Introduction to Intel® Software Guard Extensions (Intel® SGX)

April 18, 2017

Why Use Intel® SGX?

Many applications have secrets that need to be protected.

Examples of secrets include:

Healthcare records

Personally identifiable information (PII)

Biometric factors and templates

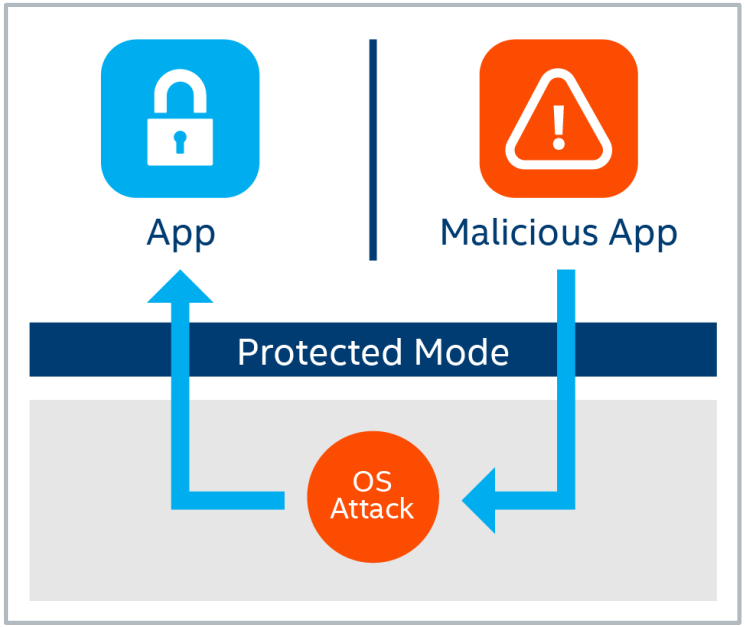
Passwords

Encryption keys

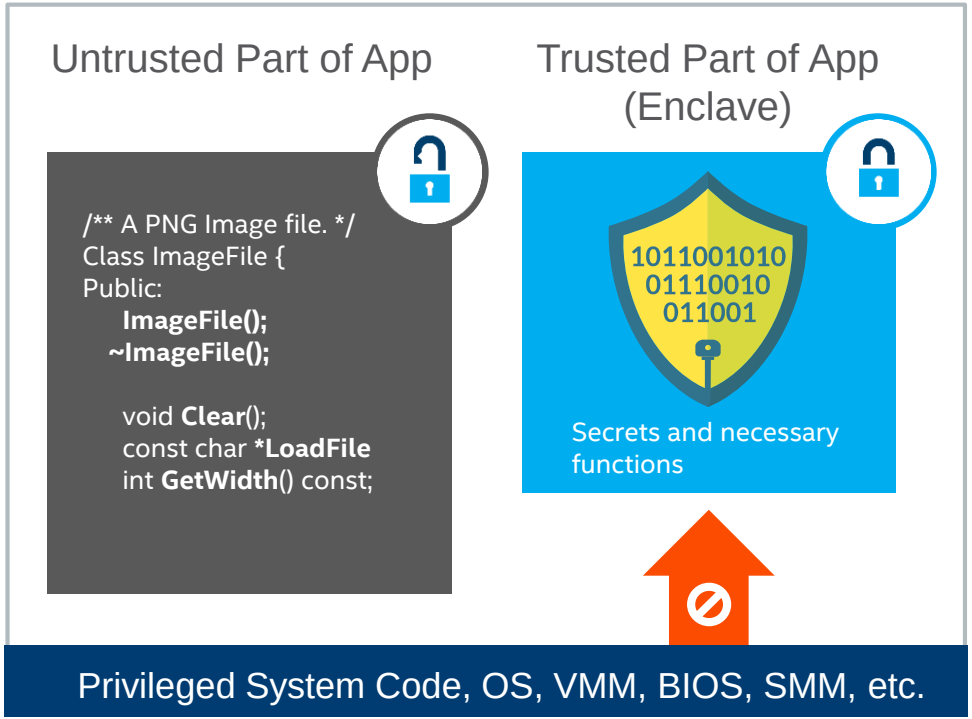
Intellectual property (IP)

What Is Intel® SGX?

- A Trusted Execution Environment from Intel for applications
- Trusted hardware: the CPU die
- Isolates a portion of physical memory to protect select code and data from view or modification
- In Intel SGX, these isolated portions of memory are called “enclaves”



Trusted and Untrusted Partitions



1. An Intel® SGX–enabled app is built with trusted and untrusted parts.

2. Best practice: the trusted partition should contain just the minimum amount of content required to protect your secrets.

What Intel® SGX Can't Protect You From

- Buffer-overflow attacks
- Side-channel attacks
- Untrusted inputs into the enclave

There is no substitute for good, secure programming practices

Intel® SGX Design Imperatives

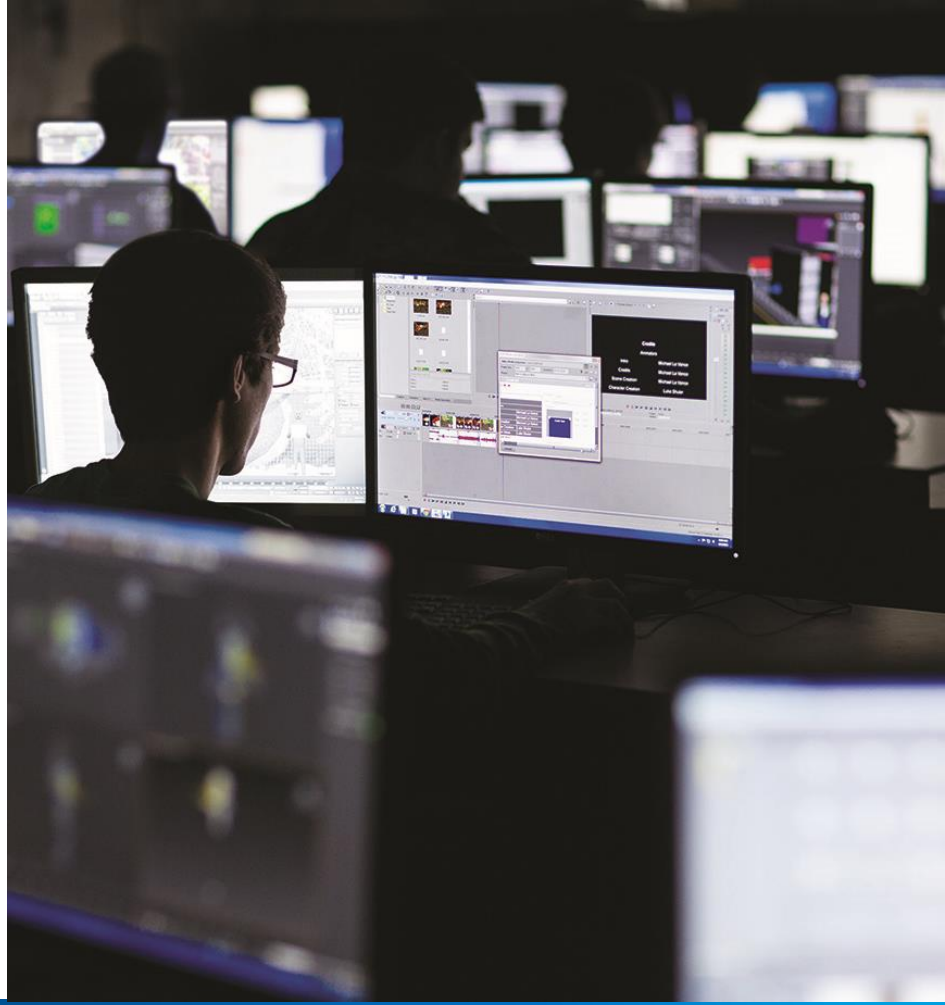
Protect sensitive data

Do not interfere with legitimate operations

Develop trusted applications using familiar tools and processes

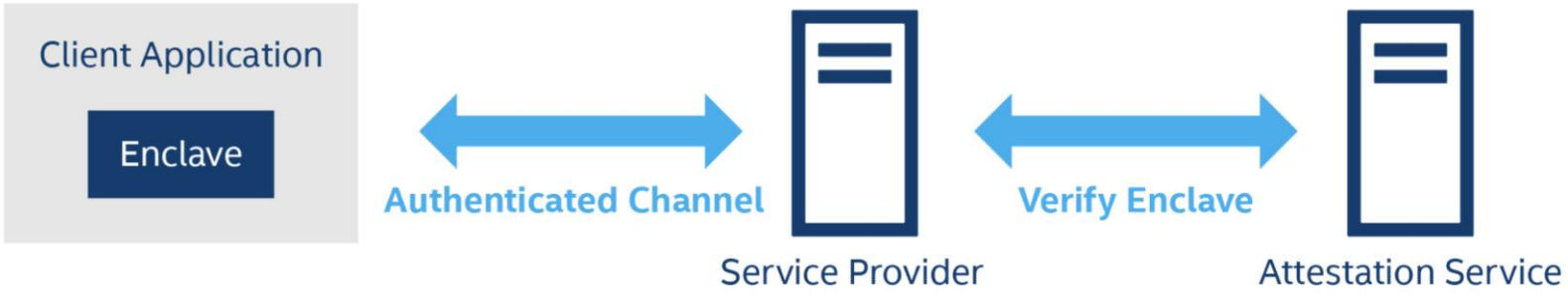
What Do You Need to Develop and Run Enclaves?

- A platform with an Intel® SGX–enabled CPU and BIOS
- Intel SGX platform software (PSW)
- The Intel SGX software-development kit (SDK)
- Microsoft Visual Studio 2015 Professional* or Eclipse™
  - Supports Intel SGX debugger



What Can You Do with Enclaves?

