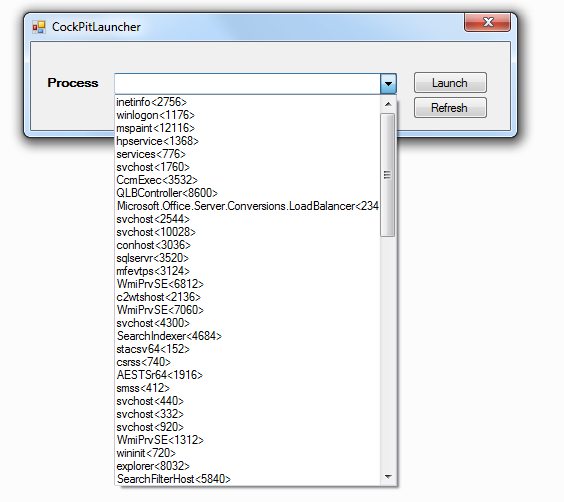
**DotNetProcessViewer User Document**

DotNetProcessViewer is implemented in two parts Cockpit (executable can with x86 or x64) UI and Debug Engine (core logic of Debugger).

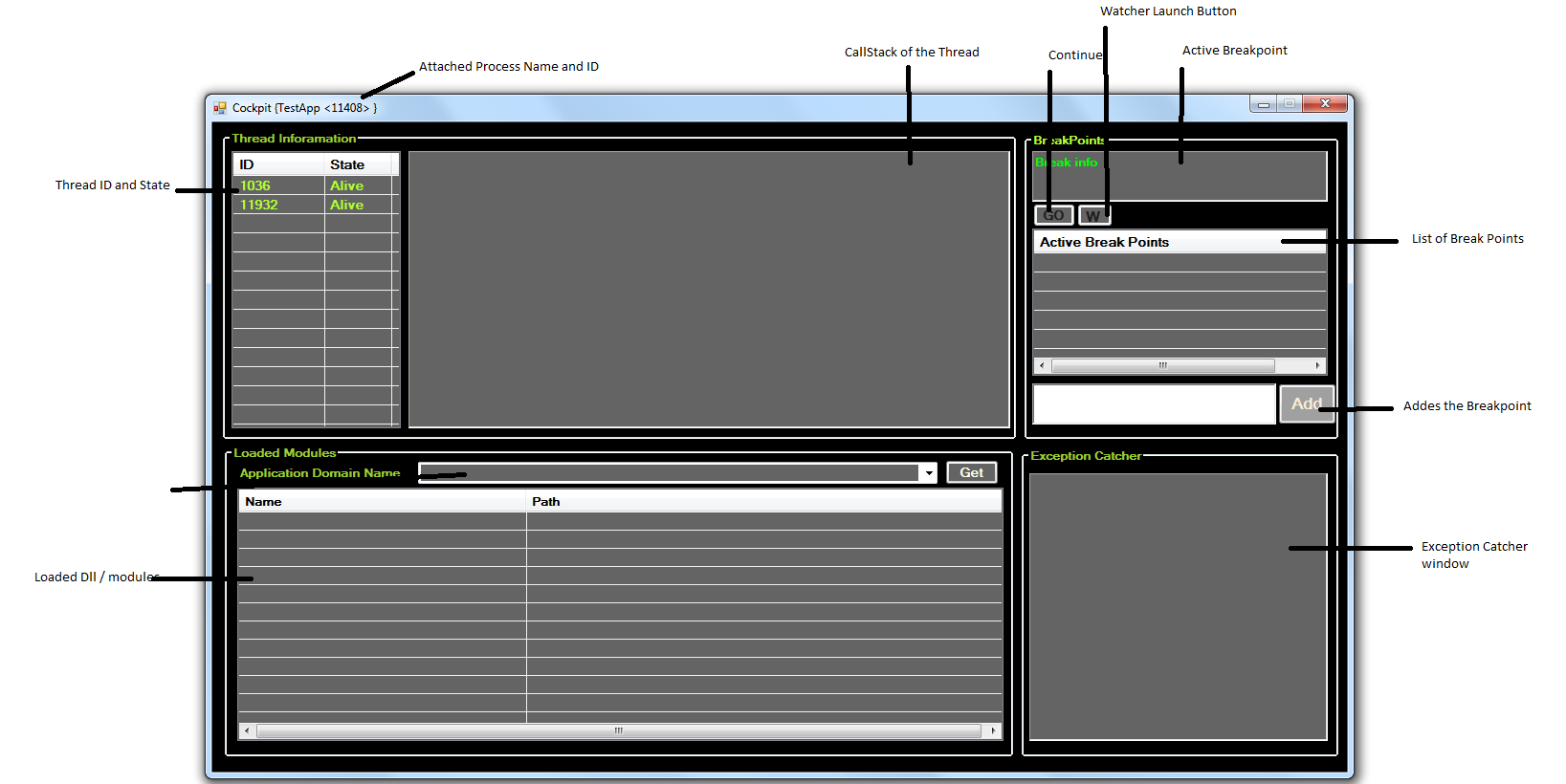
Once user starts the Cockpit it display a list of process (if Cockpit is x86 then it displays all x86 process else it display all x64 process)

