

# VISSIM COM INTERFACE

시뮬레이션 확장 및 응용

# 목차

- ▶ COM Interface 이해 및 개발환경 구성
- ▶ Python 기초
- ▶ VISSIM Indexing & Python 시뮬레이션 연동
- ▶ 시뮬레이션 실행 제어 및 데이터 전처리 보정
- ▶ 데이터베이스 연동 및 결과 처리

# 1. COM Interface 이해 및 개발환경 구성



# 1. COM Interface 이해 및 개발환경 구성

## (1) COM 인터페이스란?

COM(Component Object Model)은 서로 다른 프로그램이 바이너리 수준에서 객체를 공유하고 협업할 수 있도록 정의한 Microsoft의 기술입니다. COM을 사용하면 한 프로그램에서 다른 프로그램의 데이터와 함수를 호출하여 자동화 및 통합 작업을 수행할 수 있습니다.

## (2) Vissim에서의 COM 사용 목적

Vissim은 COM 인터페이스를 지원하는 소프트웨어로, 외부 프로그램이 Vissim 내부의 객체나 데이터를 제어하거나 읽어올 수 있도록 설계되어 있습니다. 이러한 COM 기능을 통해 시뮬레이션 실행, 네트워크 설정, 결과 추출 등의 작업을 자동화할 수 있습니다.

# 1. COM Interface 이해 및 개발환경 구성

## (3) 설치와 구성

Vissim의 COM 인터페이스는 Vissim 소프트웨어 설치 시 자동으로 포함됩니다. 설치 중 COM 서버 기능을 활성화하면 외부에서 Vissim을 자동화 서버로 사용할 수 있습니다. 단, 데모 버전에서는 이 기능이 제공되지 않기 때문에, 정식 버전을 사용하는 경우에만 COM 자동화 기능을 활용할 수 있습니다.

## (4) 언어 독립적인 COM의 특징

COM은 특정 프로그래밍 언어에 종속되지 않기 때문에 다양한 언어에서 사용할 수 있습니다. 대표적으로 COM 객체는 다음과 같은 언어에서 활용이 가능합니다.

VBA (Visual Basic for Applications)	VBScript	Python	C / C++
C#	Delphi	MATLAB	etc

실제 예제에서는 VBA를 활용한 경우가 가장 많으며, Python을 사용하는 예외적인 경우는 별도로 명시하여 제공합니다.

# 1. COM Interface 이해 및 개발환경 구성

## (5) 활용 시나리오

Vissim의 COM 인터페이스는 다음과 같은 작업을 자동화하는 데에 활용됩니다:

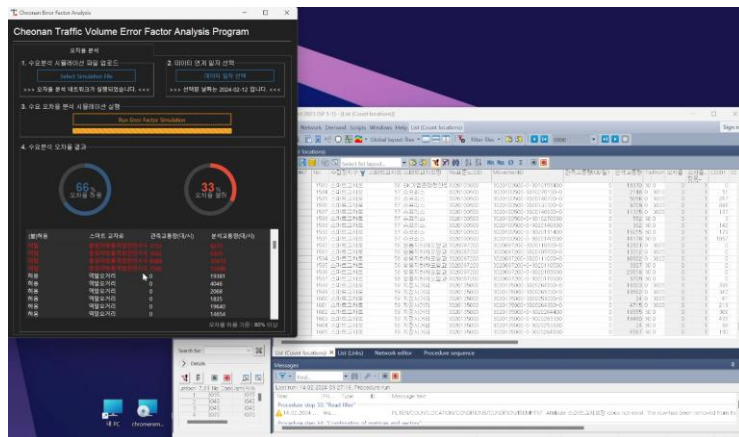
- 시뮬레이션 시나리오 실행 자동화
- 네트워크 객체 속성 값 읽기 및 설정
- 교통 신호 제어 및 타이밍 수정
- 시뮬레이션 결과 추출 및 외부 저장
- 반복 시나리오 테스트 및 배치 처리

## (6) Document

C:\WProgram Files\WPTV Vision\WPTV Vissim 2023\WDoc\WEng\WVissim 2023 - COM Intro.pdf

