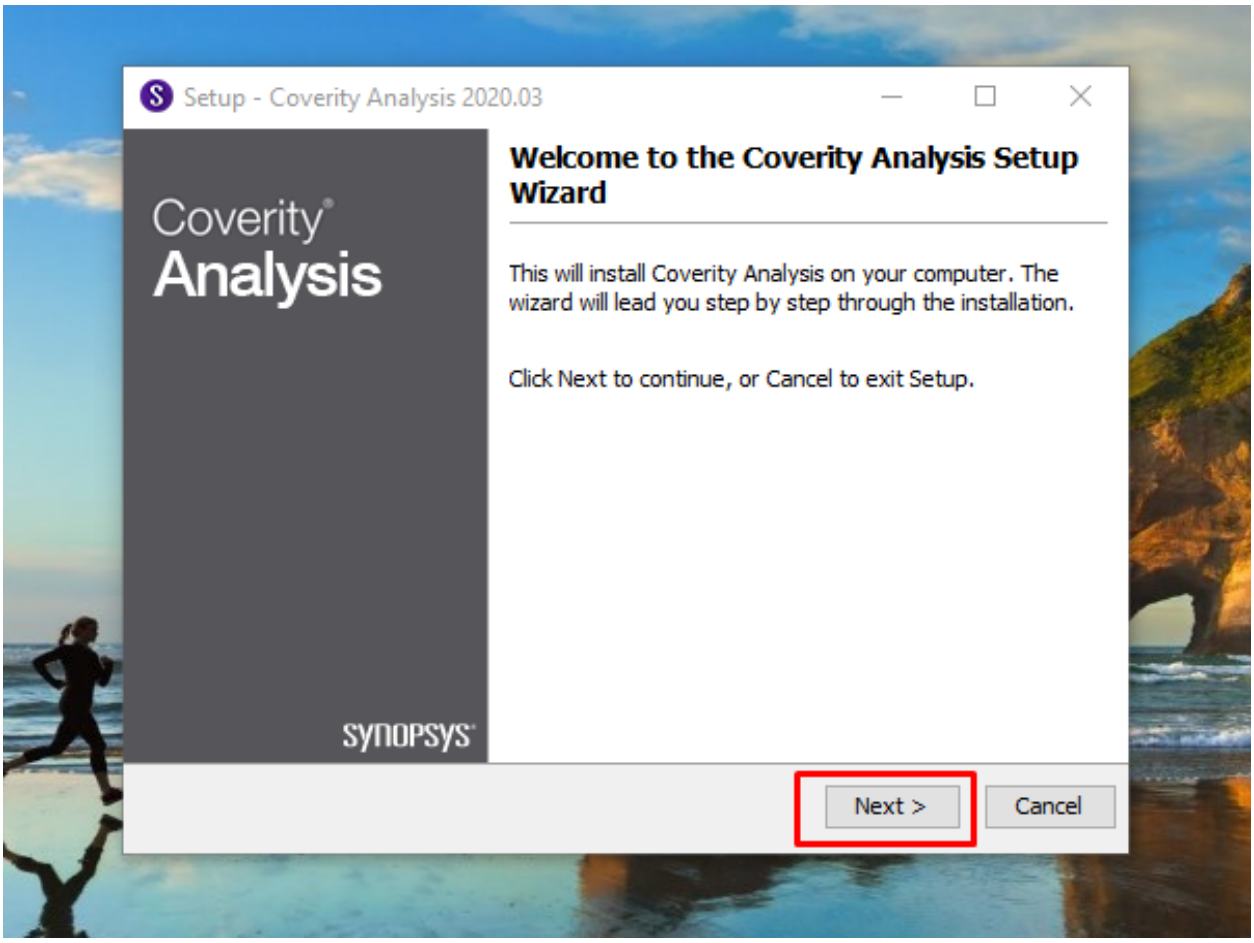
1.1. Install on Windows

Here are the steps to install **Coverity Analysis** on the Windows operating system

