

# Wisenet SSM Client SDK v2.10.7

## Programmer's Guide for C# programming

WISENET\_SSM\_CLIENT\_SDK\_PG\_KR

2.10.7

2020-05-15

### Copyright

©2010-2020 Hanwha Techwin Co., Ltd. All rights reserved.

### Trademark

**WISENET** 은 한화테크윈의 로고입니다. SSM Client SDK는 한화테크윈의 등록상표입니다. 이밖에 이 문서에 언급된 상표는 해당 회사 고유의 등록상표입니다.

### Restriction

이 문서의 저작권은 한화테크윈에게 있습니다. 한화테크윈의 공식 승인없이 이 문서의 내용을 복제, 배포 및 재생산 할 수 없고, 임의로 변경할 수 없습니다.

### Disclaimer

한화테크윈은 이 문서에 수록된 정보의 완결성과 정확성을 검증하기 위해 최대한 노력하였으나 이에 대해 보증하지 않습니다. 문서의 사용 결과에 따른 책임은 전적으로 사용자에게 있습니다. 한화테크윈은 사전 예고 없이 이 문서의 내용을 변경할 수 있습니다.

### Contact Information

HANWHA TECHWIN Co., LTD.

Hanwha techwin R&D Center, 6, Pangyo-ro 319beon-gil,  
Bundang-gu, Seongnam-si, Gyeonggi-do, Korea, 463-400

TEL: +82-70-7147-8740~60 FAX: +82-31-8018-3745

<https://step.hanwha-security.com>

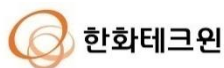
### Hanwha Techwin America Inc.

100 Challenger Road Ridgefield Park, New Jersey, 07660  
U.S.A.

### Hanwha Techwin Europe Ltd.

2nd Floor, No. 5 The Heights, Brooklands, Weybridge,  
Surrey, KT13 0NY, U.K.

Park Chertsey, Surrey, UNITED KINGDOM KT16 OPS



# 들어가며

## 목적

이 문서는 한화테크윈 (주)의 Wisenet SSM Client SDK를 사용하여 애플리케이션을 작성하는 방법을 설명합니다.

## 독자

이 문서는 Wisenet SSM Client SDK를 이용하여 애플리케이션(CMS, 뷰어(viewer) 애플리케이션 등)을 구현하는 개발자를 독자로 합니다.

## 범위

이 문서는 Wisenet SSM Client SDK를 사용하여 애플리케이션을 개발하는 방법, 샘플 프로그램 구현 방법 등에 대해 설명합니다.

## 문서 구성

이 문서는 다음과 같이 구성되어 있습니다.

- CHAPTER 1. Wisenet SSM Client SDK 소개
- CHAPTER 2. 개발 환경 설정
- CHAPTER 3. 샘플 프로그램 소개
- CHAPTER 4. Login Sample
- CHAPTER 5. PasswordChange Sample
- CHAPTER 6. Search Sample
- CHAPTER 7. Backup Sample
- CHAPTER 8. SingleLive Sample
- CHAPTER 9. PTZ Sample
- CHAPTER 10. MultiMonitoring Sample
- CHAPTER 11. Playback Sample
- CHAPTER 12. InsertLog Sample
- CHAPTER 13. LiveSnapshot Sample
- CHAPTER 14. LiveLocalRecord Sample
- CHAPTER 15. UserManagement Sample
- CHAPTER 16. Ntp Sample
- CHAPTER 17. DeviceRecord Sample
- CHAPTER 18. Wisenet DDNS Login Sample
- CHAPTER 19. URL Login Sample

## Convention

버튼 또는 메뉴 표기: 버튼명과 메뉴명은 대괄호([ ])를 사용해서 표기하였습니다.

메뉴 선택 경로 표기: 메뉴 선택 경로는 산괄호(>)를 사용해서 표기하였습니다.

# 문서 이력

문서의 버전과 개정 이력은 다음과 같습니다.

버전	개정일자	개정내역
2.0	2017. 08. 31	최초 작성
2.0	2017. 10. 31	SDK 이름 변경
2.10.1	2018. 10. 30	버전 변경
2.10.1	2018. 11. 16	기능 추가
2.10.3	2019. 04. 15	DeviceRecord 샘플 추가
2.10.4	2019. 05. 15	Active X 제거, C++/CLI wrapper example 버전
2.10.5	2019. 07. 10	Wisenet DDNS Login, URL Login 샘플 추가
2.10.5	2019. 07. 25	Event Key 를 사용한 이벤트 전송 추가
2.10.6	2020. 03. 24	Backup Sample – Track ID 설명 추가

# 목차

들어가며 .....	2
문서 이력 .....	3
목차 .....	4
그림 목차 .....	12
표 목차 .....	13
<b>CHAPTER 1 Wisenet SSM Client SDK .....</b>	<b>14</b>
개요 .....	15
SDK Architecture .....	16
SDK 구성 .....	18
시스템 요구 사양 .....	19
지원 서버 .....	20
<b>CHAPTER 2 개발 환경 설정 .....</b>	<b>21</b>
사전 지식 .....	22
프로젝트 생성 .....	22
프로젝트 설정 .....	24
DEP 문제 해결 방법 .....	25
<b>CHAPTER 3 샘플 프로그램 소개 .....</b>	<b>27</b>
위치 .....	28
샘플 프로그램 목록 .....	28
<b>CHAPTER 4 Login 샘플 프로그램 .....</b>	<b>29</b>
샘플 프로그램 소개 .....	30
API 호출 순서 .....	31

구현 방법.....	32
SsmSdkWrapper 참조 추가.....	32
SsmSdkWrapper Callback 함수 추가 .....	33
SsmSdkWrapper 초기화.....	34
로그인 .....	34
로그아웃 .....	35
SsmSdkWrapper 해제 .....	35
See also.....	35
FAQ.....	36
<b>CHAPTER 5 PasswordChange 샘플 프로그램.....</b>	<b>37</b>
샘플 프로그램 소개.....	38
API 호출 순서.....	39
구현 방법.....	40
SsmSdkWrapper 참조 추가.....	40
SsmSdkWrapper Callback 함수 추가 .....	40
SsmSdkWrapper 초기화.....	40
로그인 .....	40
비밀번호 변경 .....	41
로그아웃 .....	41
SsmSdkWrapper 해제 .....	41
See also.....	41
FAQ.....	42
<b>CHAPTER 6 Search 샘플 프로그램.....</b>	<b>43</b>
샘플 프로그램 소개.....	44
API 호출 순서.....	45
구현 방법.....	46
검색 권한 획득 .....	46
달력 검색.....	46
녹화 트랙 검색 .....	47

녹화 구간 정보 검색 .....	48
검색 권한 반환 .....	49
See also .....	49
FAQ .....	49
<b>CHAPTER 7 Backup 샘플 프로그램 .....</b>	<b>51</b>
샘플 프로그램 소개 .....	52
API 호출 순서 .....	53
구현 방법 .....	55
SsmSdkWrapper 참조 추가 .....	55
SsmSdkWrapper Callback 함수 추가 .....	55
SsmSdkWrapper 초기화 .....	56
로그인 .....	56
검색 권한 획득 .....	56
백업 시작 .....	57
백업 종료 .....	58
검색 권한 반환 .....	58
로그아웃 .....	59
SsmSdkWrapper 해제 .....	59
See also .....	59
FAQ .....	59
<b>CHAPTER 8 SingleLive 샘플 프로그램 .....</b>	<b>60</b>
샘플 프로그램 소개 .....	61
API 호출 순서 .....	62
구현 방법 .....	63
SsmSdkWrapper 초기화 .....	63
RTSP URL 얻어오기 .....	64
미디어 오픈 .....	64
미디어 클로즈 .....	65
SsmSdkWrapper 해제 .....	65

See also.....	65
FAQ.....	66
<b>CHAPTER 9 PTZ 샘플 프로그램.....</b>	<b>67</b>
샘플 프로그램 소개.....	68
API 호출 순서.....	69
구현 방법.....	71
SsmSdkWrapper 초기화.....	71
RTSP URL 얻어오기 .....	71
미디어 오픈 .....	71
PTZ 컨트롤 .....	71
프리셋 추가 .....	72
프리셋 목록 가져오기 .....	72
프리셋 실행 .....	73
미디어 클로즈 .....	73
SsmSdkWrapper 해제 .....	74
See also.....	74
FAQ.....	74
<b>CHAPTER 10 MultiMonitoring 샘플 프로그램.....</b>	<b>75</b>
샘플 프로그램 소개.....	76
API 호출 순서.....	77
구현 방법.....	78
SsmSdkWrapper 초기화.....	78
RTSP URL 얻어오기 .....	78
미디어 오픈 .....	78
미디어 클로즈 .....	79
SsmSdkWrapper 해제 .....	79
See also.....	79
FAQ.....	79
<b>CHAPTER 11 Playback 샘플 프로그램.....</b>	<b>80</b>

샘플 프로그램 소개.....	81
API 호출 순서.....	82
구현 방법.....	83
SsmSdkWrapper 초기화.....	83
RTSP URL 얻어오기 .....	83
TimeZoneInfo 얻어오기 .....	83
미디어 오픈 .....	84
미디어 클로즈 .....	84
SsmSdkWrapper 해제 .....	84
See also.....	85
FAQ.....	85
<b>CHAPTER 12 InsertLog 샘플 프로그램 .....</b>	<b>86</b>
샘플 프로그램 소개.....	87
API 호출 순서.....	88
구현 방법.....	89
이벤트 로그 전송 .....	89
Event Key를 사용하여 이벤트 로그 전송 .....	89
See also.....	90
FAQ.....	91
<b>CHAPTER 13 LiveSnapshot 샘플 프로그램 .....</b>	<b>92</b>
샘플 프로그램 소개.....	93
API 호출 순서.....	94
구현 방법.....	94
Live snapshot 요청 .....	95
See also.....	96
FAQ.....	96
<b>CHAPTER 14 LiveLocalRecord 샘플 프로그램 .....</b>	<b>97</b>
샘플 프로그램 소개.....	98



API 호출 순서.....	99
구현 방법.....	100
Local Record 시작 .....	100
Local Record 중지 .....	101
See also.....	102
FAQ.....	102
<b>CHAPTER 15 UserManagement 샘플 프로그램.....</b>	<b>103</b>
샘플 프로그램 소개.....	104
API 호출 순서.....	104
구현 방법.....	105
User Group 정보 얻기 .....	106
User Group 추가.....	106
User Group 수정 .....	106
User Group 삭제.....	107
User 정보 얻기 .....	108
User 추가.....	108
User 수정.....	109
User 삭제.....	110
See also.....	111
FAQ.....	111
<b>CHAPTER 16 NTP 샘플 프로그램.....</b>	<b>112</b>
샘플 프로그램 소개.....	113
API 호출 순서.....	114
구현 방법.....	114
NTP 설정 불러오기 .....	115
NTP 설정 변경하기 .....	116
See also.....	117
FAQ.....	117

## **CHAPTER 17 Device Record 샘플 프로그램.....119**

샘플 프로그램 소개.....	120
API 호출 순서.....	121
구현 방법.....	121
장치 수동 녹화 시작하기.....	122
장치 수동 녹화 중지하기.....	122
See also.....	123
FAQ.....	123

## **CHAPTER 18 Wisenet DDNS Login 샘플 프로그램.....124**

샘플 프로그램 소개.....	125
API 호출 순서.....	126
구현 방법.....	127
SsmSdkWrapper 참조 추가.....	127
SsmSdkWrapper Callback 함수 추가 .....	128
SsmSdkWrapper 초기화.....	129
로그인.....	129
로그아웃 .....	130
SsmSdkWrapper 해제 .....	130
See also.....	130
FAQ.....	131

## **CHAPTER 19 URL Login 샘플 프로그램.....132**

샘플 프로그램 소개.....	133
API 호출 순서.....	134
구현 방법.....	135
SsmSdkWrapper 참조 추가.....	135
SsmSdkWrapper Callback 함수 추가 .....	136
SsmSdkWrapper 초기화.....	137
로그인.....	137
로그아웃 .....	138

SsmSdkWrapper 해제 .....	138
See also .....	138
FAQ .....	139
<b>약어 .....</b>	<b>140</b>

# 그림 목차

Figure 1 Wisenet SSM Client SDK Architecture .....	16
Figure 2 LogIn 샘플 프로그램 실행 화면 .....	30
Figure 3 Password change 샘플 프로그램 실행 화면 .....	38
Figure 5 Search 샘플 프로그램 실행 화면 .....	44
Figure 4 Backup 샘플 프로그램 실행 화면 .....	52
Figure 6 SingleLive 샘플 프로그램 실행 화면 .....	61
Figure 7 PTZ 샘플 프로그램 실행 화면 .....	68
Figure 8 MultiMonitoring 샘플 프로그램 실행 화면 .....	76
Figure 9 Playback 샘플 프로그램 실행 화면 .....	81
Figure 10 Insert log 샘플 프로그램 실행 화면 .....	87
Figure 11 LiveSnapshot 샘플 프로그램 실행 화면 .....	93
Figure 12 LiveLocalRecord 샘플 프로그램 실행 화면 .....	98
Figure 13 UserManagement 샘플 프로그램 실행 화면 .....	104
Figure 14 NTP 샘플 프로그램 실행 화면 .....	113
Figure 15 DeviceRecord 샘플 프로그램 실행 화면 .....	120
Figure 16 Wisenet DDNS LogIn 샘플 프로그램 실행 화면 .....	125
Figure 17 URL LogIn 샘플 프로그램 실행 화면 .....	133

# 표 목차

Table 1 샘플 프로그램 목록 .....	28
--------------------------	----

## CHAPTER 1

# Wisenet SSM Client SDK

---

이 장은 Wisenet SSM Client SDK의 개요, 구성 및 지원 서버 등에 대해 설명합니다.

## Contents

개요

SDK Architecture

SDK 구성

시스템 요구 사양

지원 서버

# 개요

Wisenet SSM(Wisenet Smart Security Manager) Client SDK은 Microsoft의 .NET Framework 4.5를 채택하여 만든 SDK입니다.

Wisenet SSM Client SDK는 Wisenet SSM Core Server에 등록된 네트워크 장비의 영상/음성 스트리밍 및 이벤트 수신을 지원합니다.

Wisenet SSM Client SDK를 이용하여 어플리케이션을 개발하기 위해서는 SDK에서 제공하는 C DLL이 필요합니다. Wisenet SSM Client SDK 의 설치 파일을 이용하면 DLL이 지정된 위치에 저장됩니다. 설치 방법은 'SSM 매뉴얼'을 참고하시기 바랍니다.

Wisenet SSM Client SDK는 Wisenet SSM Core Server와 네트워크 연결을 통해서 기능을 제공합니다. Wisenet SSM Core Server 프로그램은 Wisenet SSM Core Server 설치 파일을 통해 설치해야 합니다. 설치 방법은 'SSM 매뉴얼'을 참고하시기 바랍니다.

Wisenet SSM Client SDK에서 제공하는 주요 기능은 다음과 같습니다.

- Core Server 로그인/로그아웃, 비밀번호 변경
- 네트워크 장비에서 발생한 이벤트 실시간 수신
- 실시간 영상 감시 및 녹화 영상 재생
- PTZ 제어
- 녹화 검색
- 녹화 영상 백업
- 사용자 관리
- 장치 녹화 제어

# SDK Architecture

Wisenet SSM Client SDK의 구조는 다음과 같습니다.

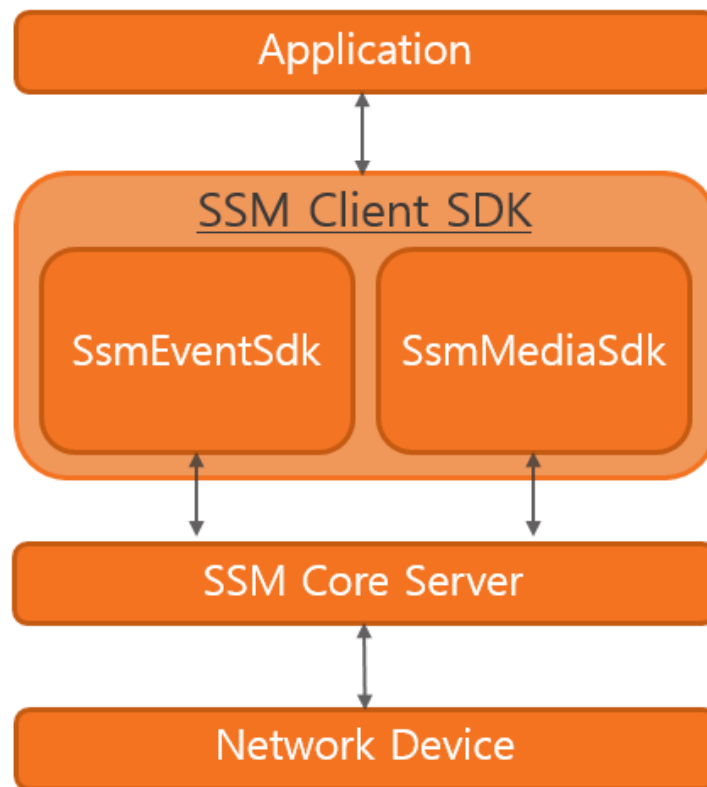


Figure 1 Wisenet SSM Client SDK Architecture

- Application
  - Wisenet SSM Client SDK을 사용하여 개발한 프로그램
- Wisenet SSM Client SDK
  - SsmEventSdk: SSM Console의 주요 기능을 포함하고 있는 라이브러리
  - SsmMediaSdk: 영상/음성/메타 데이터 수신, 디코딩, 렌더링 등 미디어 컨트롤과 관련된 기능을 수행하는 라이브러리
- Wisenet SSM Core Server
  - SSM의 메인 서버.
  - 사용자, 디바이스, 메시징 등을 관리한다.
  - SSM의 미디어 중계 서버.



- 사용자 프로그램에 미디어 데이터를 전송하고, 디바이스를 관리
- 영상, 음성 데이터 녹화
- Network Device
  - NVR, DVR 등
  - 아날로그/네트워크 카메라
  - 인코더(Encoder)

# SDK 구성

Wisenet SSM Client SDK는 다음과 같이 구성되어 있습니다.

- 라이브러리
  - 애플리케이션 개발 시 사용되는 DLL 파일.
  - Bin, Bin\_x64 폴더에 설치됨.
- 문서
  - API 가이드, 프로그래밍 가이드
  - {\$SDK path}\WDoc 폴더에 저장됨.
- 샘플 실행 파일
  - SDK를 사용하여 개발한 프로그램
  - {\$SDK path}\WBin, {\$SDK path}\WBin\_x64 폴더에 설치됨.
- 샘플 소스 코드
  - 샘플 프로그램의 소스
  - {\$SDK path}\WSample 폴더에 저장됨..

# 시스템 요구 사양

Wisenet SSM Client SDK을 사용하기 위한 시스템 사양은 다음과 같습니다.

- 운영체제(OS)
  - Microsoft Windows 7 또는 그 상위 버전
- .NET 프레임워크(.NET Framework)
  - Microsoft .NET Framework 4.5
- Wisenet SSM Core Server
  - v2.10.6
- CPU
  - 인텔 i5 이상
- 개발 환경
  - Microsoft Visual Studio 2013 이상

# 지원 서버

Wisenet SSM Client SDK와 Wisenet SSM Core Server는 동일한 버전을 사용하여 정상적인 동작이 가능합니다.

예를 들어 Wisenet SSM Core Server v2.10.6를 사용하시는 경우, Wisenet SSM Client SDK v2.10.6를 사용하셔야 합니다.

## CHAPTER 2

# 개발 환경 설정

---

이 장에서는 개발 환경을 설정하는 방법에 대해 설명합니다.

## Contents

사전 지식

프로젝트 생성

프로젝트 설정

DEP 문제 해결 방법

# 사전 지식

Wisenet SSM Client SDK의 샘플코드는 C#, C++/CLI와 .NET Framework 4.5를 기반으로 하고 있습니다. 이에 대한 관련 정보는 MSDN을 참조하시기 바랍니다.