1. **Access** secrets
2. **Attest** services
3. **Seal** secrets

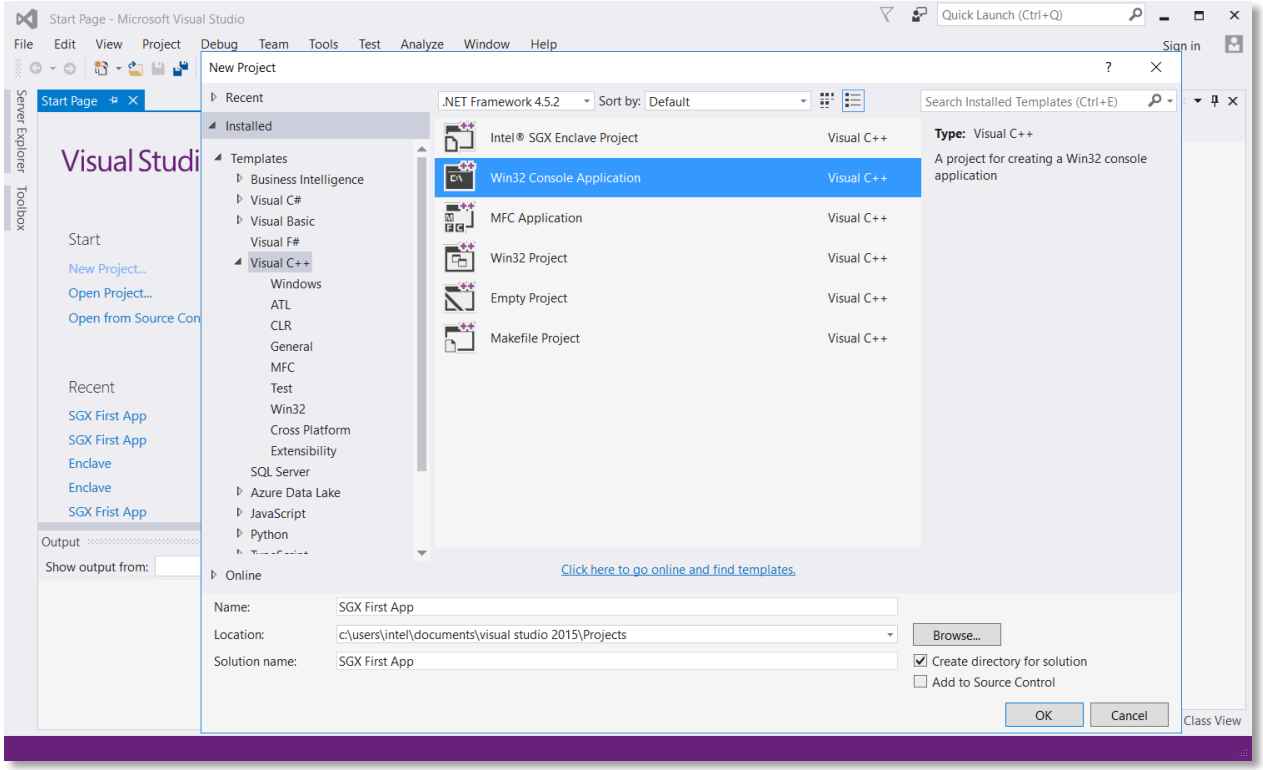


Creating a “SGX First App” Intel® SGX Application

Demonstration

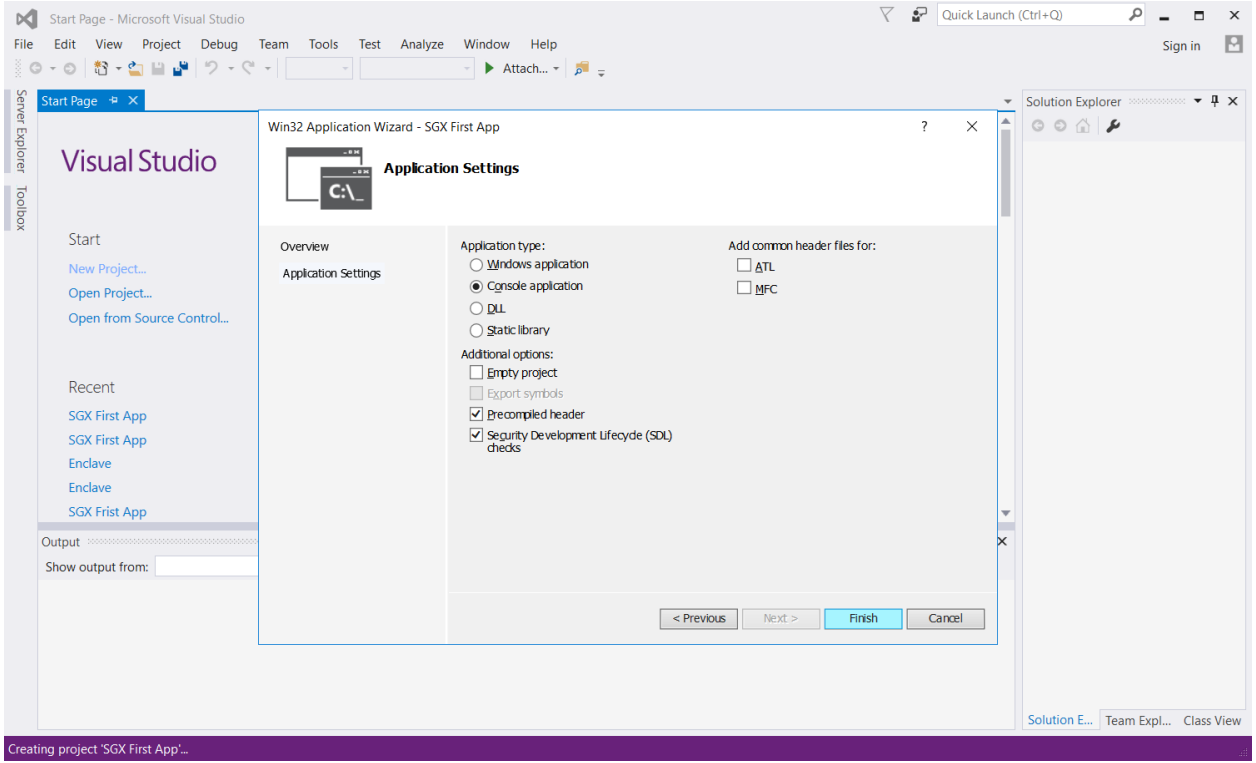
Create a Console-Application Project

Demo



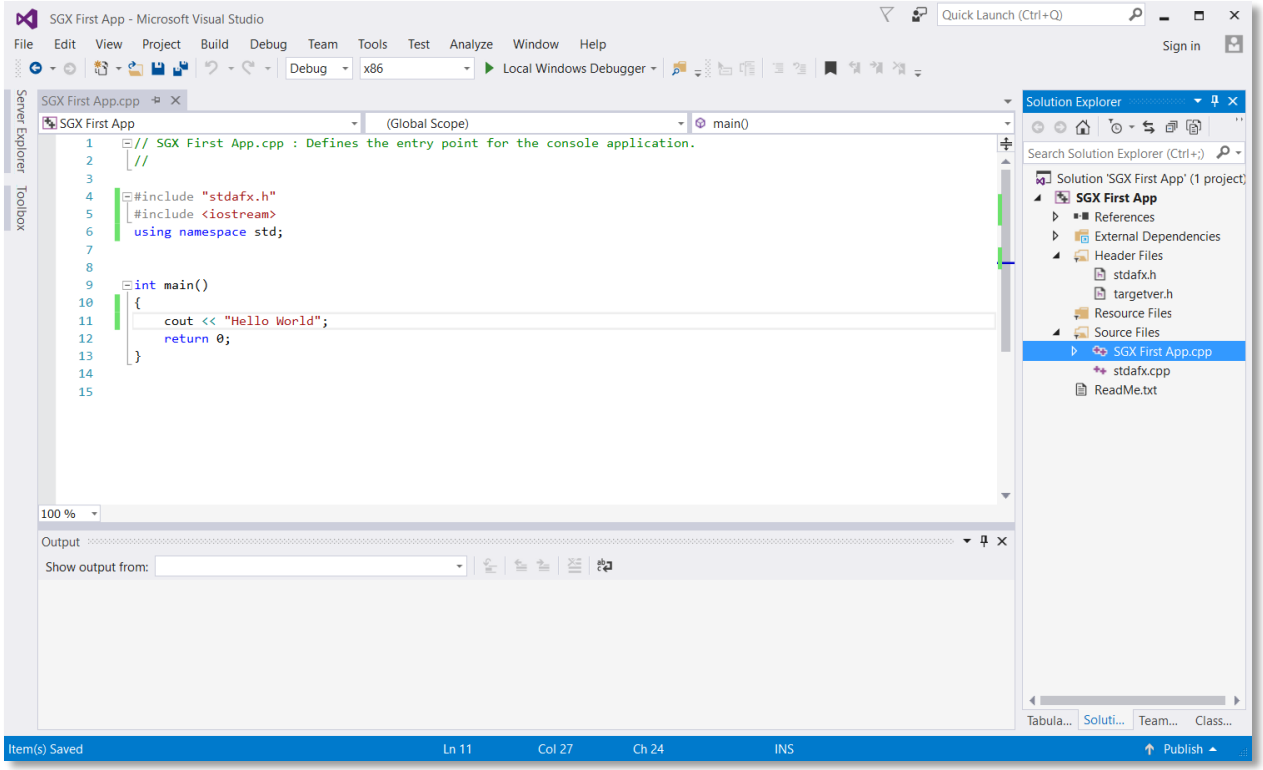