# 1. COM Interface 이해 및 개발환경 구성

## (7) 적용 사례



## 2. Python 기초





## 2. Python 기초

### (1) Python을 이용한 Vissim 자동화

Vissim은 사용자가 직접 시뮬레이션을 제어할 수 있도록 강력한 자동화 기능을 제공합니다. Python은 Vissim에서 제공하는 두 번째 주요 프로그래밍 언어로, 사용자들은 Vissim 스크립트 메뉴에서 Python 스크립트를 실행하거나 Python 인터프리터 또는 Python 스크립트 파일을 통해 수동으로 실행할 수 있습니다. 이러한 스크립트를 통해 Vissim 인스턴스를 생성하고 COM 인터페이스를 통해 다양한 제어가 가능합니다.

Excel처럼 VBA를 사용하는 프로그램과 달리, Python은 Vissim과 직접 연결하여 사용해야 합니다. PTV Vision Python 설치 패키지는 <https://cgi.ptvgroup.com/visionSetups/en/>에서 제공됩니다.

### (2) Python 설치 위치 권장

Python은 C:\WProgram Files\WPython 디렉터리보다는 C:\WPython 경로에 설치하는 것이 권장됩니다. 이는 인터프리터 및 라이브러리 호환성과 여러 개발 환경 간 충돌을 방지하기 위한 조치입니다. Python 스크립트는 다양한 텍스트 편집기에서 작성할 수 있으며, Python 언어에 대한 자세한 정보는 공식 웹사이트(<https://www.python.org/>)를 참고하면 됩니다.

## 2. Python 기초

### (3) Vissim 버전과 Python 호환성

- PTV Vissim 2020부터 Python 3.7을 공식 지원합니다.
- 현재는 Python 3.9까지 호환 가능합니다.
- Anaconda와 같은 Python 배포판은 지원하지 않습니다.
- wxPython 기반 GUI 확장 기능(Wrapper wxPython)은 지원되지 않습니다.

### (4) Python의 장점과 라이브러리

Python은 수치 계산, 데이터 처리 등 다양한 분야에서 활용되는 강력한 라이브러리를 제공합니다. 문법이 간단하면서도 유연하기 때문에 복잡한 작업도 간결하게 표현할 수 있습니다. 특히 다양한 외부 패키지와의 호환성이 뛰어나 많은 자동화 및 분석 작업에 적합합니다.

## 2. Python 기초

### (5) Vissim COM에서의 Python 속성 접근 방식

Python에서 COM 객체의 속성에 접근할 때는 두 가지 방식이 존재합니다

1. 읽기 전용 속성 (Get): 속성을 직접 불러올 수 있습니다.
2. 쓰기 가능한 속성 (Set): .SetAttValue() 메서드를 통해 값을 설정합니다.

```
number = node.AttValue("NO")  
node.SetAttValue("NO", number)
```

## 2. Python 기초

### (6) Python 변수 선언과 자료형

```
x = 10          # 정수형 (int)
y = 3.14        # 실수형 (float)
name = "Tom"    # 문자열형 (str)
is_ok = True    # 불리언형 (bool)
```

- "=" 기호로 값을 변수에 할당합니다.
- 타입 선언은 필요 없습니다. (자동 추론)
- `type(x)`로 자료형을 확인할 수 있음
- 문자열은 '문자열' 또는 "문자열" 로 작성 가능

## 2. Python 기초

### (7) Python 출력과 입력

```
print("안녕하세요!")    # 화면에 출력

name = input("이름을 입력하세요: ")    # 사용자 입력 받기
print("환영합니다,", name)
```

- Input() 함수는 항상 문자열(str)로 입력되므로 숫자로 쓰려면 형변환이 필요.
- ex) age = int(input("나이: "))

## 2. Python 기초

### (8) Python 조건문

```
age = 20

if age < 10:
    print("어린이입니다.")
elif age < 18:
    print("청소년입니다.")
else:
    print("성인입니다.")
```

연산자	의미	예시
==	같다	x == 10
!=	다르다	x != 10
<	미만	x < 10
>	초과	x > 3
<=	이하	x <= 7
>=	이상	x >= 2

