DigitObliterator

User Manual

Contents

1 Installation process

1.1 Requirements

To ensure that the calculator works properly, it is recommended that you have the **.NET 7 Desktop Runtime** installed on your computer. You can download and install it from the Microsoft website.

Please note that DigitObliterator is only guaranteed to work on the following operating systems:

- Windows 11 64-bit
- Windows 10 64-bit version 1941 or higher

If you have one of these operating systems, you should be able to run the calculator without any issues. However, if you have a different operating system or version, DigitObliterator may not work correctly.

1.2 Installation setup

To install the DigitObliterator, you first need to extract the installation files from an archive in which they may be located. Before starting the installation process, it is necessary to add the application certificate as a trusted one.

Follow steps included below:

1. Right-click on the .msix file and choose Properties. The name of the file is as follows:

   `DigitObliterator_[version]_[isa].msix`
2. Select the Digital Signatures tab.
3. Choose the certificate then press Details.
4. In the Properties pane of the MSIX file with the Digital Signatures tab selected, select View Certificate.
5. Select Install Certificate....
6. Choose Local Machine then select Next.
7. If you're prompted by User Account Control to "Do you want to allow this app to make changes to your device?", select Yes.
8. In the Certificate Import Wizard window, select Place all certificates in the following store.
9. Select Browse... and then choose the Trusted People store. Select OK to close the dialog.
10. Certificate Import Wizard window is shown while selecting the Trusted People store.
11. Select Next and then Finish. You should see a dialog that says: "The import was successful."
12. Select OK on any window opened as part of this process, to close them all.

1.3 Installation

To install, run the previously mentioned `.msix` file and the installation wizard will guide you through the process. Once installation is complete, you should be able to find the application by searching for "DigitObliterator" in your Start menu.

2 Uninstallation

To completely uninstall the application, you can use the standard uninstall function provided by the Windows operating system. Note that the instruction below may slightly vary based on your operating system version.

Uninstall from the Start menu

1. Select Start and look for the app or program in the list shown.
2. Press and hold (or right-click) on the app, then select Uninstall.

Uninstall from the Settings page

1. Select Start, then select Settings Apps Apps & features.
2. Select the app you want to remove, and then select Uninstall.

Uninstall from the Control Panel (for programs)

1. In the search box on the taskbar, type Control Panel and select it from the results.
2. Select Programs ¿ Programs and Features.
3. Press and hold (or right-click) on the program you want to remove and select Uninstall or Uninstall/Change. Then follow the directions on the screen.

3 Manual Installation

The application is built using MAUI that uses packaged compilation using MSIX. To change the the target to unpackaged application please follow the steps bellow. Note that unpackaged publishing is not officially supported and may result in unexpected behaviour.

3.1 Requirements

To compile your project in Visual Studio 2022, you will need the following:

1. Windows operating system
2. Visual Studio 2022 installed on your machine
3. .NET 7 SDK
4. WinUI 3 SDK
5. MAUI Visual Studio extension installed
6. Microsoft Visual Studio Installer Projects extension installed
7. PowerShell installed on your machine
8. pki package installed in PowerShell
9. dotnet package installed in PowerShell

3.2 Unpackaged compilation

Add following lines to **CalculatorDesktopApp.csproj** file into the **PropertyGroup** node:

```
<WindowsPackageType>None</WindowsPackageType>
<WindowsAppSDKSelfContained Condition=”'$(IsUnpackaged)'␣==␣'true'”>true</Wind
<SelfContained Condition=”'$(IsUnpackaged)'␣==␣'true'”>true</SelfContained>
```

Change your **launchSettings.json** file to look like this:

```
{
        "profiles": {
                "Windows Machine": {
                        "commandName":~"Project"
                }
        }
}
```

To build the application open **IvsProject.sln** file in Visual Studio 2022. Select the **CalculatorDesktopApp** project as startup project and click ”Build”.

3.3 Packaged compilation

Create a certificate

Open a PowerShell terminal and navigate to the directory with your project.