Create a Console-Application Project

Demo



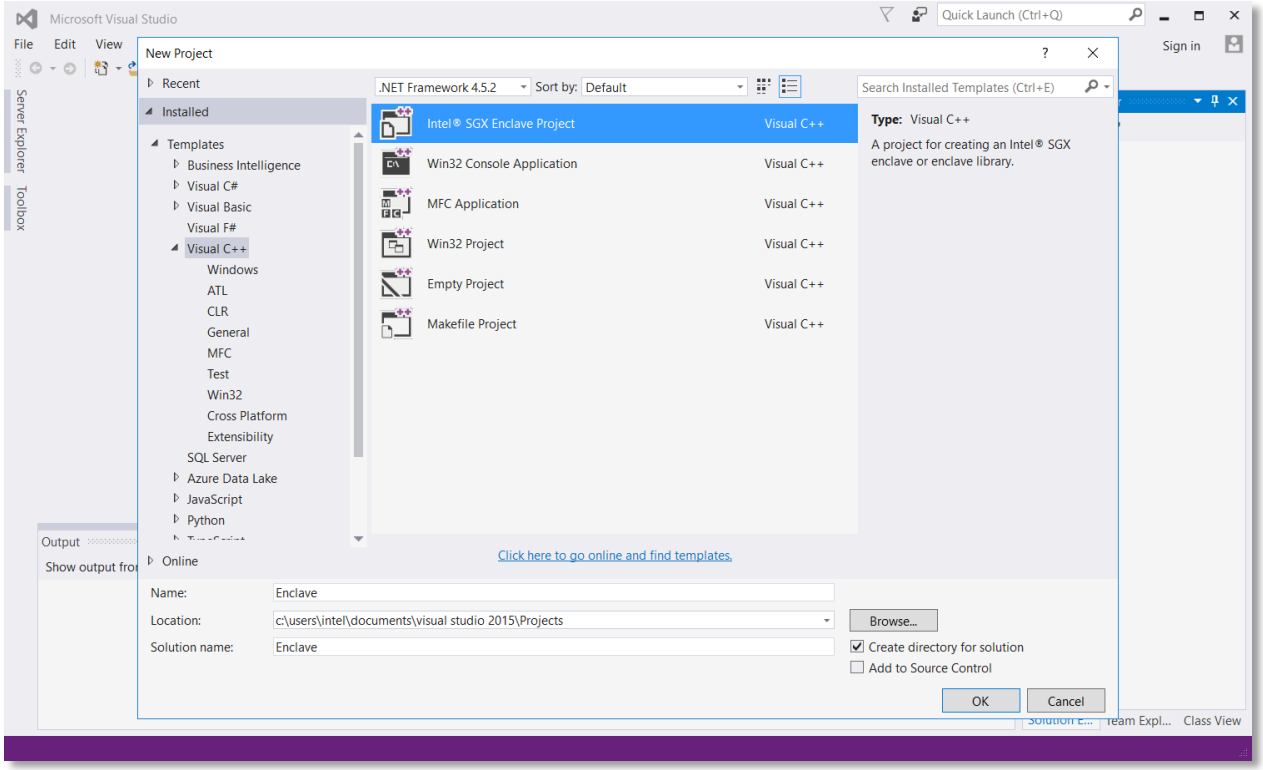
Create a Console-Application Project

Demo



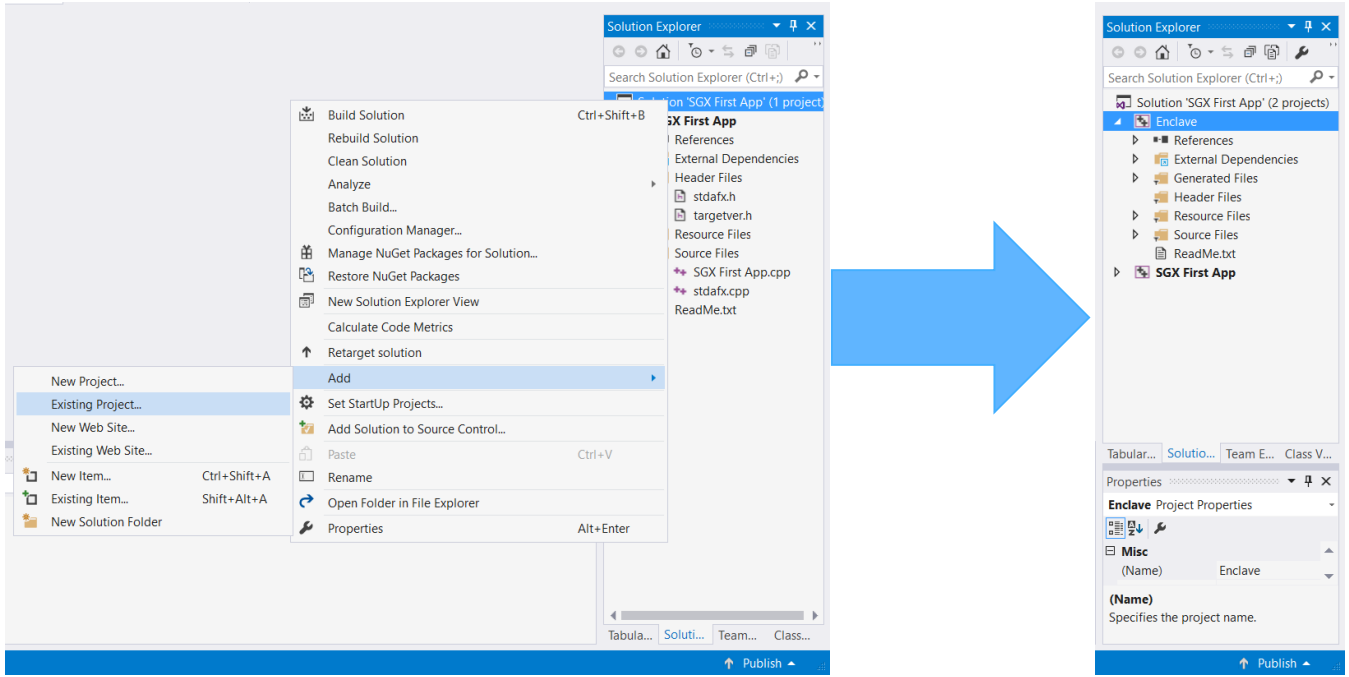
Create an Enclave-Application Project

Demo



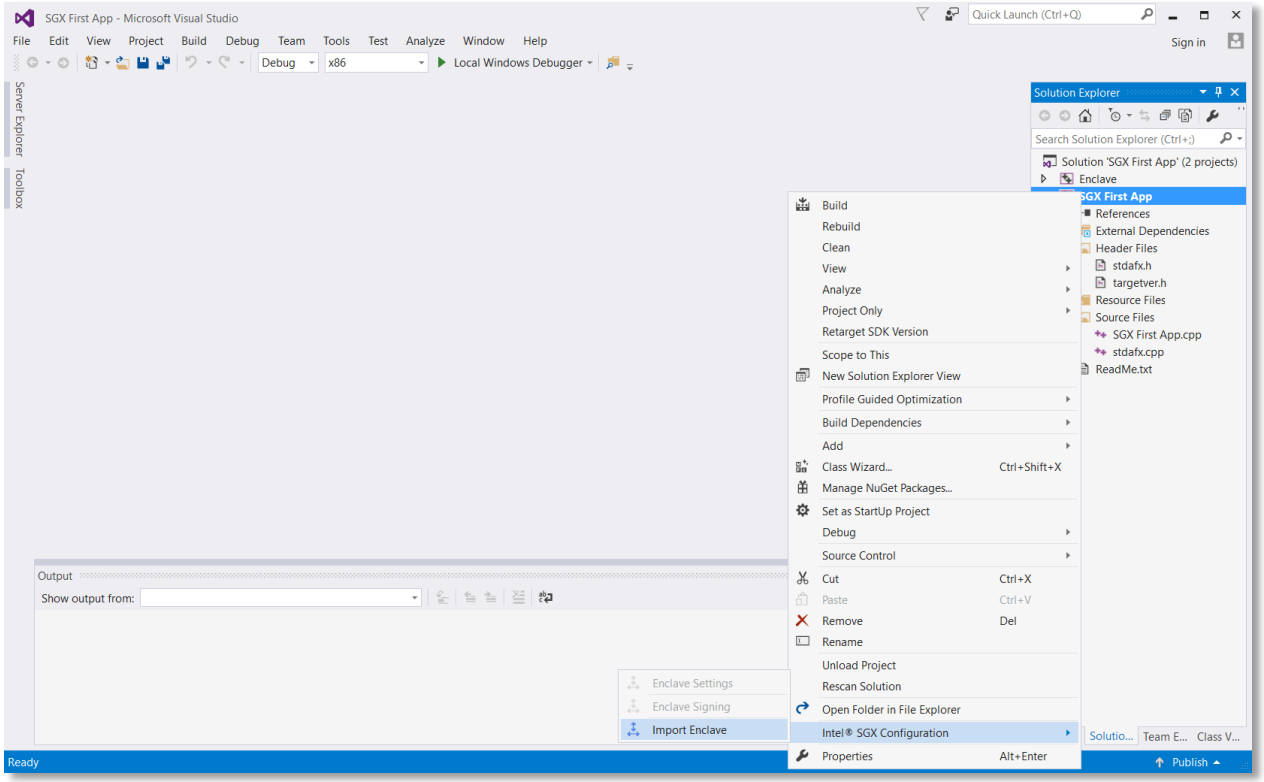
Demo

Add an Enclave Project to the Console-Application Project