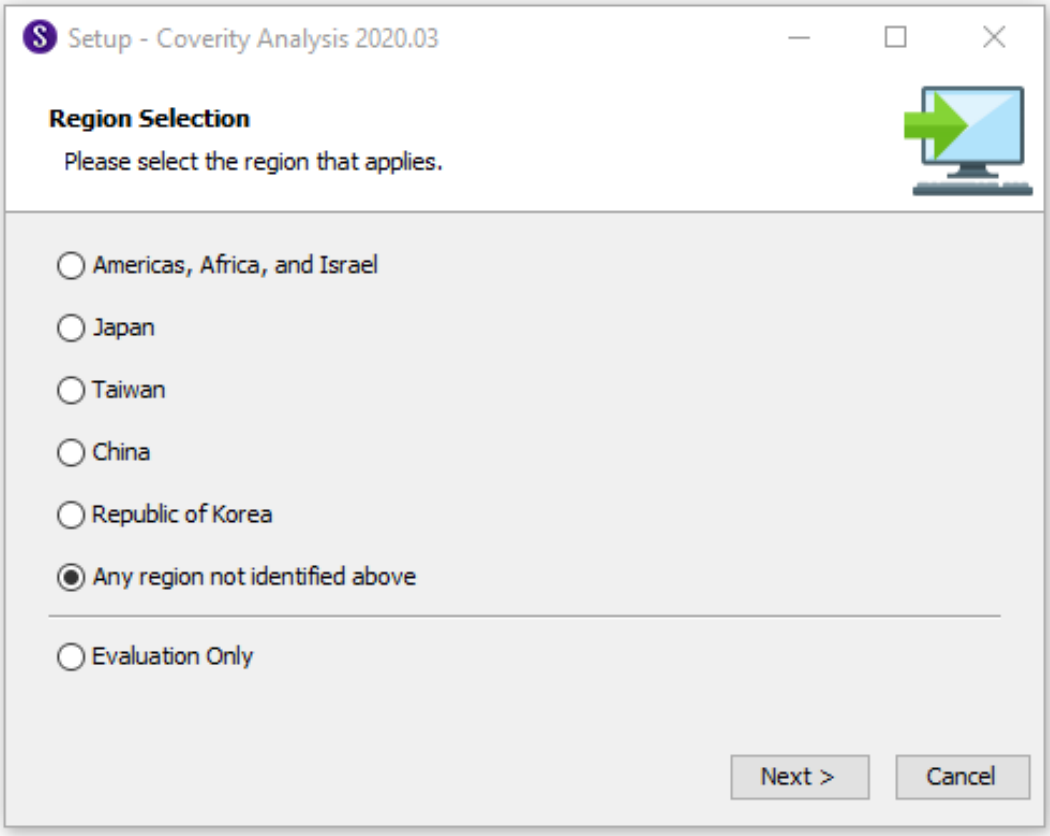
- **Step 1:** Install the Coverity file with permissions `administrator`



