

## EXHIBIT 3



## SDK: What Exactly is a Software Development Kit?

Software developers face all sorts of challenges in their daily work. Besides pure programming, **concepts** and **prototypes** must be designed for new applications, **development steps** planned and organized, and **interim results** analyzed. And wherever there are errors or weak points, these also need to be resolved. The trend of increasing complexity in software only makes these tasks more difficult. That's why modern development work can scarcely be imagined without aids like frameworks – ready-made code structures.

Another essential tool is software development kits (SDKs), which play a central role in typical work with **programming languages** as well as the **platform-specific development** and provision of software.

### Contents +

1. What is an SDK (Software Development Kit)?
2. What Does a Software Development Kit Contain?
3. What Rules or Licenses Might Apply to SDKs?

### What is an SDK (Software Development Kit)?

A software development kit, or SDK for short, is a **package of tools and information** that helps developers to create programs in a certain programming language, for a certain platform or application – or even enables them to do so in the first place. Compiling and releasing an SDK is up to the original developer of the respective language or hardware/software, who in turn has a natural interest in helping third-party software for their product or programming language to enter the market. In most cases, software development kits are therefore available to **use for free**, although the manufacturer can define certain rules and licenses.

#### Definition: Software Development Kit

Software Development Kit: An SDK (software development kit) is a collection of tools for developing applications for specific hardware/software or in a certain programming language. With some interpreted languages, the SDK can be identical to the run-time environment.

### What Does a Software Development Kit Contain?

The composition of an SDK varies from manufacturer to manufacturer. The scope of a kit also depends on whether it is intended for a language, operating system or hardware. Standard components contained in almost all software development kits include the **API** (or APIs) for connecting new software projects at the level of the source text. These **programming interfaces** are typically provided with extensive **documentation** that covers explanations on how to use the API as well as other useful information. This way interested developers can quickly get an overview of whether and how their planned **project** can be **implemented**.

#### Note

Instead of SDKs, some manufacturers use **alternative names** to refer to their software packages. For example, Oracle calls its collection for the distributed programming language Java "JDK" – Java (SE) Development Kit.

A good software development kit should also be supplied with all **technical core components** such as editors, libraries, run-time and development environments, compilers and debuggers (as well as other test/analysis tools, if available) if they are necessary for development or significantly simplify the development process. Likewise, special drivers and **network protocols** that are required should also be included. In some cases,

### Related Products



See packages ▶

### Web hosting for agencies

Provide powerful and reliable service to your clients with a web hosting package from IONOS.

View packages ▶

### Popular Articles

#### Website types

An overview of the 7 most popular website types...

#### How much does a website cost?

Use our typology to create a realistic estimate for the cost of your website...

#### Create a blog

This guide shows you how to start blogging with success in a few simple steps...

#### How to sell online?

Use our easy step-by-step guide to master the art of selling products online...

#### How to promote your website

From SEM, to display and mobile, we show you the most important online marketing methods...

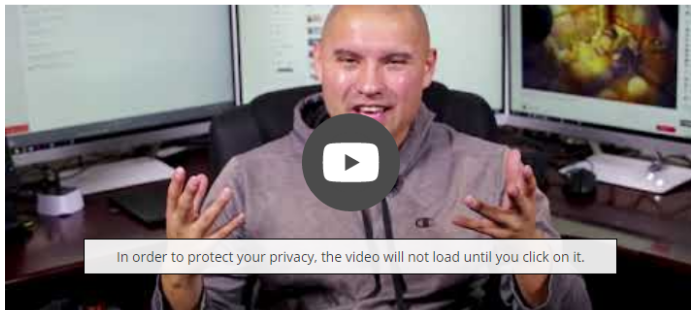
manufacturers also add examples or small test projects to their SDKs in order to make it as easy as possible for developers to get started.

Here's a summary of the components that may be found in software development kits:

- API(s)
- Documentation
- Editors
- Libraries
- Run-time and development environments
- Compilers
- Debuggers
- Drivers
- Network protocols
- Examples/test projects

## What Rules or Licenses Might Apply to SDKs?

As mentioned above, most SDKs are available for free. This is primarily because new software for a system or device represents one of the easiest ways for any manufacturer to **increase added value for users**. This is especially evident with smartphones and tablets, which would only garner a fraction of the interest they have without third-party apps. However, certain rules may be tied to the **download** and **use** of software development kits.



For example, a manufacturer may only grant access to an SDK if a user agrees not to pass on confidential information. **Non-disclosure agreements** are especially common for software with secret algorithms and products at an alpha or beta stage. To ensure that the created software is not released under another incompatible license, a software development kit can also be **subject to a license**. Both proprietary and free licenses are possible here. For this reason, developers should consider the underlying models before they start their work – as shown by the following examples:

- If an SDK has a proprietary license, it is unsuitable for developing open-source software.
- In turn, software development kits with a GNU **General Public License (GPL)** are not suitable for the development of proprietary applications.
- A software development kit provided under the free GNU Lesser **General Public License (LGPL)** can also be used for projects with proprietary code elements. However, these kinds of projects must always be created in a way that permits end users to view and modify the open-source component of the software at any time.

### Fact

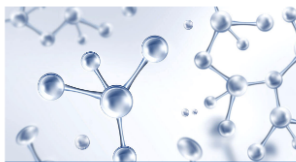
**Open-source applications** like LibreOffice or Mozilla Firefox are continuously being developed thanks to software development kits: Both amateurs and professional developers use the SDKs of these free programs to code new designs or functional expansions, which they then **make available to the whole community**.

