

CH12\_001

EXERCISE

# USING DEBUGGERS

## Exercise Files

Starter – "Kit/gpgt/server/scripts/gpgt/chapter12/exercise001.cs"

Answers – n/a

## Exercise Mission

Chapter 12: "001\_DebuggingTorque: Using Debuggers"

## Special Setup

*This exercise uses the Torsion IDE. If you do not have this application installed, please install it at this time. If you do not own Torsion, you can pick up the trial-version on the GarageGames site.*

*(<http://www.garagegames.com/products/106/>)*

*Please be aware that Torsion is a Windows ONLY tool, so if you are an OSX user, you will need to use an alternative Debugger/IDE to do this exercise. Additionally, not all steps in this exercise will match your debugger/IDE exactly.*

## Synopsis

In this exercise, we will test your understanding of the basic debugger topics that were discussed in chapter 12. To accomplish this, we will be using the Torsion IDE.

## Prerequisites

1. *ch1\_001.pdf "Using The Kit"*

## Exercises

1. *Loading The Kit in Torsion (pg 2)*
2. *Basic Breakpoints and Stepping (pg 4)*
3. *Conditional Breakpoints and Watches (pg 7)*

# USING DEBUGGERS

## 1 Loading The Kit in Torsion

**Goal:** Get the Kit loaded into Torsion so that we can start this exercise.

**Steps:**

The Kit comes with a predefined Torsion project that you can simply load into Torsion once you have installed and started it. To do this, follow these simple steps.

1. Install Torsion. You should have done this above in the "Special Setup", but I'm saying it again, just in case.
2. Click File->Open->Project (figure 1).
3. In the dialog that appears (figure 2), please locate "gpgt/Kit" (as per your installation of the accompanying disk) and load the file "gpgt.torsion".
4. When this file is loaded, the project browser pane (on the left) should look something like figure 3.

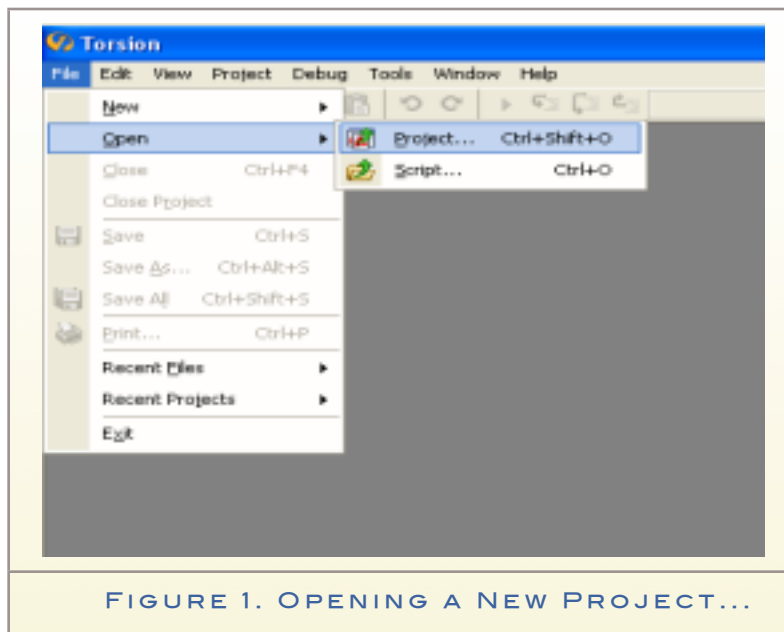
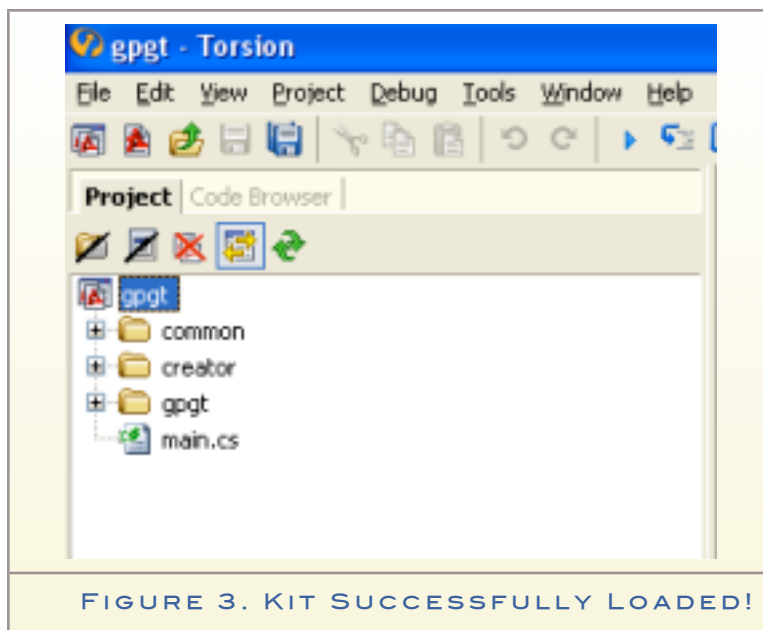
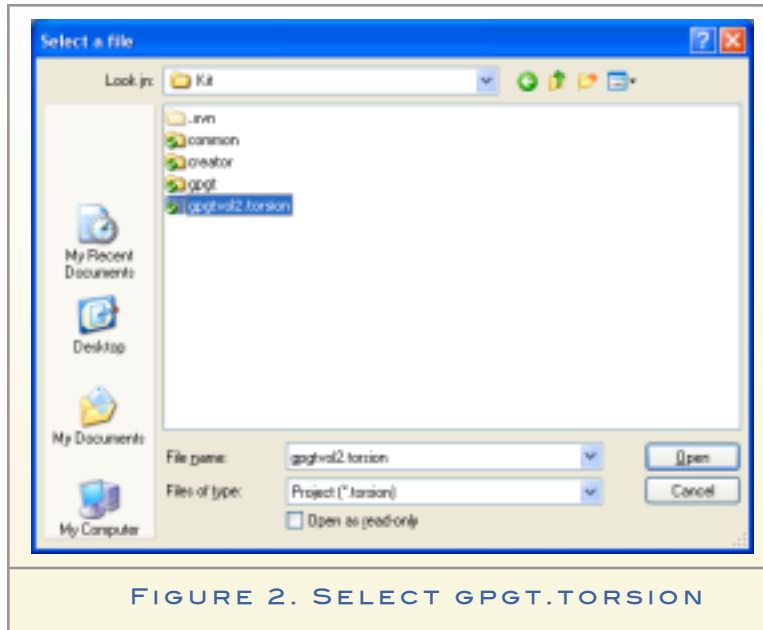