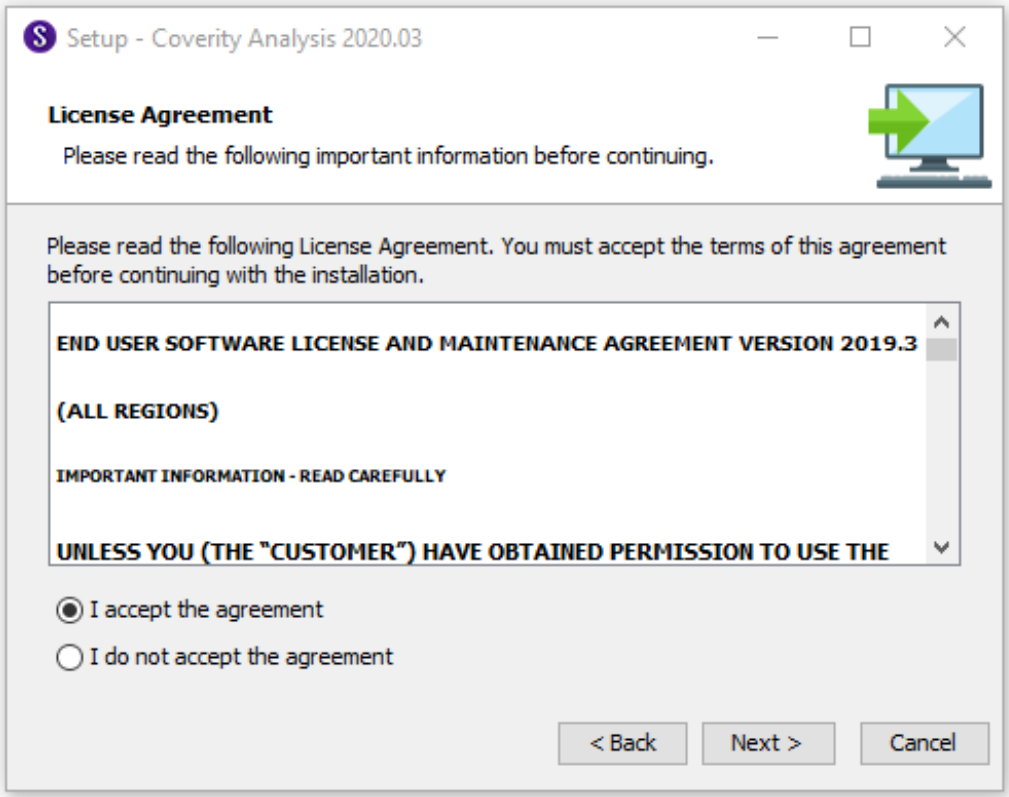
- **Step 2:** Click **Next**



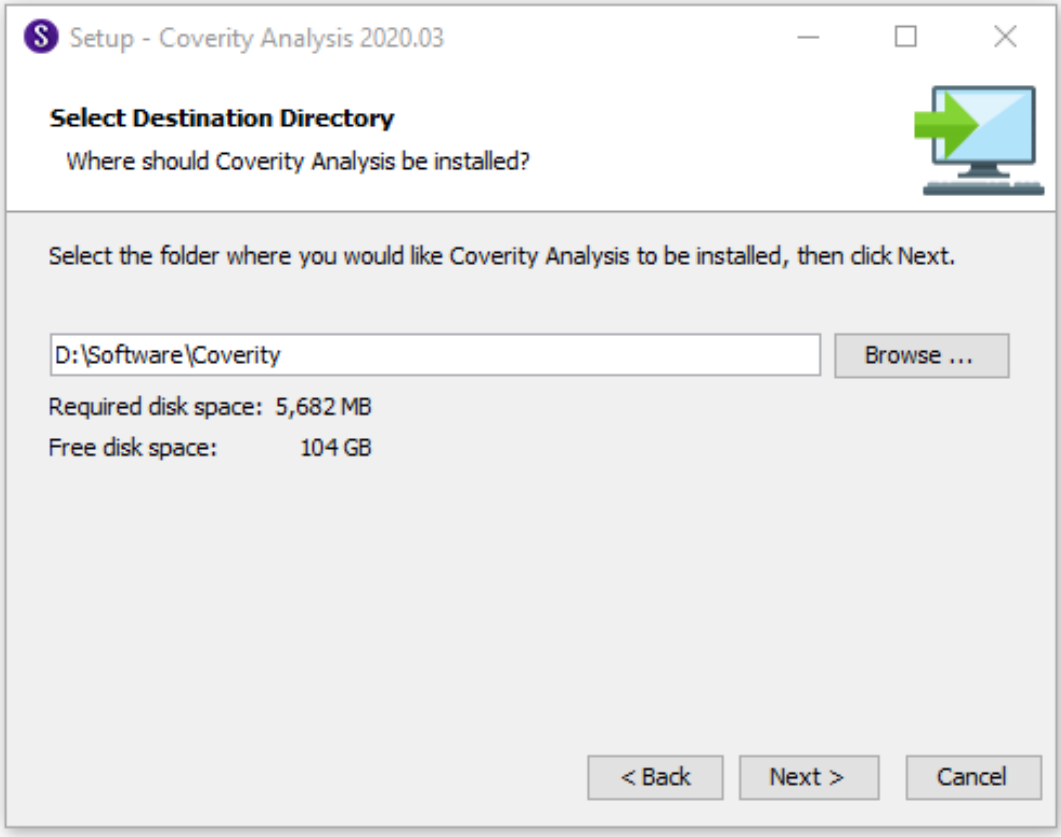
- **Step 3:** Select **Any region not identified above → Next**



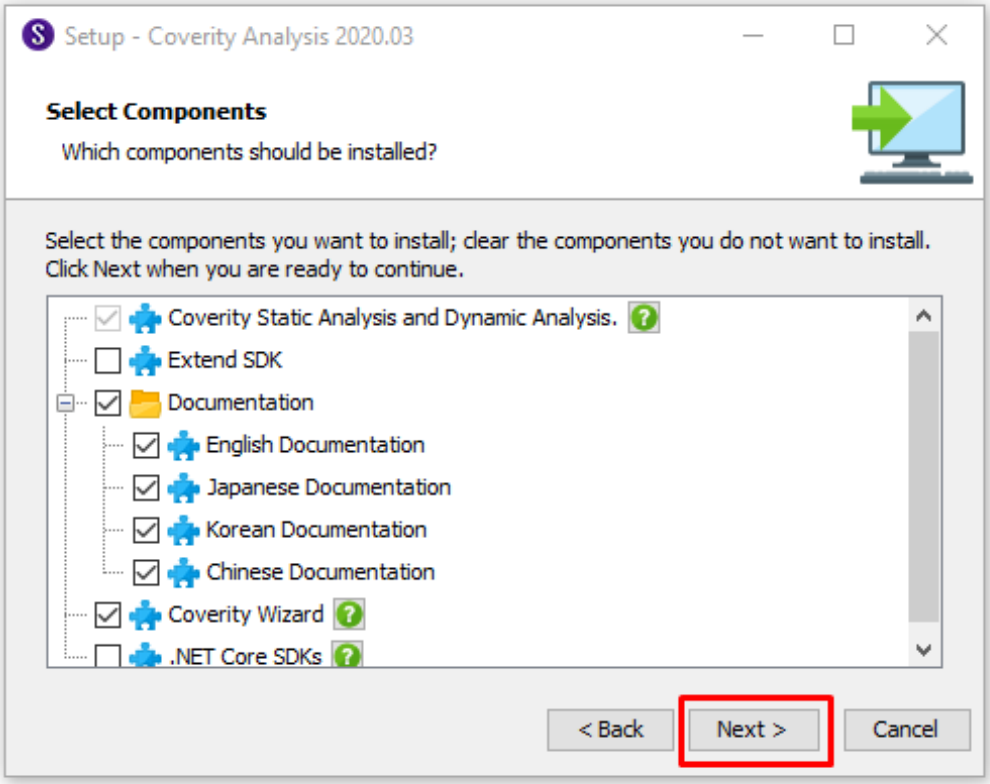
- **Step 4:** Select **I accept the agreement → Next**