07.10.19 | Web development

How did you like the article? 5 ★★★★★ (1)



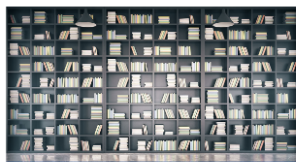
## Related articles



### Software quality assurance: module tests

04.03.2019 | Web development

Module tests are one of the most effective methods of identifying as many errors in the source code as possible early on in the software development process. The special feature: the smallest



### The most popular JavaScript frameworks and libraries

10.05.2019 | Web development

If you've ever worked with JavaScript on a website, you will know how complicated the Document Object Model (DOM) is. With their useful functions and simplifications, JavaScript frameworks



### Web frameworks – overview and classification

31.08.2017 | Web development

Frameworks are an integral part of web engineering. Instead of writing applications from scratch, programmers can rely on tried and tested software frameworks from pre-defined classes

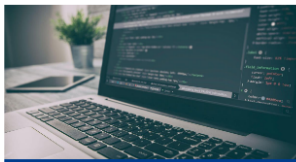
isolatable components of a program are placed in a neutral test environment. If errors appear, it's obvious they are occurring in the module being tested, which makes them easy to correct.



and libraries provide a helping hand for developers and promise a simpler programming procedure. But what are the best JavaScript libraries and frameworks? And how does a framework actually differ from a...



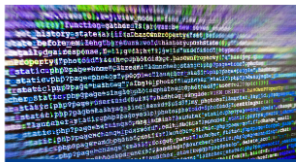
and methods for object-oriented software development. Choosing an appropriate web framework is anything but easy. We explain how different kinds of frameworks work and what you should know to find the perfect web...



### Jenkins: Tutorial for the CI-Tool

© 13.03.2019 | Web development

Continuous integration can improve the work of software development: fewer errors and more efficient, continuous integration. But this requires a simple yet comprehensive tool. That's why we're introducing Jenkins. This software makes it easier for you to work with automatic builds and individual pipelines. Our Jenkins tutorial will help you along the way!



### Web components explained

© 02.04.2020 | Web development

No web developer can manage without HTML, CSS, and JavaScript. As a result, the same elements tend to be repeated over and over again when programming, which can be both tedious and time-consuming. Web components help with this by providing encapsulated HTML tags and allowing them to be reused as often as needed.



# Microsoft® Windows® SDK Release Notes for Release Candidate 1

Version RC1.1.0.0

---

## 1. Introduction

Welcome to the Microsoft® Windows® Software Development Kit (SDK) for Release Candidate 1 (RC1). The Windows SDK contains documentation, samples, headers, libraries, and tools designed to help you develop Microsoft® Windows applications. The documentation, samples, and tools provided in the Windows SDK support application programming interfaces (APIs) available in the Windows Vista, Microsoft® Windows® XP Service Pack 2 (SP2), and Microsoft® Windows® Server 2003 Service Pack 1 (SP1) versions of Windows. Not all API support all platforms. You can look under the specific API documentation to determine which platform the API supports.

### Table of Contents

[Introduction](#)

[License Agreement](#)

[Supported Compiler and Platforms](#)

[File System Layout](#)

[Installation Instructions](#)

[Known Issues](#)

[Setup and General Development](#)

[Build Environment](#)

[Documentation](#)

[SDK Tools](#)

[Samples](#)

[Windows SDK Product Support](#)

[Feedback](#)

**Important** This version of the Windows SDK is available as both a DVD ISO image that must be burned to a DVD or mounted as a virtual drive and installed from that media, and as a web download.

The DVD ISO image of [the Windows SDK available from Download Center](#) will not install the SDK directly on to your hard disk without the use of additional tools. See the Installation Instructions below for more information. The DVD ISO image requires a minimum of 1.5 GB of free disk space to install. If you do not have that much free disk space, you can use the SDK's custom setup options to install a subset of the content.

The web download is also available from the Download Center. The web download will stream the SDK content to your local machine.

**Important** This version of the Windows SDK targets specific versions of Windows and has other restrictions. This version:

- Is supported on Windows Vista Release Candidate 1, which is available from the MSDN Universal Subscribers site on MSDN and from the Beta customer locations.
- Supports .NET Framework 3.0 Release Candidate 1 for Windows XP SP2 and Windows Server 2003 SP1, which is available publicly from the Download Center.

**Important** To optimize your Windows SDK setup experience, we strongly recommend that you install the latest updates and patches from [Microsoft Update](#) before you begin installing the Windows SDK.

**Please note that this release is for preview purposes only. The APIs, documentation, samples, headers, libraries, and tools provided in this release are preliminary and subject to change.**

- It is strongly recommended that you install this pre-release version of the Windows SDK in a test environment, not a production environment. This document identifies many known issues. Please read the Installation Instructions and Known Issues sections before installing.
- To learn more about Known Issues and to find additional information about Windows Vista, see [The MSDN Online Windows Vista Developer Center](#). The Developer Center is focused on keeping developers up-to-date on how they can take advantage of Windows Vista's rich new client features in their applications. Visit the Developer Center often to see new articles, read monthly columns, participate in the [Windows Vista Beta Newsgroups](#), download sample code and read the Editor's Web Log.

**Important** When building samples, do so in a directory outside of Program Files to which you have write access. For more information, see the Samples section below.

---

## 2. License Agreement

The contents included in the Windows SDK are licensed to you, the end user. Your use of the SDK is subject to the terms of an End User License Agreement ("EULA") accompanying the SDK and located in the \License subdirectory. You must read and accept the terms of the EULA before you access or use the SDK. If you do not agree to the terms of the EULA, you are not authorized to use the SDK.

---

## 3. Supported Compiler and Platforms

This pre-release of the Windows SDK supports the RTM release of Microsoft® Visual Studio® 2005. You can download a copy at no charge from the [Visual Studio 2005 Developer Center](#).

This pre-release of the Windows SDK supports x86, x64, and IA64 platforms for building and running samples.

---

## 4. File System Layout

By default, the Windows SDK is installed to your hard disk in the locations described in the following table. This list is not complete, but covers the most common directories.

