Complete SDK Integration Guide

> ⚠️ **IMPORTANT**: Before beginning installation, ensure you have administrator privileges and backup any important data. The installation process requires system modifications.

> 📝 **Note**: This guide assumes a clean installation. If you have previous versions installed, please uninstall them first.

Table of Contents

1. System Requirements and Prerequisites

1.1 System Check

1. Verify your Windows system type:
   - Right-click on **This PC** → **Properties**
   - Look under "System type" for 32-bit or 64-bit
   - Note: This determines which binaries you'll use (Win32 vs x64)

> ⚠️ **WARNING**: Installing incorrect binaries may cause system instability.

1.2 Disk Space Requirements

- Minimum 10GB free space:
  - Visual Studio: ~8GB
  - OpenCV: ~1GB
  - SDK and dependencies: ~1GB

> 🔍 **TIP**: Run disk cleanup before installation to ensure adequate space.

1.3 Prerequisites Check

1. Open Control Panel → Programs and Features
2. Check for existing installations:
    - Microsoft Visual Studio (2013 or 2022)
    - Microsoft Visual C++ 2013 Redistributable (x86 and x64)
    - Microsoft Visual C++ 2010 Redistributable (x86 and x64)

2. Required Components

2.1 Essential Files Checklist

- SDK Core Files:

  ```
  ├── HBISDKApi.dll
  ├── OpenCV DLLs
  │   ├── opencv_world341.dll
  │   └── opencv_world341d.dll
  ├── VC++ 2013 Runtime DLLs
  │   ├── msvcp120.dll
  │   ├── msvcp120d.dll
  │   ├── msvcr120.dll
  │   └── msvcr120d.dll
  └── VC++ 2010 Runtime DLLs (Optional)
      ├── msvcp100.dll
      ├── msvcp100d.dll
      ├── msvcr100.dll
      └── msvcr100d.dll
  ```

2.2 Download Links

1. Development Environment:

    - Visual Studio 2013 Community: Download →
    - Visual Studio 2022 Community: Download →
    - VS 2013 Update 5: Download →

2. Visual C++ Redistributables:

    - 2013 Runtime:
        - x86 Version: Download →
        - x64 Version: Download →
    - 2010 Runtime:
        - x86: Download →
        - x64: Download →

3. OpenCV:
    - OpenCV 3.4.1: Download →
    - Direct Download: opencv-3.4.1-vc14_vc15.exe

3. Visual Studio Installation

3.1 Visual Studio 2013 Installation

> ⚠️ **CAUTION**: Install Visual Studio 2013 before any other components.

1. Download Visual Studio 2013 Community Edition
2. Run the installer:
    - Select **Custom** installation type
    - Check **Visual C++** under Programming Languages
    - Select **Microsoft Foundation Classes for C++**
3. Complete installation
4. Install Update 5:
    - Download VS 2013 Update 5
    - Run the installer
    - Follow prompts to complete update

3.2 Visual Studio 2022 Installation (Recommended)

> 💡 **RECOMMENDATION**: Visual Studio 2022 is recommended for new projects due to:
>
> - Better performance and modern IDE features
> - Improved debugging capabilities
> - Support for latest Windows SDKs
> - Better compatibility with modern C++ standards

1. Download Visual Studio 2022 Community Edition

2. During installation, select:

    - **Desktop development with C++**
    - **C++ MFC for x86 and x64**
    - Under Individual Components:
        - **MSVC v120 - VS 2013 C++ build tools**

        - Windows 10 SDK

    > 🔍 **TIP**: Install the following optional components for better development experience:
    >
    > - C++ ATL for latest build tools
    > - C++ CMake tools
    > - Windows 11 SDK (backwards compatible)
    > - C++ profiling tools

3. Complete installation

4. Install any available updates

> 📝 **NOTE**: After installation, ensure you configure the following:
>
> - Set default development settings to Visual C++
> - Install any recommended security updates
> - Configure source control integration if needed

4. Visual C++ Redistributables

> 🔍 **TIP**: Install x86 versions before x64 versions to avoid compatibility issues.

4.1 Required Installations

1. Install Visual C++ 2013 Redistributable:

    - Download both x86 and x64 versions
    - Install x86 version first
    - Install x64 version second
    - Restart system if prompted

2. Install Visual C++ 2010 Redistributable (if needed):

    - Follow same process as above
    - Install x86 then x64

4.2 Verification

1. Check Control Panel → Programs and Features
2. Verify presence of:
    - Microsoft Visual C++ 2013 Redistributable (x86)
    - Microsoft Visual C++ 2013 Redistributable (x64)
    - Microsoft Visual C++ 2010 Redistributable (x86)

    - Microsoft Visual C++ 2010 Redistributable (x64)

5. OpenCV Setup

5.1 Installation from SDK Package

1. Locate `opencv341_unzipping_to_disk_D.7z` in SDK
2. Extract to `D:\` :

```
D:\opencv341\opencv\
├── build\
│   ├── include\
│   ├── x64\vc14\lib\
│   ├── x64\vc14\bin\
│   ├── x86\vc14\lib\
│   └── x86\vc14\bin\
```

5.2 Manual Installation (Recommended)

> 💡 **RECOMMENDATION**: Manual installation is preferred because:
>
> - Ensures clean installation without potential SDK package conflicts
> - Allows custom configuration options
> - Provides latest bug fixes and security updates
> - Better control over component selection