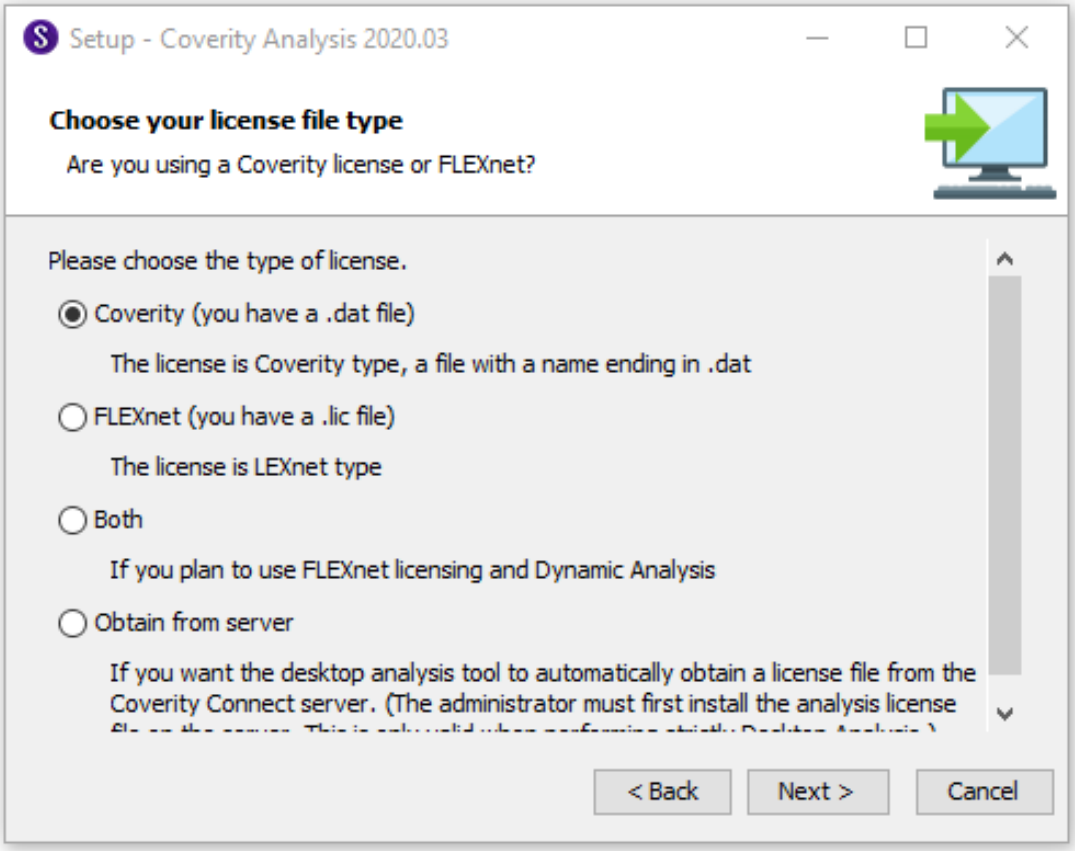
- **Step 5:** Choose **Coverity Home → Next**