Directory	Contents
\Bin	Windows SDK tools
\Help	Windows SDK documentation
\Include	Windows SDK headers
\Lib	Import libraries and TLB files
\License	Windows SDK license information
\Samples	Windows SDK samples

## 5. Installation Instructions

**Please install on a clean machine or completely uninstall any pre-releases** of the Microsoft® WinFX® Runtime Components 3.0, the WinFX SDK, the Platform SDK, the Windows SDK, the .NET Framework redistributable, Microsoft® Visual Studio®, and their dependencies before installing this release. These older components may interfere with this release, causing setup to fail or break functionality. We recommend installing this version on non-production machines (preferably in a test environment) without these previous releases to prevent incompatibilities.

### To install on Microsoft Windows XP Service Pack 2 or Microsoft Windows 2003 Server Service Pack 1:

1. If you intend to install .NET Framework 3.0 components available in the Windows SDK, install the .NET Framework 3.0 Release Candidate 1 from the Download Center. **Note** If you download the DVD ISO image, burn the image to a DVD or mount as a virtual drive and install the SDK on your local hard disk. The details page for the Windows SDK in the Download Center has additional information about tools you can use to create the setup image without burning the image to a DVD. You can also install the Windows SDK by using web setup.
2. If you create a DVD ISO image or mount a virtual drive and launch setup.exe, choose the appropriate option to install the Windows SDK. Insert the SDK DVD into a DVD-ROM drive and click setup.exe, or after the virtual drive is mounted, launch the setup.exe from within that virtual drive.
3. Follow the instructions in the Windows SDK Setup wizard.
4. **Optional** Install Microsoft® Visual Studio® 2005.
5. **Optional** Install the Visual Studio 2005 Extensions for .NET Framework 3.0 from the Download Center.

The .NET Framework 3.0 is available by default in Windows Vista.

**Note** If you intend to use Visual Studio to develop .NET Framework 3.0 applications, you must install the Windows SDK before installing the Visual Studio 2005 Extensions for .NET Framework 3.0. It is essential that you use the version of Visual Studio identified in the Installation Instructions in .NET Framework 3.0 setup.

Access the Windows SDK through the Start menu at Microsoft Windows SDK. The Start menu folder contains pointers to the documentation (which contains the samples for .NET Framework 3.0), tools, debug and release build environments. Documentation for the Windows SDK is also available on MSDN Online at <http://windowssdk.msdn.microsoft.com/library>.

## 6. Known Issues

This release has the following known issues. Known issues are categorized by content type.

### 6.1 Setup and General Development

#### 6.1.1 Windows SDK Disk Space Requirements

The complete DVD ISO installation of the Windows SDK requires 1.5 GB or more disk space to install successfully. Please verify that the computer you are installing to has at least the minimum required disk space before beginning setup. If the minimum required disk space is not available, setup will return a fatal error.

#### 6.1.2 Several SDK Components Must be Uninstalled Separately in Add/Remove Programs

The Windows SDK installs both its core components and external applications as separate MSI files. Please note that several applications will need to be uninstalled from Add/Remove Programs in order to remove all SDK components. Those components are:

- Debugging Tools for Windows
- Microsoft Windows Software Development Kit
- Microsoft Windows Software Development Kit Headers and Libraries
- Microsoft .NET Compact Framework 2.0
- Microsoft FxCop 1.3.5
- Windows PowerShell

Due to SDK componentization, we recommend re-running the SDK setup.exe from your download location and choosing which features to install. For example, if you choose "Microsoft Windows Software Development Kit" from Add/Remove Programs and then "Change," you will see a different screen that does not show all the internal and external content installed as MSI files.

If you originally only installed the x86 development environment and now want to install the x64 development environment, you need to relaunch the SDK setup.exe versus launching the "Change" functionality for the "Microsoft Windows Software Development Kit" located in Add/Remove Programs in the Control Panel.

#### 6.1.3 VSS Software Provider ID is required to Use Some APIs

The VSS Software Provider ID is needed to use various APIs, including the shadow copy storage management interfaces (IVSSSnapshotMgmt).

To add this provider ID, define a const GUID and use that in your code.

```
// {b5946137-7b9f-4925-af80-51abd60b20d5}
```

```
static const GUID VSS_SWPRV_ProviderID =
```

```
{ 0xb5946137, 0x7b9f, 0x4925, { 0xaf, 0x80, 0x51, 0xab, 0xd6, 0xb, 0x20, 0xd5 } };
```

In a few cases, you may need to use the Software Provider GUID to access some APIs (GetProviderMgmtInterface method on the IVSSSnapshotMgmt interface).

Vssadmin list providers can also obtain the GUID.

#### 6.1.4 Known issues with Windows Workflow Foundation (WF)

There are a few known issues with Windows Workflow Foundation (WF) in this SDK release. See the online [Windows Workflow Foundation Runtime Components Readme](#) for a complete list.

#### 6.1.5 Visual Studio 2005 is Required for .NET Compact Framework Development

The SDK provides documentation and sample code that targets the .NET Compact Framework. However, actual development of .NET Compact Framework applications requires Visual Studio 2005.

#### 6.1.6 Windows Presentation Foundation Imaging Component Does Not Support Metadata That Spans Multiple APP Segments

JPEG APP headers do not support metadata segments larger than 64K. The Windows Presentation Foundation Imaging Component does not support metadata that spans multiple APP segments. If you add a color profile larger than 64K to the APP2 metadata marker of a JPEG image, the color profile may become corrupt if the image or the metadata are modified.

There is no workaround for this issue.

## 6.2 Build Environment

The following section describes and offers possible workarounds for known issues with compilers and known issues with headers and libraries included in the Windows SDK.

#### 6.2.1 Headers and Libraries: Using the Windows SDK to Develop Win32 C++ Applications With Visual Studio 2005

In order to utilize Windows SDK headers, libraries, and tools within Visual Studio 2005, the SDK-provided Visual Studio registration tool must first be run. The Visual Studio registration tool must run in elevated administrator mode on a Windows Vista computer. When running this tool on a Windows Vista machine it must be run for each user.

To run the Visual Studio Registration Tool, go to

Start --> All Programs --> Microsoft Windows SDK --> Visual Studio Registration--> Register Windows SDK Directories with Visual Studio 2005.