1. Download OpenCV 3.4.1 from official source:

    - Go to: OpenCV 3.4.1 Release
    - Download: opencv-3.4.1-vc14_vc15.exe

2. Pre-installation checks:

    - Verify minimum 1GB free space on D: drive
    - Close any applications using OpenCV
    - Backup any existing OpenCV installations

3. Installation steps:

    - Run self-extracting archive as Administrator
    - Choose custom installation path: `D:\opencv341`
    - Select components:

```
✓ Build folder
✓ Include folder
✓ Java bindings (optional)
✓ Python bindings (optional)
```

4. Post-installation verification:

   - Verify directory structure:

     ```
     D:\opencv341\opencv\
     ├── build\
     │   ├── include\         # Header files
     │   ├── x64\vc14\lib\    # 64-bit libraries
     │   ├── x64\vc14\bin\    # 64-bit DLLs
     │   ├── x86\vc14\lib\    # 32-bit libraries
     │   └── x86\vc14\bin\    # 32-bit DLLs
     ```

   - Test sample programs in `\samples` directory
   - Verify all required DLLs are present

> 🔍 TIPS:
>
> - Keep the original installer for future reference
> - Document any custom configurations made
> - Consider creating a backup of the successful installation

6. SDK Installation

6.1 Directory Structure

Create the following structure:

```
project_directory/
├── output/
│   ├── Win32/
│   └── x64/
├── DemoDll/
│   └── DemoDll/
├── Demolib/
│   └── Demolib/
└── HB_SDK_DEMO2013/
```

6.2 File Deployment

1. Copy SDK DLLs to project directories:
    - `HBISDKApi.dll`
    - `opencv_world341.dll`
    - `opencv_world341d.dll`
2. Copy runtime DLLs:
    - All VC++ runtime DLLs listed in section 2.1

7. Environment Configuration

> 🔍 **TIP**: Log out and log back in after modifying environment variables.

7.1 System PATH Configuration

1. Open System Properties:
    - Right-click This PC → Properties
    - Click Advanced system settings
    - Click Environment Variables
2. Edit System PATH:
    - Add OpenCV binary path:

      ```
      D:\opencv341\opencv\build\x64\vc14\bin
      ```

    - Add SDK binary path if needed

7.2 Verify Environment

1. Open Command Prompt
2. Type `echo %PATH%`
3. Verify added paths are present

8. Project Configuration

8.1 Initial Project Setup

1. Open project in Visual Studio
2. For VS 2022 only:
    - Project Properties → General
    - Set Platform Toolset to `v120`

> ⚠️ **WARNING**: Using incorrect platform toolset will cause build failures.

8.2 Include Directories

1. Project Properties → C/C++ → General
2. Additional Include Directories:

   ```
   D:\opencv341\opencv\build\include
   ```

8.3 Library Directories

1. Project Properties → Linker → General
2. Additional Library Directories:
   - For x64:

     ```
     D:\opencv341\opencv\build\x64\vc14\lib
     ```

   - For Win32:

     ```
     D:\opencv341\opencv\build\x86\vc14\lib
     ```

8.4 Dependencies

1. Project Properties → Linker → Input
2. Additional Dependencies:

   ```
   opencv_world341.lib
   ```

8.5 Build Configurations

1. Debug Configuration:

   - Use debug DLLs (`*d.dll`)
   - Set Runtime Library to Multi-threaded Debug DLL

2. Release Configuration:

   - Use release DLLs
   - Set Runtime Library to Multi-threaded DLL

9. Build and Test

9.1 OpenCV Test Program

```
#include <opencv2/opencv.hpp>
#include <iostream>

int main() {
```

```
    cv::Mat img = cv::imread("test.jpg");
    if (img.empty()) {
        std::cout << "Image not found or failed to load!" << std::endl;
        return -1;
    }
    cv::imshow("Display Window", img);
    cv::waitKey(0);
    return 0;
}
```

9.2 Build Verification

1. Build Configurations:
   - Debug Win32
   - Release Win32
   - Debug x64
   - Release x64
2. Check Output Directories:
   - `project_directory\output\Win32`
   - `project_directory\output\x64`

9.3 Runtime Testing

1. Place test image in executable directory
2. Run test program
3. Verify image display works
4. Test SDK demo application

10. Troubleshooting

10.1 Common Issues

1. DLL Not Found:
   - Verify PATH environment variable
   - Check DLL presence in system directory
   - Use Dependencies.exe to check DLL dependencies
2. Build Errors:
   - Check Platform Toolset setting
   - Verify include/library paths

   - Confirm matching runtime library settings

3. OpenCV Issues:

   - Verify OpenCV installation
   - Check binary compatibility
   - Confirm debug/release DLL matching

10.2 Visual Studio 2022 Specific

1. Platform Toolset Issues:

   - Verify VS 2013 build tools installation
   - Check v120 toolset selection

2. Compatibility Problems:

   - Install older platform toolsets
   - Update project settings

10.3 System Requirements Issues

1. Disk Space:

   - Clean up unnecessary files
   - Verify available space

2. Access Rights:

   - Run Visual Studio as Administrator
   - Check folder permissions

10.4 Support Resources

1. SDK Documentation
2. Visual Studio Logs:
   - Check Output window
   - Review Error List
3. System Event Logs

> 📝 **Note**: Keep this guide handy for reference during the installation process. If you encounter any issues not covered in the troubleshooting section, contact technical support.