C#

[https://msdn.microsoft.com/en-us/library/kx37x362\(v=vs.120\).aspx](https://msdn.microsoft.com/en-us/library/kx37x362(v=vs.120).aspx)

.NET Framework 4.5

[https://msdn.microsoft.com/en-US/library/k1s94fta\(v=vs.100\).aspx](https://msdn.microsoft.com/en-US/library/k1s94fta(v=vs.100).aspx)

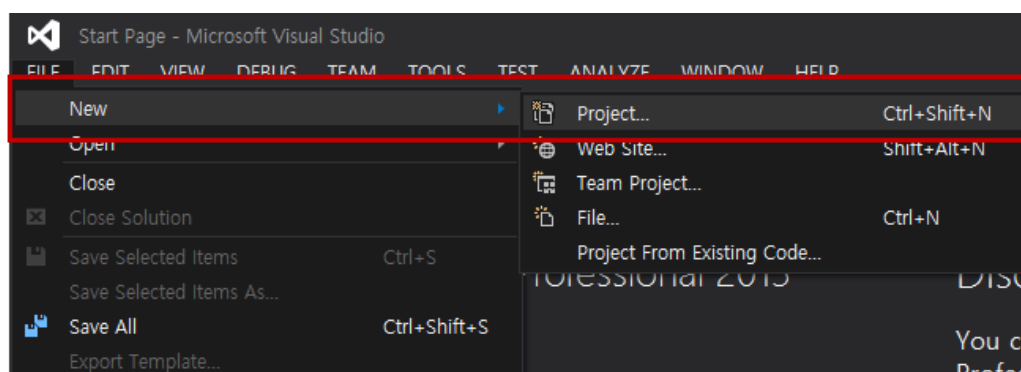
# 프로젝트 생성

애플리케이션을 개발하는데 필요한 프로젝트는 다음과 같은 방법으로 생성합니다. (본 문서는 Visual Studio 2013 버전을 기준으로 설명합니다.)

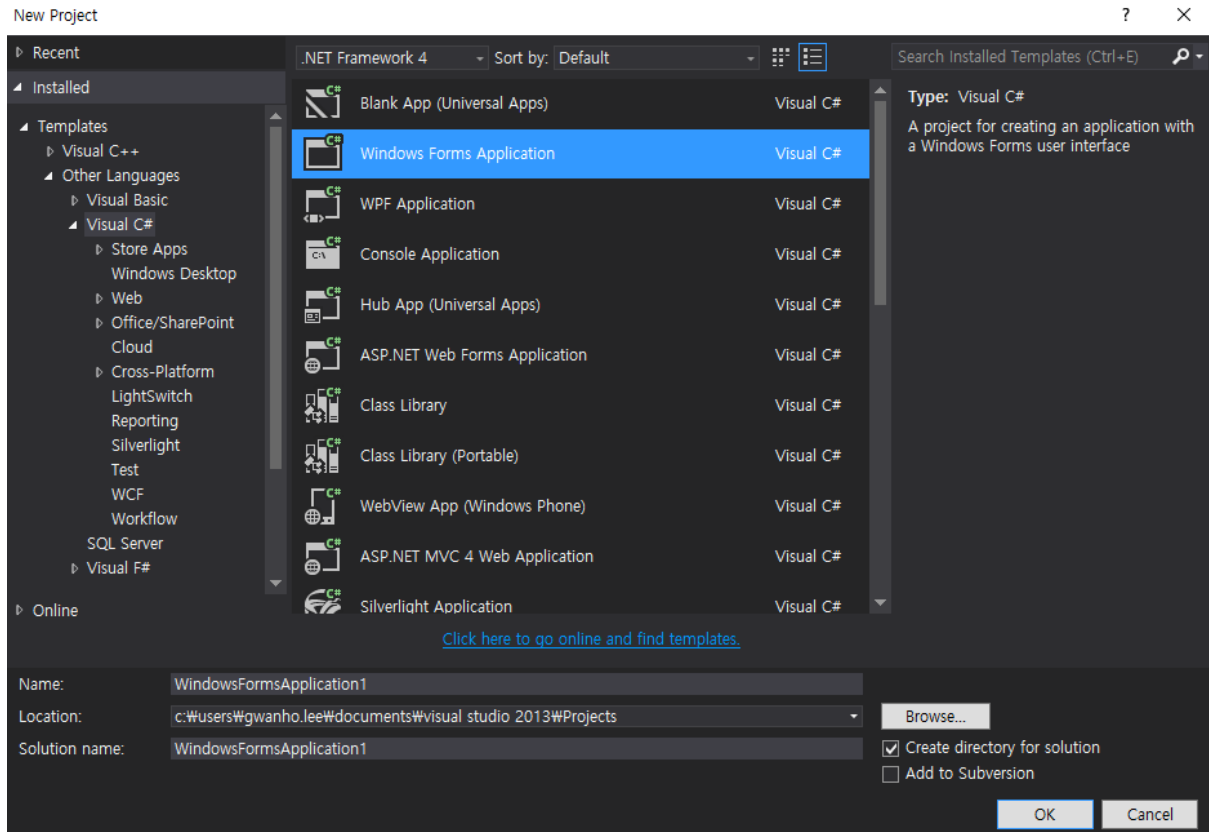
## Visual Studio 2013

### Procedures

- Step 1.** Visual Studio 2013을 실행합니다.
- Step 2.** [File] > [New] > [Project]를 선택합니다.



- Step 3.** [New Project] 다이얼로그에서,
- A. “Visual C#”, “Windows Forms Application” 템플릿을 선택합니다.
  - B. 프로젝트의 이름과 경로를 지정합니다.
  - C. [OK]를 클릭합니다.



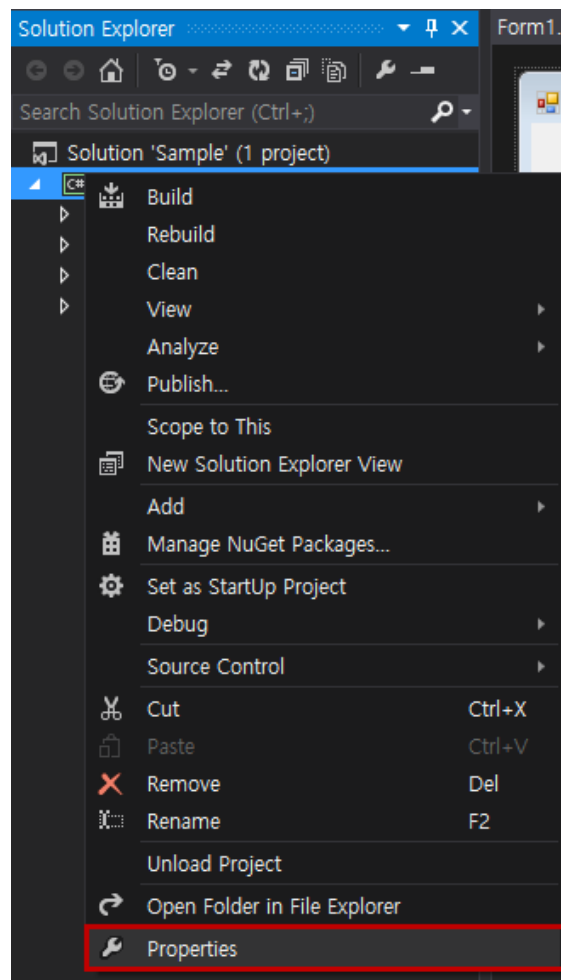
# 프로젝트 설정

애플리케이션을 작성한 후, 빌드 하기 위해서는 프로젝트의 프로퍼티(property)를 설정해야 합니다. Visual Studio 2013에서 프로젝트 프로퍼티를 설정하는 방법은 아래와 같습니다.

## Visual Studio 2013

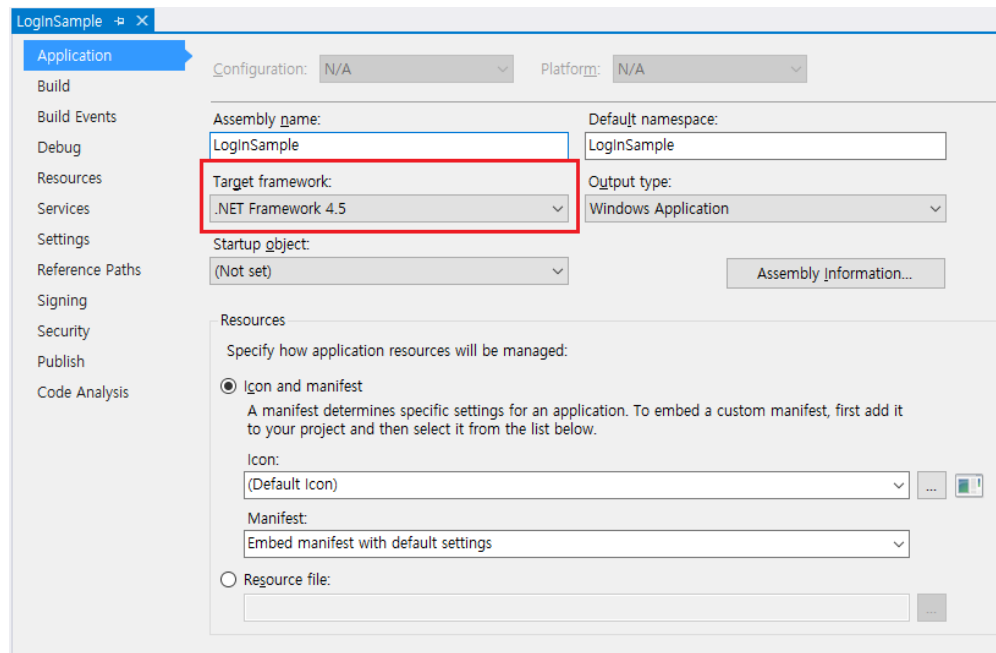
### 절차

- Step 1.** [Solution Explorer]에서 프로젝트를 선택한 후, 마우스 오른쪽 버튼을 클릭하여 [Properties] 메뉴를 선택합니다.



- Step 2.** 왼쪽 트리 메뉴에서 [Application]을 선택한 후, [대상 프레임워크] 필드에 [.NET Framework 4.5]를 선택합니다. [확인]을 클릭합니다.

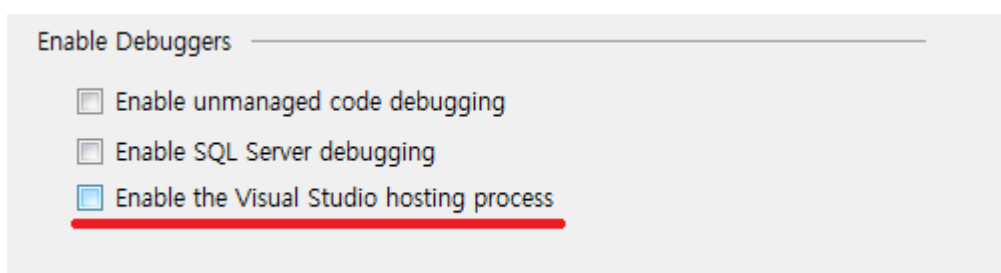




## XP 이상 OS 에서 C# Debug 문제 해결 방법

### 절차

- Step 1.** [properties] – [Debug] 탭에서, 만약 하단의 그림과 같이 Enable the Visual Studio hosting process 부분이 체크가 되어있다면 해제합니다.



## DEP 문제 해결 방법

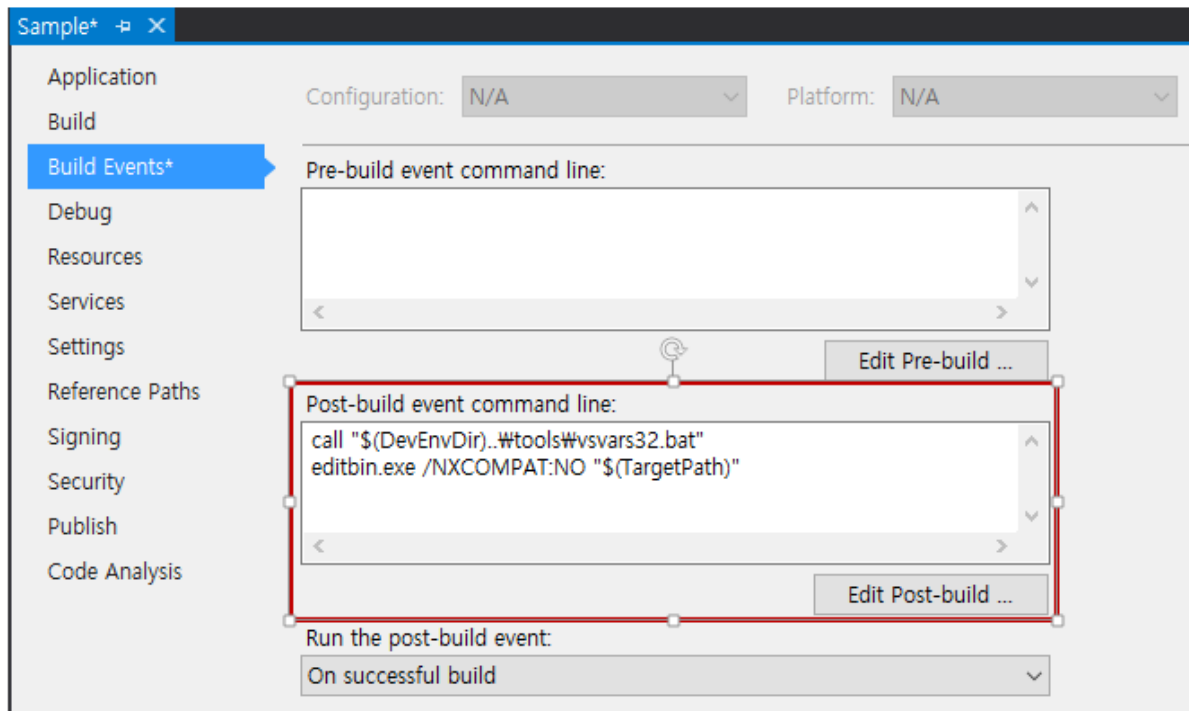
DEP(데이터 실행 방지)는 프로그램을 모니터링하여 프로그램이 시스템 메모리를 안전하게 사용하게 함으로써 바이러스 및 다른 보안 위험으로부터 컴퓨터가 손상되는 것을 방지해 줄 수 있는 Windows의 보안 기능입니다. DEP 기능 설정 때문에 SDK 프로그램이 올바르게 실행되지 않을 수 있습니다. 이 경우에는 다음과 같이 개발환경에서 직접 DEP를 해제할 수 있습니다.

## C# 프로젝트 설정

### 절차

- Step 1.** [Solution Explorer]에서 프로젝트를 선택한 후, 마우스 오른쪽 버튼을 클릭하여 [Properties] 메뉴를 선택합니다.
- Step 2.** [Build Events] 탭을 선택한 후, 아래와 같이 명령어를 입력합니다.

```
call "$(DevEnvDir)..\tools\vsvars32.bat"
editbin.exe /NXCOMPAT:NO "$(TargetPath)"
```



## CHAPTER 3

# 샘플 프로그램 소개

---

이 장에서는 Wisenet SSM Client SDK가 제공하는 샘플 프로그램을 소개합니다. 샘플 프로그램은 SsmEventSdk 및 SsmMediaSdk 라이브러리를 사용하여 사용자가 구현할 수 있는 프로그램의 예시를 보여줍니다.

## Contents

위치

샘플 프로그램 목록

# 위치

샘플 프로그램은 {\$SDK path}\samples 폴더에서 확인하실 수 있습니다.

## 샘플 프로그램 목록

Table 1 샘플 프로그램 목록

Sample Program	Description
Login	SSM Core Server에 로그인하는 방법을 설명합니다.
PasswordChange	사용자 비밀번호 변경 방법을 설명합니다.
Backup	녹화 영상을 백업하는 방법을 설명합니다.
Search	녹화된 영상과 관련된 정보를 검색하는 방법을 설명합니다.
SingleLive	1 채널 라이브 영상을 수신하는 방법을 설명합니다.
PTZ	PTZ 카메라의 Pan, Tilt, Zoom, Preset 기능을 사용하는 법을 설명합니다.
MultiMonitoring	다중 라이브 영상을 수신하는 방법을 설명합니다.
Playback	서버로부터 녹화 영상을 수신하는 방법을 설명합니다.
InsertLog	서버로 이벤트를 송신하여 기록하는 방법을 설명합니다.
LiveSnapshot	라이브 영상의 스냅샷을 생성합니다.
LiveLocalRecord	라이브 영상을 로컬 PC 에 저장합니다.
UserManagement	사용자 정보를 관리합니다.
NTP	NTP 설정을 변경합니다.
Device Record	장치의 녹화를 시작/중지 합니다.

## CHAPTER 4

# LogIn 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Core Server에 로그인하는 방법을 설명합니다.

## Contents

샘플 프로그램 소개

API 호출 순서

구현 방법

See Also

FAQ

# 샘플 프로그램 소개

LogIn 샘플 프로그램은 Wisenet SSM Core Server에 로그인하는 방법을 설명하기 위한 예제입니다.

샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 로그인 버튼을 클릭하여 System Manager 서버로 접속 되는지 확인합니다.
- Step 3.** 프로그램을 종료합니다.

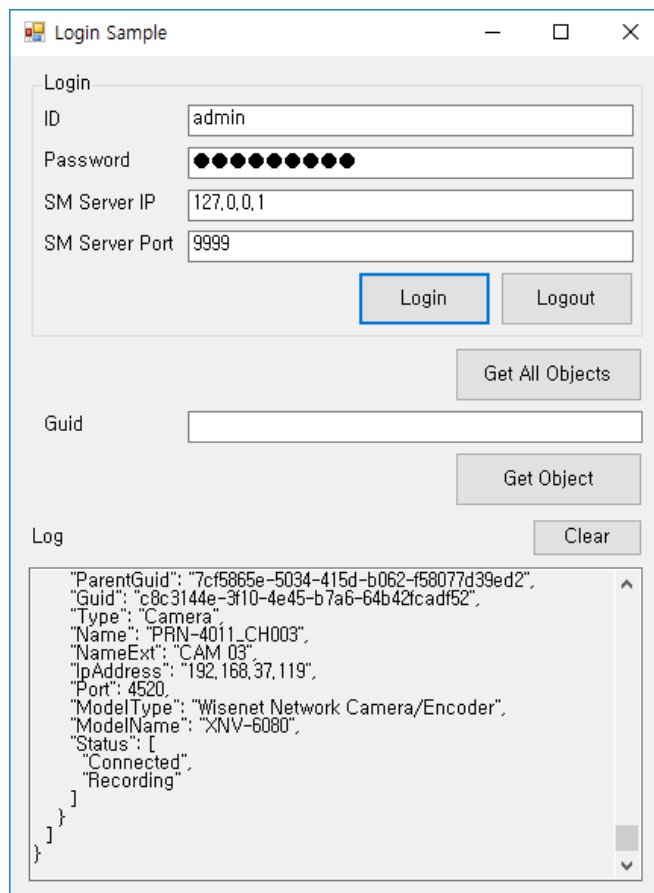


Figure 2 LogIn 샘플 프로그램 실행 화면

# API 호출 순서

API 호출 순서는 다음과 같습니다.

Preprocessing

1. `ssmSdkWrapper.InitializeEventEvent()`

Login

2. `ssmSdkWrapper.Login()`

Response of Login

3. `ssmSdkWrapper.OnResponse`

Receive Event: Login

4. `ssmSdkWrapper.OnEvent`

Logout

5. `ssmSdkWrapper.Logout()`

Release

6. `ssmSdkWrapper.ReleaseEvent()`

# 구현 방법

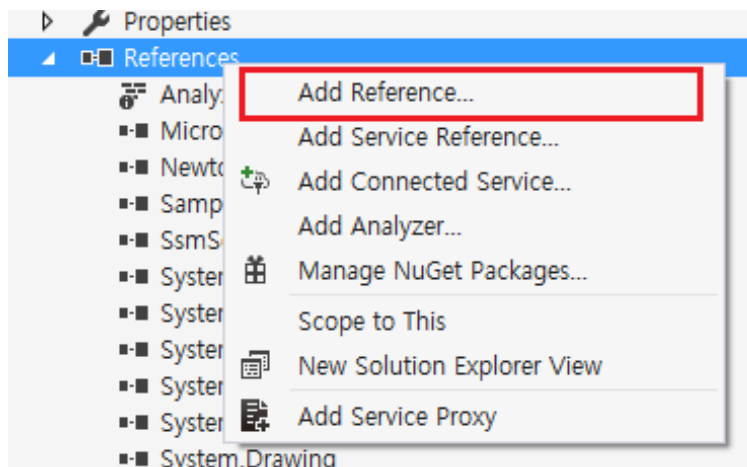
이 절에서는 LogIn 샘플 프로그램 구현 방법을 상세히 설명합니다.