#### 6.2.2 Compilers: The J# Compiler is Not Available in This Version of the Windows SDK

J# samples will not build using the Windows SDK because there is no appropriate build environment.

There is no workaround. This edition of the Windows SDK does not support building J# applications.

#### 6.2.3 Compilers: Debugging Windows Presentation Foundation (WPF) Applications with Kernel Debugger Active May Result In an Error

When debugging WPF applications, the following error message may appear:

Debugging is not possible because a kernel debugger is enabled on the system.

This message occurs while debugging managed code on a system running Windows<sup>®</sup> NT, Windows<sup>®</sup> 2000, or Windows XP that has been started in debug mode.

#### 6.2.4 Msbuild Reports a Dependency upon the Microsoft .NET Framework SDK 2.0

The Windows SDK does not set the HKLM\Software\Microsoft\.NETFramework\sdkInstallRootv2.0 registry key to a string value containing the root directory of the Windows SDK installation. However, some MSBuild tasks may expect this registry key to be set. If you already have the .NET Framework SDK 2.0 or Visual Studio 2005 set this key would be set and you should not encounter a problem. However, if you install the Windows SDK without either the .NET Framework SDK or Visual Studio 2005, you may receive an error message from MSBuild tasks with a dependency on this key.

To work around this issue set the string value of this key to the root directory of the Windows SDK installation. By default, this is: C:\Program Files\Microsoft SDKs\Windows\v6.0

## 6.3 Documentation

#### 6.3.1 Some Links Are Broken In the SDK Documentation

Some links in the class library and other areas of SDK documentation are broken. These will be fixed in future releases. There is no workaround for this issue.

#### 6.3.2 No "How Do I" Articles Are Available From the Windows SDK Documentation Viewer

The How Do I feature might be removed in a future release. There is no workaround for this issue.

#### 6.3.3 Some In-Page Language Filter Options May Filter Out Required Syntax Blocks in WPF Reference Pages

In some cases, both XAML and either C# or VB code are required to successfully build and run samples. Applying either the C#, VB, or XAML filters alone may exclude necessary syntax from the page view. This is currently by design, and the design may be revisited in a later release.

#### 6.3.4 Some Filters May Not Work As Expected

Filters may not work as expected. This is a known issue and will be fixed in a future release.

### 6.3.5 WM\_GETDLGCODE now uses wParam parameter.

Previously, the wParam parameter of the WM\_GETDLGCODE notification was not used. That parameter now indicates a virtual key pressed by the user. It is important that the handler to react to this parameter. For example, the handler could selectively handle VK\_RETURN, but delegate VK\_TAB to the owner window.

### 6.3.6 WmpBitmapEncoder Property Type Changes

A late change to the code has changed the WmpBitmapEncoder properties HorizontalTileSlices and VerticalTileSlices from a System.UInt16 to a System.Int16 value for CLS compliance.

## 6.4 SDK Tools

This section describes known issues with Windows SDK tools and possible workarounds.

### 6.4.1 Windows Powershell Can Be Downloaded Directly

Windows PowerShell was available to customers and partners as a part of the Windows SDK. This is no longer the case. Now, Windows PowerShell can be downloaded directly.

- [X86](#)
- [X64](#)
- [IA64](#)

For more information, see the [Windows PowerShell team blog](#).

### 6.4.2 Tb3x.Exe Has Been Deprecated and Removed From the Windows SDK

Tb3x.exe has been deprecated and removed from the Windows SDK. There is no known workaround for this issue.

### 6.4.3 The Application Verifier tool is Not Available in the Windows SDK

The Application Verifier tool is not available in the Windows SDK. It ships as part of the Windows Application Compatibility Toolkit.

To work around this issue, download the Windows Application Compatibility Toolkit from Download Center.

### 6.4.4 Uispy.Exe Fails to Run and Displays a Strong Name Validation Error

UISpy sometimes fails to run and displays a strong name validation error.

This error occurs because UISpy.exe was delay signed and never resigned. By default, this causes the .NET runtime to fail the execution of UISpy.exe because the strong name signature in the executable is not valid. However, UISpy will run correctly if you configure the runtime to ignore the strong name signature for UISpy.exe.

To work around this issue, register UISpy.exe for strong name verification skipping.

1. Open a Windows SDK command prompt.
2. Navigate to the /bin directory. By default, this is C:\Program Files\Microsoft SDKs\Windows\v1.0\bin.
3. Run the command `SN -Vr Uispy.exe`. For example:

```
C:\Program Files\Microsoft SDKs\Windows\v1.0\Bin>sn -Vr Uispy.exe
```

The successfully run command will produce a message indicating that the verification entry was added for the UISpy assembly.

### 6.4.5 Possible Application Exception after Removing localIssuer Address Using SvcConfigEditor.exe

If you use SvcConfigEditor.exe to delete the address value in the localIssuer element of clientCredential in the WCF Endpoint Behavior, the tool will create an invalid configuration file and your application will throw an exception. This is because SvcConfigEditor.exe doesn't remove localIssuer when the localIssuer value is an empty string (an unexpected value).

To work around this issue, manually remove the localIssuer element using some other editor, such as Notepad.

### 6.4.6 SvcTraceViewer.exe Does Not Display Arabic Characters Correctly

In traces that contain Arabic characters, the XML view in SvcTraceViewer.exe may render the characters in reverse order.

To work around this issue use the Formatted View, which displays the traces in the correct character order. You can also copy the text from the XML view and paste it into some other editor that displays the characters in the correct order.

### 6.4.7 Managed C++ Code Generated Using SvcUtil.exe Tool May Not Compile

You can use the SvcUtil.exe tool to generate code for web service proxies and data types from metadata. However, there are known issues with the C++ code provider in Visual Studio 2005 and the .NET Framework 2.0 SDK that can cause the tool to generate Managed C++ code that may not compile.

### 6.4.8 Code Generation Language Support in SvcUtil.exe

SvcUtil.exe can generate code for web service proxies and data types from metadata in any language that has a managed code provider. SvcUtil.exe has been tested with the C#, Visual Basic, and C++ managed code providers. Other code providers have not been tested and may generate code that does not compile or may be otherwise unusable.