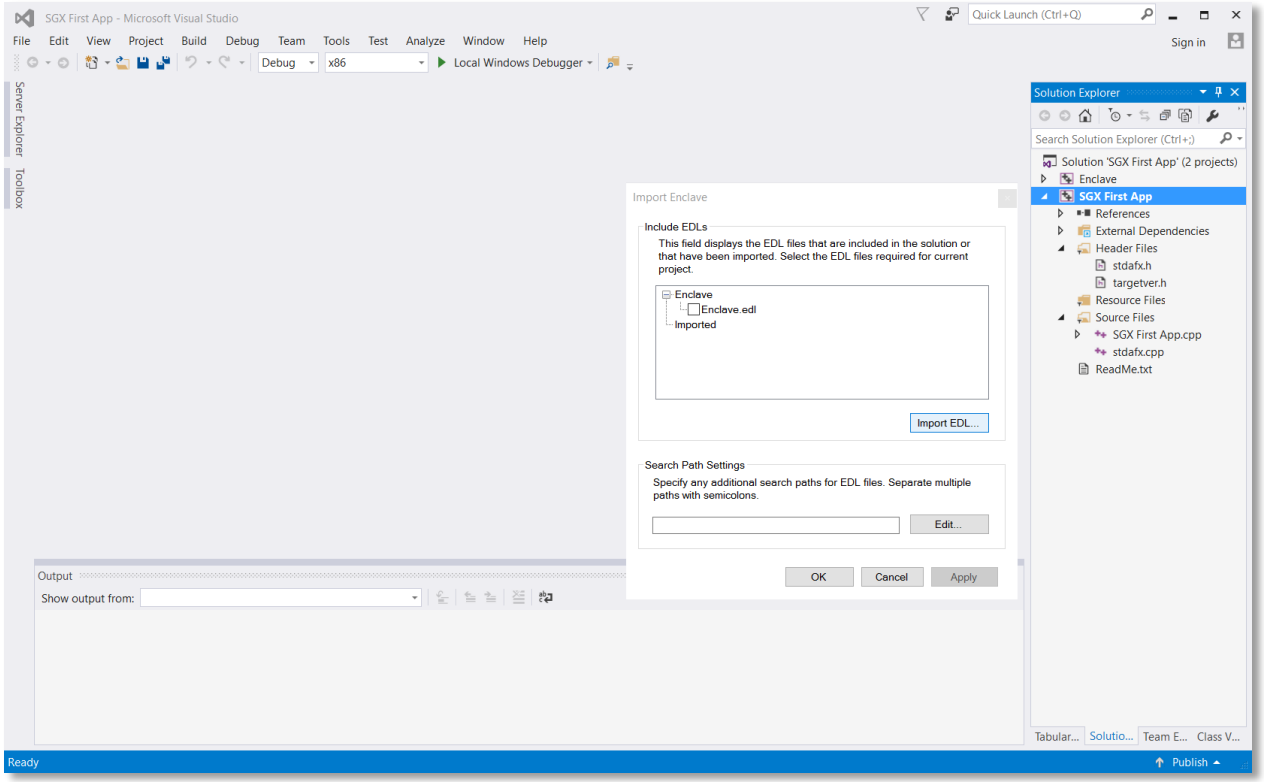
Import an Enclave into a Console Application

Demo



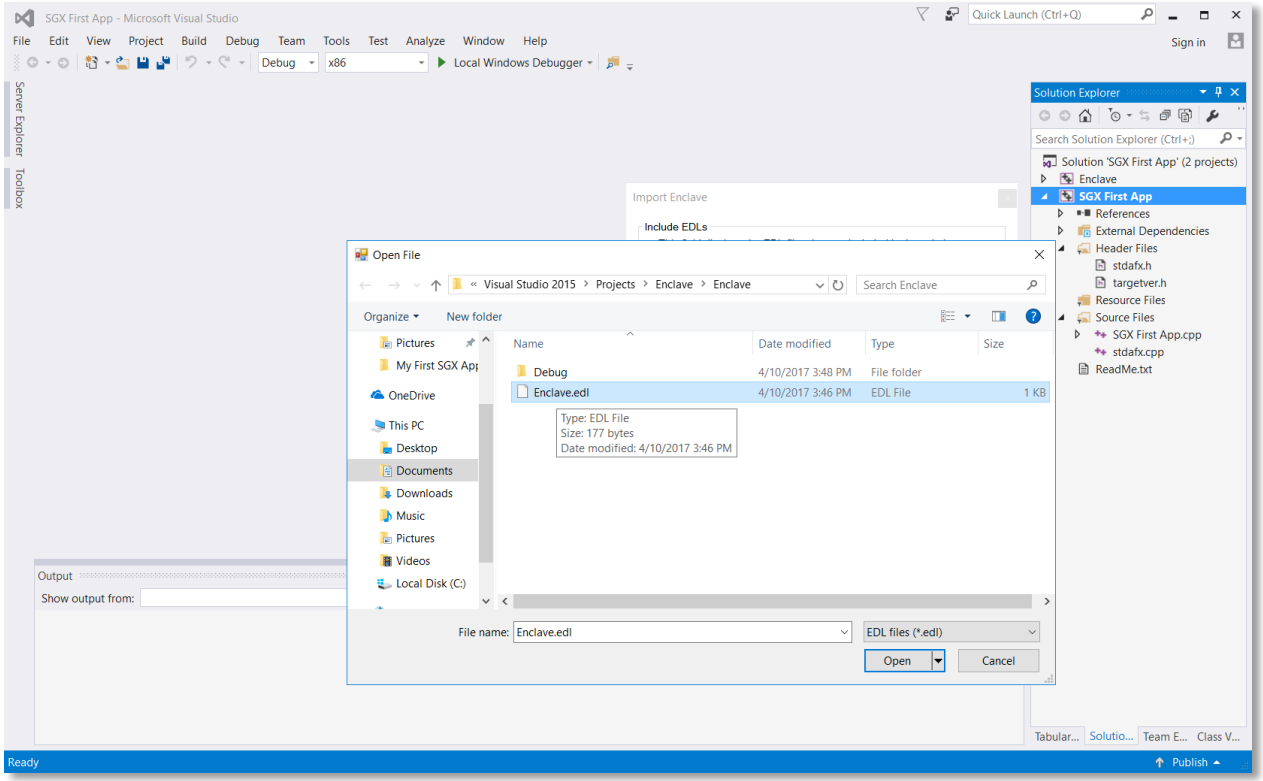
Import an Enclave into a Console Application

Demo



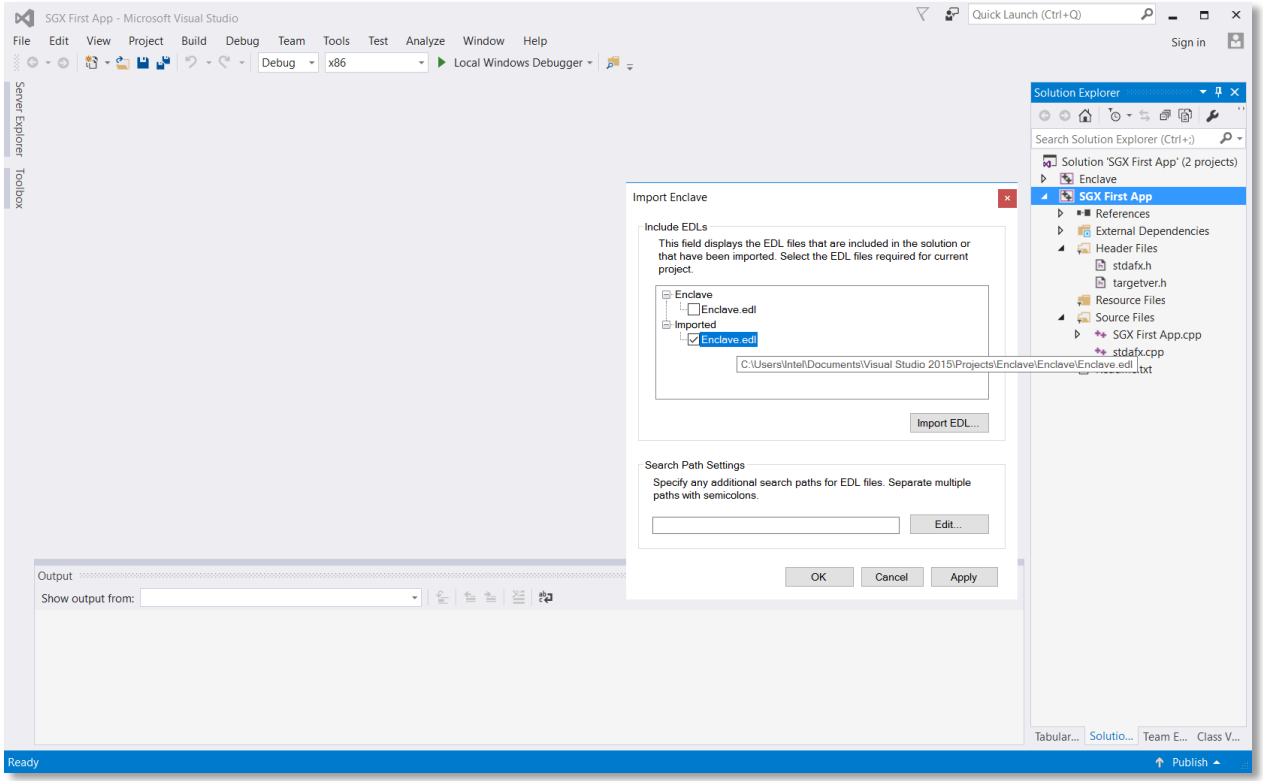
Import an Enclave into a Console Application