FIGURE 1. OPENING A NEW PROJECT...

## USING DEBUGGERS



# USING DEBUGGERS

## 2 Basic Breakpoints and Stepping

**Goals:** This part of the exercise has several goals, including:

1. Demonstrate the ability to set a static breakpoint(s).
2. Demonstrate the ability to step.
3. Demonstrate the ability to re-start execution during a loop for 'faster' stepping.

**Starter Code:** For this part of the exercise, you are provided with two different functions, `debugTest0()`, and `debugTest1()`.

```
function debugTest0()
{
    %local0  = 100;
    $global0 = "Torque Rocks!";
    echo("In debugTest0 %local0 == ", %local0 );
    echo("In debugTest0 $global0 == ", $global0 );
}

function debugTest1()
{
    for(%i = 0x1; % < 0x10; %i = %i << 1 )
    {
        switch( %i )
        {
            case 0x1:
                echo("%i == 1");
            case 0x2:
                echo("%i == 2");
            case 0x3:
                echo("%i == 3");
            case 0x4:
                echo("%i == 4");
            case 0x8:
                echo("%i == 8");
        }
    }
}
```

It is your task to use Torsion and these functions to demonstrate the ability to set breakpoints, step, and restart. At this time, please be sure you have the starter file loaded (in Torsion) and then continue with the exercise steps below.

## USING DEBUGGERS

### debugTest0()

Let's test your ability to set breakpoints and to step.

#### Steps:

1. Please set a static breakpoint on the line (in debugTest0), where the local variable is set.
2. Now, click the run button on the top toolbar, or press F5.
3. Once the Kit has started, please start the mission associated with this exercise.
4. Once the mission is loaded, please open the console (~) and type "debugTest0();".
5. When you press return, the function will execute and Torsion should come to the front of the screen. At this point, the kit is halted and Torsion has stopped the function at the breakpoint.
6. Now, please step through the code line by line.
7. Please answer questions 1..3 below.
8. Now resume running by pressing F5.

#### Questions:

1. What key did you use to step, line by line?
2. How many times did you press this key before the function completed and a new file was loaded?
3. What new file was loaded?

### debugTest1()

Let's test your ability to use breakpoints to solve a simple coding problem. The problem we are trying to solve involves the question, "What case statement(s) in our script are not being executed?" Yes, we could determine this by inspection, but sometimes you need to use a debugger instead. This part of the exercises demonstrates the basic steps you would take to do this.

#### Steps:

1. Please set a static breakpoint on each of the echo statements in debugTest1().
2. At this point, the console should still be open from the last exercise. If it is not, please re-open the console (~).
3. Now type "debugTest1();".
4. Please answer question 1 below.
5. Please resume the kit by pressing F5 and every time Torsion hits a break point, please write down the line number that Torsion stops on. Please do this until Torsion doesn't stop the kit anymore.
6. Please answer questions 2 and 3 below.
7. Please stop the kit by pressing SHIFT+F5.



## USING DEBUGGERS

**Questions:**

1. Which line (in the script) did Torsion stop the code on?
2. How many lines did Torsion stop on while running debugTest1()?
3. What line(s) (breakpoint) did Torsion NOT stop on?

# USING DEBUGGERS

## 3 Conditional Breakpoints and Watches

**Goal:** Demonstrate the ability to set up and use conditional breakpoints, as well as the ability to use the watch features to inspect values in our code as we debug.

**Starter Code:** For this part of the exercise, you are provided with one function, `debugTest2()`. This function loops from 0 to 100 and calculates the logarithm of that value.

```
function debugTest2()
{
    for(%i = 0x0; %i < 100; %i++ )
    {
        %j = mLog( i );
    }
}
```

### Steps:

1. Please set a conditional breakpoint on the line "`%j = mLog( i );`" such that the break point will only trigger the 32<sup>nd</sup> time the line is about to execute.
2. Now, please start the kit (F5) and run the mission associated with this exercise.
3. When the mission is loaded, please open the console (~) and run "`debugTest2()`".
4. Answer questions 1 through 3 below.
5. Please resume by pressing F5.
6. Now, please add the local variables `%i` and `%j` to the watch pane.
7. Please modify the breakpoint to stop when `%j` is greater than 4.0.
8. Now re-run "`debugTest2()`;" and answer the remaining questions below.

### Questions:

1. What conditional setting did you use for this breakpoint?
2. Before this line executes, what is the value in `%i` and `%j`? After?
3. Since you were not instructed to use a watch variable yet, what feature did you use to inspect the value of `%j`?
4. What conditional setting did you use for this breakpoint?
5. When the breakpoint triggers, what is the value of `%i`? `%j`?
6. What happens if you press F5?