Refresh button will all get the latest running process and updates

Launch button will attach the target process as Debuggee

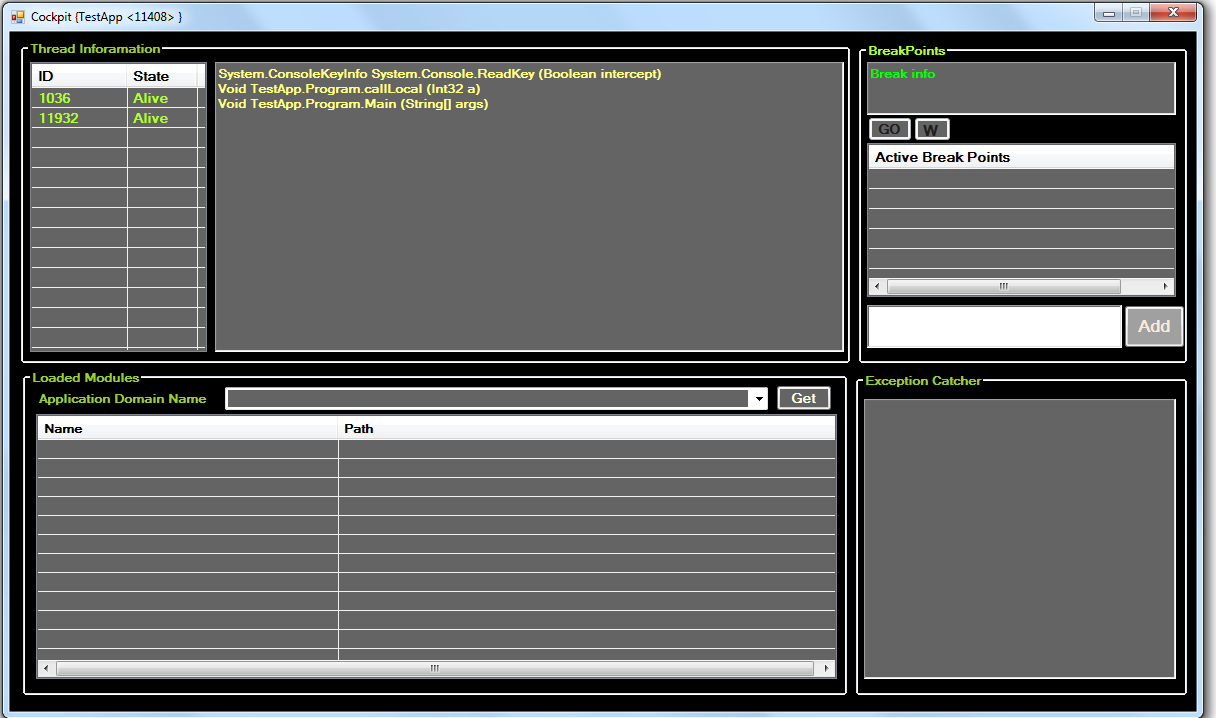


**Cockpit View:-**



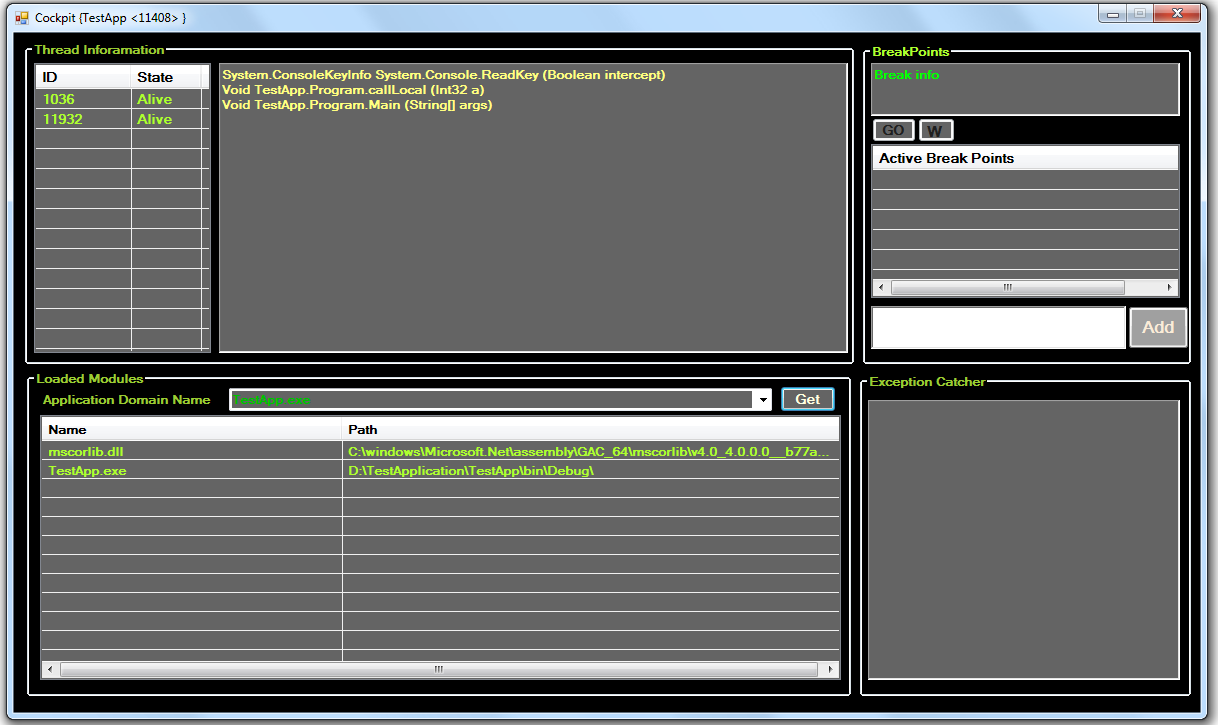
**Display Call stack**

Once it launches it displays list of running thread by double click on the thread id row we find the thread call stack



**List the Loaded Modules**

User find the list of loaded modules by select the specific appdomain and click on Get