Demo



Import an Enclave into a Console Application

Demo



Technical Discussion: EDL

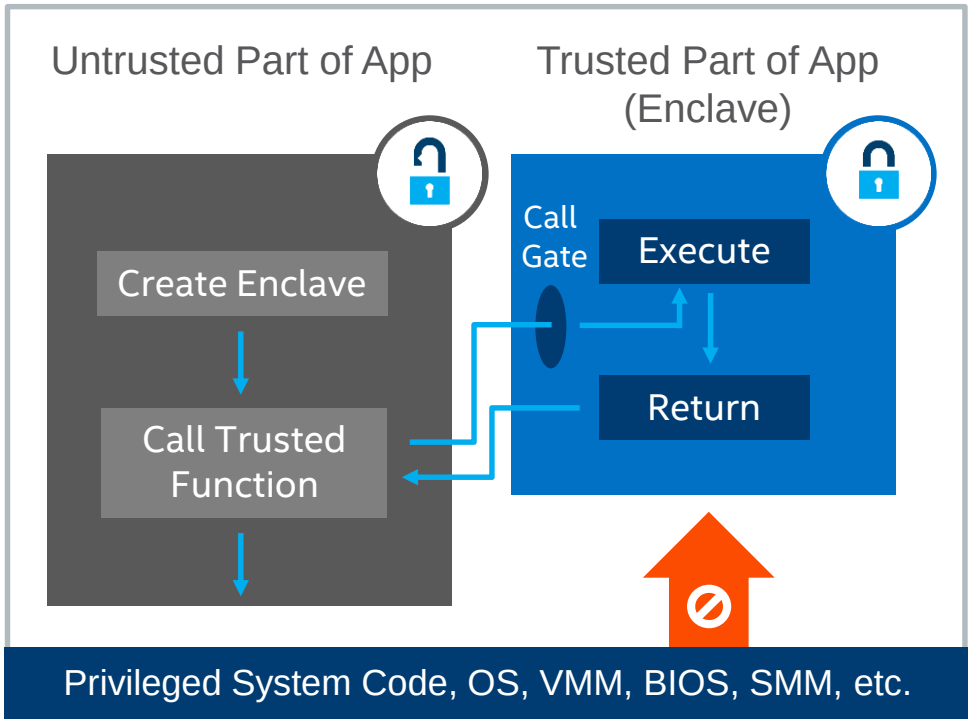
A Brief Introduction to EDL

EDL stands for *Enclave Definition Language.*

It defines the interface between the trusted and untrusted portions of your applications.

```
enclave {
     //Include files

     //Import other edl files

     //Data structure declarations to be used as parameters of the
     //function prototypes in edl

     trusted {
          //Include header files if any
          //Will be includedd in enclave_t.h

          //Trusted function prototypes

     };

     untrusted {
          //Include header files if any
          //Will be included in enclave_u.hhead

          //Untrusted function prototypes

     };
};
```

Trusted and Untrusted Application Code

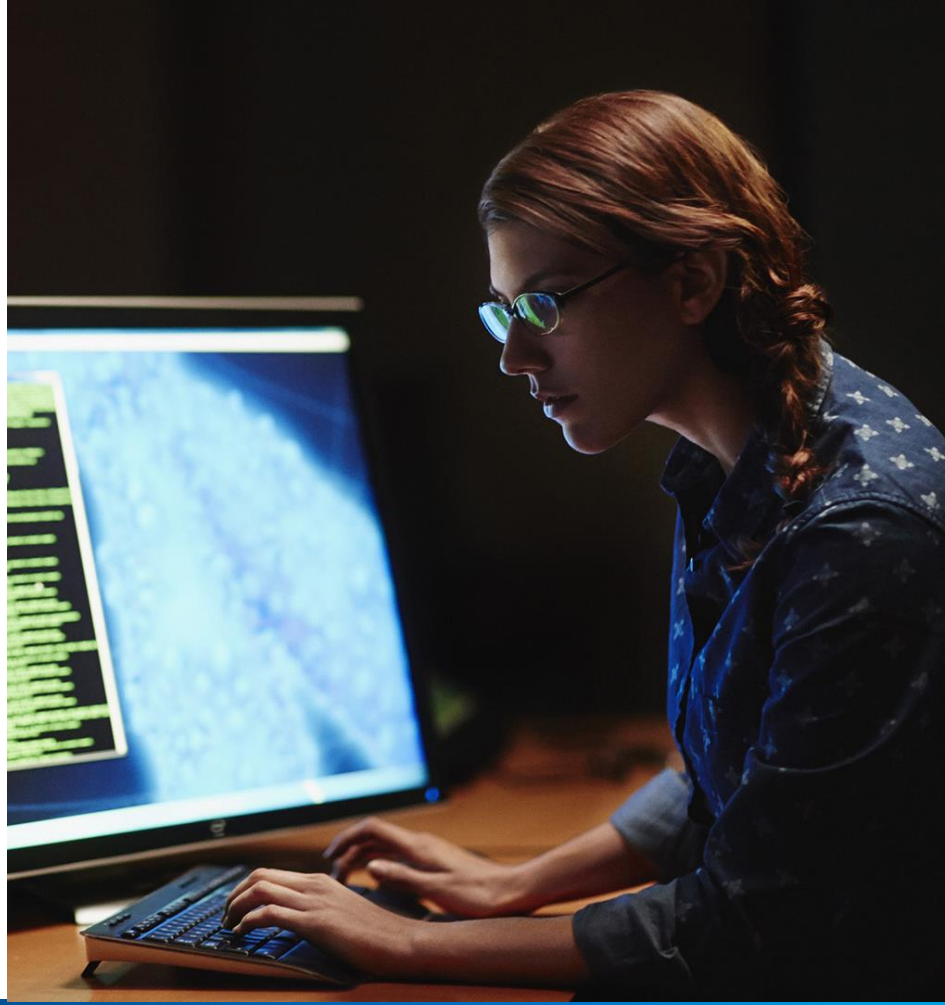


1. An Intel® SGX–enabled app is built with trusted and untrusted parts.
2. The app runs and creates an enclave, which is placed in trusted memory.
3. A trusted function is called, code running inside the enclave sees the data as clear text, and external access to the data is denied.
4. The function returns, and the enclave data remains in trusted memory.

Edge Code Auto-generation

The Intel® SGX SDK auto-generates edge routines defined by EDL through use of the Edger8r* tool:

- Intel SGX uses only plain C, so wrapper functions are necessary.
- It auto-generates the code the first time you build a project.
- It produces edge routines for accessing enclaves.
- Edge routines are specified by the developer in the EDL file.