### 6.4.9 SvcConfigEditor.exe Reloads the Config File When It Is Changed by Another Entity

If you're using SvcConfigEditor.exe to edit a config file and another process or editor accesses that file, SvcConfigEditor.exe will automatically reload the file. For example, this can happen when anti-virus software scans the config file. The reload will cause you to lose any changes made to the config file since SvcConfigEditor.exe last saved it.

To work around this issue, make sure SvcConfigEditor.exe is the only process that accesses the file during a session.

## 6.5 Samples

This section describes known issues with Windows SDK samples and possible workarounds.

**Note** Some .NET Framework 3.0 samples have specific dependencies or setup requirements. If you find that a sample in the samples .zip files is not working properly, first check the sample page in the documentation for instructions.

### 6.5.1 Building Samples from a Directory Other Than the Sample Directory

When building Samples, do so in a directory outside of Program Files to which you have write access. To build from the command line:

1. Open either the CMD Shell or PowerShell command prompt.
2. Copy the sample to a working folder not under Program Files. Copying to a location other than Program Files makes it possible to maintain a pristine copy of the samples installed by the SDK as well as avoid issues when writing to files and directories located under Program Files.
3. Build the sample from the command line as follows:
  - Build a makefile by typing `nmake`.
  - Build a .csproj file by typing `msbuild mysample.csproj`.
  - Build a .vbproj file by typing `msbuild mysample.vbproj`.
  - Build a .vcproj by typing `vcbuild mysample.vcproj`.

### 6.5.2 Not All C++ Samples with Visual C++ 2005 Project Files Have Configurations for X64

Not all samples with Visual C++ 2005 project files have configurations to build for X64. To workaround this just load the sample in Visual C++ 2005 and update Configuration Manager under Project | Properties. Additionally, if you do not install libraries for all CPU architectures, some samples with Visual C++ 2005 project files may fail to build (link fails with "fatal error LNK1181: cannot open input file") for all configurations in the project file. For example, if a sample has an x86 configuration and no x86 libraries were installed (the default install setting when installing the SDK on a X64 machine), that sample's x86 configuration will fail to build.

### 6.5.3 Where to Find .NET Framework 3.0 Samples

Samples demonstrating .NET Framework 3.0 can be found individually in the SDK documentation. Each sample has its own description page. From each sample page, you can view the source files for those samples as well as individually download all the files of that sample to a chosen location. The .NET Framework 3.0 samples can be accessed in bulk by directly opening the sample .zip files located in %MSSDK%\Samples.

### 6.5.4 Building an Application with the SDK and ATL or MFC

The following samples require Visual Studio 2005 to build either because an ATL or MFC header is required that Visual Studio 2005 ships.

The SDK does not come with either ATL or MFC headers, libraries, or runtimes. However, if you have Visual C++ 2005 (non-Express editions) installed on your system, you can use the SDK to build using ATL or MFC.

1. Open an SDK command window.
2. Type `cd %MSSDK%\Setup .`
3. Run `VCIntegrate.exe .`
4. From this point forward, command windows will now have the proper environment settings to build an application that takes advantage of the SDK.

**Note** If you want to disable support, run `VCIntegrate /u` to uninstall changes and return to command windows with no ATL or MFC support.

The following samples require an ATL or MFC header that Visual Studio 2005 ships, but the Windows SDK does not ship.

- `\com\administration\explore.vc`
- `\multimedia\audio\aecmicarray`
- `\multimedia\audio\multichan\`
- `\multimedia\gameux`
- `\multimedia\gdi\icm\devicemodelplugin`
- `\multimedia\gdi\icm\gamutmapmodelplugin`
- `\multimedia\wia\getimage`
- `\multimedia\windowsmediaservices9\authentication`
- `\multimedia\windowsmediaservices9\cacheproxy\cplusplus`
- `\multimedia\windowsmediaservices9\datasource`
- `\multimedia\windowsmediaservices9\eventnotification`
- `\multimedia\windowsmediaservices9\playlist`

- \multimedia\wpd\wpdapisample
- \netds\adsi\samples\activedir\propsheethost\
- \netds\adsi\samples\dsui\scpext\scpcmenu\
- \netds\adsi\samples\dsui\scpext\scpproppage\
- \netds\adsi\samples\dsui\scpext\scpwizard\
- \netds\adsi\samples\dsui\userext\userproppage\
- \netds\adsi\samples\dsui\userext\userwizard\
- \netds\adsi\samples\general\adqi\
- \netds\adsi\samples\general\adserror\
- \netds\adsi\samples\general\adsidump\
- \netds\eap\eaphost
- \netds\fax\devicesetting\
- \netds\fax\faxaccount\
- \netds\fax\faxjoboperations\
- \netds\fax\faxreassign\
- \netds\fax\faxsecurity\
- \netds\fax\foldersenum\
- \netds\fax\outboundrouting\
- \netds\fax\routingextension\
- \netds\fax\sendfax\
- \netds\fax\serverconfig\
- \netds\messagequeuing\c\_draw
- \netds\messagequeuing\imp\_draw\
- \netds\messagequeuing\mqapitst
- \netds\messagequeuing\mqapitst\
- \netds\messagequeuing\mqf\_draw
- \netds\messagequeuing\mqpers\
- \netds\nap\sdqec
- \netds\nap\shv
- \netds\netmon\experts\blrplate\
- \netds\ras\customdial\
- \netds\ras\phoneentry\
- \netds\rras\eap\
- \netds\rtc\client\cpp\rtcincoming\
- \netds\rtc\client\cpp\rtcoutgoing\
- \netds\rtc\client\cpp\rtcsample\
- \netds\tapi\tapi3\cpp\msp\samplemsp\
- \netds\tapi\tapi3\cpp\pluggable\
- \netds\upnp\dco\_dimmerservice\
- \netds\upnp\genericucp\cpp\
- \netds\upnp\registerdevice\
- \netds\wlan\autoconfig
- \security\authorization\azman\azmigrate
- \security\capicom\cpp\isapicertpolicy
- \security\certservices\exit\c++\windows.net\
- \security\certservices\exit\c++\windows2000\
- \security\certservices\policy\c++\windows.net\
- \security\certservices\policy\c++\windows2000\
- \security\credentialprovider
- \web\bits\bits\_ie\
- \web\bits\upload\
- \web\iis\components\cpp\intermediate
- \web\iis\components\cpp\simple\
- \web\iis\rscs
- \web\networking\asyncdemo\
- \winbase\cluster\win2003\clipbookserver\clipbook\_serverex\
- \winbase\cluster\win2003\filesharesample\file share sampleex\
- \winbase\rdc
- \winbase\storage\optical\imapi2sample
- \winbase\vss\vshadow