Use the **New-SelfSignedCertificate** command to generate a self-signed certificate. The **PublisherName** value is displayed to the user when they install your app, supply your own value and omit the ”()” characters. You can set the **Name** parameter to any string of text you want.

```
New–SelfSignedCertificate –Type Custom –Subject ”CN=(PublisherName)”
–KeyUsage DigitalSignature
–FriendlyName ”Name”
–CertStoreLocation ”Cert:\CurrentUser\My” ‘
–TextExtension @(”2.5.29.37={text}1.3.6.1.5.5.7.3.3”, ”2.5.29.19={text}”)
```

Use the following PowerShell command to query the certificate store for the certificate that was created:

```
Get–ChildItem ”Cert:\CurrentUser\My” | Format–Table Subject, FriendlyName, Thu
```

Compile and publish the application

Navigate to the folder with **CalculatorDesktopApp** project in PowerShell and use the following command to build the project. Replace ”(cert)” with your certificate.

```
dotnet publish -f net7.0-windows10.0.19041.0 -c Release
-p:RuntimeIdentifierOverride=win10-x64 -p:PackageCertificateThumbprint=(cert)
```

You should find the path to the packaged application in the terminal path.

4 Navigation

4.1 Main Window

The calculators main window is the primary interface through which users can input calculations and receive their results. It consists of a number of buttons for entering values, selecting operations, memory functions and a field for displaying the result of the current calculation.
In the top left corner users can navigate to menu and in-application documentation.

4.2 Settings

The settings of a calculator allow users to customize the applications theme and behaviour. The currently supported settings are:

- Theme (Light/Dark): Changes the calculators color scheme.
- Number of digits: Changes the number of displayed digits.

4.3 In-Application Documentation

The calculator also includes smaller and concise documentation.

5 Usage

5.1 Introduction

Once the user is on the main screen of the calculator, he is presented with a set of buttons that include numbers, mathematical operators, functions and memory functions.

For example, to perform a simple calculation, you can start by pressing the numbers you want to use, followed by the mathematical operator you want to apply, and then the next number. For example, to calculate 5 + 3, you would press the number 5, followed by the plus sign (+), and then the number 3.

This calculator also has memory functions that allow you to store a number and retrieve it later. For instance, if you want to add 5 to a number that you previously stored in memory, you can recall that number by pressing the memory recall button, and then add 5 to it.

5.2 Operations

Please note that the calculator does not support IEEE 754 floating-point representations, such as NaN (Not a Number) and infinities. These features have been omitted from the calculator's functionality as they have been deemed inaccurate and confusing.

- Plus (+):

  Adds two numbers together.

- Minus (-):

  Subtracts the second number from the first number.

- Multiply (×):

  Multiplies two numbers together.

- Division (÷):

  Divides the first number by the second number.
  Raises an error when the second number is zero.

- Factorial (!):

  Calculates the factorial of a number, which is the product of all positive integers up to and including that number.
  Raises an error when number is not natural.

- Root ($\sqrt{\ }$):

  Calculates the square root of a number, which is a value that, when multiplied by itself, gives the original number.
  The result must not be imaginary.

- Power ($\wedge$):

  Raises the first number to the power of the second number.
  Raises an error when second number is not natural.

- Modulo (%):

  Divides the first number by the second number and returns the remainder.
  Raises an error when the second number is zero.

5.3 Memory functions

It is important to note that the calculator's memory functions work as a stack. This means that each time you use a memory function, the current value on the calculator's display is pushed onto the memory stack, and the memory value becomes the new current value. This stack-based memory functionality allows you to perform multiple memory operations in sequence.

- Memory Store (MS): Stores the current value on the calculator's display in memory.
- Memory Recall (MR): Recalls the value stored in the calculator's memory.
- Memory Clear (MC): Clears the value stored in the calculator's memory.
- Memory Add (M+): Adds the current value on the calculator's display to the value stored in memory.
- Memory Subtract (M-): Subtracts the current value on the calculator's display from the value stored in memory.

5.4 Keyboard controls

In addition to the traditional button interface, the calculator also supports keyboard controls. This can be particularly useful for users who prefer typing or for those who need to perform calculations quickly.

- Numbers (0-9): Inputs a number
- Left Alt: Calculates the expression
- Basic Operations: Inputs an operation
- Ctrl+C: Copies the result
- F1-F5: Memory functions
- Backspace: Deletes a character
- Tab+Enter: Standard tab control

6 References

https://support.microsoft.com/en-us/windows/uninstall-or-remove-apps-and-programs-in-windows-4b55f974-2cc6-2d2b-d092-5905080eaf98#ID0EBD=Windows_10
https://learn.microsoft.com/en-us/dotnet/maui/windows/deployment/publish-cli