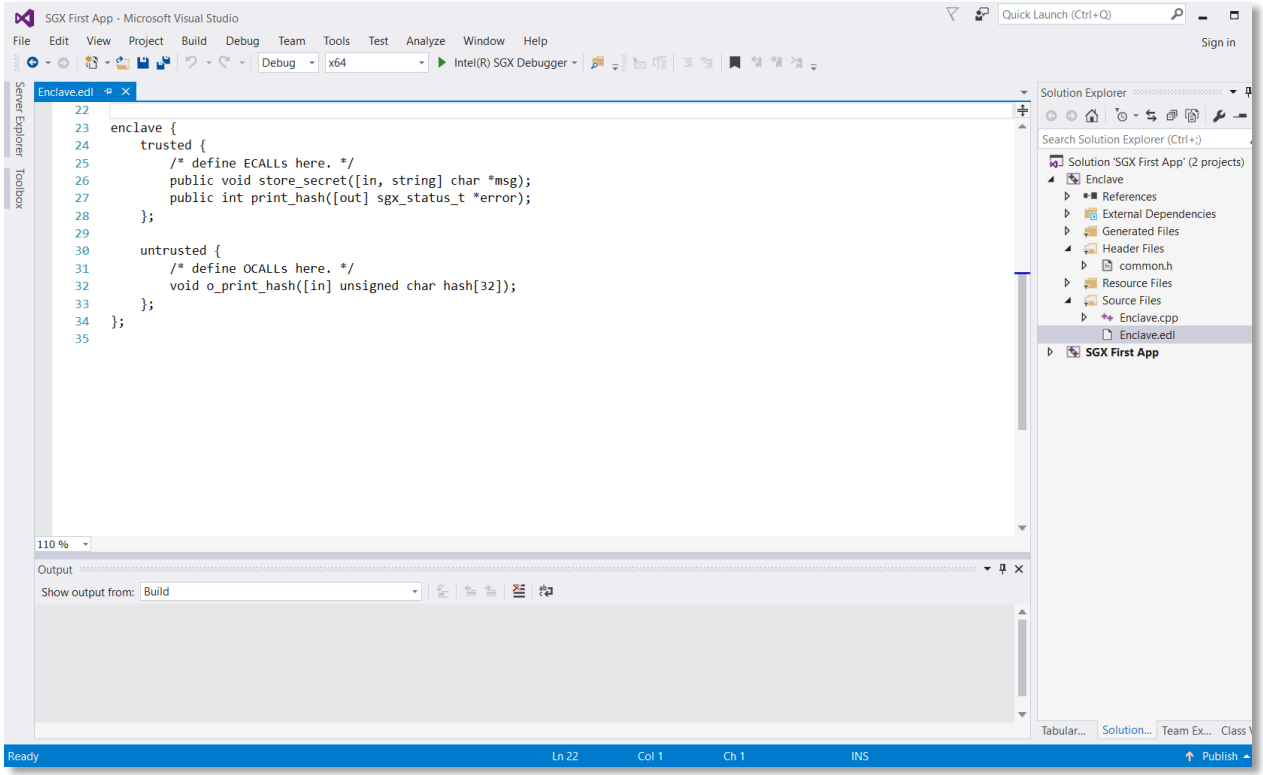


Completing the “SGX First App” Enclave Application

Demonstration

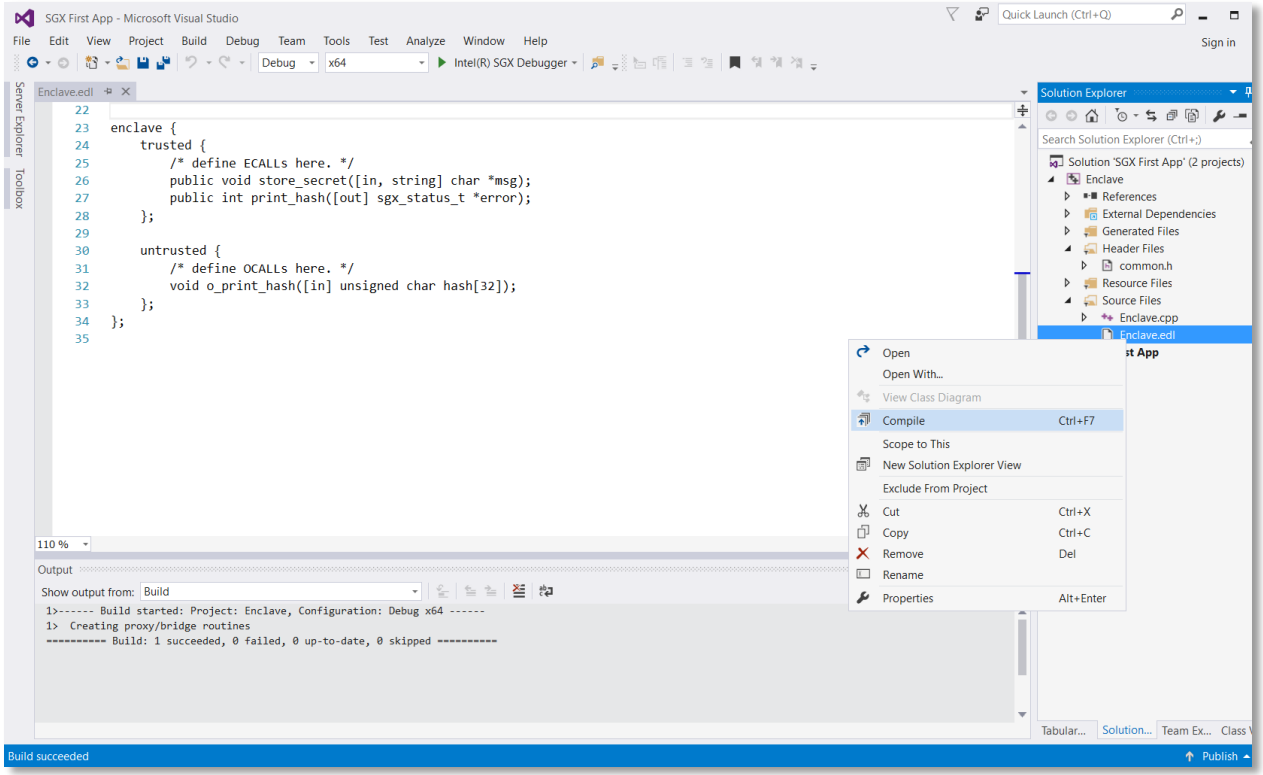
Add a Trusted Call to an Enclave Project

Demo



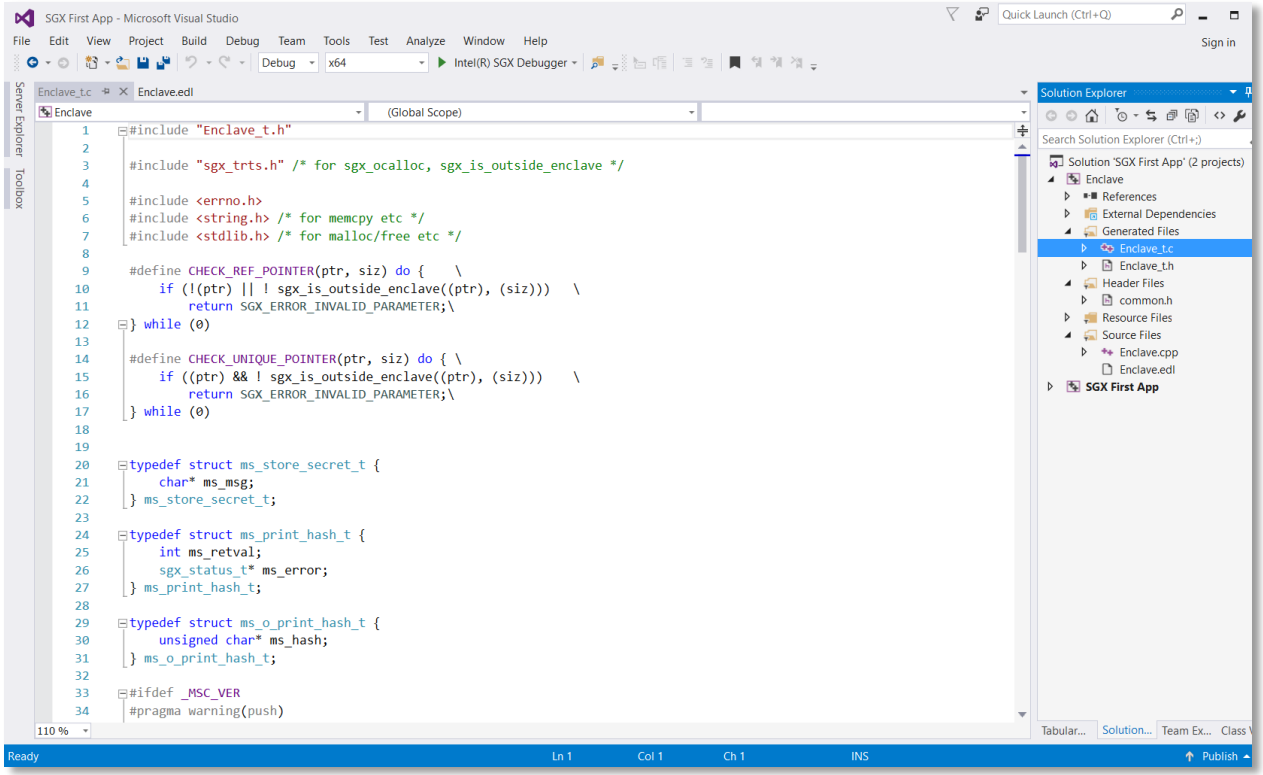
Add a Trusted Call to an Enclave Project

Demo



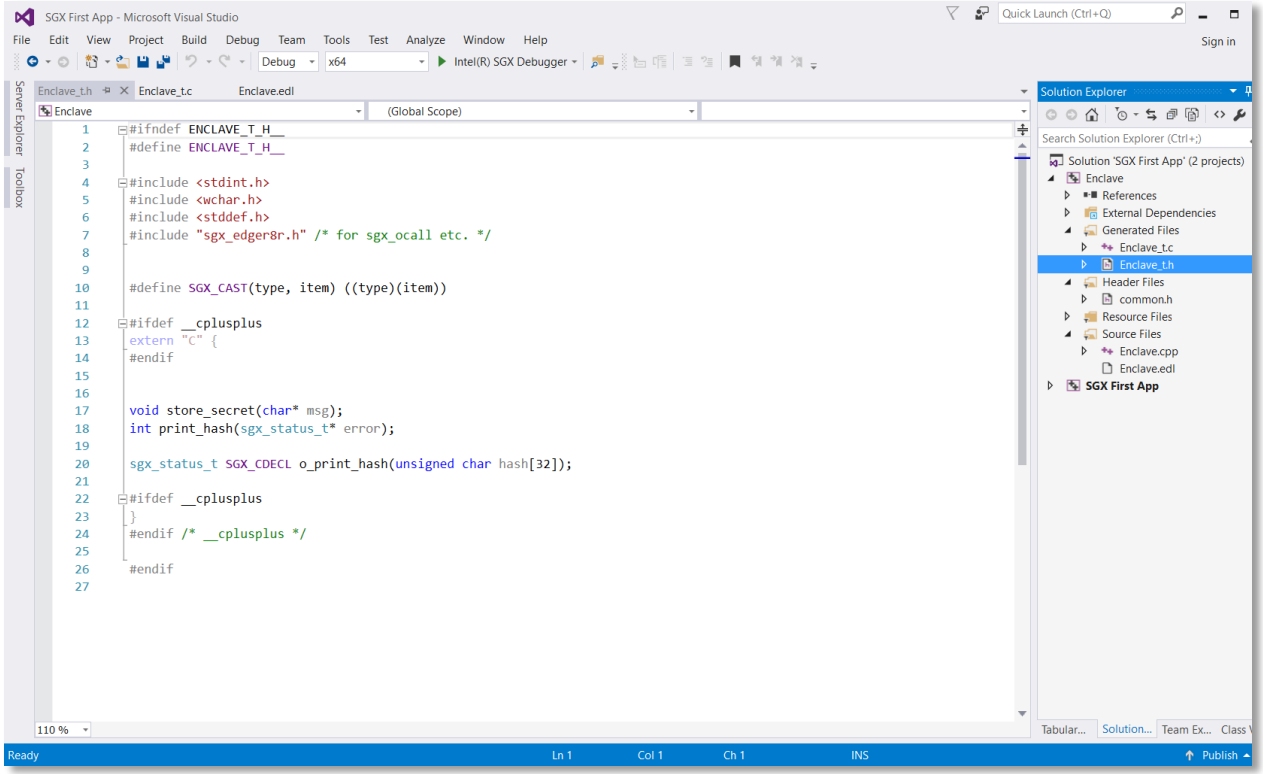
Add a Trusted Call to an Enclave Project