## SsmSdkWrapper 참조 추가

### 절차

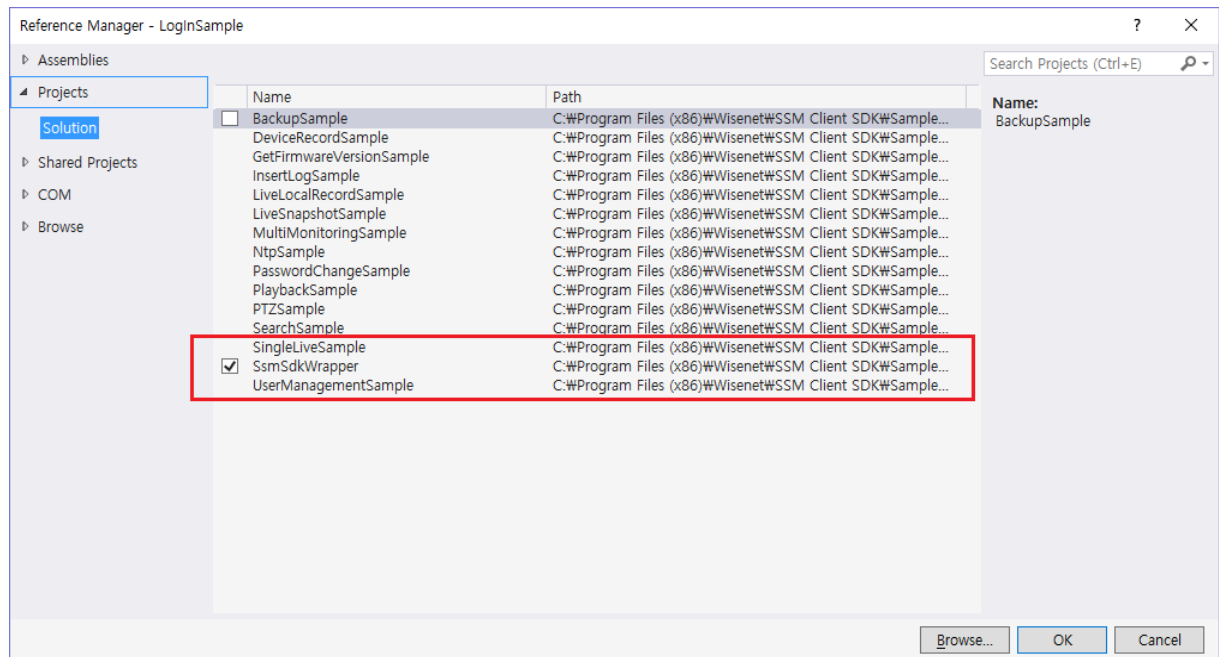
Visual Studio 2013에서 Windows Forms를 생성한 후, SsmSdkWrapper를 참조 추가합니다.

- Step 1.** Windows Forms를 생성합니다.
- Step 2.** Visual Studio 프로젝트 References에서 오른쪽 마우스 버튼을 눌러 Add Reference를 선택합니다.



- Step 3.** Reference Manager에서 SsmSdkWrapper 프로젝트를 추가합니다.





## SsmSdkWrapper Callback 함수 추가

명령에 대한 응답, 이벤트를 받기 위해서는 Callback 함수를 구현해야 합니다.

### Callback 함수 구현: OnResponse

LoginSample 클래스의 OnResponse를 구현합니다.

```
// LoginSample.cs
private void OnResponse(UInt32 commandID, UInt32 errorCode, UInt32 sequenceID, string info)
{
    _logger.WLOGD(
        "OnResponse():" +
        " Command ID=" + commandID +
        " Result=" + errorCode +
        " Sequence ID= " + sequenceID +
        " Info=" + info);
}
```

## Callback 함수 구현: OnEvent

LoginSample 클래스의 OnEvent를 구현합니다.

```
// LoginSample.cs
private void OnEvent(UInt32 eventID, String info)
{
    _logger.WLOGD(
        "OnEvent():" +
        " Event ID=" + eventID +
        " Info=" + info);
}
```

---

## SsmSdkWrapper 초기화

SsmSdkWrapper를 생성하고 ssmSdkWrapper.InitializeEvent()를 호출하여 초기화합니다.  
LoginSample 클래스의 멤버 함수 LoginSample\_Load() 안에서 InitializeEvent() 메서드를 호출합니다. InitializeEvent()를 통해 SsmSdkWrapper의 서비스 사용을 시작합니다.

```
// LoginSample.cs
private SsmSdkWrapper ssmSdkWrapper = null;

private void LoginSample_Load(object sender, EventArgs e)
{
    ssmSdkWrapper = new SsmSdkWrapper(this.OnResponse, this.OnEvent);
    ssmSdkWrapper.InitializeEvent();
}
```

---

## 로그인

로그인 버튼 Event Handler를 생성하여 로그인을 수행합니다. 로그인을 위해서는 Wisenet SSM Core Server 서버의 IP주소와 Port번호, ID와 Password가 필요합니다.

```
// LoginSample.cs
private void btnLogin_Click(object sender, EventArgs e)
{
    uint resCode = ssmSdkWrapper.Login(
        txtID.Text,
```

```

        txtPassword.Text,
        txtIPAddress.Text,
        Convert.ToUInt32(txtPort.Text));
    _logger.WLOGD("Login()::Result=" + resCode);
}

```

## 로그아웃

로그아웃 버튼 Event Handler를 생성하여 로그아웃을 수행합니다.

```

// LoginSample.cs
private void btnLogout_Click(object sender, EventArgs e)
{
    uint resCode = ssmSdkWrapper.Logout();
    _logger.WLOGD("Logout()::Result=" + resCode);
}

```

## SsmSdkWrapper 해제

SsmSdkWrapper의 사용이 끝난 후에는 서비스를 중단하고 자원을 해제해야 합니다.

LoginSample 클래스의 멤버 함수 LoginSample\_Closed() 안에서 ReleaseEvent() 메서드를 호출합니다.

```

// LoginSample.cs
private void LoginSample_Closed(object sender, EventArgs e)
{
    ssmSdkWrapper.ReleaseEvent();
}

```

## See also

Wisenet SSM Client SDK API Reference (v2.10.6)

# FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 5

# PasswordChange 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 사용자 계정의 비밀번호를 변경하는 방법을 설명합니다.

## Contents

[샘플 프로그램 소개](#)

[API 호출 순서](#)

[구현 방법](#)

[See Also](#)

[FAQ](#)

# 샘플 프로그램 소개

PasswordChange 샘플 프로그램은 SSM 사용자 계정의 비밀번호를 변경하는 예제입니다.

샘플 프로그램 사용법은 다음과 같습니다.

**Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.

서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

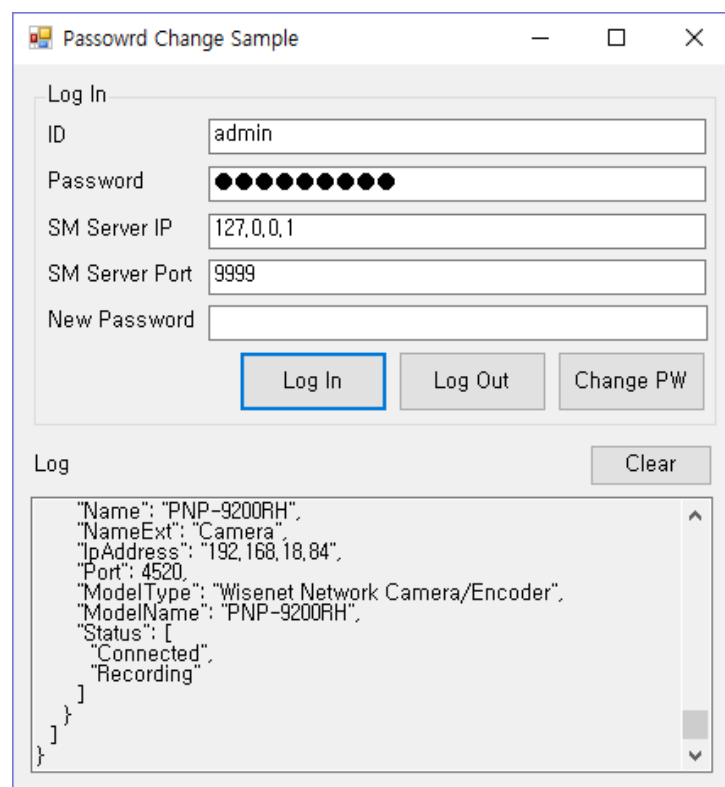


Figure 3 Password change 샘플 프로그램 실행 화면

**Step 2.** OnResponse의 ResultCode를 통해 로그인이 정상적으로 이루어졌는지 확인합니다.

**Step 3.** [New Password] 창에 변경할 신규 비밀번호를 입력합니다.

**Step 4.** [Change PW] 버튼을 클릭하여 비밀번호 변경을 요청합니다.

**Step 5.** [로그 리스트] 창에서 서버로부터 수신한 응답을 확인합니다.

**Step 6.** [Logout] 버튼을 클릭하여 서버와 접속을 해제합니다.

**Step 7.** 프로그램을 종료합니다.

# API 호출 순서

API 호출 순서는 다음과 같습니다.

## Preprocessing

1. `ssmSdkWrapper.InitializeEventEvent()`

## Login

2. `ssmSdkWrapper.Login()`

## Response of Login

3. `ssmSdkWrapper.OnResponse`

## Receive Event: Login

4. `ssmSdkWrapper.OnEvent`

## Password change

5. `ssmSdkWrapper.ChangePassword()`

## Response of Password change

6. `ssmSdkWrapper.OnResponse`

## Logout

7. `ssmSdkWrapper.Logout()`

## Release

8. `ssmSdkWrapper.ReleaseEvent()`

# 구현 방법

이 절에서는 PasswordChange 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다.

---

## SsmSdkWrapper 참조 추가

### 참고

SsmSdkWrapper 참조 추가 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

---

## SsmSdkWrapper Callback 함수 추가

### Callback 함수 구현: OnResponse

Callback 함수 구현: OnResponse 작성 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

### Callback 함수 구현: OnEvent

Callback 함수 구현: OnEvent 작성 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

---

## SsmSdkWrapper 초기화

### 참고

SsmSdkWrapper 초기화 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

---

## 로그인

### 참고



로그인 구현 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을 (32쪽) 참고 바랍니다.

## 비밀번호 변경

비밀번호 변경요청 버튼클릭 Event Handler를 생성하여, 비밀번호 변경 요청 명령 전송을 수행합니다. ID와 현재 비밀번호, 그리고 새로운 비밀번호가 메서드의 argument로 필요합니다.

```
// PasswordChangeSample.cs
private void btnChangePassword_Click(object sender, EventArgs e)
{
    UInt32 sequenceID = 0;
    uint resCode = ssmSdkWrapper.ChangePassword(
        txtID.Text,
        txtPassword.Text,
        txtNewPassword.Text,
        ref sequenceID);

    _logger.WLOGD("ChangePassword()::Result=" + resCode + ", SequenceID=" +
sequenceID);
}
```

## 로그아웃

### 참고

로그아웃 구현 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

## SsmSdkWrapper 해제

### 참고

SsmSdkWrapper 해제 구현 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

See also

Login 샘플 프로그램

Wisenet SSM Client SDK API Reference (v2.10.6)

## FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 6

# Search 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 녹화 영상과 관련된 정보를 얻어오는 방법을 설명합니다.

## Contents

[샘플 프로그램 소개](#)

[API 호출 순서](#)

[구현 방법](#)

[See Also](#)

[FAQ](#)

# 샘플 프로그램 소개

Search 샘플 프로그램은 네트워크 장비로부터 녹화 영상과 관련된 정보를 얻어오는 예제입니다. 샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

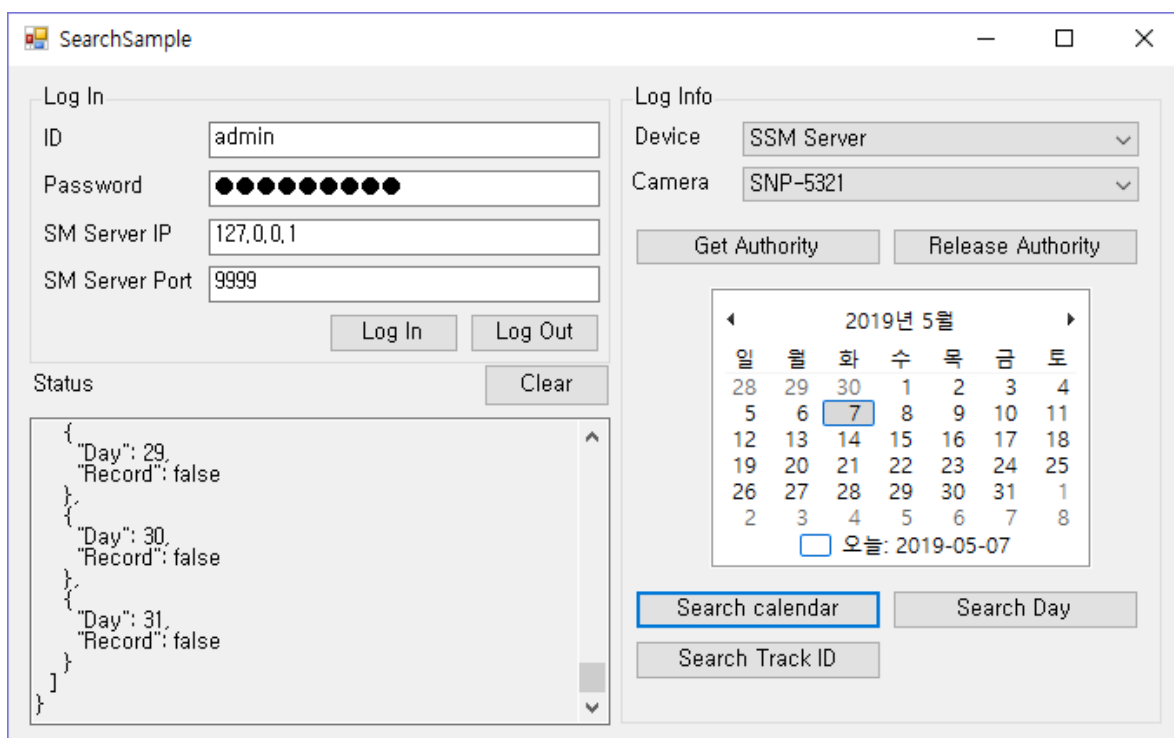


Figure 4 Search 샘플 프로그램 실행 화면

- Step 3.** OnResponse의 ResultCode를 통해 로그인이 정상적으로 이루어졌는지 확인합니다.
- Step 4.** 로그인이 정상적으로 이루어지면 [Device]와 [Camera] ComboBox에 현재 연결 된 장비와 카메라 이름이 나타납니다.
- Step 5.** Device와 Camera를 각각 ComboBox에서 선택합니다.
- Step 6.** [Get Authority] 버튼을 클릭하여 선택한 Device의 검색 권한을 획득합니다.
- Step 7.** [Search Calendar] 버튼을 클릭하여, 선택한 달에서 영상이 녹화된 날짜들을 획득합니다.
- Step 8.** [Search Day] 버튼을 클릭하여, 선택한 날짜에서 영상이 녹화된 시간 구간과 레코드 타입 등을 획득합니다.
- Step 9.** [Search TrackID] 버튼을 클릭하여, 선택한 녹화 구간의 트랙 ID 정보를 얻어옵니다.
- Step 10.** [Release Authority] 버튼을 클릭하여 검색 권한을 반환합니다.

- Step 11.** [Logout] 버튼을 클릭하여 서버와 접속을 해제합니다.
- Step 12.** 프로그램을 종료합니다.

## API 호출 순서

이 샘플 프로그램은 Wisenet SSM Client SDK가 제공하는 라이브러리를 사용하여 녹화 영상 검색 기능을 구현하였습니다. 샘플 프로그램의 API 호출 순서는 다음과 같습니다.

Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`

Login

2. `ssmSdkWrapper.Login()`

Response of Login

3. `ssmSdkWrapper.OnResponse`

Receive Event: Login

4. `ssmSdkWrapper.OnEvent`

Receive Event: ObjectConnected

5. `ssmSdkWrapper.OnEvent`

GetSearchAuthority

6. `ssmSdkWrapper.GetSearchAuthority`

Response of GetSearchAuthority

7. `ssmSdkWrapper.OnResponse`

Search Calendar

8. `ssmSdkWrapper.SearchCalendar()`

Response of SearchCalendar

9. `ssmSdkWrapper.OnResponse`

Search TrackId

10. `ssmSdkWrapper.SearchTrackID()`

Response of SearchDay

11. `ssmSdkWrapper.OnResponse`

Search Day

12. `ssmSdkWrapper.SearchDay()`

Release Authority

13. `ssmSdkWrapper.ReleaseAuthority()`

Log out

14. `ssmSdkWrapper.Logout()`

Release

15. `ssmSdkWrapper.ReleaseEvent()`

## 구현 방법

이 절에서는 Backup 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다

---

## 검색 권한 획득

### 참고

검색 권한 획득 방법은 본 문서의 CHAPTER 6에 기술된 Backup 샘플 프로그램의 '구현 방법' 절을(55쪽) 참고 바랍니다.

---

## 달력 검색

달력 검색 버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 `SearchCalendar()` 메서드를 통해 달력 검색 요청을 수행합니다. 선택한 달에서의 날짜별 녹화 여부는 `OnResponse`를 통해 받을 수 있습니다.

```
// SearchSample.cs
```

```
private void btnSearchCalendar_Click(object sender, EventArgs e)
{
    int year = monthCalendar.SelectionStart.Year;
    int month = monthCalendar.SelectionStart.Month;
    int trackID = -1;
    UInt32 sequenceID = 0;

    try
    {
        uint result = ssmSdkWrapper.SearchCalendar(
            _deviceUuid,
            _cameraUuid,
            year.ToString(),
            month.ToString(),
            ref sequenceID);
        _logger.WLOGD("SearchCalendar()::Result=" + result + ", SequenceID=" +
sequenceID);
    }
    catch (Exception ex)
    {
        _logger.WLOGD(ex.Message);
    }
}
```

## 녹화 트랙 검색

버튼 클릭 Event Handler를 생성하고, SsmSdkWrapper의 SearchTrackID() 메서드를 통해 지정한 녹화 구간의 트랙 정보를 획득합니다. 트랙 정보는 OnResponse를 통해 받을 수 있습니다.

```
// BackupSample.cs
private void btnSearchTrackID_Click(object sender, EventArgs e)
{
    var startDate = monthCalendar.SelectionStart.Date;
    var endDate = monthCalendar.SelectionEnd.Date.AddHours(23.59);
    string startTime = startDate.ToLocalTime().ToString(UtcFormat);
    string endTime = endDate.ToLocalTime().ToString(UtcFormat);
    UInt32 sequenceID = 0;
```

```
try
{
    uint result = ssmSdkWrapper.SearchTrackID(
        _deviceUuid,
        startTime,
        endTime,
        ref sequenceID);
    _logger.WLOGD("SearchCalendar()::Result=" + result + ", SequenceID=" +
sequenceID);
}
catch (Exception ex)
{
    _logger.WLOGD(ex.Message);
}
}
```

## 녹화 구간 정보 검색

일(day) 검색 버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 SearchDay() 메서드를 통해 선택한 날짜의 녹화 구간 정보를 검색합니다. 녹화 구간 정보는 OnResponse를 통해 받을 수 있습니다.

```
// SearchSample.cs
private void btnSearchDay_Click(object sender, EventArgs e)
{
    var startDate = monthCalendar.SelectionStart.Date;
    var endDate = monthCalendar.SelectionEnd.Date.AddHours(23.59);
    string startTime = startDate.ToLocalTime().ToString(UtcFormat);
    string endTime = endDate.ToLocalTime().ToString(UtcFormat);
    UInt32 sequenceID = 0;

    try
    {
        uint result = ssmSdkWrapper.SearchDay(
            _deviceUuid,
            _cameraUuid,
            trackId,
            startTime,
```



```

false,      // DST
endTime,
false,      // DST
(uint)RecordType.ALL,
(uint)IVEventType.ALL,
ref sequenceID);

_logger.WLOGD("SearchDay()::Result=" + result + ", SequenceID=" +
sequenceID);
}
catch (Exception ex)
{
    _logger.WLOGD(ex.Message);
}
}

```

## 검색 권한 반환

### 참고

SsmSdkWrapper 초기화 방법은 본 문서의 CHAPTER 6에 기술된 Backup 샘플 프로그램의 '구현 방법' 절을(55쪽) 참고 바랍니다.

