

גירסה 1.01 – 24.3.2010
גירסה 1.00 – 4.4.2003

עבודה עם קבצי טקסט בשפת C#

מסמך זה הורד מהאתר [http://www.underwar.co.il/](http://www.underwar.co.il).
אין להפיץ מסמך זה במדיה כלשהי, ללא אישור מפורש מאת המחבר.
מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר עשה את מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר.

כל הזכויות שמורות לניר אדר

Nir Adar

Email: nir@underwar.co.il

Home Page: <http://www.underwar.co.il>

אנא שלחו תיקונים והערות אל המחבר.

הקדמה

קבצי טקסט הינם קבצים בפורמט מיוחד – הם קריאים וניתנים להבנה גם על ידי בני אדם וגם על ידי תוכנות מחשב.

עקב חשיבותם, סביבת .Net מכילה מחלקות עזר המאפשרות לעבוד עם קבצי טקסט בקלות

עבודה עם קבצי טקסט מורכבת מהשלבים הבאים:

1. פתיחת הקובץ.
2. קריאה/כתיבה לקובץ.
3. סגירת הקובץ.

שלבים אלו מוכרים לכל מי שעבד עם קבצים בשפות עיליות כגון C/C++. צורת העבודה עם קבצים בשפות אלו דומה לצורת העבודה עם קבצים בשפת C#.

יצירת קובץ טקסט

נציג את תהליך יצירת קובץ וכתיבה לתוכו בעזרת דוגמא.
נביט בקוד הבא:

```
using System;
using System.IO;

namespace WriteFileExample
{
    class TextFileWriter
    {
        static void Main(string[] args)
        {
            // create a writer and open the file
            TextWriter tw = new StreamWriter("my_file.txt");

            // write a line of text to the file
            tw.WriteLine(DateTime.Now);

            // close the stream
            tw.Close();
        }
    }
}
```

תוכנית 1

תוכנית זו תיצור קובץ כאשר נריץ אותה. בספרייה בה התוכנית ממוקמת ייוצר קובץ בשם my_file.txt. אם נביט בתוכן הקובץ, נראה כי הוא מכיל תאריך בפורמט הבא: 4/4/2003 3:34:51 AM. תאריך זה הוא התאריך בו הורצה התוכנית בפעם האחרונה.

כתיבה וקריאה של טקסט ב- .Net מתבצעות בעזרת המחלקות StreamWriter ו- StreamReader. מחלקות אלו נורשות מהמחלקות האבסטרקטיות TextWriter ו- TextReader.

המשימה הראשונה שתוכנית 1 מבצעת היא פתיחת קובץ לכתובה. התוכנית מבצעת זאת על ידי יצירת אובייקט מסוג `StreamWriter`. האובייקט מושם בתוך משתנה מסוג `TextWriter`. באותה מידה היינו יכולים לשים את האובייקט החדש גם בתוך משתנה מסוג `StreamWriter`. כאשר יצרנו את האובייקט, השתמשנו בבנאי (Constructor) שלו המקבל פרמטר אחד בלבד – מחרוזת המציינת את שם הקובץ הרצוי. מלבד בנאי זה, למחלקה 6 בנאים נוספים. הבנאי יוצר קובץ ריק בשם המבוקש. במידה ולא קיים קובץ כזה ייוצר קובץ חדש. אם לעומת זאת קובץ בשם המבוקש כבר היה קיים, תוכנו המקורי יימחק.

בהמשך הקוד אנו כותבים את התאריך הנוכחי אל תוך הקובץ. המחלקה `TextWriter` מספקת מגוון פונקציות לגישה אל הקובץ. שיטת העבודה עם `TextWriter` מזכירה במידה רבה את צורה העבודה עם `Console`.

כתיבת התאריך נעשית על ידי שימוש במאפיין הסטטי `Now` של המחלקה `DateTime`.

אחרי שסיימנו את הכתיבה לקובץ, יש לסגור את הקובץ כדי לדאוג שהכתיבה בעצם תתבצע. אנו עושים זאת בשורה השלישית, על ידי קריאה לשיטה `Close()`.

קריאת קובץ טקסט

נראה כעת כיצד קוראים נתונים מקובץ. נכתוב תוכנית שתפתח את הקובץ שיצרנו בתוכנית 1, תקרא את השורה שכתבנו לתוכו ותציג אותו על המסך.

```
using System;
using System.IO;

namespace ReadFileExample
{
    class TextFileReader
    {
        static void Main(string[] args)
        {
            // create reader & open file
            TextReader tr = new StreamReader("my_file.txt");

            // read a line of text
            Console.WriteLine(tr.ReadLine());

            // close the stream
            tr.Close();
        }
    }
}
```

תוכנית 2

פתיחת הקובץ נעשית בצורה דומה לצורה בה פתחנו את הקובץ לקריאה, מלבד העובדה שבקריאה אנו משתמשים במחלקות `TextReader` ו-`StreamReader`.

השורה השנייה בתוכנית קוראת שורה אחת מהקובץ, ואז מדפיסה אותה אל המסך בעזרת אובייקט Console-ה.
אנו משתמשים בפונקציה ReadLine() על מנת לקרוא את השורה שכתבנו.
ל-TextReader פונקציות שימושיות נוספות: Read() מאפשרת קריאת הקלט תו תו, במקרה קריאה של שורה שלמה. Peek() מאפשרת לנו לבדוק מהו התו הבא, מבלי לשלוף אותו מערוץ הקובץ.
לאחר שאנו מסיימים לקרוא נתונים מהקובץ, אנו סוגרים אותו בעזרת הפונקציה Close().

EOF