Demo



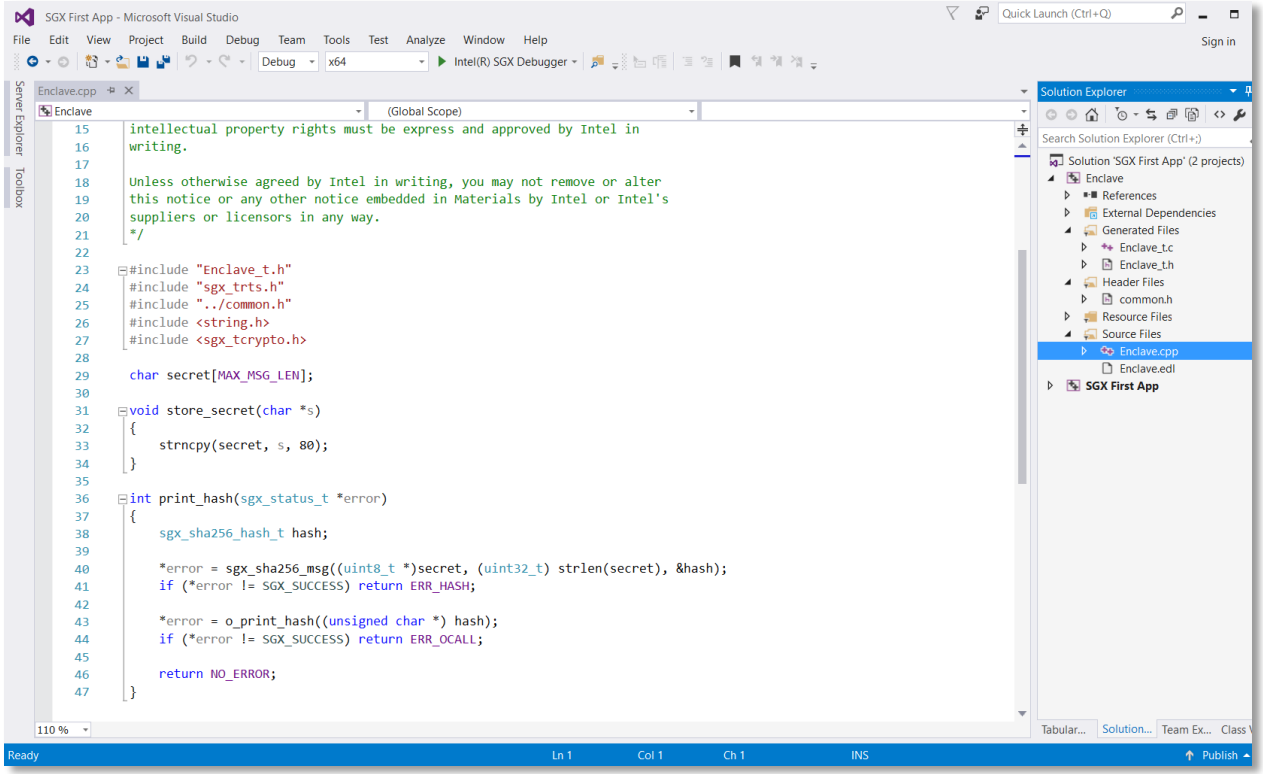
Add a Trusted Call to an Enclave Project