## See also

Login 샘플 프로그램

Backup 샘플 프로그램

Wisenet SSM Client SDK API Reference (v2.10.6)

## FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 7

# Backup 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 녹화 영상을 백업하는 방법을 설명합니다.  
예제에서는 AVI 포맷을 사용합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ

# 샘플 프로그램 소개

Backup 샘플 프로그램은 네트워크 장비로부터 1채널 녹화 영상을 수신하여 파일로 백업하는 예제입니다. 샘플 프로그램은 AVI 파일을 생성하는 예제만을 포함합니다. 샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

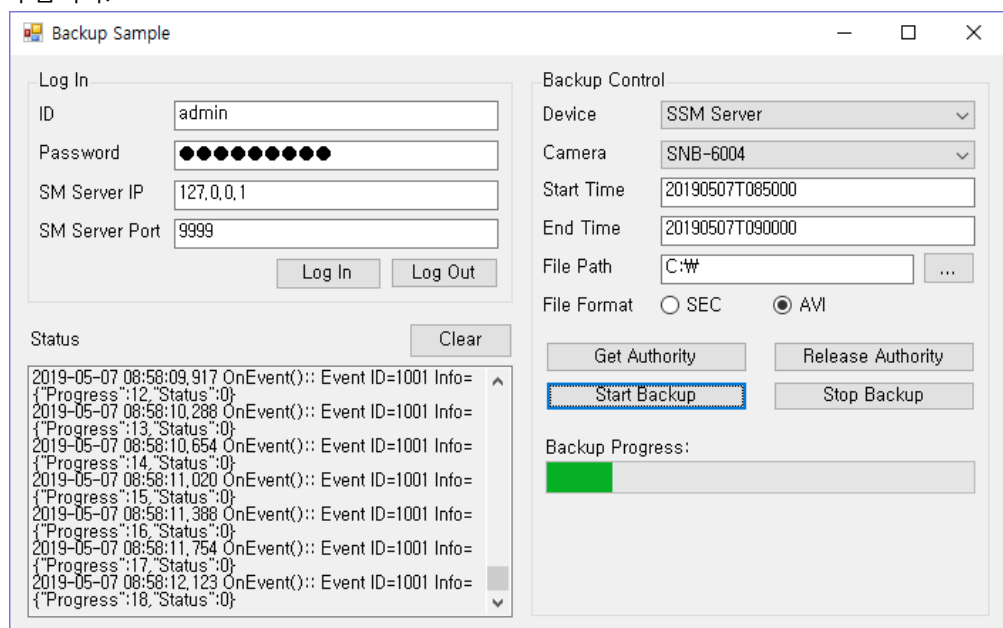


Figure 5 Backup 샘플 프로그램 실행 화면

- Step 3.** OnResponse의 ResultCode를 통해 로그인에 정상적으로 이루어졌는지 확인합니다.
- Step 4.** 로그인이 정상적으로 이루어지면 [Device]와 [Camera] ComboBox에 현재 연결된 장비와 카메라 이름이 나타납니다.
- Step 5.** Device와 Camera를 각각 ComboBox에서 선택합니다.
- Step 6.** [Get Authority] 버튼을 클릭하여 선택한 Device의 검색 권한을 획득합니다.
- Step 7.** [Start Time] 텍스트 박스에 백업 시작 시간을 입력합니다. 포맷은 "yyyyMMddThhmmss"입니다.
- Step 8.** [End Time] 텍스트 박스에 백업 종료 시간을 입력합니다. 포맷은 "yyyyMMddThhmmss"입니다.

- Step 9. [File Path] 컨트롤을 클릭하여 백업파일 저장 경로를 지정합니다.
- Step 10. [Start Backup] 버튼을 클릭하여 백업을 시작합니다.
- Step 11. [Status] 텍스트 박스를 통해, BackupStatusChanged 이벤트로 전달되는 값을 확인합니다.
- Step 12. [Progress Bar]를 통해, 백업 진행률을 확인합니다.
- Step 13. [Status] 텍스트 박스를 통해, 백업 종료 이벤트가 발생하는 것을 확인합니다.
- Step 14. [Release Authority] 버튼을 클릭하여 검색 권한을 반환합니다.
- Step 15. [Logout] 버튼을 클릭하여 서버와 접속을 해제합니다.
- Step 16. 프로그램을 종료합니다.

## API 호출 순서

API 호출 순서는 다음과 같습니다.

Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`

Login

2. `ssmSdkWrapper.Login()`

Response of Login

3. `ssmSdkWrapper.OnResponse`

Receive Event: LogIn

4. `ssmSdkWrapper.OnEvent`

Receive Event: ObjectConnected

5. `ssmSdkWrapper.OnEvent`

GetSearchAuthority

6. `ssmSdkWrapper.GetSearchAuthority`

Response of GetSearchAuthority

7. `ssmSdkWrapper.OnResponse`

SearchTrackID

8. `ssmSdkWrapper.SearchTrackID()`  
SearchDay
9. `ssmSdkWrapper.SearchDay()`  
StartBackup
10. `ssmSdkWrapper.StartBackup()`  
Response of StartBackup
11. `ssmSdkWrapper.OnResponse`  
Receive Event: BackupStatusChanged
12. `ssmSdkWrapper.OnEvent`  
Release Authority
13. `ssmSdkWrapper.ReleaseSearchAuthority()`  
Logout
14. `ssmSdkWrapper.Logout()`  
Release
15. `ssmSdkWrapper.ReleaseEvent()`

# 구현 방법

이 절에서는 Backup 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다.

## SsmSdkWrapper 참조 추가

### 참고

SsmSdkWrapper 참조 추가 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

## SsmSdkWrapper Callback 함수 추가

### Callback 함수 구현: OnResponse

Callback 함수 구현: OnResponse 작성 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

### Callback 함수 구현: OnEvent

BackupSample 클래스의 OnEvent()에 Login와 BackupStatusChanged 케이스를 구현합니다. Login 이벤트가 발생하면, 현재 연결된 Device와 Camera 목록을 콤보박스에 업데이트 합니다. BackupStatusChanged 이벤트가 발생하면, ProgressBar에 백업 진행률을 업데이트합니다.

```
// BackupSample.cs
private void OnEvent(UInt32 eventID, String info)
{
    _logger.WLOGD(
        "OnEvent()::" +
        " Event ID=" + eventID +
        " Info=" + info);

    if (eventID == (uint)EventID.Login)
    {
        SetObjectList(info);
    }
}
```

```
        UpdateComboDeives();
        UpdateComboCameras();
    }
    else if (eventID == (uint)EventID.BackupStatusChanged)
    {
        UpdateBackupStatusProgressbar(info);
    }
}
```

---

## SsmSdkWrapper 초기화

### 참고

SsmSdkWrapper 초기화 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

---

## 로그인

### 참고

로그인 구현 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

---

## 검색 권한 획득

GetAuthority 버튼 클릭 이벤트 핸들러를 생성하여 검색 권한 획득 명령 전송 요청을 수행합니다. Device Uuid와 멀티패스워드 정보가 메서드의 파라미터로 필요합니다. 멀티패스워드를 사용하지 않는다면, 빈 문자열("")을 전달합니다.

```
// BackupSample.cs
private void btnGetAuthority_Click(object sender, EventArgs e)
{
    try
    {
        UInt32 sequenceID = 0;
        uint result = ssmSdkWrapper.GetSearchAuthority(_deviceUuid, "", ref
sequenceID);
        _logger.WLOGD("GetAuthority():Result=" + result + ", SequenceID=" +
sequenceID);
    }
}
```



```
        catch (Exception ex)
        {
            _logger.WLOGD(ex.Message);
        }
    }
```

---

## 백업 시작

StartBackup 버튼 클릭 이벤트 핸들러에서 백업 시작을 요청합니다.

```
// BackupSample.cs
private void btnStartBackup_Click(object sender, EventArgs e)
{
    uint result = 0;
    int trackID = -1;
    uint fileSizeLimit = 0;
    uint timePeriodLimit = 0;
    uint sequenceID = 0;
    try
    {
        result = ssmSdkWrapper.StartBackup(
            _cameraUuid,
            trackID,
            txtStartTime.Text,
            false,
            txtEndTime.Text,
            false,
            (uint)_fileFormat,
            txtFilePath.Text,
            fileSizeLimit,
            timePeriodLimit,
            ref _medialD,
            ref sequenceID);
    }
    catch (Exception ex)
    {
        _logger.WLOGD(ex.Message);
    }
}
```

```
_logger.WLOGD("StartBackup()::Result=" + result + ", SequenceID=" + sequenceID);  
}
```

---

## 백업 종료

백업 도중 작업을 중단하고 싶다면, 백업 종료 명령을 전송합니다.

```
// BackupSample.cs  
private void btnStopBackup_Click(object sender, EventArgs e)  
{  
    uint result = 0;  
    uint sequenceID = 0;  
    try  
    {  
        result = ssmSdkWrapper.StopBackup(_cameraUuid, _medialD, ref sequenceID);  
    }  
    catch (Exception ex)  
    {  
        _logger.WLOGD(ex.Message);  
    }  
    _logger.WLOGD("StartBackup()::Result=" + result + ", SequenceID=" + sequenceID);  
}
```

---

## 검색 권한 반환

Release Authority 버튼 클릭 이벤트 핸들러를 생성하여 검색 권한 반환 명령을 수행합니다. 권한을 반환할 장비의 Uuid를 메서드의 파라미터로 전달합니다.

```
// BackupSample.cs  
private void btnReleaseAuthority_Click(object sender, EventArgs e)  
{  
    try  
    {  
        UInt32 sequenceID = 0;  
        uint result = ssmSdkWrapper.ReleaseSearchAuthority(_deviceUuid, ref  
sequenceID);  
        _logger.WLOGD("ReleaseAuthority()::Result=" + result + ", SequenceID=" +  
sequenceID);  
    }  
}
```

```
catch (Exception ex)
{
    _logger.WLOGD(ex.Message);
}
}
```

---

## 로그아웃

### 참고

로그아웃 구현 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

---

## SsmSdkWrapper 해제

### 참고

SsmSdkWrapper 해제 구현 방법은 본 문서의 CHAPTER 4에 기술된 LogIn 샘플 프로그램의 '구현 방법' 절을(32쪽) 참고 바랍니다.

## See also

LogIn 샘플 프로그램

Wisenet SSM Client SDK API Reference (v2.10.6)

## FAQ

StartBackup에서 TrackID에 채워야 하는 값은?

- ➔ 예제에서는 -1을 넣었지만, Search TrackID, Search Day의 응답으로 받은 타임라인 구간에서 얻은 TrackID를 넣어줘야 합니다.

## CHAPTER 8

# SingleLive 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 1채널 라이브 영상을 화면에 출력하는 방법을 설명합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ

# 샘플 프로그램 소개

SingleLive 샘플 프로그램은 특정 네트워크 장비로부터 1채널 라이브 영상을 수신하여 화면에 보여주는 예제입니다. 샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

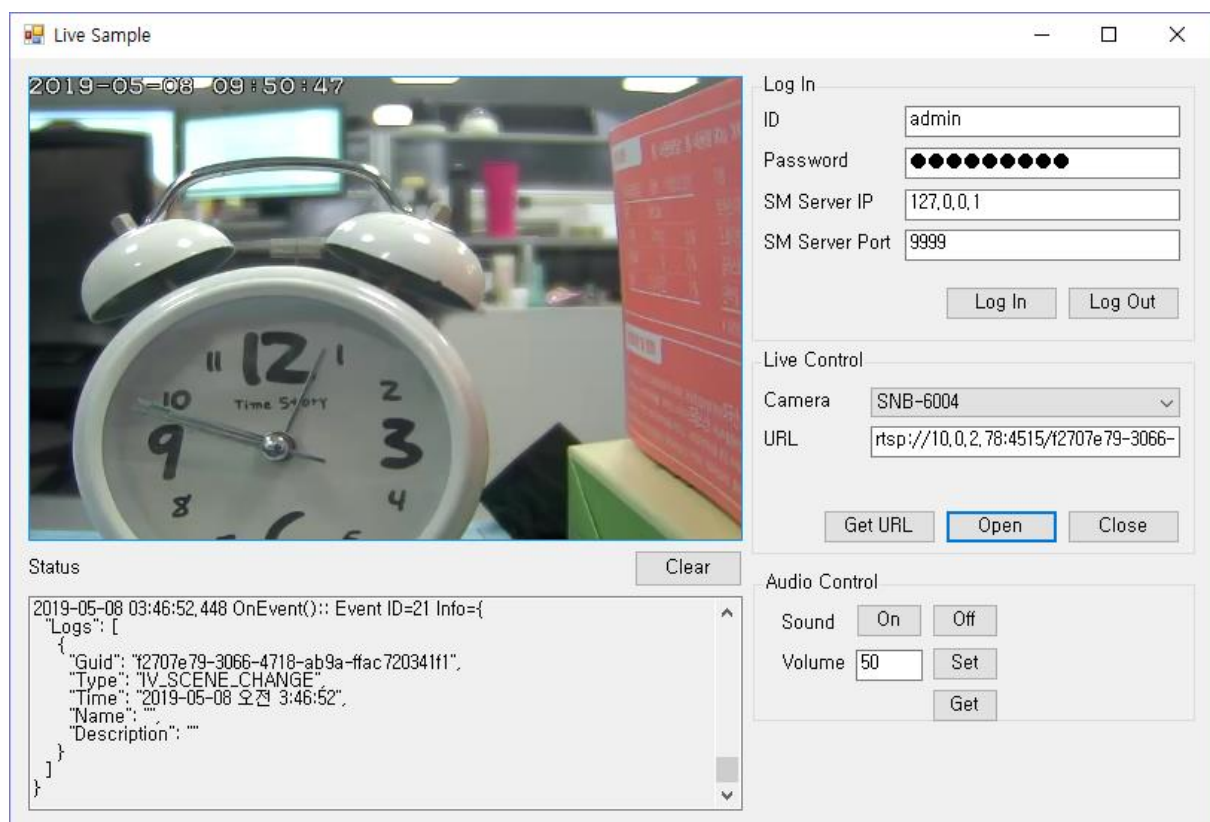


Figure 6 SingleLive 샘플 프로그램 실행 화면

- Step 3.** 로그인에 정상적으로 이루어지면 [Camera] ComboBox에 현재 연결 된 카메라 이름이 나타 납니다.
- Step 4.** ComboBox에서 연결할 Camera를 선택합니다.
- Step 5.** [Get Rtp Url] 버튼을 클릭하여 선택된 카메라의 URL을 얻어옵니다.
- Step 6.** [Open] 버튼을 클릭하여 얻어온 URL을 통해, 선택한 카메라로부터 미디어 스트림을 수신 합니다.
- Step 7.** 좌측 영상 창에 라이브 영상이 표시되는 것을 확인합니다.

**Step 8.** [Close] 버튼을 클릭하여, 미디어 스트림 수신을 중단하고 리소스를 해제합니다.

**Step 9.** [Logout] 버튼을 클릭하여 서버와 접속을 해제합니다.

**Step 10.** 프로그램을 종료합니다.

## API 호출 순서

이 샘플 프로그램은 Wisenet SSM Client SDK가 제공하는 라이브러리를 사용하여 1채널 라이브 영상 출력 기능을 구현하였습니다. 샘플 프로그램의 API 호출 순서는 다음과 같습니다.

### Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`
2. `ssmSdkWrapper.InitailizeMedia()`

### Login

3. `ssmSdkWrapper.Login()`

### Response of Login

4. `ssmSdkWrapper.OnResponse`

### Receive Event: LogIn

5. `ssmSdkWrapper.OnEvent`

### Receive Event: ObjectConnected

6. `ssmSdkWrapper.OnEvent`

### GetRtspUrl

7. `ssmSdkWrapper.GetRtspUrl()`

### Response of GetRtspUrl

8. `ssmSdkWrapper.OnResponse`

### Open Media

9. `ssmSdkWrapper.MediaOpen()`

### Close Media

10. `ssmSdkWrapper.MediaClose()`

### Logout

11. SsmSdkWrapper.Logout()
- Release
12. ssmSdkWrapper.ReleaseEvent()
13. ssmSdkWrapper.ReleaseMedia()

## 구현 방법

이 절에서는 SingleLive 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다

---

## SsmSdkWrapper 초기화

LiveSample 클래스의 멤버 함수 LiveSample\_Load() 안에서 SsmSdkWrapper를 생성하고 ssmSdkWrapper.InitializeEvent(), ssmSdkWrapper.InitializeMedia()를 호출하여 초기화합니다. 영상을 화면에 출력하려면 InitailizeMedia의 첫번째 파라미터에 윈도우 핸들을 두번째 파라미터를 0으로 합니다. 1, 2로 설정하면, 영상 Open시 OnVideo, OnAudio 콜백함수가 호출됩니다. (API 문서 4장 참고). 한편, Media ID는 media를 구분할 때 사용하는 값이므로 반드시 저장해둡니다.

```
// SingleLiveSample.cs
private SsmSdkWrapper ssmSdkWrapper = null;

private void LiveSample_Load(object sender, EventArgs e)
{
    ssmSdkWrapper = new SsmSdkWrapper(this.OnResponse, this.OnEvent, this.OnVideo,
    this.OnAudio);
    ssmSdkWrapper.InitializeEvent();

    ssmSdkWrapper.InitializeMedia(pictureBox1.Handle, 0, ref this.mediaId);
    ssmSdkWrapper.MoveWindow((UInt32)this.mediaId, 0, 0, pictureBox1.Width,
    pictureBox1.Height);
}
```

## RTSP URL 얻어오기

버튼 클릭 이벤트 핸들러를 생성하여 GetRtspUrl 명령을 수행합니다.

SsmSdkWrapper의 GetRtspUrl() 메서드를 통해, 선택한 카메라의 URL을 얻어옵니다. 이 샘플에서는 프로파일 값은 HIGH(=1), 스트림 타입은 LIVE(=2), 프로토콜은 TCP(=1)로 설정합니다.

얻어온 URL 주소는 SsmMediaSdk 컨트롤의 Open() 메서드에서 사용됩니다.

```
// SingleLiveSample.cs
private void btnGetRtspUrl_Click(object sender, EventArgs e)
{
    UInt32 sequenceID = 0;
    uint resCode = ssmSdkWrapper.GetRtspUrl(_cameraUuid, 2, 2, 1, ref sequenceID); //
    MEDIUM, LIVE, TCP
    _logger.WLOGD("GetRtspUrl()::Result=" + resCode + ", SequenceID=" + sequenceID);
}
```

## 미디어 오픈

미디어 오픈 버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 MediaOpen() 메서드를 호출하여 미디어 스트림 수신을 시작합니다.

InitailizeMedia()시 Display 모드를 선택했다면, 화면에 렌더링 된 영상이 보여질 것입니다.

Relay 모드를 선택했다면, 수신된 미디어 스트림은 OnVideo와 OnAudio 이벤트로 전달됩니다.

Media Type을 Live로 선택한다면, StartTime, TimeZone, DST는 사용하지 않습니다. 따라서, 각각 "0", "", false 값을 파라미터로 전달하면 됩니다.

```
// SingleLiveSample.cs
private void btnOpen_Click(object sender, EventArgs e)
{
    ssmSdkWrapper.MediaOpen(
        txtRtspUrl.Text,
        txtID.Text,
        txtPW.Text,
        _cameraName,
        (uint)MediaType.Live,
        "0", // start-time
        "", // Time zone
        false); // DST
}
```



---

## 미디어 클로즈

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 MediaClose() 메서드를 통해 미디어 스트림 수신을 중단합니다.

```
// SingleLiveSample.cs

private void btnClose_Click(object sender, EventArgs e)
{
    ssmSdkWrapper.MediaClose();
}
```

---

## SsmSdkWrapper 해제

SsmSdkWrapper의 사용이 끝난 후에는 서비스를 중단하고 자원을 해제해야 합니다.