Exception Catcher display all exception happened at runtime

<< image need to be added>>

**Breakpoint**:-

Breakpoint structure ApplicationName!FullClassname#MethodName

Example Source of TestApp.exe

namespace TestApp

{

class Class1

{

public void MyTest(int myindex){

//Source

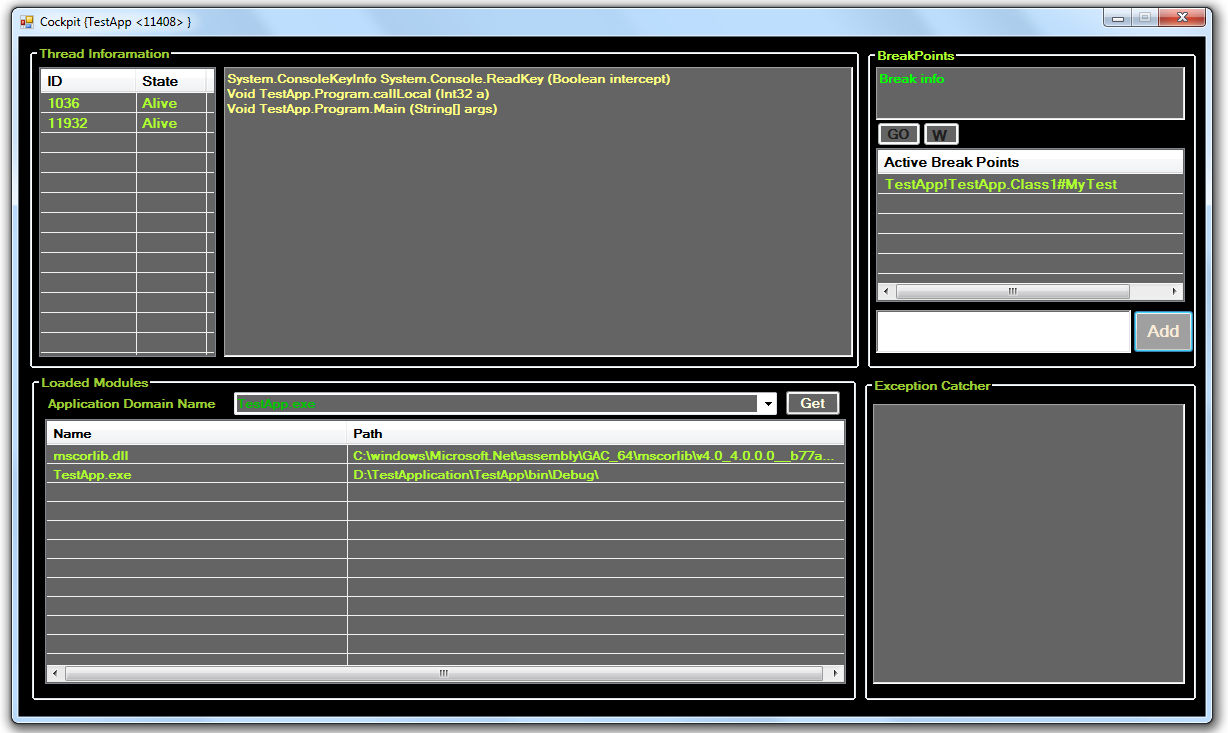
}

}

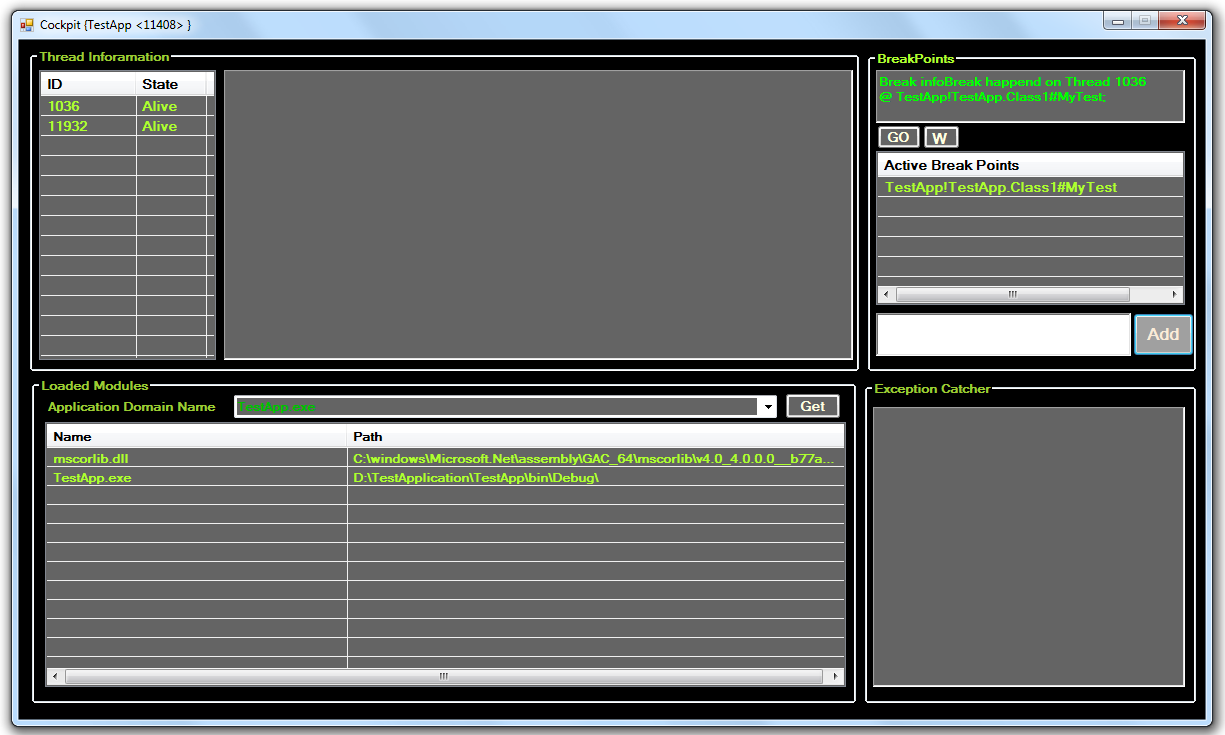
}

Breakpoint: - TestApp**!TestApp.Class1#MyTest**

Enter the breakpoint text as above format and click on Add, it will validate and add it to the list