Demo



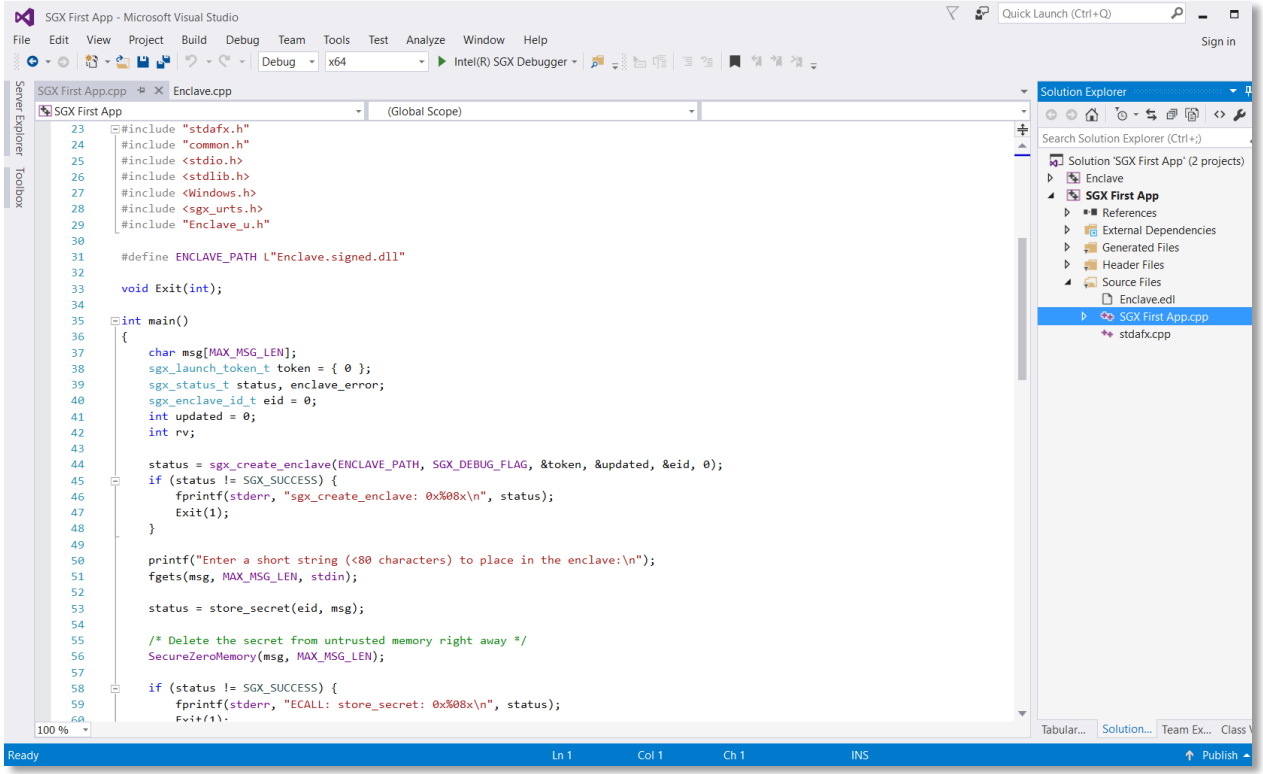
Implement Enclave Functions

Demo



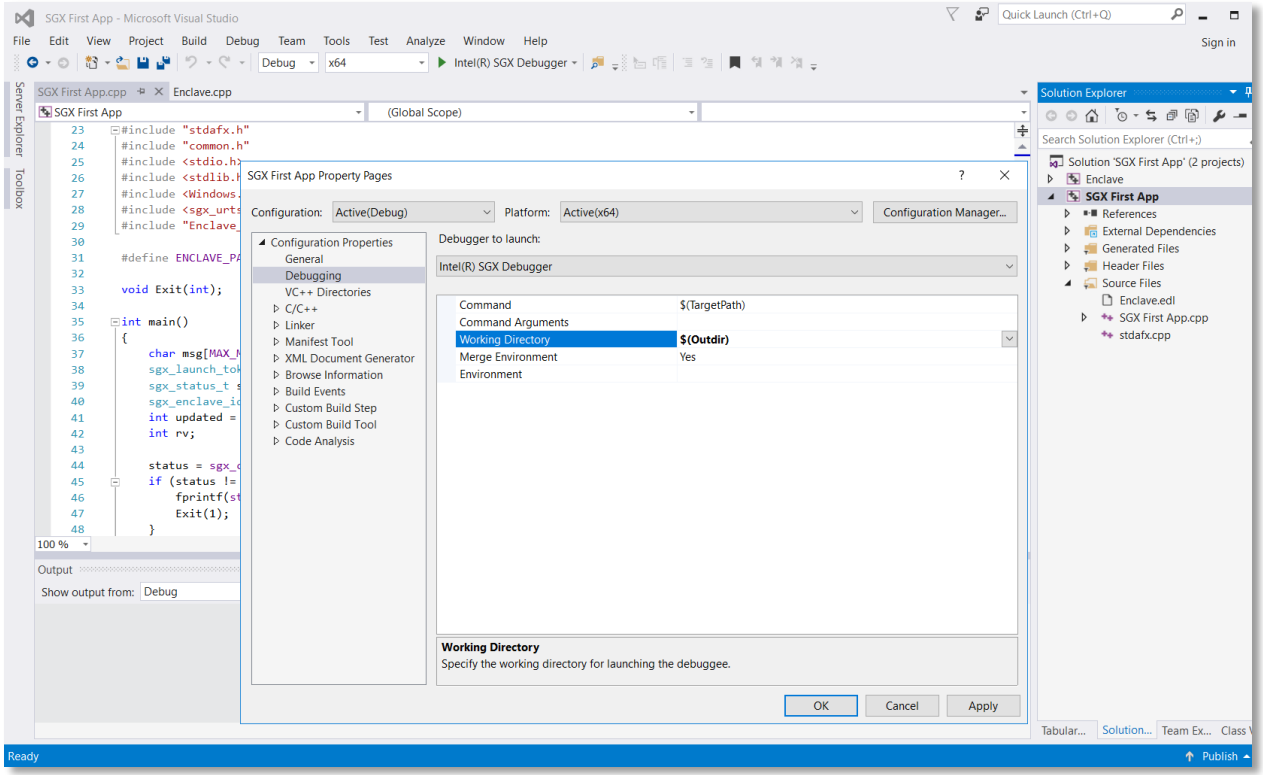
Implement the Enclave Application

Demo



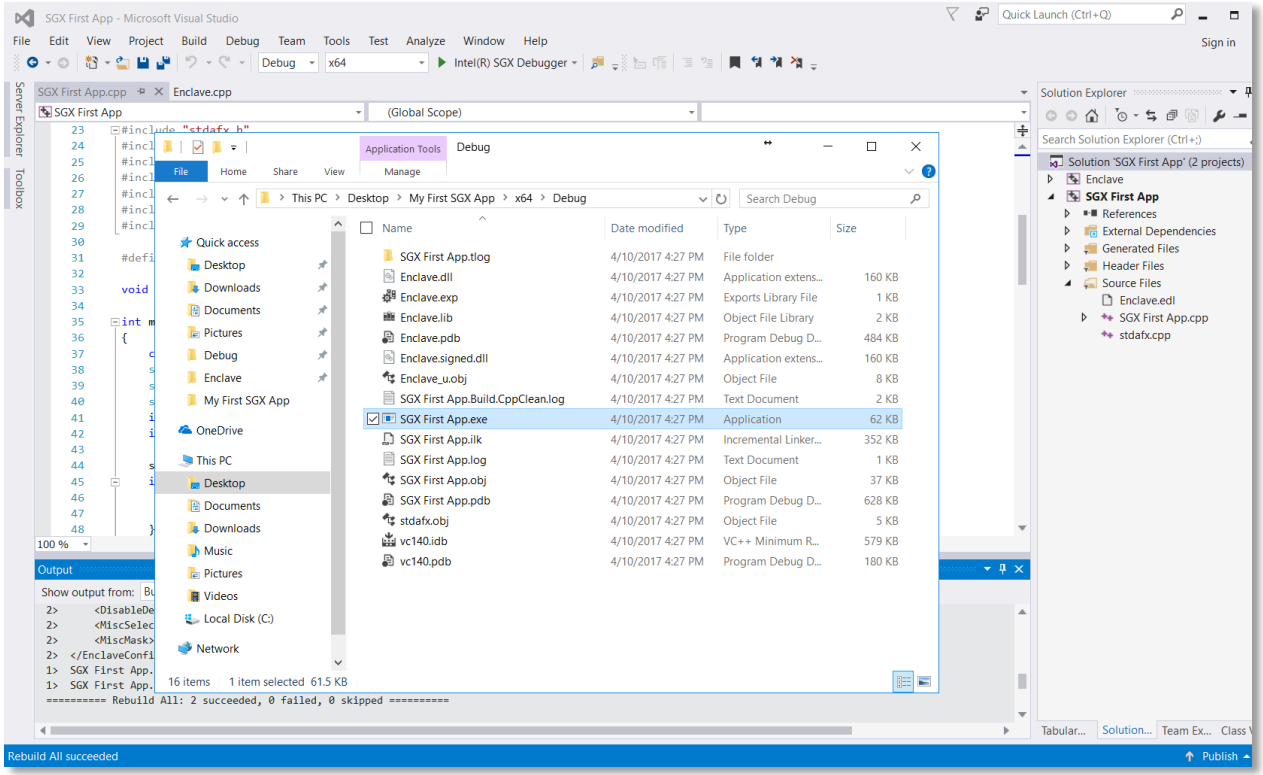
Implement the Enclave Application

Demo



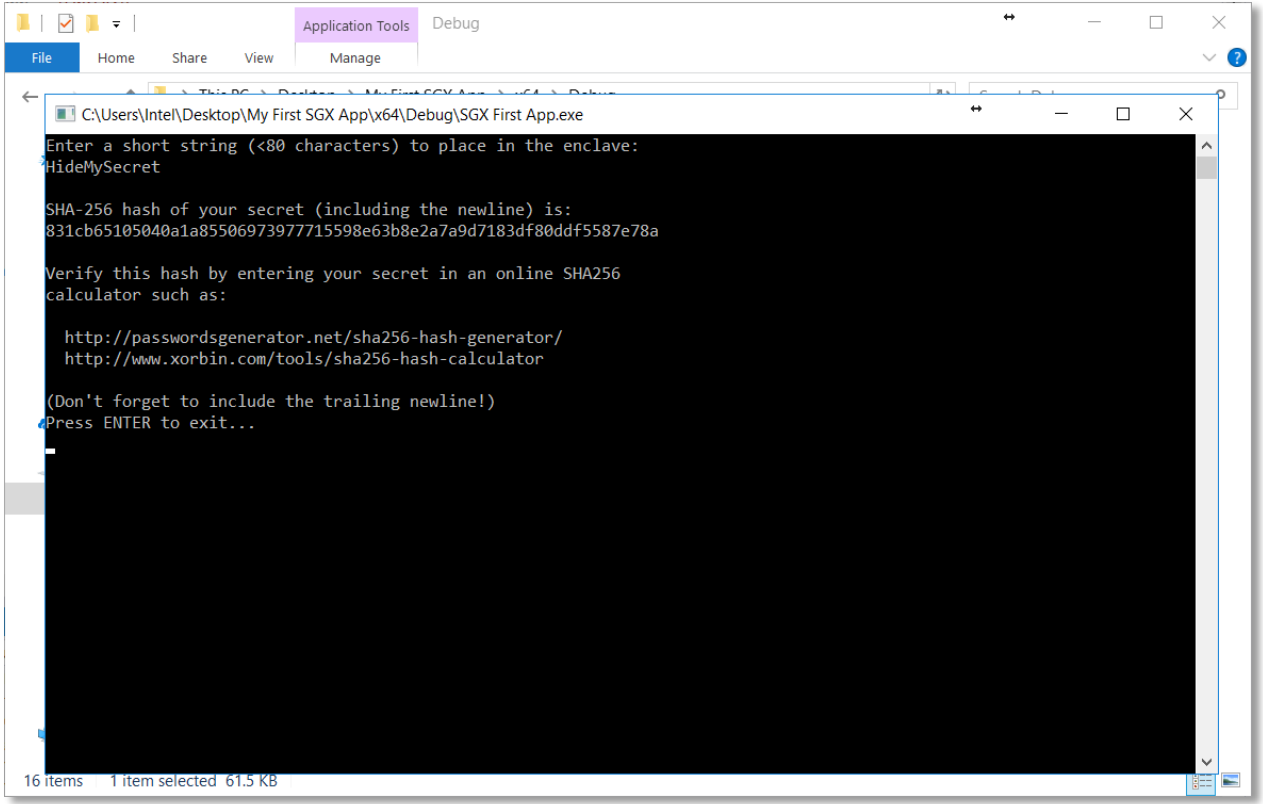
Implement the Enclave Application

Demo



Implement the Enclave Application

Demo



Conclusion

Wrap-up

- Hardware-based protection for sensitive data
- SDK auto-generates many necessary wrapper functions
- Build trusted apps with familiar tools
- Visit the links at the end of the webinar to learn more and see other examples

Additional Resources

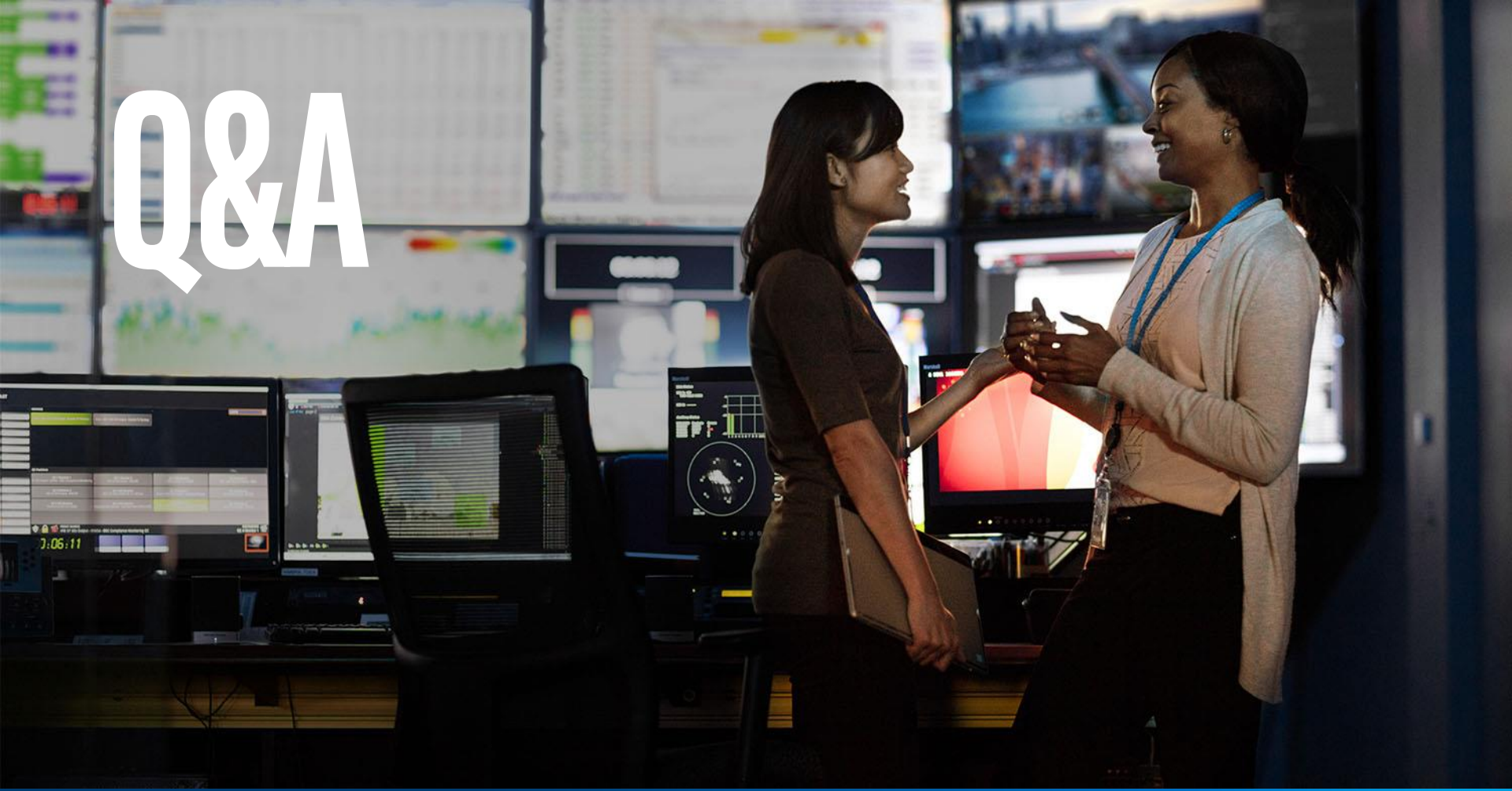
Intel® SGX sample code and tutorials: https://software.intel.com/en-us/sgx/code-samples

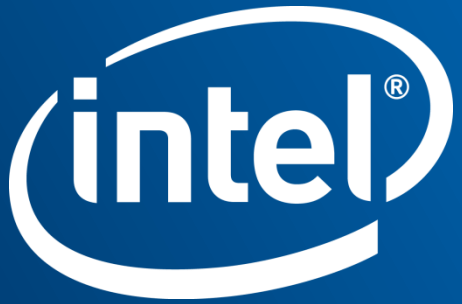
Intel SGX forum: https://software.intel.com/en-us/forums/intel-software-guard-extensions-intel-sgx

Intel SGX for Linux*: https://01.org/intel-software-guard-extensions

Intel SGX at the Intel® Developer Zone: https://software.intel.com/en-us/sgx

Q&A




intel®

Legal Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel, the Intel logo, and Intel Core are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.