LiveSample 클래스의 멤버 함수 LogInSample\_Closing() 안에서 ReleaseEvent(), ReleaseMedia() 메서드를 호출합니다.

```
// SingleLiveSample.cs

private void LiveSample_Closing(object sender, FormClosingEventArgs e)
{
    ssmSdkWrapper.ReleaseEvent();

    ssmSdkWrapper.MediaClose(this.mediaId);
    ssmSdkWrapper.ReleaseMedia(this.mediaId);
}
```

## See also

LogIn 샘플 프로그램

Wisenet SSM Client SDK API Reference (v2.10.6)

# FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 9

# PTZ 샘플 프로그램

---

PTZ 샘플 프로그램은 Wisenet SSM Client SDK를 사용하여 Pan, Tilt, Zoom, Preset 기능을 사용하는 방법을 설명하기 위한 예제입니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ

# 샘플 프로그램 소개

PTZ 샘플 프로그램은 PTZ 기능을 지원하는 네트워크 카메라의 Pan, Tilt, Zoom, Preset 기능을 사용하는 예제입니다. 샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

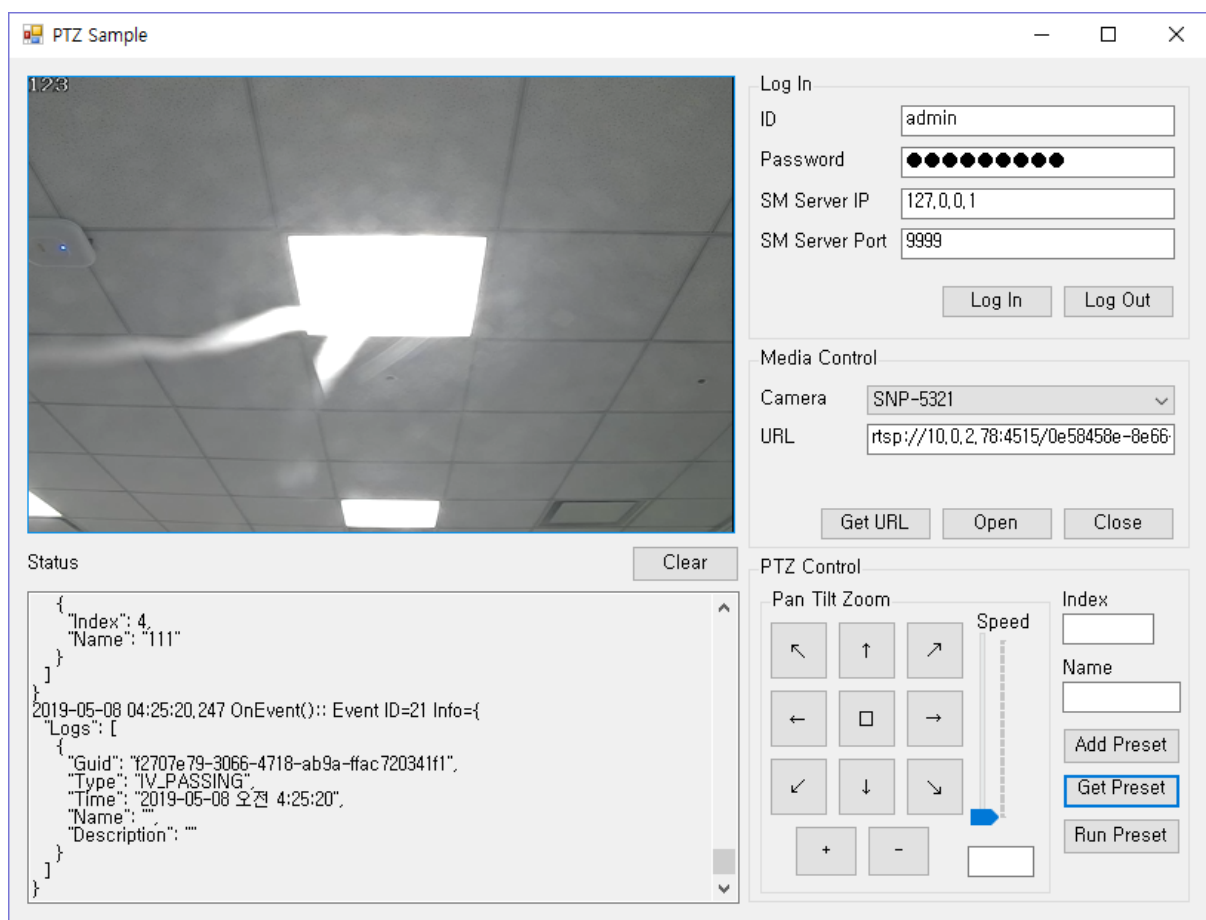


Figure 7 PTZ 샘플 프로그램 실행 화면

- Step 3.** 로그인에 정상적으로 이루어지면 [Camera] ComboBox에 현재 연결 된 카메라 이름이 나타납니다.
- Step 4.** ComboBox에서 연결할 Camera를 선택합니다.
- Step 5.** [Get Rtp Url] 버튼을 클릭하여 선택된 카메라의 URL을 얻어옵니다.
- Step 6.** [Open] 버튼을 클릭하여 얻어온 URL을 통해, 선택한 카메라로부터 미디어 스트림을 수신합니다.

- Step 7. 좌측 영상 창에 라이브 영상이 표시되는 것을 확인합니다.
- Step 8. Pan/Tilt/Zoom 버튼을 눌러 해당 기능이 동작하는지 확인합니다.
- Step 9. □ 버튼을 눌러 PTZ 명령을 종료합니다.
- Step 10. Speed 바를 조정하여 PTZ 동작 속도를 변경합니다.
- Step 11. [Add Preset list] 버튼을 클릭하여 Preset을 추가합니다.
- Step 12. [Get Preset list] 버튼을 클릭하여 Preset 리스트를 가져옵니다.
- Step 13. Preset 번호와 인덱스 이름을 입력하고 [Run Preset] 버튼을 클릭하여 Preset을 로드합니다.
- Step 14. [Close] 버튼을 클릭하여, 미디어 스트림 수신을 중단하고 리소스를 해제합니다.
- Step 15. [Logout] 버튼을 클릭하여 서버와 접속을 해제합니다.
- Step 16. 프로그램을 종료합니다.

## API 호출 순서

이 샘플 프로그램은 Wisenet SSM Client SDK가 제공하는 라이브러리를 사용하여 PTZ 명령 기능을 구현하였습니다. 샘플 프로그램의 API 호출 순서는 다음과 같습니다.

### Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`
2. `ssmSdkWrapper.InitailizeMedia()`

### Login

3. `ssmSdkWrapper.Login()`

### Response of Login

4. `ssmSdkWrapper.OnResponse`

### Receive Event: Login

5. `ssmSdkWrapper.OnEvent`

### Receive Event: ObjectConnected

6. `ssmSdkWrapper.OnEvent`

### GetRtspUrl

7. `ssmSdkWrapper.GetRtspUrl()`

Response of `GetRtspUrl`

8. `ssmSdkWrapper.OnResponse`

Open Media

9. `ssmSdkWrapper.MediaOpen()`

ControlPtz

10. `ssmSdkWrapper.ControlPtz()`

AddPreset

11. `ssmSdkWrapper.AddPreset()`

Response of `AddPreset`

12. `ssmSdkWrapper.OnResponse`

GetPresetList

13. `ssmSdkWrapper.GetPresetList()`

Response of `GetPresetList`

14. `ssmSdkWrapper.OnResponse`

RunPreset

15. `ssmSdkWrapper.RunPreset()`

Reponse of `RunPreset`

16. `ssmSdkWrapper.OnResponse`

Close Media

17. `ssmSdkWrapper.MediaClose()`

Logout

18. `SsmSdkWrapper.Logout()`

Release

19. `ssmSdkWrapper.ReleaseEvent()`

20. `ssmSdkWrapper.ReleaseMedia()`

# 구현 방법

이 절에서는 PTZ 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다

## SsmSdkWrapper 초기화

### 참고

SsmSdkWrapper 초기화 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

## RTSP URL 얻어오기

### 참고

RTSP URL 얻어오기 구현 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

## 미디어 오픈

### 참고

미디어 오픈 구현 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

## PTZ 컨트롤

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 ControlPtz() 메서드를 사용해 Pan/Tilt/Zoom 기능을 수행합니다. nAction 값으로 PTZ 상세 명령(Up, Left, Zoom In, Stop, etc.)을 설정합니다. nSpeed 값으로 제어 속도를 설정합니다. PTZ 명령 전송 후엔, 반드시 STOP 명령을 전송해야 이전 PTZ 명령 수행이 중단됩니다.

```
// PTZSample.cs
```

```
private void btnPTZLeft_Click(object sender, EventArgs e)
{
    ssmSdkWrapper.ControlPtz(
```

```
_cameraUuid,  
(uint)PtzAction.Left,  
(ushort)trackBarPTZSpeed.Value);  
}
```

---

## 프리셋 추가

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 AddPreset() 메서드를 사용해 카메라에 해당 프리셋을 추가합니다. 프리셋 번호와 이름을 함께 설정해야 합니다.

```
// PTZSample.cs  
private void btnAddPreset_Click(object sender, EventArgs e)  
{  
    try  
    {  
        UInt32 sequenceID = 0;  
        uint resCode = ssmSdkWrapper.AddPreset(  
            _cameraUuid,  
            Convert.ToUInt32(txtPresetIndex.Text),  
            txtPresetName.Text,  
            ref sequenceID);  
        _logger.WLOGD("AddPreset()::Result=" + resCode + ", SequenceID=" +  
sequenceID);  
    }  
    catch (Exception ex)  
    {  
        _logger.WLOGD(ex.Message);  
    }  
}
```

---

## 프리셋 목록 가져오기

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 GetPresetList() 메서드를 사용해 카메라에 설정된 모든 프리셋 목록을 가져옵니다. 프리셋 목록은 SsmSdkWrapper의 OnResponse 이벤트로 수신됩니다.

```
// PTZSample.cs  
private void btnGetPresetList_Click(object sender, EventArgs e)  
{
```



```
try
{
    UInt32 sequenceID = 0;
    uint resCode = ssmSdkWrapper.GetPresetList(_cameraUuid, ref sequenceID);
    _logger.WLOGD("GetPresetList():Result=" + resCode + ", SequenceID=" +
sequenceID);
}
catch (Exception ex)
{
    _logger.WLOGD(ex.Message);
}
}
```

---

## 프리셋 실행

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 RunPreset() 메서드를 통해 카메라에 선택한 프리셋을 실행합니다.

```
// PTZSample.cs
private void btnRunPreset_Click(object sender, EventArgs e)
{
    try
    {
        UInt32 sequenceID = 0;
        uint resCode = ssmSdkWrapper.RunPreset(_cameraUuid,
Convert.ToUInt32(txtPresetIndex.Text), ref sequenceID);
        _logger.WLOGD("RunPreset():Result=" + resCode + ", SequenceID=" +
sequenceID);
    }
    catch (Exception ex)
    {
        _logger.WLOGD(ex.Message);
    }
}
```

---

## 미디어 클로즈

참고

미디어 클로즈 구현 방법은 본 문서의 CHAPTER 7에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

---

## SsmSdkWrapper 해제

### 참고

SsmSdkWrapper 해제 구현 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

## See also

LogIn 샘플 프로그램

SingleLive 샘플 프로그램

Wisenet SSM Client SDK API Reference (v2.10.6)

## FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.

현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 10

# MultiMonitoring 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 4채널 라이브 영상을 화면에 출력하는 방법을 설명합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ

# 샘플 프로그램 소개

MultiMonitoring 샘플 프로그램은 네트워크 장비로부터 4채널 라이브 영상을 수신하여 화면에 표시하는 예제입니다.

샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.
- Step 3.** 로그인에 정상적으로 이루어지면 [Camera] ComboBox에 현재 연결 된 카메라 이름이 나타납니다.

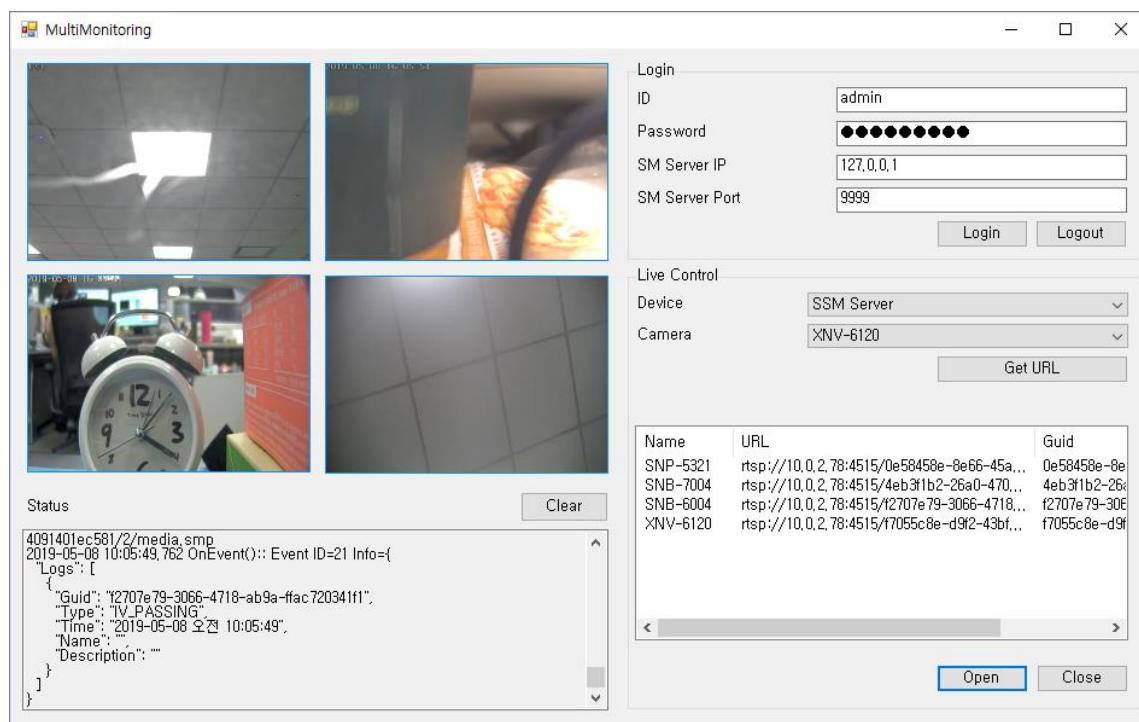


Figure 8 MultiMonitoring 샘플 프로그램 실행 화면

- Step 4.** [Get whole cameras' url] 버튼을 클릭하여, Device에 연결된 카메라의 url 리스트를 가져옵니다. url 리스트는 ListView에 나타납니다.
- Step 5.** ListView에서 카메라 이름을 선택하고 [Open] 버튼을 클릭해, 미디어 스트림을 수신을 시작합니다.
- Step 6.** 좌측 영상 창에 라이브 영상이 표시되는 것을 확인합니다. 선택한 순서대로 표시됩니다.
- Step 7.** [Close] 버튼을 클릭하여, 모든 미디어 스트림 수신을 중단하고 리소스를 해제합니다.

**Step 8.** [Logout]버튼을 클릭하여 서버와 접속을 해제합니다.

**Step 9.** 프로그램을 종료합니다.

## API 호출 순서

이 샘플 프로그램은 Wisenet SSM Client SDK가 제공하는 라이브러리를 사용하여 1채널 라이브 영상 출력 기능을 구현하였습니다. 샘플 프로그램의 API 호출 순서는 다음과 같습니다.

### Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`
2. `ssmSdkWrapper.InitailizeMedia()`

### Login

3. `ssmSdkWrapper.Login()`

### Response of Login

4. `ssmSdkWrapper.OnResponse`

### Receive Event: LogIn

5. `ssmSdkWrapper.OnEvent`

### Receive Event: ObjectConnected

6. `ssmSdkWrapper.OnEvent`

### GetRtspUrl

7. `ssmSdkWrapper.GetRtspUrl()`

### Response of GetRtspUrl

8. `ssmSdkWrapper.OnResponse`

### Open Media

9. `ssmSdkWrapper.MediaOpen()`

### Close Media

10. `ssmSdkWrapper.MediaClose()`

### Logout

11. `SsmSdkWrapper.Logout()`

Release

12. `ssmSdkWrapper.ReleaseEvent()`
13. `ssmSdkWrapper.ReleaseMedia()`

## 구현 방법

이 절에서는 MultiMonitoring 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다

---

### SsmSdkWrapper 초기화

#### 참고

SsmSdkWrapper 초기화 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

InitializeMedia()를 타일 개수만큼 호출하고 Media ID를 저장해둡니다.

---

### RTSP URL 얻어오기

#### 참고

RTSP URL 얻어오기 구현 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

---

### 미디어 오픈

#### 참고

미디어 오픈 구현 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

---

## 미디어 클로즈

### 참고

미디어 클로즈 구현 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

---

## SsmSdkWrapper 해제

### 참고

SsmSdkWrapper 해제 구현 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

## See also

LogIn 샘플 프로그램

SingleLive 샘플 프로그램

Wisenet SSM Client SDK API Reference (v2.10.6)

## FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 11

# Playback 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 녹화된 영상을 화면에 표시하는 방법을 설명합니다. 예제에서는 H.264 프로파일을 사용합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ



## 샘플 프로그램 소개

Playback 샘플 프로그램은 네트워크 장비로부터 녹화 영상을 수신하여 화면에 표시하는 예제입니다. 샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

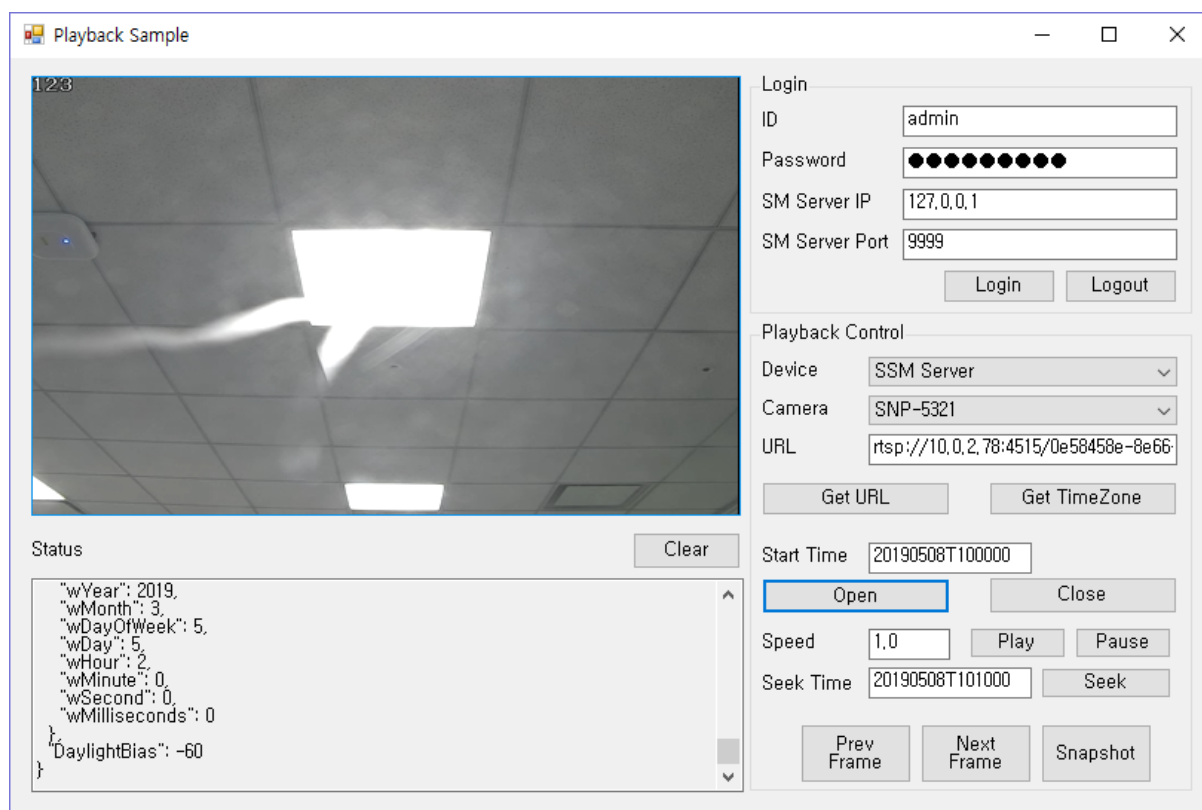


Figure 9 Playback 샘플 프로그램 실행 화면

- Step 3.** 로그인에 정상적으로 이루어지면 [Device]와 [Camera] ComboBox에 현재 연결 된 장비와 카메라 이름이 나타납니다.
- Step 4.** Device와 Camera를 각각 ComboBox에서 선택합니다.
- Step 5.** [Get URL] 버튼을 클릭하여 선택한 카메라의 URL을 얻어옵니다.
- Step 6.** [Get TimeZone] 버튼을 클릭하여, 디바이스의 TimeZone 정보를 얻어옵니다.
- Step 7.** [Start Time] 텍스트 박스에 녹화 영상의 시작 시간을 입력합니다. 포맷은 "yyyyMMddThhmmss"입니다.

- Step 8.** [Open] 버튼을 클릭하여, 선택한 카메라로부터 미디어 스트림 수신을 시작합니다.
- Step 9.** 좌측 영상 창에 녹화 영상이 표시되는 것을 확인합니다.
- Step 10.** [Close] 버튼을 클릭하여, 미디어 스트림 수신을 중단하고 리소스를 해제합니다.
- Step 11.** [Logout] 버튼을 클릭하여 서버와 접속을 해제합니다.
- Step 12.** 프로그램을 종료합니다.

## API 호출 순서

이 샘플 프로그램은 Wisenet SSM Client SDK가 제공하는 라이브러리를 사용하여 녹화 영상을 검색 및 출력하는 기능을 구현하였습니다. 샘플 프로그램의 API 호출 순서는 다음과 같습니다.

### Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`
2. `ssmSdkWrapper.InitailizeMedia()`

### Login

3. `ssmSdkWrapper.Login()`

### Response of Login

4. `ssmSdkWrapper.OnResponse`

### Receive Event: Login

5. `ssmSdkWrapper.OnEvent`

### Receive Event: ObjectConnected

6. `ssmSdkWrapper.OnEvent`

### GetRtspUrl

7. `ssmSdkWrapper.GetRtspUrl()`

### Response of GetRtspUrl

8. `ssmSdkWrapper.OnResponse`

### Open Media

9. `ssmSdkWrapper.MediaOpen()`

### Close Media

```
10. ssmSdkWrapper.MediaClose()

Logout

11. SsmSdkWrapper.Logout()

Release

12. ssmSdkWrapper.ReleaseEvent()

13. ssmSdkWrapper.ReleaseMedia()
```

## 구현 방법

이 절에서는 Playback 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다

---

### SsmSdkWrapper 초기화

#### 참고

SsmSdkWrapper 초기화 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

---

### RTSP URL 얻어오기

#### 참고

RTSP URL 얻어오기 구현 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

---

### TimeZoneInfo 얻어오기

버튼 클릭 Event Handler를 생성하고, SsmSdkWrapper의 GetTimeZoneInfo() 메서드를 통해 선택한 장비의 TimeZone 정보를 얻어옵니다.

```
// PlaybackSample.cs
```

```
private void btnGetTimeZone_Click(object sender, EventArgs e)
{
    _timeZoneInfo = ssmSdkWrapper.GetTimeZoneInfo(_deviceId);
    _logger.WLOGD("GetTimeZoneInfo::Result=" + _timeZoneInfo);
}
```

---

## 미디어 오픈

버튼 클릭 Event Handler를 생성하고, SsmSdkWrapper의 MediaOpen() 메서드를 통해 Playback 미디어 스트림 수신을 시작합니다. Playback 미디어 오픈 할 때는, StartTime, TimeZoneInfo, DST 정보가 추가적으로 필요합니다. TimeZoneInfo() 메서드를 통해 얻어온 TimeZoneInfo 값을 파라미터로 넘겨줍니다.

```
// PlaybackSample.cs

private void btnOpen_Click(object sender, EventArgs e)
{
    ssmSdkWrapper.MediaOpen(
        txtRtspUrl.Text,
        txtID.Text,
        txtPW.Text,
        _cameraName,
        (uint)MediaType.Playback,
        txtStartTime.Text, // StartTime
        _timeZoneInfo,    // TimeZoneInfo
        false);           // DST
}
```

---

## 미디어 클로즈

### 참고

미디어 클로즈 구현 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

---

## SsmSdkWrapper 해제

### 참고

SsmSdkWrapper 해제 구현 방법은 본 문서의 CHAPTER 8에 기술된 SingleLive 샘플 프로그램의 '구현 방법' 절을(63쪽) 참고 바랍니다.

## See also

LogIn 샘플 프로그램

SingleLive 샘플 프로그램

Wisenet SSM Client SDK API Reference (v2.10.6)

## FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.

현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 12

# InsertLog 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 Wisenet SSM Core Server 서버로 이벤트 정보를 전송하는 방법을 설명합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ

# 샘플 프로그램 소개

InsertLog 샘플 프로그램은 Wisenet SSM Core Server 서버로 이벤트 정보를 전송하는 예제입니다. 샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

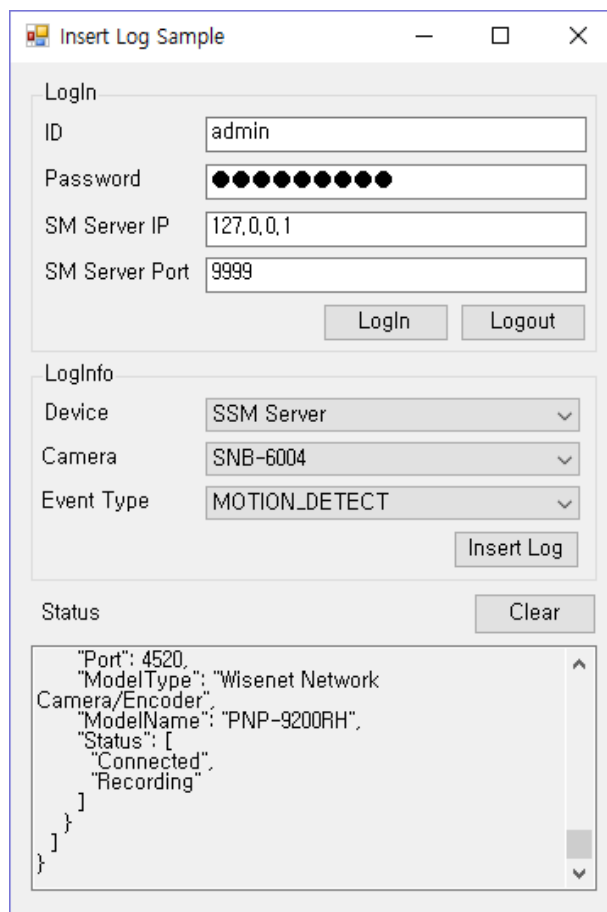


Figure 10 Insert log 샘플 프로그램 실행 화면

- Step 3.** 로그인이 정상적으로 이루어지면 [Device]와 [Camera] ComboBox에 현재 연결 된 장비와 카메라 이름이 나타납니다.
- Step 4.** Device와 Camera를 각각 ComboBox에서 선택합니다.
- Step 5.** 이벤트 타입을 선택합니다.
- Step 6.** [Insert Log] 버튼을 클릭하여 이벤트를 전송합니다.

- Step 7.** 전송한 이벤트가 수신되었는지 확인합니다.
- Step 8.** [Logout] 버튼을 클릭하여 서버와 접속을 해제합니다.
- Step 9.** 프로그램을 종료합니다.

## API 호출 순서

이 샘플 프로그램은 Wisenet SSM Client SDK가 제공하는 라이브러리를 사용하여 서버로 이벤트를 전송하는 기능을 구현하였습니다. 샘플 프로그램의 API 호출 순서는 다음과 같습니다.

### Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`

### Login

2. `ssmSdkWrapper.Login()`

### Response of Login

3. `ssmSdkWrapper.OnResponse`

### Receive Event: LogIn

4. `ssmSdkWrapper.OnEvent`

### Receive Event: ObjectConnected

5. `ssmSdkWrapper.OnEvent`
6. `ssmSdkWrapper.InsertExternalLog()`

### Response of InsertExternalLog

7. `ssmSdkWrapper.OnResponse`

### Recevice Event: Log

8. `ssmSdkWrapper.OnEvent`

### Logout

9. `ssmSdkWrapper.Logout()`

### Release

10. `ssmSdkWrapper.ReleaseEvent()`



# 구현 방법

이 절에서는 InsertLog 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다.

---

## 이벤트 로그 전송

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 InsertExternalLog() 메서드를 통해 이벤트 로그를 전송합니다.

```
// InsertLogSample.cs
private void btnInsertLog_Click(object sender, EventArgs e)
{
    uint resCode = 0;
    UInt32 sequenceID = 0;
    try
    {
        resCode = ssmSdkWrapper.InsertExternalLog(
            _deviceUuid,
            _cameraUuid,
            _logTypeDictionary[_logType],
            ref sequenceID);
    }
    catch (Exception ex)
    {
        _logger.WLOGD(ex.Message);
    }
    _logger.WLOGD("InsertLog()::Result=" + resCode + ", SequenceID=" + sequenceID);
}
```

---

## Event Key를 사용하여 이벤트 로그 전송

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 InsertExternalLogEventKey() 메서드를 통해 이벤트 로그를 전송합니다.

```
// InsertLogSample.cs
private void btnInsertLogWithEventKey_Click(object sender, EventArgs e)
{
    uint resCode = 0;
    UInt32 sequenceID = 0;

    UInt32 eventKey = 0;

    if (!UInt32.TryParse(this.txtEventKey.Text, out eventKey))
    {
        _logger.WLOGD("[ERROR] Check Event Key");
        return;
    }

    try
    {
        resCode = ssmSdkWrapper.InsertExternalLogEventKey(
            _deviceUuid,
            _cameraUuid,
            eventKey,
            ref sequenceID);
    }
    catch (Exception ex)
    {
        _logger.WLOGD(ex.Message);
    }

    _logger.WLOGD("InsertLog()::Result=" + resCode + ", SequenceID=" + sequenceID);
}
```

## See also

Login 샘플 프로그램

Backup 샘플 프로그램

Wisenet SSM Client SDK API Reference (v2.10.6)

## FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 13

# LiveSnapshot 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 Live 영상을 캡처하는 방법을 설명합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ

## 샘플 프로그램 소개

LiveSnapshot 샘플 프로그램은 라이브 영상을 캡처하여 파일로 저장하는 예제입니다. 샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

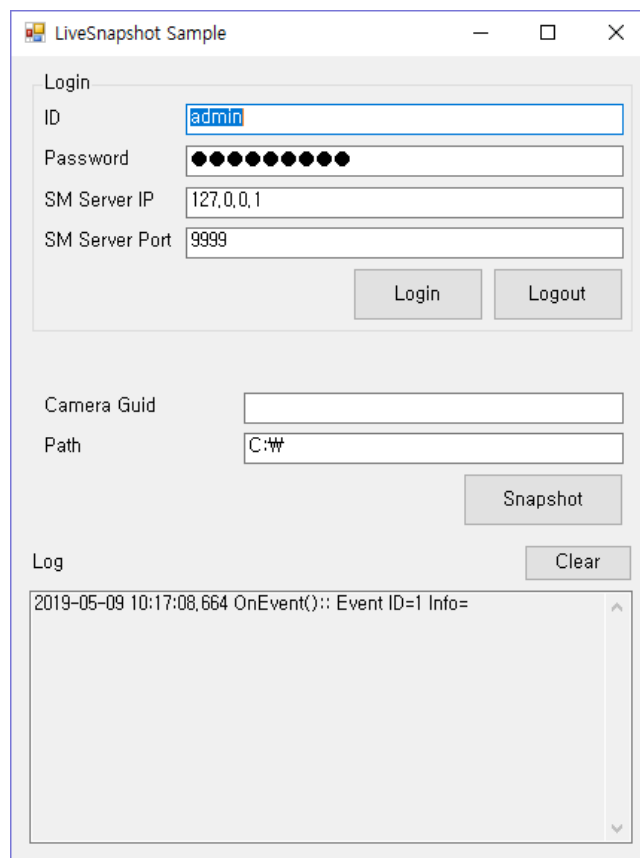


Figure 11 LiveSnapshot 샘플 프로그램 실행 화면

- Step 3.** Camera의 Guid와 Path를 입력합니다.
- Step 4.** [Snapshot] 버튼을 클릭하여 명령을 실행합니다.
- Step 5.** [Logout] 버튼을 클릭하여 서버와 접속을 해제합니다.
- Step 6.** 프로그램을 종료합니다.

# API 호출 순서

이 샘플 프로그램은 Wisenet SSM Client SDK가 제공하는 라이브러리를 사용하여 라이브 캡처 이미지를 생성하는 기능을 구현하였습니다. 샘플 프로그램의 API 호출 순서는 다음과 같습니다.

Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`

Login

2. `ssmSdkWrapper.Login()`

Response of Login

3. `ssmSdkWrapper.OnResponse`

Receive Event: LogIn

4. `ssmSdkWrapper.OnEvent`

Receive Event: ObjectConnected

5. `ssmSdkWrapper.OnEvent`

Get snapshot

6. `ssmSdkWrapper.GetSnapshot`

Logout

7. `ssmSdkWrapper.Logout()`

Release

8. `ssmSdkWrapper.ReleaseEvent()`

# 구현 방법

이 절에서는 LiveSnapshot 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다.

## Live snapshot 요청

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 GetSnapshot() 메서드를 통해 Live snapshot 요청을 전송합니다.

```
// LiveSnapshotSample.cs
private void Snapshot_Click(object sender, EventArgs e)
{
    String guidText = textBoxUnitID.Text;
    Guid guid = Guid.Empty;
    UInt32 sequenceID = 0;
    UInt32 resCode = 0;

    if (Guid.TryParse(guidText, out guid))
    {
        SnapshotModel snapshotRequest = new SnapshotModel();

        snapshotRequest.CameraGuid = textBoxUnitID.Text;
        snapshotRequest.ProfileType = 1; // HIGH
        snapshotRequest.FilePath = textBoxFilePath.Text;

        String json = JsonConvert.SerializeObject(snapshotRequest,
Formatting.Indented);
        resCode = ssmSdkWrapper.GetSnapshot(json, ref sequenceID);
        _logger.WLOGD("GetSnapshot():Result=" + resCode + ", SequenceID=" +
sequenceID);
    }
    else
    {
        MessageBox.Show("Check the guid");
    }
}
```

## See also

[Login 샘플 프로그램](#)

[Backup 샘플 프로그램](#)

[Wisenet SSM Client SDK API Reference \(v2.10.6\)](#)

## FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.



## CHAPTER 14

# LiveLocalRecord 샘플 프로그램 램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 Live 영상을 .avi 파일로 저장 하는 방법을 설명합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ

# 샘플 프로그램 소개

LiveLocalRecord 샘플 프로그램은 라이브 영상을 동영상 파일로 저장하는 예제입니다. 샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

The screenshot shows the 'LiveLocalRecord Sample' application window. It has a 'Login' section with fields for ID (admin), Password (masked with dots), SM Server IP (127.0.0.1), and SM Server Port (9999). There are 'Login' and 'Logout' buttons. Below this is a section for recording settings with fields for Camera Guid, Path (C:\W), and File name (Record.avi), along with 'Record Start' and 'Record Stop' buttons. At the bottom is a 'Log' section with a 'Clear' button and a text area displaying JSON data: {"Type": "Camera", "Name": "PNP-9200RH", "NameExt": "Camera", "IpAddress": "192.168.18.84", "Port": 4520, "ModelType": "Wisenet Network Camera/Encoder", "ModelName": "PNP-9200RH", "Status": [{"Connected": true, "Recording": true}]}.

Figure 12 LiveLocalRecord 샘플 프로그램 실행 화면

- Step 3.** Camera의 Guid와 Path, File name을 입력합니다.
- Step 4.** [Record start] 버튼을 클릭하여 명령을 실행합니다.
- Step 5.** [Record stop] 버튼을 클릭하여 명령을 실행합니다.
- Step 6.** [Logout] 버튼을 클릭하여 서버와 접속을 해제합니다.

Step 7. 프로그램을 종료합니다.

## API 호출 순서

이 샘플 프로그램은 Wisenet SSM Client SDK가 제공하는 라이브러리를 사용하여 라이브 영상을 .avi 파일로 저장하는 기능을 구현하였습니다. 샘플 프로그램의 API 호출 순서는 다음과 같습니다.

Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`

Login

2. `ssmSdkWrapper.Login()`

Response of Login

3. `ssmSdkWrapper.OnResponse`

Receive Event: LogIn

4. `ssmSdkWrapper.OnEvent`

Receive Event: ObjectConnected

5. `ssmSdkWrapper.OnEvent`

Start Live local record

6. `ssmSdkWrapper.StartLiveLocalRecording`

Stop Live local record

7. `ssmSdkWrapper.StopLiveLocalRecording`

Logout

8. `ssmSdkWrapper.Logout()`

Release

9. `ssmSdkWrapper.ReleaseEvent()`

# 구현 방법

이 절에서는 LiveLocalRecord 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다.

## Local Record 시작

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 StartLiveLocalRecording() 메서드를 통해 Local Record 시작 요청을 전송합니다.

```
// LiveLocalRecordSample.cs
private void btnRecordStart_Click(object sender, EventArgs e)
{
    String guidText = textBoxUnitID.Text;
    Guid guid = Guid.Empty;
    uint resCode = 0;
    UInt32 sequenceID = 0;

    if (Guid.TryParse(guidText, out guid))
    {
        LocalRecordModel localRecordModel = new LocalRecordModel();

        localRecordModel.CameraGuid = textBoxUnitID.Text;
        localRecordModel.ProfileType = 1; // HIGH
        localRecordModel.FilePath = textBoxFilePath.Text;
        localRecordModel.FileName = textBoxFileName.Text;

        String json = JsonConvert.SerializeObject(localRecordModel,
Formatting.Indented);

        resCode = ssmSdkWrapper.StartLiveLocalRecording(json, ref sequenceID);
        _logger.WLOGD("StartLiveLocalRecording()::Result=" + resCode + ",
SequenceID=" + sequenceID);
    }
    else
```

```

    {
        MessageBox.Show("Check the guid");
    }
}

```

## Local Record 중지

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 StopLiveLocalRecording() 메서드를 통해 Local Record 중지 요청을 전송합니다.

```

// LiveLocalRecordSample.cs
private void btnRecordStop_Click(object sender, EventArgs e)
{
    String guidText = textBoxUnitID.Text;
    Guid guid = Guid.Empty;
    uint resCode = 0;
    UInt32 sequenceID = 0;

    if (Guid.TryParse(guidText, out guid))
    {
        LocalRecordModel localRecordModel = new LocalRecordModel();

        localRecordModel.CameraGuid = textBoxUnitID.Text;
        localRecordModel.ProfileType = 1; // HIGH

        String json = JsonConvert.SerializeObject(localRecordModel,
Formatting.Indented);

        resCode = ssmSdkWrapper.StopLiveLocalRecording(json, ref sequenceID);
        _logger.WLOGD("StopLiveLocalRecording()::Result=" + resCode + ",
SequenceID=" + sequenceID);
    }
    else
    {
        MessageBox.Show("Check the guid");
    }
}

```

## See also

LogIn 샘플 프로그램

Backup 샘플 프로그램

Wisenet SSM Client SDK API Reference (v2.10.6)

## FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 15

# UserManagement 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 사용자 그룹, 사용자 정보를 관리하는 방법을 설명합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ

# 샘플 프로그램 소개

User management 샘플 프로그램은 사용자 그룹, 사용자를 추가, 수정, 삭제하는 예제입니다. 샘플 프로그램 사용법은 다음과 같습니다.

**Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.

서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

**Step 2.** User Group, User 관련 버튼을 클릭합니다.

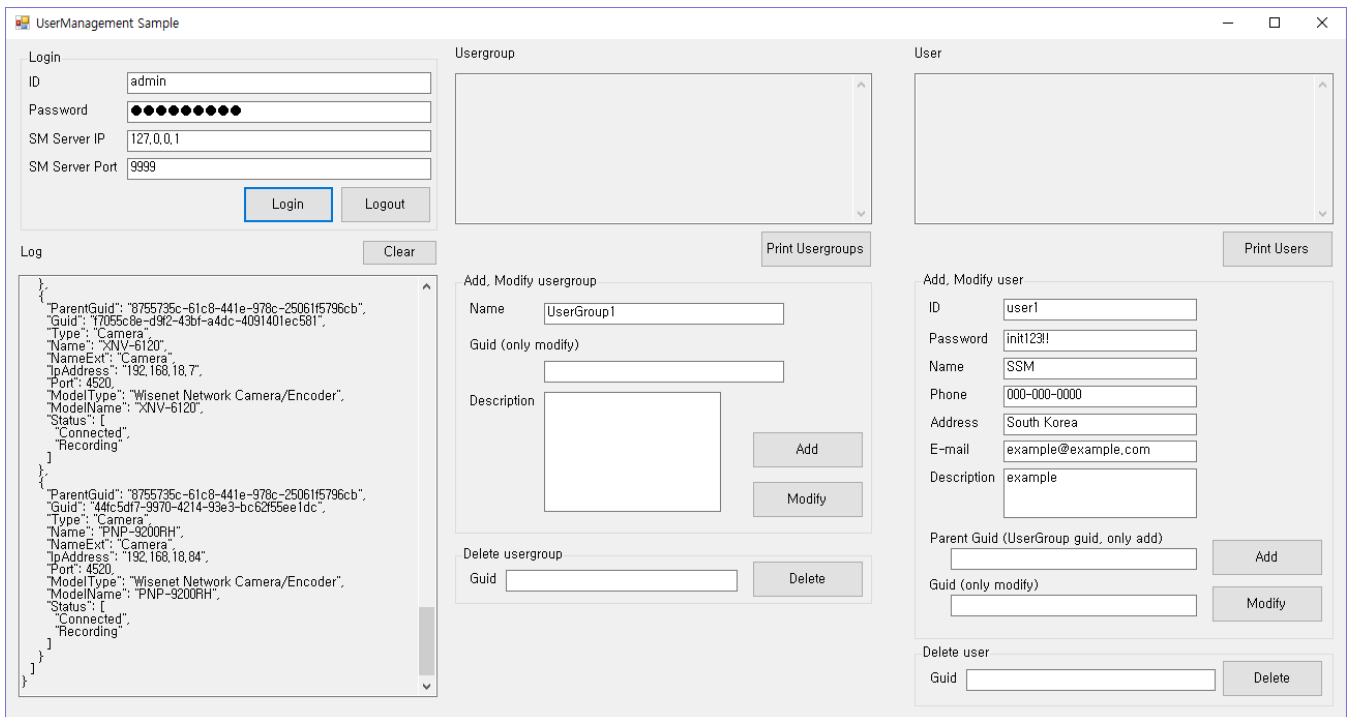


Figure 13 UserManagement 샘플 프로그램 실행 화면

**Step 3.** [Logout] 버튼을 클릭하여 서버와 접속을 해제합니다.

**Step 4.** 프로그램을 종료합니다.

## API 호출 순서



이 샘플 프로그램은 Wisenet SSM Client SDK가 제공하는 라이브러리를 사용자 그룹, 사용자를 관리하는 기능을 구현하였습니다. 샘플 프로그램의 API 호출 순서는 다음과 같습니다.

Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`

Login

2. `ssmSdkWrapper.Login()`

Response of Login

3. `ssmSdkWrapper.OnResponse`

Receive Event: LogIn

4. `ssmSdkWrapper.OnEvent`

Receive Event: ObjectConnected

5. `ssmSdkWrapper.OnEvent`

User group, User get, add, modify, delete

6. `ssmSdkWrapper.GetUserGroupInfo`  
`ssmSdkWrapper.AddUserGroup`  
`ssmSdkWrapper.ModifyUserGroup`  
`ssmSdkWrapper.DeleteUserGroup`  
`ssmSdkWrapper.GetUserInfo`  
`ssmSdkWrapper.AddUser`  
`ssmSdkWrapper.ModifyUser`  
`ssmSdkWrapper.DeleteUser`

Logout

7. `ssmSdkWrapper.Logout()`

Release

8. `ssmSdkWrapper.ReleaseEvent()`

## 구현 방법

이 절에서는 User Management 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다.

## User Group 정보 얻기

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 GetUserGroupInfo() 메서드를 통해 사용자 그룹 정보를 얻습니다.

```
// UserManagementSample.cs
private void PrintUserGroup_Click(object sender, EventArgs e)
{
    String str = ssmSdkWrapper.GetUserGroupInfo();
    this.textBoxUserGroupList.Text = str;
}
```

## User Group 추가

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 AddUserGroup() 메서드를 통해 사용자 그룹을 추가합니다.

```
// UserManagementSample.cs
private void AddUsergroupButton_Click(object sender, EventArgs e)
{
    UserGroupModel userGroupModel = new UserGroupModel();

    userGroupModel.Name = textBoxAddUsergroupName.Text;
    userGroupModel.Permission.MonitoringViewer = true;
    userGroupModel.Permission.Live = true;

    String json = JsonConvert.SerializeObject(userGroupModel, Formatting.Indented);
    UInt32 resCode = 0;
    UInt32 sequenceID = 0;

    resCode = ssmSdkWrapper.AddUserGroup(json, ref sequenceID);
    _logger.WLOGD("AddUserGroup::Result=" + resCode + ", SequenceID=" + sequenceID);
}
```

## User Group 수정

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 ModifyUserGroup() 메서드를 통해 사용자 그룹을 수정합니다.

```
// UserManagementSample.cs
private void ModifyUsergroupButton_Click(object sender, EventArgs e)
{
    UserGroupModel userGroupModel = new UserGroupModel();

    userGroupModel.Guid = textBoxModifyUsergroupGuid.Text;
    userGroupModel.Name = textBoxAddUsergroupName.Text;
    userGroupModel.Description = textBoxModifyUsergroupDescription.Text;

    userGroupModel.Permission.MonitoringViewer = true;
    userGroupModel.Permission.Live = true;

    String json = JsonConvert.SerializeObject(userGroupModel, Formatting.Indented);
    UInt32 resCode = 0;
    UInt32 sequenceID = 0;

    resCode = ssmSdkWrapper.ModifyUserGroup(json, ref sequenceID);
    _logger.WLOGD("ModifyUserGroup():Result=" + resCode + ", SequenceID=" +
sequenceID);
}
```

## User Group 삭제

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 DeleteUserGroup() 메서드를 통해 사용자 그룹을 삭제합니다.

```
// UserManagementSample.cs
private void DeleteUsergroupButton_Click(object sender, EventArgs e)
{
    UserGroupModel userGroupModel = new UserGroupModel();

    userGroupModel.Guid = textBoxDeleteUsergroupGuid.Text;
    String json = JsonConvert.SerializeObject(userGroupModel, Formatting.Indented);
    UInt32 resCode = 0;
    UInt32 sequenceID = 0;

    resCode = ssmSdkWrapper.DeleteUserGroup(json, ref sequenceID);
}
```

```

        _logger.WLOGD("DeleteUserGroup()::Result=" + resCode + ", SequenceID=" +
sequenceID);
    }

```

## User 정보 얻기

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 GetUserInfo() 메서드를 통해 사용자 정보를 얻습니다.

```

// UserManagementSample.cs
private void PrintUsers_Click(object sender, EventArgs e)
{
    String str = ssmSdkWrapper.GetUserInfo();

    this.textBoxUserList.Text = str;
}

```

## User 추가

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 AddUser() 메서드를 통해 사용자를 추가합니다.

```

// UserManagementSample.cs
private void AddUserButton_Click(object sender, EventArgs e)
{
    if (textBoxAddUserID.Text == String.Empty)
    {
        MessageBox.Show("ID is empty.");
        return;
    }

    Guid usergroupGuid = Guid.Empty;
    if (!Guid.TryParse(textBoxAddUserParentGuid.Text, out usergroupGuid))
    {
        MessageBox.Show("Check parent guid(usergroup guid).");
        return;
    }

    UserModel userModel = new UserModel();
}

```

```

        userModel.UserGroupGuid = usergroupGuid.ToString();
        userModel.ID = textBoxAddUserID.Text;
        userModel.UserName = textBoxAddUserName.Text;
        userModel.Password = textBoxAddUserPassword.Text;
        userModel.PhoneNumber = textBoxAddUserPhoneNumber.Text;
        userModel.Email = textBoxAddUserEmail.Text;
        userModel.Address = textBoxAddUserAddress.Text;
        userModel.Description = textBoxAddUserDescription.Text;

        String json = JsonConvert.SerializeObject(userModel, Formatting.Indented);
        UInt32 resCode = 0;
        UInt32 sequenceID = 0;

        resCode = ssmSdkWrapper.AddUser(json, ref sequenceID);
        _logger.WLOGD("AddUser()::Result=" + resCode + ", SequenceID=" + sequenceID);
    }

```

## User 수정

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 ModifyUser() 메서드를 통해 사용자를 수정합니다.

```

// UserManagementSample.cs
private void ModifyUserButton_Click(object sender, EventArgs e)
{
    if (textBoxAddUserID.Text == String.Empty)
    {
        MessageBox.Show("ID is empty.");
        return;
    }

    Guid userGuid = Guid.Empty;
    if (!Guid.TryParse(textBoxModifyUserGuid.Text, out userGuid))
    {
        MessageBox.Show("Check guid.");
        return;
    }

    UserModel userModel = new UserModel();

```

```

        userModel.Guid = userGuid.ToString();
        userModel.ID = textBoxAddUserID.Text;
        userModel.UserName = textBoxAddUserName.Text;
        userModel.Password = textBoxAddUserPassword.Text;
        userModel.PhoneNumber = textBoxAddUserPhoneNumber.Text;
        userModel.Email = textBoxAddUserEmail.Text;
        userModel.Address = textBoxAddUserAddress.Text;
        userModel.Description = textBoxAddUserDescription.Text;

        String json = JsonConvert.SerializeObject(userModel, Formatting.Indented);
        UInt32 resCode = 0;
        UInt32 sequenceID = 0;

        resCode = ssmSdkWrapper.ModifyUser(json, ref sequenceID);
        _logger.WLOGD("ModifyUser()::Result=" + resCode + ", SequenceID=" + sequenceID);
    }

```

## User 삭제

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 DeleteUser() 메서드를 통해 사용자를 삭제합니다.

```

// UserManagementSample.cs
private void DeleteUserButton_Click(object sender, EventArgs e)
{
    UserModel userModel = new UserModel();

    userModel.Guid = textBoxDeleteUserGuid.Text;
    String json = JsonConvert.SerializeObject(userModel, Formatting.Indented);
    UInt32 resCode = 0;
    UInt32 sequenceID = 0;

    resCode = ssmSdkWrapper.DeleteUser(json, ref sequenceID);
    _logger.WLOGD("DeleteUser()::Result=" + resCode + ", SequenceID=" + sequenceID);
}

```

## See also

LogIn 샘플 프로그램

Backup 샘플 프로그램

Wisenet SSM Client SDK API Reference (v2.10.6)

## FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 16

# NTP 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 NTP 설정을 불러오거나 변경하는 방법을 설명합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ



# 샘플 프로그램 소개

NTP 샘플 프로그램은 NTP 설정을 불러오고, 변경하는 예제입니다. 샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

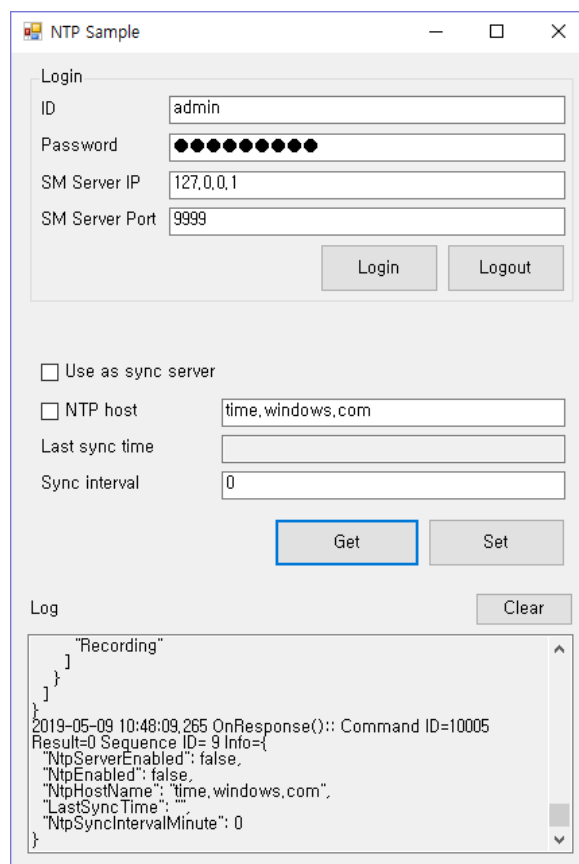


Figure 14 NTP 샘플 프로그램 실행 화면

- Step 3.** [Get] 버튼을 클릭하여 현재 NTP 설정을 가져옵니다.
- Step 4.** 정보 등을 변경하여 [Set] 버튼을 클릭합니다..
- Step 5.** [Logout] 버튼을 클릭하여 서버와 접속을 해제합니다.
- Step 6.** 프로그램을 종료합니다.

# API 호출 순서

이 샘플 프로그램은 Wisenet SSM Client SDK가 제공하는 라이브러리를 사용하여 NTP 설정을 불러오거나 변경하는 기능을 구현하였습니다. 샘플 프로그램의 API 호출 순서는 다음과 같습니다.

## Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`

## Login

2. `ssmSdkWrapper.Login()`

## Response of Login

3. `ssmSdkWrapper.OnResponse`

## Receive Event: Login

4. `ssmSdkWrapper.OnEvent`

## Receive Event: ObjectConnected

5. `ssmSdkWrapper.OnEvent`

## Get Ntp

6. `ssmSdkWrapper.GetNtp`

## Set Ntp

7. `ssmSdkWrapper.SetNtp`

## Logout

8. `ssmSdkWrapper.Logout()`

## Release

9. `ssmSdkWrapper.ReleaseEvent()`

# 구현 방법

이 절에서는 NTP 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다.

## NTP 설정 불러오기

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 GetNtp() 메서드를 통해 NTP 설정 불러오기 요청을 전송합니다.

```
// NtpSample.cs
private void NtpGet_Click(object sender, EventArgs e)
{
    UInt32 sequenceID = 0;
    UInt32 resCode = 0;

    resCode = ssmSdkWrapper.GetNtp(ref sequenceID);

    _logger.WLOGD("GetNtp()::Result=" + resCode + ", SequenceID=" + sequenceID);
}

private void OnResponse(UInt32 commandID, UInt32 errorCode, UInt32 sequenceID, string info)
{
    _logger.WLOGD(
        "OnResponse():" +
        " Command ID=" + commandID +
        " Result=" + errorCode +
        " Sequence ID= " + sequenceID +
        " Info=" + info);

    if (commandID == (uint)CommandID.GetNtp)
    {
        UpdateNtp(info);
    }
}

public delegate void DeleUpdateNtp(String json);
private void UpdateNtp(String json)
{
    NtpInfoModel ntpInfoModel = JsonConvert.DeserializeObject<NtpInfoModel>(json);

    if (this.InvokeRequired)
    {

```

```
        DeleUpdateNtp deUpdateNtp = new DeleUpdateNtp(this.UpdateNtp);
        this.Invoke(deUpdateNtp, new object[] { json });
        return;
    }
    else
    {
        this.ntpServerEnable.Checked = ntpInfoModel.NtpServerEnabled;
        this.ntpHost.Checked = ntpInfoModel.NtpEnabled;
        this.textBoxNtpHost.Text = ntpInfoModel.NtpHostName;
        this.textBoxLastSyncTime.Text = ntpInfoModel.LastSyncTime;
        this.textBoxSyncInterval.Text = ntpInfoModel.NtpSyncIntervalMinute.ToString();
    }
}
```

## NTP 설정 변경하기

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 SetNtp() 메서드를 통해 NTP 설정 변경 요청을 전송합니다.

```
// NtpSample.cs
private void NtpSet_Click(object sender, EventArgs e)
{
    String json = String.Empty;
    UInt32 sequenceID = 0;
    UInt32 resCode = 0;

    NtpInfoModel ntpInfoModel = new NtpInfoModel();

    ntpInfoModel.NtpServerEnabled = this.ntpServerEnable.Checked;
    ntpInfoModel.NtpEnabled = this.ntpHost.Checked;
    ntpInfoModel.NtpHostName = this.textBoxNtpHost.Text;

    Int32 interval = 0;
    Int32.TryParse(textBoxSyncInterval.Text, out interval);
    ntpInfoModel.NtpSyncIntervalMinute = interval;

    json = JsonConvert.SerializeObject(ntpInfoModel, Formatting.Indented);

    resCode = ssmSdkWrapper.SetNtp(json, ref sequenceID);
}
```

```
_logger.WLOGD("SetNtp()::Result=" + resCode + ", SequenceID=" + sequenceID);  
}
```

## See also

[LogIn 샘플 프로그램](#)

[Backup 샘플 프로그램](#)

[Wisenet SSM Client SDK API Reference \(v2.10.6\)](#)

## FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.



## CHAPTER 17

# Device Record 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Client SDK를 사용하여 카메라 수동 녹화를 시작/중지하는 방법을 설명합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ

# 샘플 프로그램 소개

DeviceRecord 샘플 프로그램은 카메라를 수동으로 녹화 시작/중지하는 예제입니다. 샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 서버 IP address, 서버 포트, ID/Password 등의 서버 연결 정보를 설정한 후 [Login] 버튼을 누릅니다.

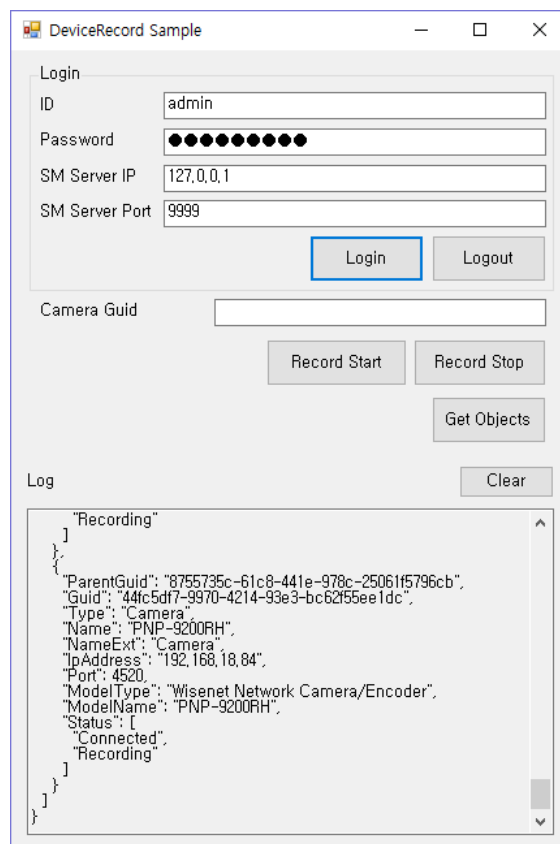


Figure 15 DeviceRecord 샘플 프로그램 실행 화면

- Step 3.** 카메라의 Guid를 입력합니다.
- Step 4.** [Record Start] 버튼을 클릭하여 녹화를 시작합니다.
- Step 5.** Get Objects 버튼을 클릭하면 Log창에 장치 상태가 표시됩니다.
- Step 6.** [Record Stop] 버튼을 클릭하여 녹화를 중지합니다.
- Step 7.** 프로그램을 종료합니다.



# API 호출 순서

이 샘플 프로그램은 Wisenet SSM Client SDK가 제공하는 라이브러리를 사용하여 카메라 수동 녹화 시작/중지기능을 구현하였습니다. 샘플 프로그램의 API 호출 순서는 다음과 같습니다.

## Preprocessing

1. `ssmSdkWrapper.InitializeEvent()`

## Login

2. `ssmSdkWrapper.Login()`

## Response of Login

3. `ssmSdkWrapper.OnResponse`

## Receive Event: LogIn

4. `ssmSdkWrapper.OnEvent`

## Receive Event: ObjectConnected

5. `ssmSdkWrapper.OnEvent`

## Start Recording

6. `ssmSdkWrapper.StartDeviceRecording`

## Stop Recording

7. `ssmSdkWrapper.StopDeviceRecording`

## Logout

8. `ssmSdkWrapper.Logout()`

## Release

9. `ssmSdkWrapper.ReleaseEvent()`

# 구현 방법

이 절에서는 DeviceRecord 샘플 프로그램 구현 방법을 상세히 설명합니다.

SsmSdkWrapper 참조 추가, Callback 함수 추가, 초기화, 로그인, 로그아웃, SsmSdkWrapper 해

제는 CHAPTER 4에 기술된 '구현 방법'을 참고 바랍니다.

---

## 장치 수동 녹화 시작하기

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 StartDeviceRecording() 메서드를 통해 녹화 시작 요청을 전송합니다.

```
// DeviceRecordSample.cs
private void btnRecordStart_Click(object sender, EventArgs e)
{
    String guidText = textBoxUnitID.Text;
    Guid guid = Guid.Empty;
    UInt32 sequenceID = 0;

    if (Guid.TryParse(guidText, out guid))
    {
        uint resCode = ssmSdkWrapper.StartDeviceRecording(textBoxUnitID.Text, ref
sequenceID);
        _logger.WLOGD("StartDeviceRecording()::Result=" + resCode + ",
SequenceID=" + sequenceID);
    }
    else
    {
        MessageBox.Show("Check the guid.");
    }
}
```

---

## 장치 수동 녹화 중지하기

버튼 클릭 이벤트 핸들러를 생성하고, SsmSdkWrapper의 StopDeviceRecording() 메서드를 통해 녹화 중지 요청을 전송합니다.

```
// DeviceRecordSample.cs
private void btnRecordStop_Click(object sender, EventArgs e)
{
    String guidText = textBoxUnitID.Text;
    Guid guid = Guid.Empty;
    UInt32 sequenceID = 0;
```

```
        if (Guid.TryParse(guidText, out guid))
        {
            uint resCode = ssmSdkWrapper.StopDeviceRecording(textBoxUnitID.Text, ref
sequenceID);
            _logger.WLOGD("StopDeviceRecording()::Result=" + resCode + ",
SequenceID=" + sequenceID);
        }
        else
        {
            MessageBox.Show("Check the guid.");
        }
    }
}
```

## See also

[LogIn 샘플 프로그램](#)

[Backup 샘플 프로그램](#)

[Wisenet SSM Client SDK API Reference \(v2.10.6\)](#)

## FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.

현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 18

# Wisenet DDNS Login 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Core Server에 Wisenet DDNS ID로 로그인하는 방법을 설명합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ

# 샘플 프로그램 소개

Wisenet DDNS Login 샘플 프로그램은 Wisenet DDNS ID로 Wisenet SSM Core Server에 로그인 하는 방법을 설명하기 위한 예제입니다.

샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 로그인 버튼을 클릭하여 System Manager 서버로 접속 되는지 확인합니다.
- Step 3.** 프로그램을 종료합니다.