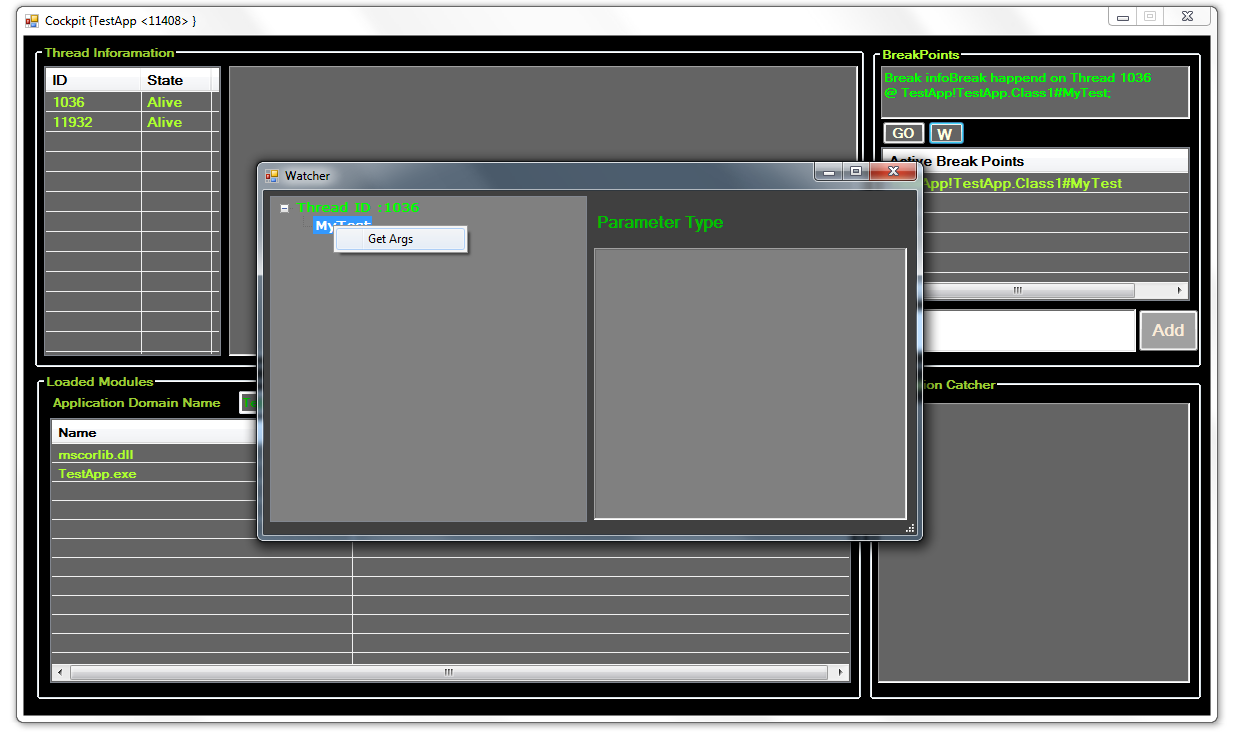


Once Debugee hit the Breakpoint source it will get active and GO and W buttons are enabled