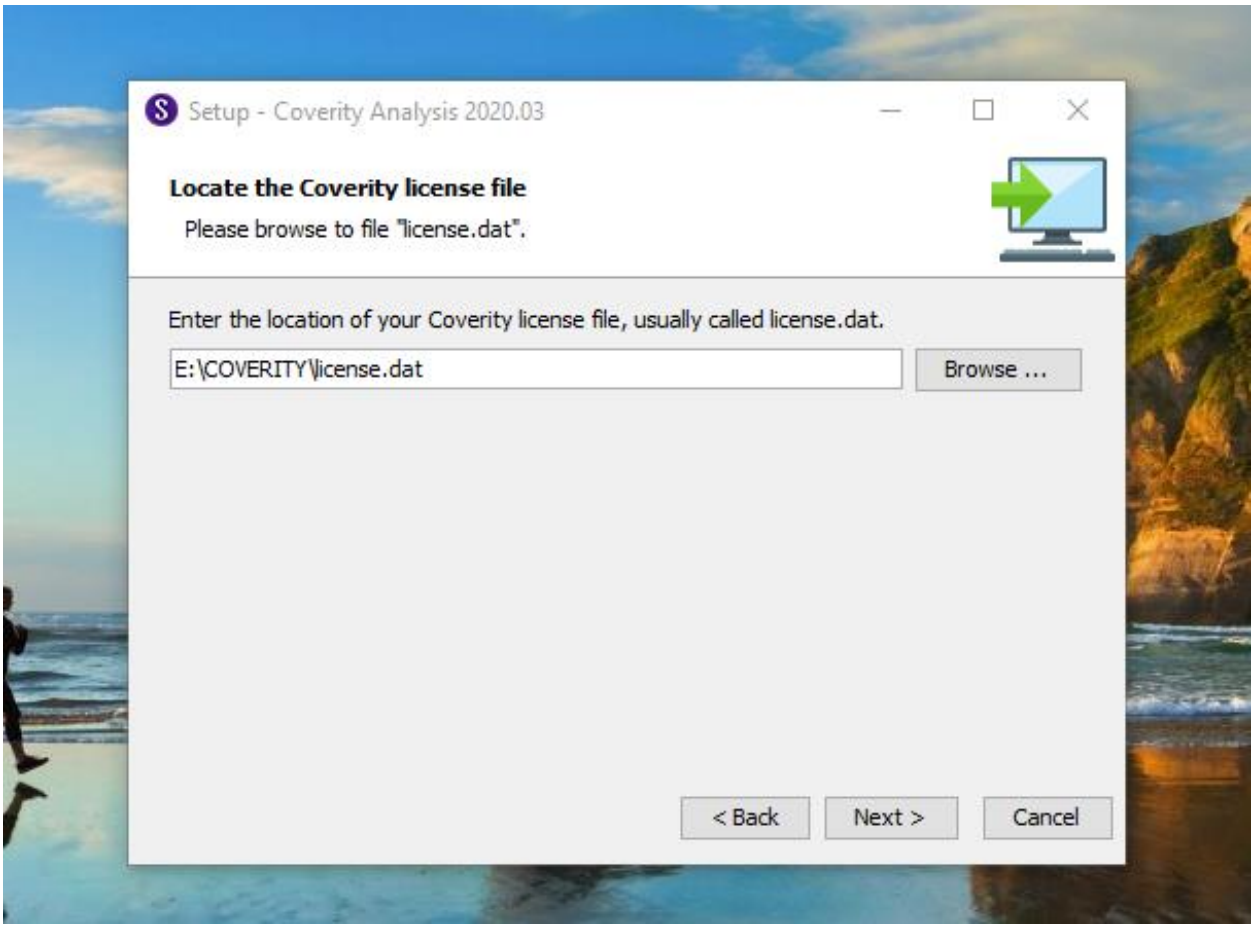
- **Step 6:** Click **Next**



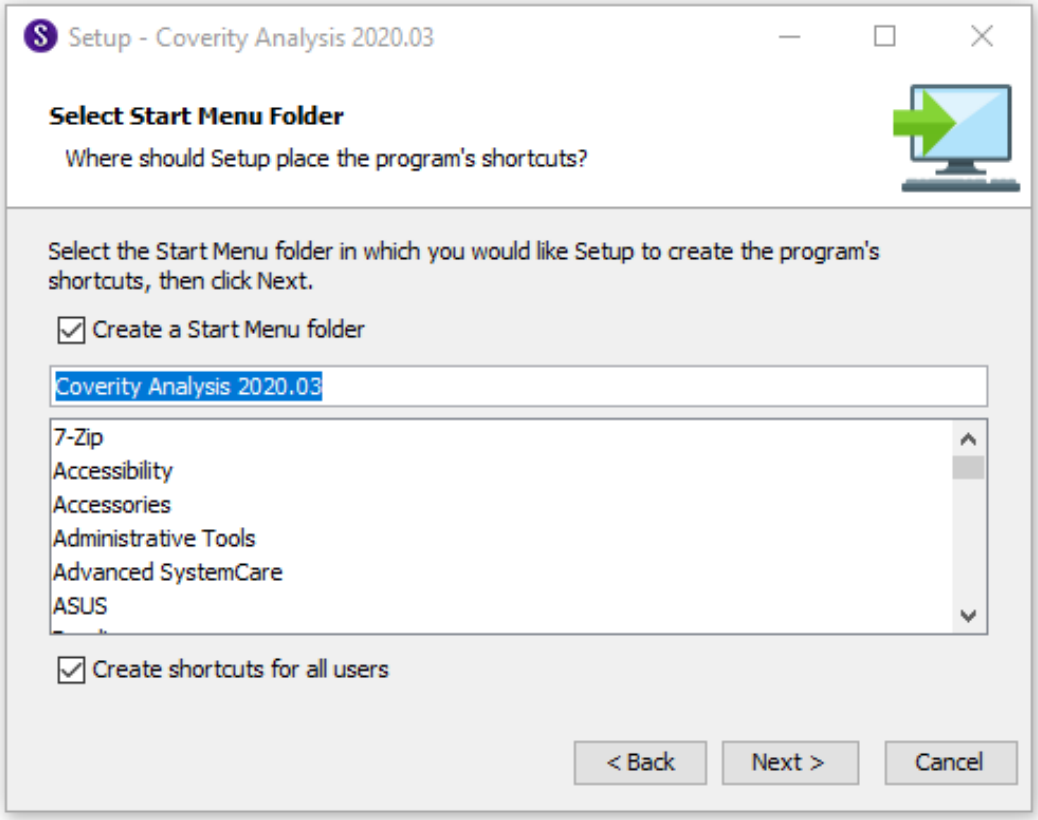
- **Step 7:** Select **Coverity** with the **.dat file**