- \winbase\vss\vsssampleprovider
- \winbase\wtsapi\wtsapi\
- \winui\globaldev\cssamp\
- \winui\globaldev\globaldev\
- \winui\muisampleapp
- \winui\pictures\acquisitionplugin
- \winui\pictures\pictureacquisition
- \winui\pictures\pictureacquisitionplugin
- \winui\shell\columnprovider\
- \winui\shell\shelliconoverlay\
- \winui\sideshow\pictures\
- \winui\sideshow\tasks\
- \winui\tsfapp\
- \winui\tsfcompart\

### 6.5.5 Must Install Peer Networking Feature on Microsoft Windows XP SP2 Before Running WCF Peer Channel Samples

The documentation for three of the WCF samples is missing instructions on how to install the Peer Name Resolution Protocol (PNRP) on a computer running Windows XP SP2. These samples are:

- Net Peer TCP Sample
- Peer Channel Secure Chat Sample
- Peer Channel Custom Authentication Sample

To install PNRP on Windows XP SP2 (one-time setup):

1. In Control Panel, double click Add or Remove Programs.
2. In the Add or Remove Programs dialog box, click Add/Remove Windows Components.
3. In the Windows Components Wizard, select the Networking Services check box, and then click Details.
4. Check the Peer-to-Peer check box and click OK.
5. Click Next in the Windows Components Wizard.
6. When installation completes, click Finish.
7. From a command shell, start the PNRP service using the following command

```
net start pnrpsvc
```

### 6.5.6 Some Netds/Peer To Peer Samples Need To Be Run From A Non-Elevated Window

The following samples cannot be run from an elevated window:

- Contacts
- Invitation
- People Near Me

To work around this issue, run these applications from a non-elevated window.

### 6.5.7 Run Error in Windows Presentation Foundation Blackjack Demo Sample

The Windows Presentation Foundation Blackjack demo sample has a known platform animation issue. When a card is drawn for either the player or the dealer, it may end up being placed on the source deck, rather than the player's or dealer's deck about 10% of the time. There is no workaround.

### 6.5.8 Need to Change Visual Studio 2005 Project Files

When building in Visual Studio 2005, the project files for the following .NET Framework 3.0 samples only support building for Win32 | Debug only after you actively select that configuration. The other configurations do not correctly build.

- BEREncoding
- DsmI\_BatchRequest
- DsmI\_Exceptions
- DsmI\_PagedSearch
- DsmI\_ReadRootDSE
- DsmI\_SendRequest
- Ldap\_AsyncSend
- Ldap\_Exceptions
- Ldap\_PagedSearch
- Ldap\_PrintDomain
- Ldap\_ReadRootDSE
- Ldap\_Referral

- Ldap\_SendRequest

### 6.5.9 The SchemaReader Sample Requires Windows Media Player 11

The Windows Media Player SchemaReader sample requires Windows Media Player 11 to be installed.

### 6.5.10 Digital Signal Processing (DSP) Plug-Ins Created By the Windows Media Player Plug-In Wizard Might Not Work With Windows Media Player 11

Digital signal processing (DSP) plug-ins created using the Windows Media Player plug-in wizard are single-threaded. This release of Windows Media Player 11 requires that DSP plug-ins be multi-threaded. Attempting to load a single-threaded DSP plug-in into Windows Media Player 11 may yield unexpected results.

Microsoft is aware of this issue and is working to provide support for existing single-threaded DSP plug-ins in future releases of Windows Media Player 11.

### 6.5.11 Windows Media Player Online Store Plug-In Wizard Does Not Create Type 1 Plug-Ins

The Windows Media Player online store plug-in wizard can be used to create a plug-in for a type 2 online store, but it cannot be used to create a plug-in for a type 1 online store.

Microsoft is aware of this issue and is working to provide support for type 1 plug-ins in a future release of the wizard. To work around this issue, you can create a type 1 plug-in manually.

### 6.5.12 MMC 3.0 Samples Require that MMC 3.0 is Installed

When the Windows SDK is installed on an x86 Windows Server 2003 SP1 machine, MMC 3.0 samples do not compile. To work around this issue, install MMC 3.0 components.

### 6.5.13 The Security Certificate Services Ocspsample Does Not Build On or For Windows Vista

There is no workaround for this issue.

### 6.5.14 The Following Windows Media 9 Samples Require Additional Steps to Build

The following samples require additional steps to build:

- \Samples\Multimedia\WindowsMediaServices9\Logging
- \Samples\Multimedia\WindowsMediaServices9\CacheProxy\csharp

To build the Logging and CacheProxy samples:

1. Load each of the samples' .sln file into Visual Studio 2005 and allow it to upgrade the project.
2. Add a reference to microsoft.windowsmediaservices.dll by going to Project | Add Reference... | Browse and then browse to the SDK's include directory and select microsoft.windowsmediaservices.dll and click OK.
3. Build the solution.