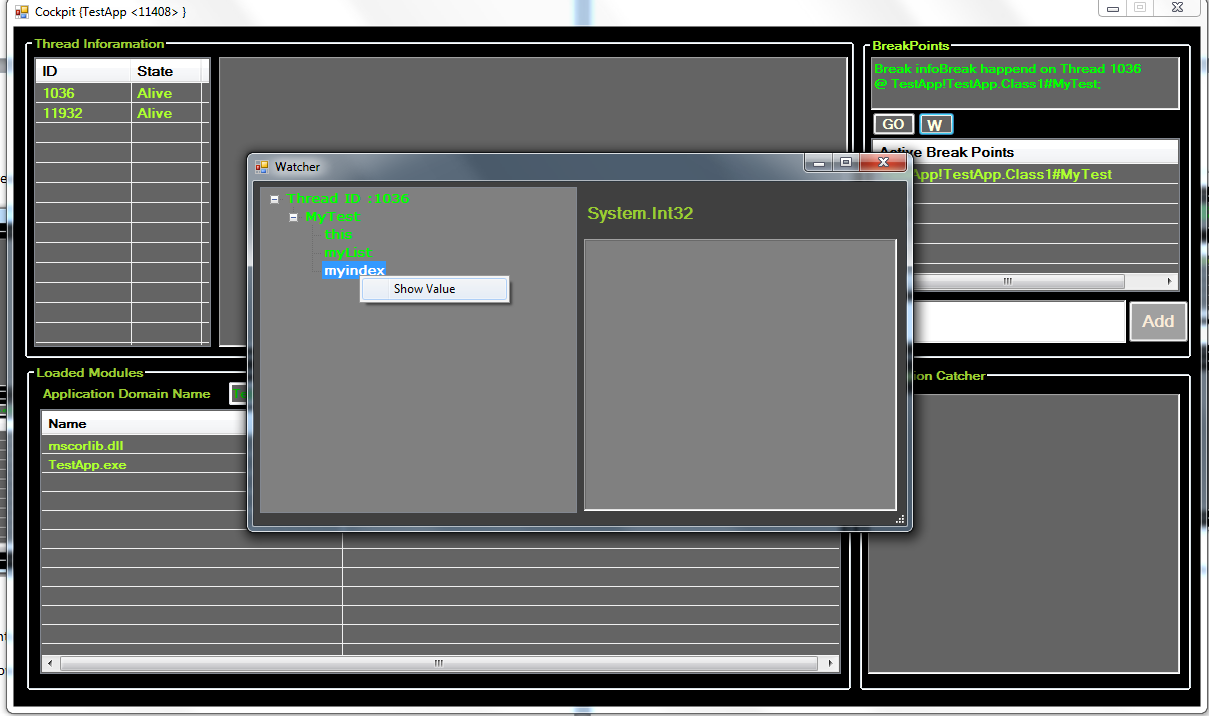


To release the breakpoint user can click **Go** Button so execution proceeds further