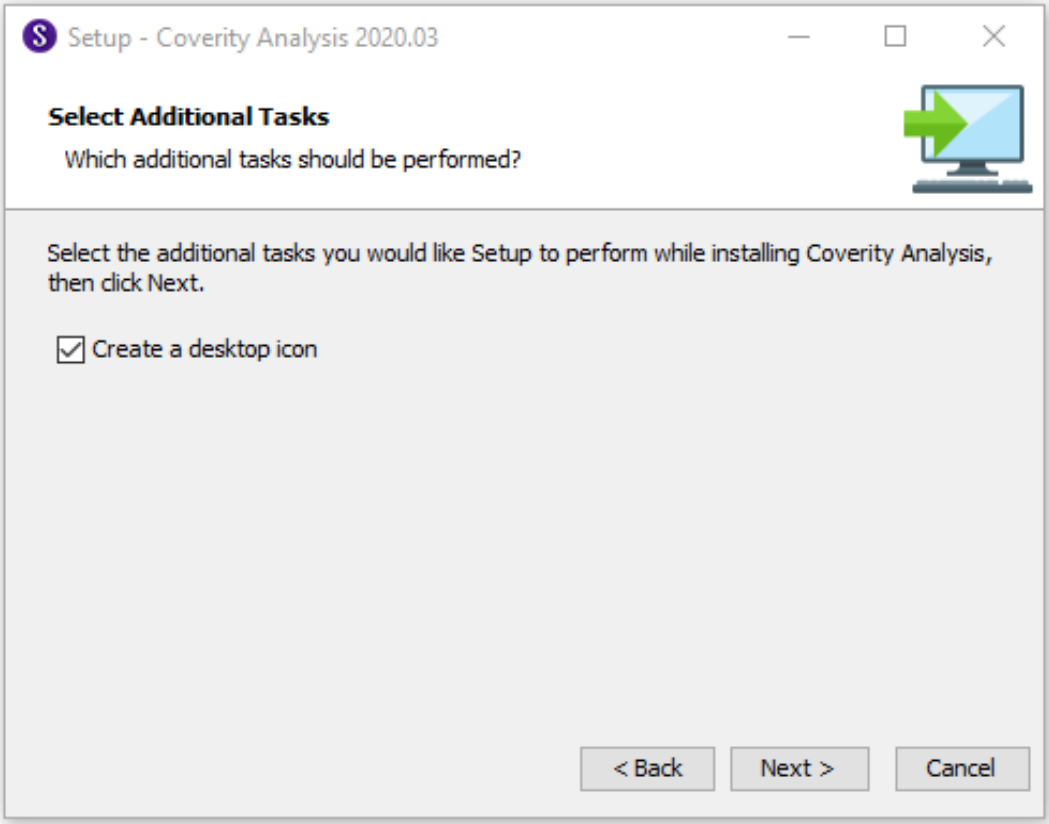
- **Step 8:** Select **the license.dat file that you download on step above**



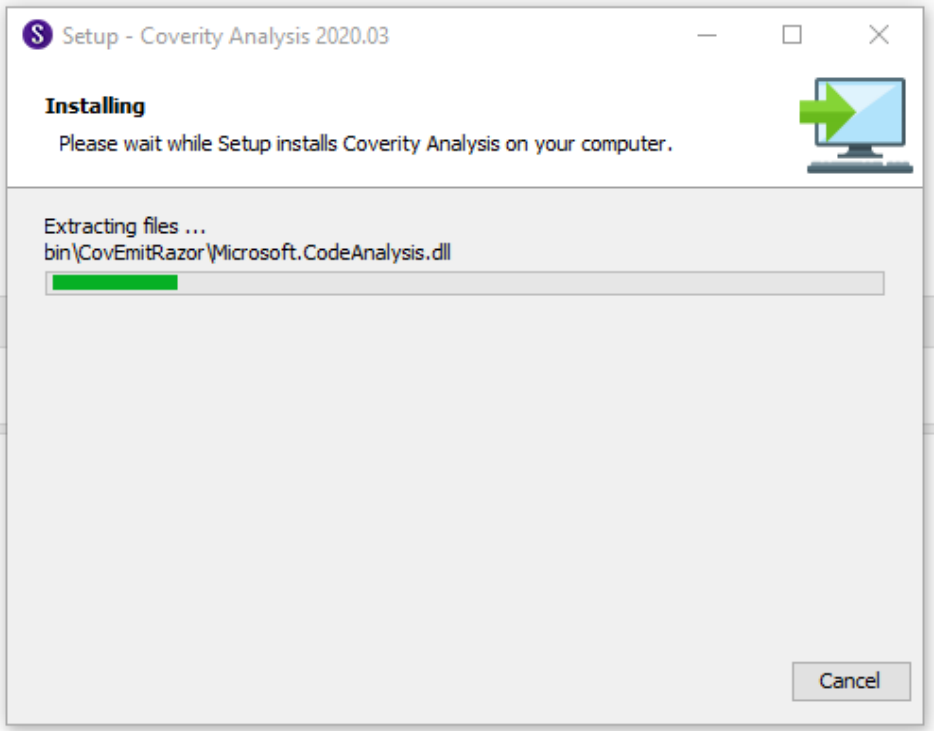
- **Step 9:** Click **Next**



- **Step 10:** Click **Next**



- **Step 11:** Wait for the process to finish



1.2. Install on Linux

Here are the steps to install **Coverity Analysis** on the Linux operating system.
After downloading the corresponding Coverity Analysis to your computer, we execute the following steps:

Step 1: Grant permission to execute and execute files
+ From the terminal, cd to the downloaded Coverity Analysis container and run **the** following command:

```
$ chmod +x cov-analysis-<OS>-<Version>.sh
```

+ File Execution

```
$ ./cov-analysis-<OS>-<Version>.sh
```

Example :

```
$ chmod +x cov-analysis-linux64-2020.03.sh
```

```
$ ./cov-analysis-linux64-2020.03.sh
```

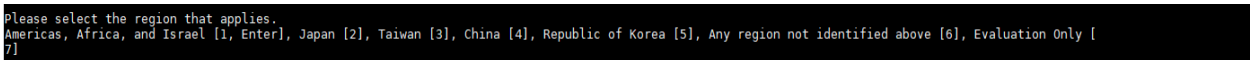
Where:

- ***OS:*** *is the operating system where you will install Coverity Analysis*
- ***Version****: is the Coverity Analysis version*

Note:

Step 2: Enter -> o

Step 3: Enter 6 →



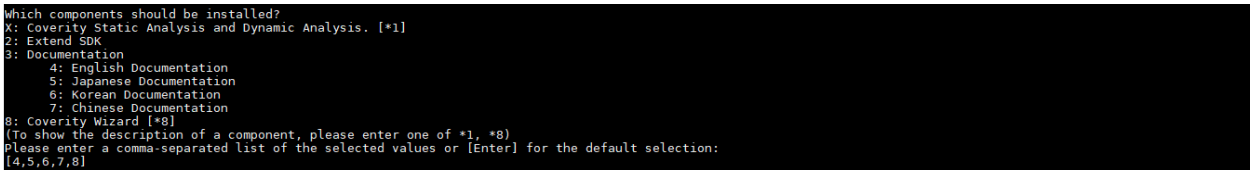
Step 4: Enter several times until the dialog box ends

Step 5: Enter 1 →**.**

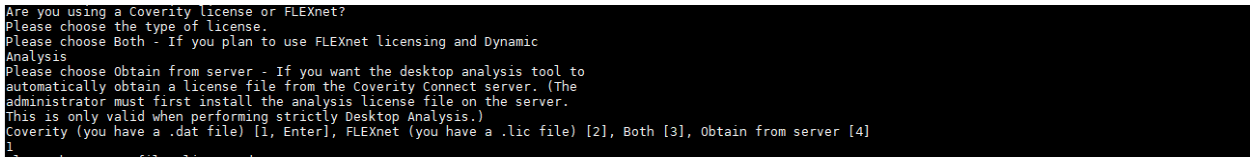
Step 6: Set where to save Coverity Home

Note: Enter then Coverity will use the Coverity recommend folder as the Coverity Home container.

Step 7: Enter -> x



Step 8: Enter -> `1`

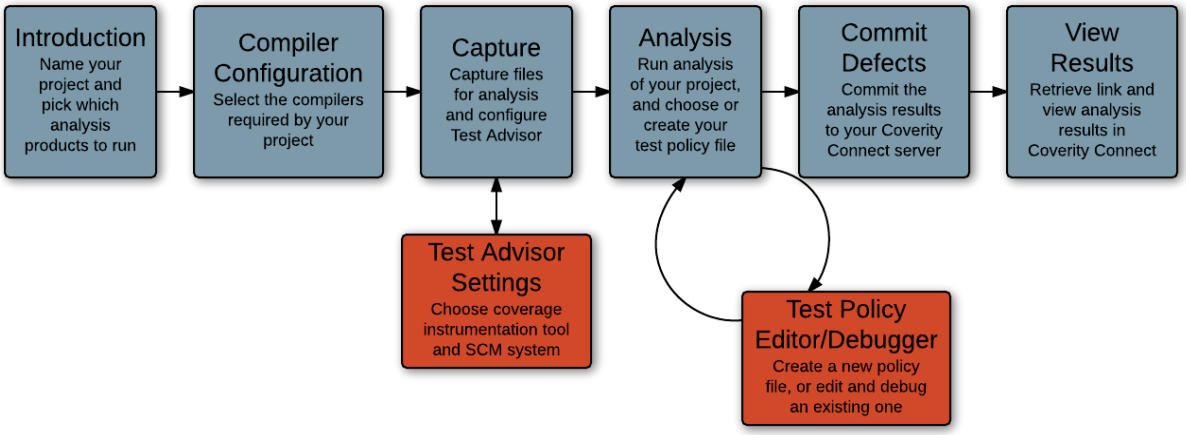


Step 9: Fill **`in .dat`** file

Step 10: Wait for the following announcement and end.