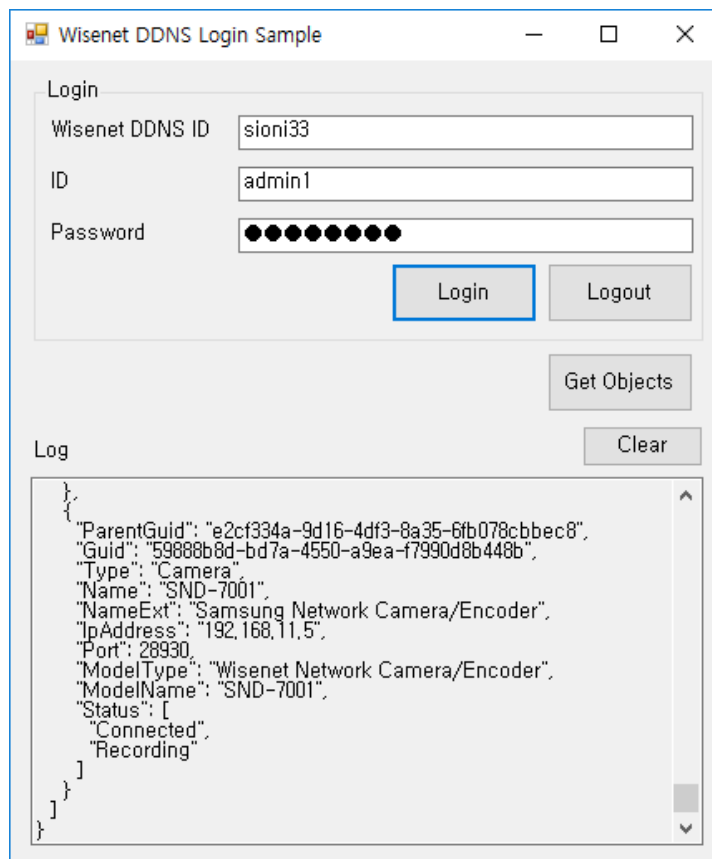


Figure 16 Wisenet DDNS Login 샘플 프로그램 실행 화면

# API 호출 순서

API 호출 순서는 다음과 같습니다.

## Preprocessing

1. `ssmSdkWrapper.InitializeEventEvent()`

## Login

2. `ssmSdkWrapper.DdnsLogin()`

## Response of Login

3. `ssmSdkWrapper.OnResponse`

## Receive Event: LogIn

4. `ssmSdkWrapper.OnEvent`

## Receive Event: ObjectConnected

5. `ssmSdkWrapper.OnEvent`

## Logout

6. `ssmSdkWrapper.Logout()`

## Release

7. `ssmSdkWrapper.ReleaseEvent()`

# 구현 방법

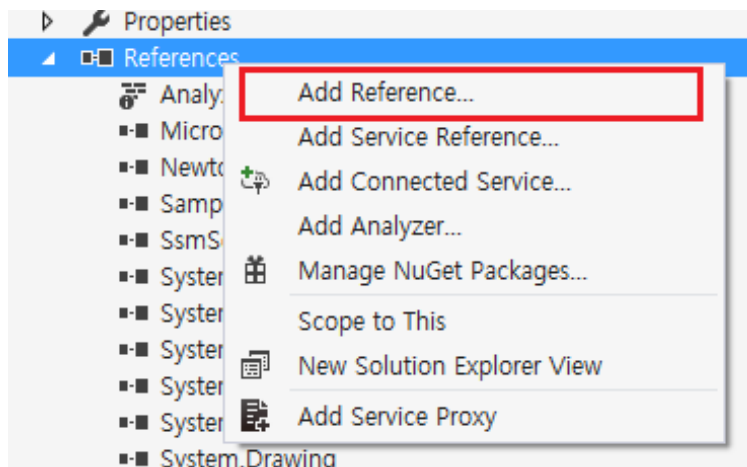
이 절에서는 Wisenet DDNS Login 샘플 프로그램 구현 방법을 상세히 설명합니다.

## SsmSdkWrapper 참조 추가

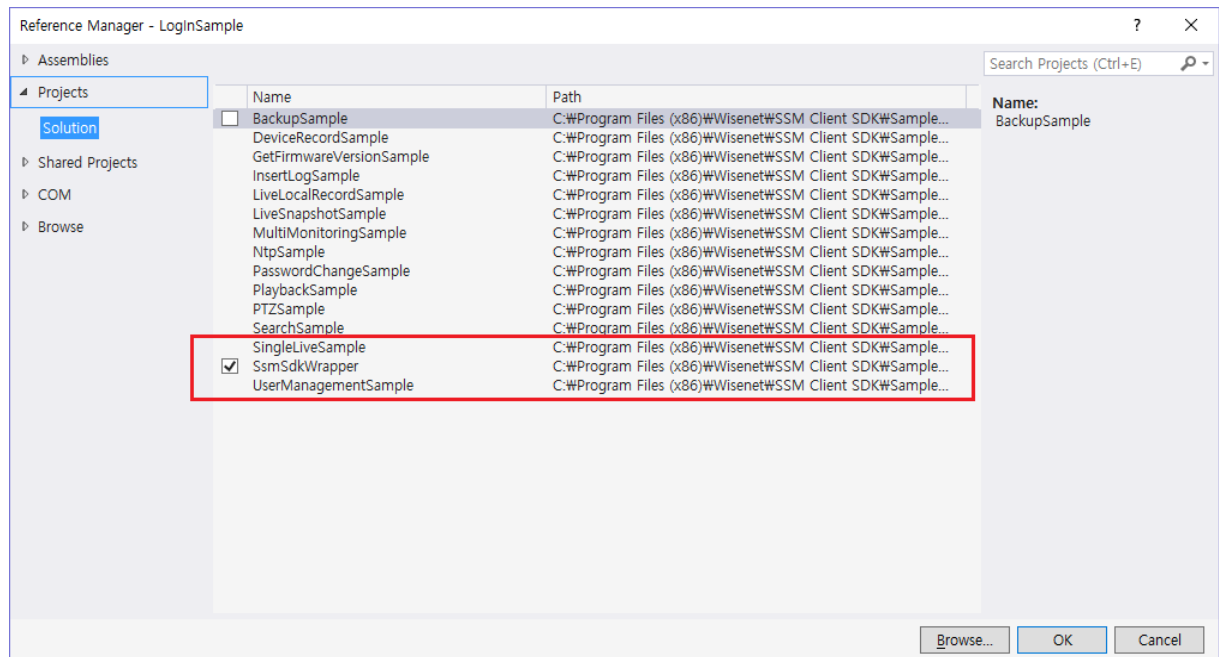
### 절차

Visual Studio 2013에서 Windows Forms를 생성한 후, SsmSdkWrapper를 참조 추가합니다.

- Step 1.** Windows Forms를 생성합니다.
- Step 2.** Visual Studio 프로젝트 References에서 오른쪽 마우스 버튼을 눌러 Add Reference를 선택합니다.



- Step 3.** Reference Manager에서 SsmSdkWrapper 프로젝트를 추가합니다.



## SsmSdkWrapper Callback 함수 추가

명령에 대한 응답, 이벤트를 받기 위해서는 Callback 함수를 구현해야 합니다.

### Callback 함수 구현: OnResponse

DdnsLoginSample 클래스의 OnResponse를 구현합니다.

```
// DdnsLoginSample.cs
private void OnResponse(UInt32 commandID, UInt32 errorCode, UInt32 sequenceID, string info)
{
    _logger.WLOGD(
        "OnResponse():" +
        " Command ID=" + commandID +
        " Result=" + errorCode +
        " Sequence ID= " + sequenceID +
        " Info=" + info);
}
```



## Callback 함수 구현: OnEvent

DdnsLoginSample 클래스의 OnEvent를 구현합니다.

```
// DdnsLoginSample.cs
private void OnEvent(UInt32 eventID, String info)
{
    _logger.WLOGD(
        "OnEvent::" +
        " Event ID=" + eventID +
        " Info=" + info);
}
```

## SsmSdkWrapper 초기화

SsmSdkWrapper를 생성하고 ssmSdkWrapper.InitializeEvent()를 호출하여 초기화합니다. DdnsLoginSample 클래스의 멤버 함수 DdnsLoginSample\_Load() 안에서 InitializeEvent() 메서드를 호출합니다. InitializeEvent()를 통해 SsmSdkWrapper의 서비스 사용을 시작합니다.

```
// DdnsLoginSample.cs
private SsmSdkWrapper ssmSdkWrapper = null;

private void DdnsLoginSample_Load(object sender, EventArgs e)
{
    ssmSdkWrapper = new SsmSdkWrapper(this.OnResponse, this.OnEvent);
    ssmSdkWrapper.InitializeEvent();
}
```

## 로그인

로그인 버튼 Event Handler를 생성하여 로그인을 수행합니다. 로그인을 위해서는 Wisenet DDNS ID, SSM ID와 Password가 필요합니다.

```
// DdnsLoginSample.cs
private void btnLogin_Click(object sender, EventArgs e)
{
    uint resCode = ssmSdkWrapper.DdnsLogin(
        txtID.Text,
```

```

        txtPassword.Text,
        txtDdnsId.Text);
    _logger.WLOGD("Login()::Result=" + resCode);
}

```

## 로그아웃

로그아웃 버튼 Event Handler를 생성하여 로그아웃을 수행합니다.

```

// DdnsLoginSample.cs
private void btnLogout_Click(object sender, EventArgs e)
{
    uint resCode = ssmSdkWrapper.Logout();
    _logger.WLOGD("Logout()::Result=" + resCode);
}

```

## SsmSdkWrapper 해제

SsmSdkWrapper의 사용이 끝난 후에는 서비스를 중단하고 자원을 해제해야 합니다.

DdnsLoginSample 클래스의 멤버 함수 DdnsLoginSample\_Closed() 안에서 ReleaseEvent() 메서드를 호출합니다.

```

// DdnsLoginSample.cs
private void DdnsLoginSample_Closed(object sender, EventArgs e)
{
    ssmSdkWrapper.ReleaseEvent();
}

```

## See also

Wisenet SSM Client SDK API Reference (v2.10.6)

# FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.

## CHAPTER 19

# URL Login 샘플 프로그램

---

이 챕터에서는 Wisenet SSM Core Server의 URL로 로그인하는 방법을 설명합니다.

## Contents

- 샘플 프로그램 소개
- API 호출 순서
- 구현 방법
- See Also
- FAQ

# 샘플 프로그램 소개

URL Login 샘플 프로그램은 Wisenet SSM Core Server의 외부 URL 로그인하는 방법을 설명하기 위한 예제입니다.

샘플 프로그램 사용법은 다음과 같습니다.

- Step 1.** 샘플 프로그램 빌드 후 정상적으로 실행되는지 확인합니다.
- Step 2.** 로그인 버튼을 클릭하여 System Manager 서버로 접속 되는지 확인합니다.
- Step 3.** 프로그램을 종료합니다.

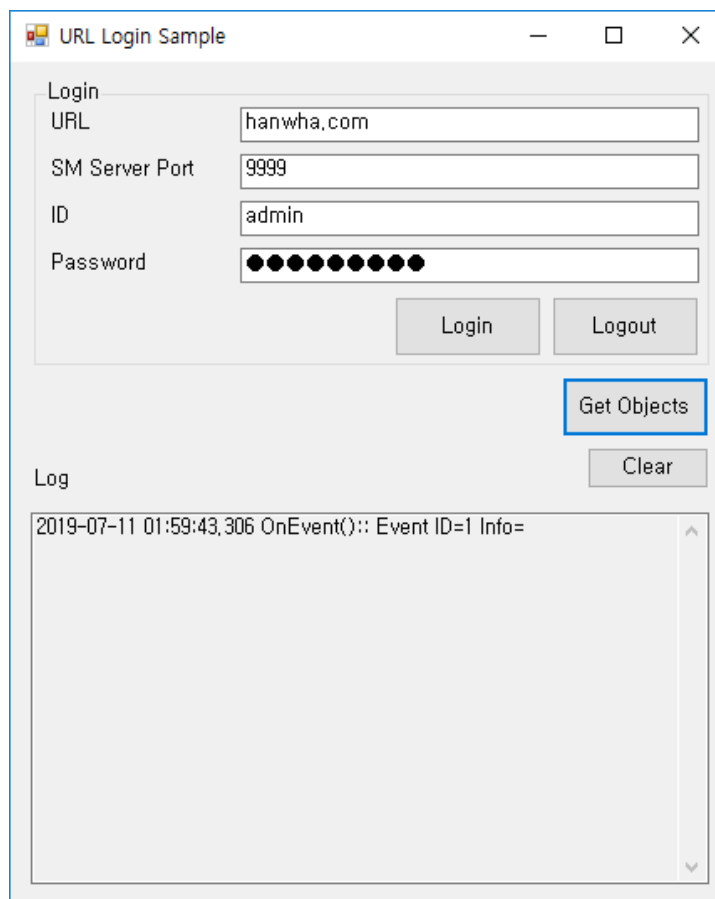


Figure 17 URL Login 샘플 프로그램 실행 화면

# API 호출 순서

API 호출 순서는 다음과 같습니다.

## Preprocessing

1. `ssmSdkWrapper.InitializeEventEvent()`

## Login

2. `ssmSdkWrapper.UrlLogin()`

## Response of Login

3. `ssmSdkWrapper.OnResponse`

## Receive Event: Login

4. `ssmSdkWrapper.OnEvent`

## Receive Event: ObjectConnected

5. `ssmSdkWrapper.OnEvent`

## Logout

6. `ssmSdkWrapper.Logout()`

## Release

7. `ssmSdkWrapper.ReleaseEvent()`

# 구현 방법

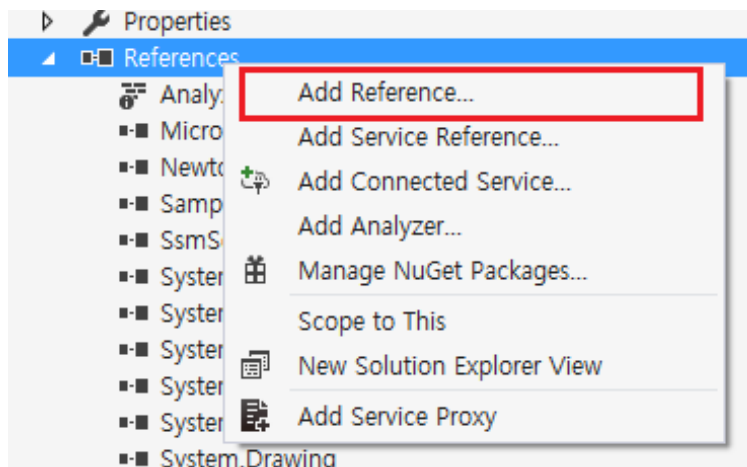
이 절에서는 URL Login 샘플 프로그램 구현 방법을 상세히 설명합니다.

## SsmSdkWrapper 참조 추가

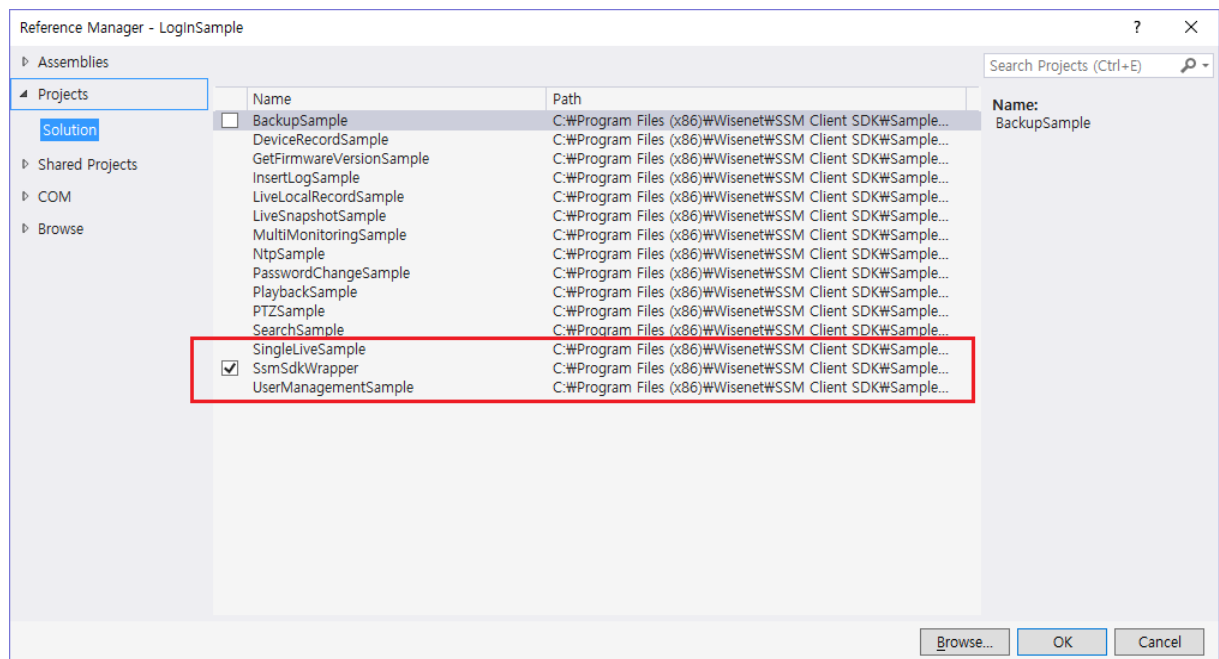
### 절차

Visual Studio 2013에서 Windows Forms를 생성한 후, SsmSdkWrapper를 참조 추가합니다.

- Step 1.** Windows Forms를 생성합니다.
- Step 2.** Visual Studio 프로젝트 References에서 오른쪽 마우스 버튼을 눌러 Add Reference를 선택합니다.



- Step 3.** Reference Manager에서 SsmSdkWrapper 프로젝트를 추가합니다.



## SsmSdkWrapper Callback 함수 추가

명령에 대한 응답, 이벤트를 받기 위해서는 Callback 함수를 구현해야 합니다.

### Callback 함수 구현: OnResponse

UrlLoginSample 클래스의 OnResponse를 구현합니다.

```
// UrlLoginSample.cs
private void OnResponse(UInt32 commandID, UInt32 errorCode, UInt32 sequenceID, string info)
{
    _logger.WLOGD(
        "OnResponse():" +
        " Command ID=" + commandID +
        " Result=" + errorCode +
        " Sequence ID= " + sequenceID +
        " Info=" + info);
}
```



## Callback 함수 구현: OnEvent

UrlLoginSample 클래스의 OnEvent를 구현합니다.

```
// UrlLoginSample.cs
private void OnEvent(UInt32 eventID, String info)
{
    _logger.WLOGD(
        "OnEvent():" +
        " Event ID=" + eventID +
        " Info=" + info);
}
```

---

## SsmSdkWrapper 초기화

SsmSdkWrapper를 생성하고 ssmSdkWrapper.InitializeEvent()를 호출하여 초기화합니다.  
UrlLoginSample 클래스의 멤버 함수 UrlLoginSample\_Load() 안에서 InitializeEvent() 메서드를 호출합니다. InitializeEvent()를 통해 SsmSdkWrapper의 서비스 사용을 시작합니다.

```
// UrlLoginSample.cs
private SsmSdkWrapper ssmSdkWrapper = null;

private void UrlLoginSample_Load(object sender, EventArgs e)
{
    ssmSdkWrapper = new SsmSdkWrapper(this.OnResponse, this.OnEvent);
    ssmSdkWrapper.InitializeEvent();
}
```

---

## 로그인

로그인 버튼 Event Handler를 생성하여 로그인을 수행합니다. 로그인을 위해서는 URL, 서버 port, SSM ID와 Password가 필요합니다.

```
// UrlLoginSample.cs
private void btnLogin_Click(object sender, EventArgs e)
{
    uint resCode = ssmSdkWrapper.UrlLogin(
        txtID.Text,
```

```
txtPassword.Text,  
txtUrl.Text,  
Convert.ToUInt32(txtPort.Text));  
_logger.WLOGD("Login()::Result=" + resCode);  
}
```

---

## 로그아웃

로그아웃 버튼 Event Handler를 생성하여 로그아웃을 수행합니다.

```
// UrlLoginSample.cs  
private void btnLogout_Click(object sender, EventArgs e)  
{  
    uint resCode = ssmSdkWrapper.Logout();  
    _logger.WLOGD("Logout()::Result=" + resCode);  
}
```

---

## SsmSdkWrapper 해제

SsmSdkWrapper의 사용이 끝난 후에는 서비스를 중단하고 자원을 해제해야 합니다.

UrlLoginSample 클래스의 멤버 함수 UrlLoginSample\_Closed() 안에서 ReleaseEvent() 메서드를 호출합니다.

```
// UrlLoginSample.cs  
private void UrlLoginSample_Closed(object sender, EventArgs e)  
{  
    ssmSdkWrapper.ReleaseEvent();  
}
```

## See also

Wisenet SSM Client SDK API Reference (v2.10.6)

# FAQ

이 절은 샘플 프로그램에 대한 문의 중 자주 들어오는 문의와 답변을 정리하는 항목입니다.  
현재까지는 접수된 문의가 없으므로 비워둡니다.

# | 약어

## D

---

### DEP

Data Execution Prevention

### DLL

Dynamic Linking Library

### DVR

Digital Video Recorder

### NVRz

Network Video Recorder

## P

---

### PTZ

Pan/Tilt/Zoom

## S

---

### SDK

Software Development Kit

### SSM

Smart Security Manager