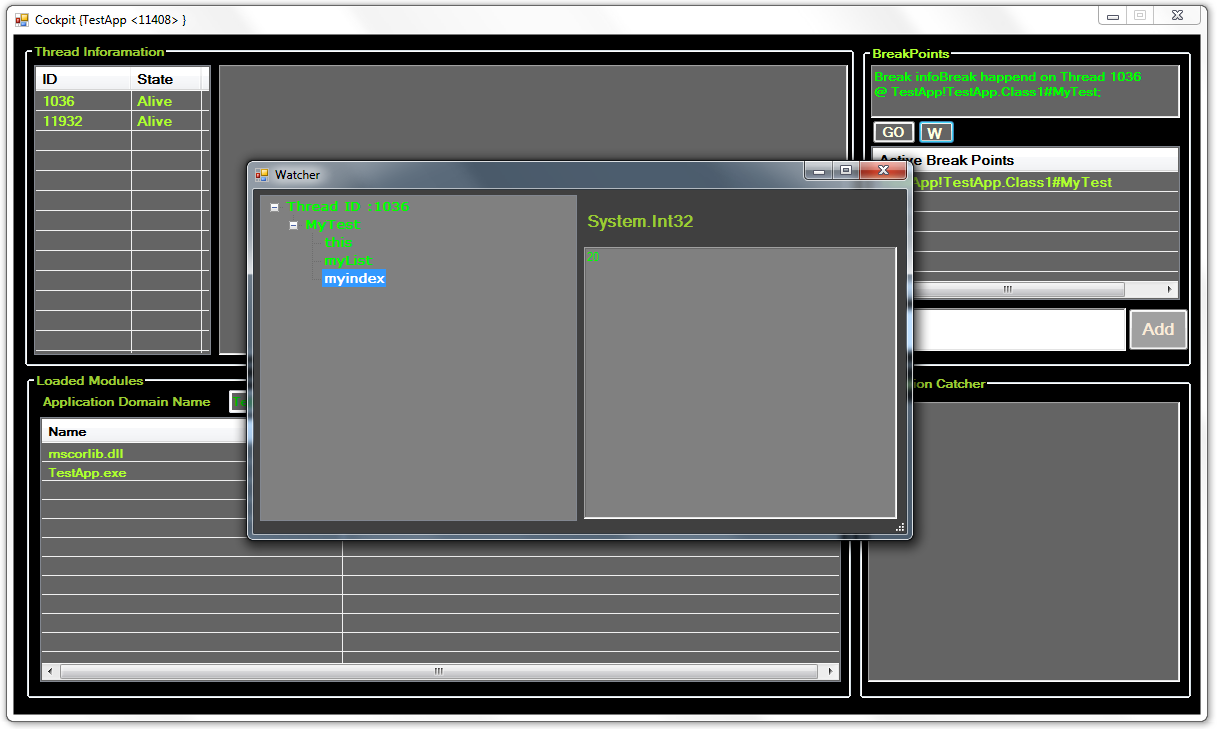
To launch watcher window user can click on **W** Button



Watcher window shows the method name and user can get the method parameters by clicking on the **GetArgs** menu item from the popup



To get the parameter / arguments value user can select the item and click show Value menu item.



Value will be displayed in the right hand side.

Once user close the Cockpit window it will safely detach from the target process