2. Analyze your source code with Coverity Analysis via the command line interface

Here is the basic workflow of Coverity, the red part showing for the separate steps for configuring **Test Advisor**



As you saw in the Coverity Analysis activity flow above, to analyze the source code of a project you need to perform the following steps:

- **(1**) Prepare the necessary requirements: license, Coverity Connect stream, and memory (RAM) for analysis.
- **(2)** Configure the compiler corresponding to the language used in the project
- **(3)** Capture problems in the source code for analysis
- **(4)** Conduct source code analysis
- **(5)** Upload the result to the Coverity server **(Optional, In this guideline, we only read the Coverity log)**
- **(6)** View results

In which steps (2), (**3**), (**4**), **(5)** you can do as follows:

2.1. Language and compiler configuration

Before running the analysis, you need to create a configuration for the compiler and/or language used. This statement needs to be used before you go through the build steps of the analysis process. `cov-configure`

Example:

You want to analyze the source code for the Embedded application running on the C language with ARM GCC compiler, and the compiler binary is named arm-none-eabi-gcc *(make sure you installed the arm-none-eabi-gcc (version 2018) compiler on your machine - contact the trainer if this step isn't clear with you).*

First, you need to perform the configuration to be able to analyze the C language (this is the command to configure the C language with ARM GCC compiler, for more information please click here: Synopsys Software Integrity Customer Community)

Go to the folder that contains your code, open the terminal (cmd), and run the command below:

```
cov-configure --template --compiler arm-none-eabi-gcc --comptype gcc -c
config.xml
```

After that, Coverity Analysis will configure in the config `config`.xml containing the appropriate configuration information for C language analysis

2.2. Build source code

After creating the necessary configurations, you need to build your source code.

```
cov-build --dir <path-to-output-folder> -c <path-to-config-file-xml> <build-
command>
```

With:

- "`--dir <path-to-output-folder>`" is the output folder of the build step above
- "`<path-to-config-file-xml>`" is a file that will contain the appropriate configuration information for analysis
- "<build-command>" is the build command

Example:

```
cov-build --dir covoutput -c config.xml arm-none-eabi-gcc example.c
```

You can also refer to the options for commands and in the Coverity 2021.9.0 Command Reference document `cov-build`

2.3. Source code analysis

After the Build step, you need to perform a source code analysis to find software errors. You can use commands to parse source code and scripts in languages supported by Coverity. `cov-analyze`

Example

After performing, run the build with using or and give the result in the temporary folder. To know if your source code contains security flaws. You need to run to analyze the captured results at the capture step. You can execute the command as below:

```
$ cov-analyze --dir <path-to-output-folder> --security --distrust-all --
enable MISRA_CAST --coding-standard-config <path-to-the-misra-config>
```

Where:

- "cov-output" is the folder where temporary analysis results are stored, which will be used in the analysis step.
- "<path-to-the-misra-config>" Path to the configuration file for the misra rules

Example:

- step 1: Create the misra configuration file "misrac2012-all.config" with the content below:

config

```
// Configuration for MISRA C-2012, rule categories: Required, Advisory,
Mandatory
{
    "version": "2.0",
    "standard": "misrac2012",
    "title": "MISRA C-2012 All Rules",
    "deviations": []
}
```

- Step 2: Open the terminal and run the command (should open the terminal in the folder that contains the example.c, config file)

```
cov-analyze --dir  covoutput --security --distrust-all --enable MISRA_CAST --
coding-standard-config misrac2012-all.config
```

The output of example as bellow:

```
Defects/Coding rule violations found : 32 Total
                                       1 MISRA C-2012 Directive 4.10
                                       1 MISRA C-2012 Directive 4.5
                                      10 MISRA C-2012 Directive 4.6
                                       1 MISRA C-2012 Rule 2.2
                                       4 MISRA C-2012 Rule 2.3
                                       1 MISRA C-2012 Rule 2.4
                                       2 MISRA C-2012 Rule 2.5
                                       1 MISRA C-2012 Rule 8.1
                                       4 MISRA C-2012 Rule 8.4
                                       2 MISRA C-2012 Rule 8.6
                                       5 MISRA C-2012 Rule 8.7
```

The detail logs for each violation can be found in "`cov-output`" folder.

Example: *covoutput\output\MISRA_C-2012_Directive_4.10.errors.xml*

```
<event>
<main>true</main>
<tag>misra_c_2012_directive_4_10_violation</tag>
<description>{CovLStrv2{{t{Header files should be guarded against multiple inclusion. (Cause:
{0})}{code in header outside guard}}}}</description>
<line>1</line>
</event>
```

Where:

- **<tag></tag>** contain name of violation rule
- **<description></ description >** contain description of the violation. {0} equal to {code in header outside guard}
- **<line></line>** contain line of code which has violation

You can also refer to the options for the command in the Coverity 2021.9.0 Command Reference document `cov-analyze`