

FAQs and Troubleshooting

In this section:

- [How can I get quick access to information about usage on the command line?](#)
- [Why do I get notified that Parasoft DTP Plugin is slowing Visual Studio 2017?](#)
- [Why are some Parasoft options not available in the Solution Explorer context menu in Visual Studio 2017?](#)
- [Why does Visual Studio 2015 not display code markers?](#)
- [How can I prevent my machine ID from floating?](#)
- [How can I work with C/C++test via proxy?](#)
- [Why are suppressions of some rules no longer available on DTP after C/C++test was upgraded to a newer version?](#)

How can I get quick access to information about usage on the command line?

Use the `-help` command line switch:

```
cpptestcli.exe -help
```

Why do I get notified that Parasoft DTP Plugin is slowing Visual Studio 2017?

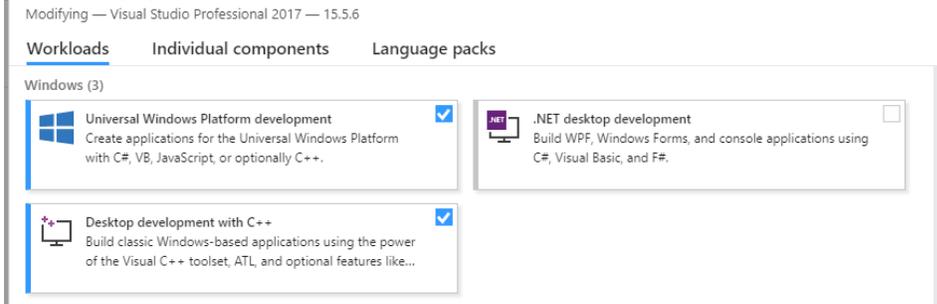
Visual Studio 2017 introduced new performance management options to monitor and manage extensions that can affect startup time. As a result, a pop-up alert may appear when Parasoft DTP Plugin is installed to indicate that Visual Studio startup time has increased. You can click the Manage Visual Studio Performance link in the pop-up window to review the startup times and disable displaying the alert.

Why are some Parasoft options not available in the Solution Explorer context menu in Visual Studio 2017?

To have access to all Parasoft options in the Solution Explorer context menu, ensure that one of the following workloads is installed with your Visual Studio 2017:

- *Desktop development with C++*
- *Universal Windows Platform development* or *.NET desktop development*

1. Run the Visual Studio Installer.
2. Select the appropriate workloads when prompted.



Why does Visual Studio 2015 not display code markers?

If the Parasoft DTP Plugin for Visual Studio is reinstalled, Visual Studio 2015 may fail to display code markers that indicate analysis findings or code coverage in the editor. This is caused by a Visual Studio known issue (see [MEFCache bug breaks text adornments in any package](#) for details).

To ensure that code markers are properly shown, clear the Visual Studio MEF cache each time you reinstall or updated the Parasoft DTP Plugin. Use one of the following options:

- Use the Clear MEF Component Cache extension that will automatically clear the MEF cache (download [here](#)).
- Manually delete the cache directory. Depending on your installation details, the cache directory may be available in the following locations:
 - If the Plugin has been installed with the default options: `%localappdata%\Microsoft\VisualStudio\14.0\ComponentModelCache`
 - If the Visual Studio registry suffix has been specified with the `<SUFFIX>` parameter during installation: `%localappdata%\Microsoft\VisualStudio\14.0<suffix_name>\ComponentModelCache`

How can I prevent my machine ID from floating?

Changes in the network environment may affect the interface that is used to compute your machine ID and result in machine ID instability. You can use the `PARASOFT_SUPPORT_NET_INTERFACES` environment variable to specify a stable interface and prevent the machine ID from floating.

1. Set up the **PARASOFT_SUPPORT_NET_INTERFACES** environment variable.
2. Set the variable value to a stable Ethernet network interface. Do not use virtual, temporary or loopback interfaces.
 - On Windows: Set the value to the MAC address of your network card. You can use the `ipconfig -all` command to obtain the address. For example:

```
SET PARASOFT_SUPPORT_NET_INTERFACES=00-10-D9-27-AC-85
```

- On Linux: Set the value to one of the network interfaces from the "inet" or "inet6" family. You can use the `ifconfig` command to obtain the list of available interfaces. For example:

```
export PARASOFT_SUPPORT_NET_INTERFACES=eth1
```

If the problem persists, you can obtain diagnostic information by setting up the environment variable **PARASOFT_DEBUG_NET_INTERFACES** and setting its value to true. This will print to the standard output the checking procedure that can be shared with technical support, as well as the interface that is used to compute your machine ID. The interface will be marked with the [SELECTED] prefix.

How can I work with C/C++test via proxy?

Typically, if you connect through a proxy server, you need to configure the connection by passing protocol-specific system properties to the JVM – using the `-D` command line option.

To work with C/C++test, ensure that the system properties for the HTTPS protocol (`https.proxyHost` and `https.proxyPort`) are configured. Your command line may resemble the following:

```
java -Dhttps.proxyHost=myserver.example.com -Dhttps.proxyPort=8080
```

-  If you use C/C++test on desktop with Eclipse, the proxy settings are automatically detected and do not need to be configured in the command line.
-  The proxy mode is not supported for Visual Studio.

Why are suppressions of some rules no longer available on DTP after C/C++test was upgraded to a newer version?

Suppressions associated with rules whose messages changed between releases may not be available on DTP and the rules must be re-suppressed.

-  You can restore legacy messages for BD category rules in version 10.4.1 and later by configuring the following settings in your `.properties` file:
`flowanalysis.legacy.messages.for.<rule_ID>=true.`

For example:

```
flowanalysis.legacy.messages.for.BD.PB.ARRAY=true  
flowanalysis.legacy.messages.for.BD.PB.ZERO=true
```

To successfully restore the legacy messages, ensure that you remove the `.cpptest` folder that contains the local C/C++test cache data from the workspace (the custom workspace or the workspace you configured with the `-workspace` command line switch).