### 6.5.15 The Following TAPI Samples Depend Upon the Directshow Baseclasses

The following TAPI samples depend upon the DirectShow BaseClasses (found in the SDK Sample directory). If the sample directory is not installed to the default location, the makefile for the following samples will need to be adjusted to refer to the new location:

- \netds\tapi\tapi3\cpp\pluggable
- \netds\tapi\tapi3\cpp\tapirecv
- \netds\tapi\tapi3\cpp\tapisend

### 6.5.16 The 3d9exsample Sample Requires DirectX SDK to Build

The sample multimedia\direct3d\d3d9exsample requires the DirectX SDK be installed and the INCLUDE and LIB environment variables updated to build. The DirectX SDK include directory should be appended to the INCLUDE environment variable, from within a SDK command prompt. Additionally the DirectX SDK library directory also must be appended to the LIB environment variable, from within a SDK command prompt. This will allow the sample to be built using the supplied makefile.

### 6.5.17 Samples Needing Administrative Privilege to Build

The following samples need administrative privilege to build on Windows Vista:

- multimedia\windowsmediaservices9\authentication
- multimedia\windowsmediaservices9\authorization
- multimedia\windowsmediaservices9\eventnotification
- multimedia\windowsmediaservices9\cacheproxy\cplusplus
- multimedia\windowsmediaservices9\datasource
- multimedia\windowsmediaservices9\playlist
- web\iis\components\cpp\intermediate
- winbase\rdc

### 6.5.18 WMI Decoupled Sample Missing precomp.h

The sample sysmgmt\wmi\vc\decoupled reports that the file precomp.h is missing and thus fails to build. The file precomp.h is located in sysmgmt\wmi\vc\decoupled and should instead be located under sysmgmt\wmi\vc\decoupled\include in order for the sample to build.

### 6.5.19 Workflow Samples

The build instructions for many of the Workflow samples say to use the Microsoft .NET Framework SDK 2.0 Command prompt. In fact, all of these samples are buildable using the Windows SDK command prompt.

### 6.5.20 Cross-Technology Sample Build Issues

Two "Cross-Technology" samples have build issues.

- SafePad does not build with the current version of the SDK.
- RecipeViewer ConsoleHost.exe builds, but does not run. The workaround is to open the configuration file app.config in the ConsoleHost project and delete any attributes or elements that are related to the behaviors element, then rebuild sample.

### 6.5.21 WCF Samples Setup Batch File May Not Work with IIS7

The WCF samples setup script, Setupvroot.bat, works on Windows Vista only if you have Internet Information Services (IIS) 6.0 compatibility installed. To do this, either use the manual setup instructions provided in the SDK or set up IIS 6.0 compatibility by doing the following:

1. Click the Start button and select Control Panel.
2. Select Programs in the Control Panel dialog.
3. Select Turn Windows Features on or off.
4. The User Account Control dialog is displayed. Click Continue.
5. The Windows Features dialog is displayed. Expand the item labeled Internet Information Services.
6. Expand the item labeled World Wide Web Services.
7. Expand the item labeled Application Development Features.
8. Make sure the following items are selected:
  - .Net Extensibility
  - ASP.NET
  - ISAPI Extensions
  - ISAPI Filters
9. Expand the item labeled Web Management Tools and make sure IIS Management Console is selected.
10. Expand the item labeled IIS 6 Management Compatibility and make sure IIS 6 Scripting Tools is selected (other options auto-select when you do this).
11. Click OK.

### 6.5.21 Where to Get Working Versions of WCF Samples That Do Not Build or Have Errors

The following WCF samples do not build in the Windows SDK RC1. Go to <http://wcf.netfx3.com> using the links in the following table to get versions that build.

Non-Building WCF Sample	Link to Sample That Builds
TechnologySamples\Basic\Management\ETWTracing\CS	<a href="http://wcf.netfx3.com/files/folders/manageability_sample/entry5132.aspx">http://wcf.netfx3.com/files/folders/manageability_sample/entry5132.aspx</a>
TechnologySamples\Extensibility\DataContract\Surrogate\CS	<a href="http://wcf.netfx3.com/files/folders/defining_and_using_contracts/entry5131.aspx">http://wcf.netfx3.com/files/folders/defining_and_using_contracts/entry5131.aspx</a>
TechnologySamples\Extensibility\Security\SupportingTokens\CS	<a href="http://wcf.netfx3.com/files/folders/tokens/entry5133.aspx">http://wcf.netfx3.com/files/folders/tokens/entry5133.aspx</a>

The TechnologySamples\Extensibility\Transport\Udp\CS sample hangs on client.Close(). The working version of the sample is available at [http://wcf.netfx3.com/files/folders/transport\\_channels/entry5235.aspx](http://wcf.netfx3.com/files/folders/transport_channels/entry5235.aspx).

The Basic\Binding\WS\MessageSecurity\Username\*\Web.config has an incorrect value for the serviceCertificate findValue attribute. The full line should read: <serviceCertificate findValue="localhost" storeLocation="LocalMachine" storeName="My" x509FindType="FindBySubjectName" />

A full refresh of all WCF samples is available at <http://wcf.netfx3.com/content/resources.aspx> under "Documentation Updates."

## 7. Windows SDK Product Support

The pre-release Windows SDK is provided "as-is" and is not supported by Microsoft. If you have questions or need assistance, [several newsgroups are available](#) to help you. If you have questions or need assistance, see [The MSDN Online Windows Vista Developer Center](#) and the [Windows Vista Beta Newsgroups](#) for additional information and updates to the known issues.

## 8. Feedback

Your feedback is important to us. Your participation and feedback through the Windows Vista Beta Newsgroups is appreciated.

Copyright © 2006 Microsoft Corporation. All rights reserved. [Legal Notices](#).

[Blog](#) > [What is a SDK and an API?](#)

Sep 14, 2020

## What is a SDK and an API?

Lat. 42.351994 Long. -71.047663

Posted by  
**Oleg Kamenetsky**[Read All Posts](#)

All mobile applications utilize several APIs (Application Programming Interfaces) and SDKs (Software Development Kits) to increase standardizations and add new useful features. These software packages allow for the addition of unique capabilities that most developers would not otherwise be able to deploy without many dedicated resources. With this being said, there is often confusion between what an SDK and API are, and why an app would want to integrate either of these third-party software options. In this post, we will define what APIs and SDKs are and discuss the differences between them.

### What is an API?

Broadly speaking, an **Application Programming Interface (API)** is a set of subroutine definitions, communication protocols, and tools for building software. In general terms, it is a set of clearly defined methods of communication among various components.

There are two main types of APIs, server-to-server or client-to-server, and component-to-component. Server-to-server and client-to-server APIs are commonly a stateless interface to a remote server to *set* or *get* data. Typically, these types of APIs use a "RESTful" (representational state) protocol which requires them to be "stateless". Stateless protocols require that any call to the API be completely independent from any past or future calls to the API. The API defines a communication protocol and "expected behavior" that allows a client to make requests and receive responses to and from a server. It is the developer's responsibility to integrate an API into their solution and use the API correctly. Component-to-component APIs can act a bit differently. These APIs allow different software packages to communicate on the same device and are not communicating with an external server or device. A good example of this would be Android's set of APIs.

APIs work on all operating systems and are commonly used on IoT devices and for device tracking & management platforms. This makes them a great solution for customers that use proprietary operating systems and protocols as well as end-to-end vertical solution providers.

Skyhook offers an API as one of four flexible integration options to access its Precision Location software. When utilizing this location API, the developer is required to source the Wi-Fi, GPS, and cellular data, and then implement the API request according to the specifications outlined.

### **What is an SDK?**

A **Software Development Kit** (SDK) is commonly defined as a set of tools that can be used to create and develop applications. In general, an SDK refers to a full-suite software module that includes everything developers need for a specific module within an app. They are often compared to a code library, which is a collection of non-volatile resources, used by computer programs, often for software development. They often “wrap” (fully or partially implement) an API and utilize the API’s communication protocols in the appropriate and intended manner. They include examples, documentation and, the methodology necessary to perform specific functions.

For example, Skyhook’s Precision Location SDK provides all the tools and documentation needed to perform highly accurate geo-positioning on a mobile device using the Android operating system. The app developer simply needs to load, or dynamically pull, the SDK library into their development environment and use the available interfaces to utilize all the capabilities Skyhook provides. No extra code is needed to source the Wi-Fi, GPS, or cellular data to calculate an accurate position - this is all handled by the SDK.

Additional benefits of Skyhook’s SDK include location smoothing, a fast time-to-fix, MAC address collapsing, stationary detection, offline location, and power optimization. See how each of these features can add value to connected devices:

- Location Smoothing – Algorithms that use historic location data and patterns to inform device locations in the future.
- MAC Address Collapsing – Intelligent filtering to detect multiple MAC addresses associated with one router. This helps evenly weight all access points used for location, without creating a bias towards a particular access point.
- Stationary Detection – Patented method for determining if a connected device is stationary, through measuring the Doppler frequency of received satellite signals.
- Offline Location – Determine device location, even when offline, by collecting a token that gets replayed once devices are back online.
- Power Optimization – Technology to minimize the number of access points that need to be scanned to yield accurate location results and reducing the time it takes to complete scans for access points.

The SDK is a great solution for applications across smart and feature phones, desktops, laptops, wearable devices, cameras, and point-of-sale devices. It has broad integration options across operating systems as well, with compatibility including custom variants of the Android operating system, Linux, Windows, MacOS, and Raspberry Pi.

### **Why use an API?**

APIs are typically much lighter weight than SDKs since a full library does not need to be integrated into the app or the APIs are native to the development platform. APIs are useful when you only need one or two specific functions of an SDK and a fully integrated experience is not needed.

Developers should keep in mind that implementing an API often requires more programming upfront as they are less “plug-and-play” than SDKs. Finally, APIs are also useful for applications where a codebase is programmed on atypical platforms.

**Why use an SDK?**

SDKs provide developers with an array of tools needed to provide end-users of an app a unique and standardized interface. Developers also don’t need to reinvent the wheel when it comes to standard app features, such as location, data storage, geofencing, user authorization, etc. SDKs have a set of robust functionality -- error handling, consistent performance, code reuse -- that takes a lot of the complexity away for software developers. SDKs also ensure that any API provided is implemented correctly, reducing the headache that could be associated with implementing an API. The SDK often goes further and provides enhanced features on top of APIs that allow for improved user experiences within your app. Finally, SDKs also allow for easier upgrade paths and handling of deprecations for certain lower-level APIs.

## The Bottom Line

No matter the type of app you are creating, SDKs and APIs provide vital services for ease in development and time to market. They provide functionality that improves user experience and common packages to address typical use cases.

Skyhook provides solutions that will accurately locate connected devices via the [Precision Location SDK](#) and API, and geofencing solutions via the [Context SDK](#), all in a battery-efficient manner. Check out [all of our available integration options here](#) or learn more on the [Skyhook website](#).



### COMMENTS

First Name\*

Last Name

Email\*

Comment



protected by reCAPTCHA  
[Privacy](#) - [Terms](#)

**SUBMIT COMMENT**

- [→ Help Center](#)
- [→ Submit an Access Point](#)
- [→ Schedule a Demo](#)

---

Products

- [Precision Location](#)
- [Geospatial Insights](#)

---

Industries

- [Chips, Modules, Operating Systems](#)
- [Devices & Apps](#)
- [IoT](#)
- [Network Operators](#)
- [Hospitality](#)
- [Airlines](#)
- [Sports & Events](#)
- [Auto](#)
- [QSR](#)
- [Retail](#)
- [Financial Services](#)
- [Real Estate](#)

---

Company

- [About](#)
- [Careers](#)
- [Products](#)
- [Patents](#)
- [Press](#)

---

Resources

- [Documentation](#)
- [Coverage Map](#)

[Data Privacy](#)

[Privacy Shield Notice](#)

---

Sign Up For Updates:

Email address



---

FOLLOW SKYHOOK



---

© 2020 Skyhook | [Privacy Policy](#) | [Do Not Sell My Personal Information](#) | [Opt Out](#)