- ":" 기호와 들여쓰기(4칸)가 필수이며, 중괄호 {}는 사용하지 않음

## 2. Python 기초

### (10) Python 자료형

- 리스트 (List) : mutable

```
colors = ["red", "green", "blue"]  
colors[0] = "yellow" # 수정 가능
```

- 튜플 (Tuple) : immutable

```
point = (3, 5)  
# point[0] = 10 → 오류 발생
```

- 딕셔너리 (Dictionary) : 키(key) - 값(Value) 쌍으로 구성

```
person = {"name": "Tom", "age": 25}  
print(person["name"]) # Tom
```

## 2. Python 기초

### (9) Python 반복문

```
for i in range(5): # 0부터 4까지 (5는 포함되지 않음)
    print(i)
```

```
for i in range(5): # 0부터 4까지 (5는 포함되지 않음)
    print(i)
```

```
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(fruit)
```



## 2. Python 기초

### (11) Python 함수

```
def greet(name):  
    return f"안녕하세요, {name}님"
```

- "def" 로 함수 정의 시작
- "name"은 파라미터(매개변수) : 함수에 전달하는 입력값
- "return"은 함수의 출력값을 반환함

```
message = greet("지민")    # → message에 문자열이 저장됨  
print(message)            # 안녕하세요, 지민님
```

- 함수는 필요할 때마다 호출 가능
- 여러 개의 파라미터 사용 가능. ex) def add(x, y): return x + y

## 2. Python 기초

### (12) Python 논리 연산자

연산자	의미	예시
and	둘 다 참이면 참	<code>x &gt; 0 and x &lt; 10</code>
or	하나라도 참이면 참	<code>x &lt; 0 or x &gt; 100</code>
not	부정	<code>not is_ok</code>

- 조건을 결합할 때 사용

### 3. VISSIM Indexing & Python 시뮬레이션 연동



### 3. VISSIM Indexing & Python 시뮬레이션 연동

#### (1) COM Object & COM Command Test 실습

- COM 인터페이스 이름 규칙
  - Vissim의 COM 인터페이스에서 사용되는 객체 식별자(Identifier)는 Vissim 클래스 이름앞에 'I'를 붙인 형태로 구성됩니다. 여기서 'I'는 Interface를 의미합니다.  
예) 클래스 이름이 'Link'라면, COM 인터페이스 상의 식별자는 'Ilink'입니다.
  - 이러한 접두사 'I'는 타입 선언을 할 때는 필요하지만, Vissim 객체를 통해 직접 접근할 때는 생략됩니다.
  - Vissim을 통해 객체에 접근할 때

```
Vissim.Net.Links
```

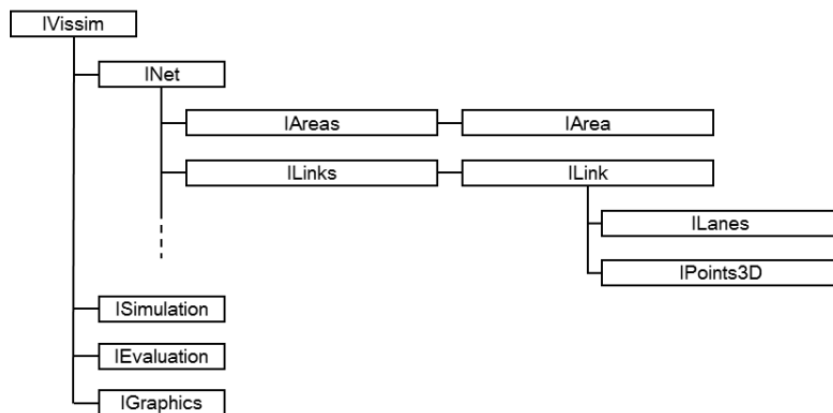
- 객체를 변수로 선언할 때

```
Dim linkObj As ILink
```

### 3. VISSIM Indexing & Python 시뮬레이션 연동

#### (2) Vissim 객체 모델 (The Vissim Object Model)

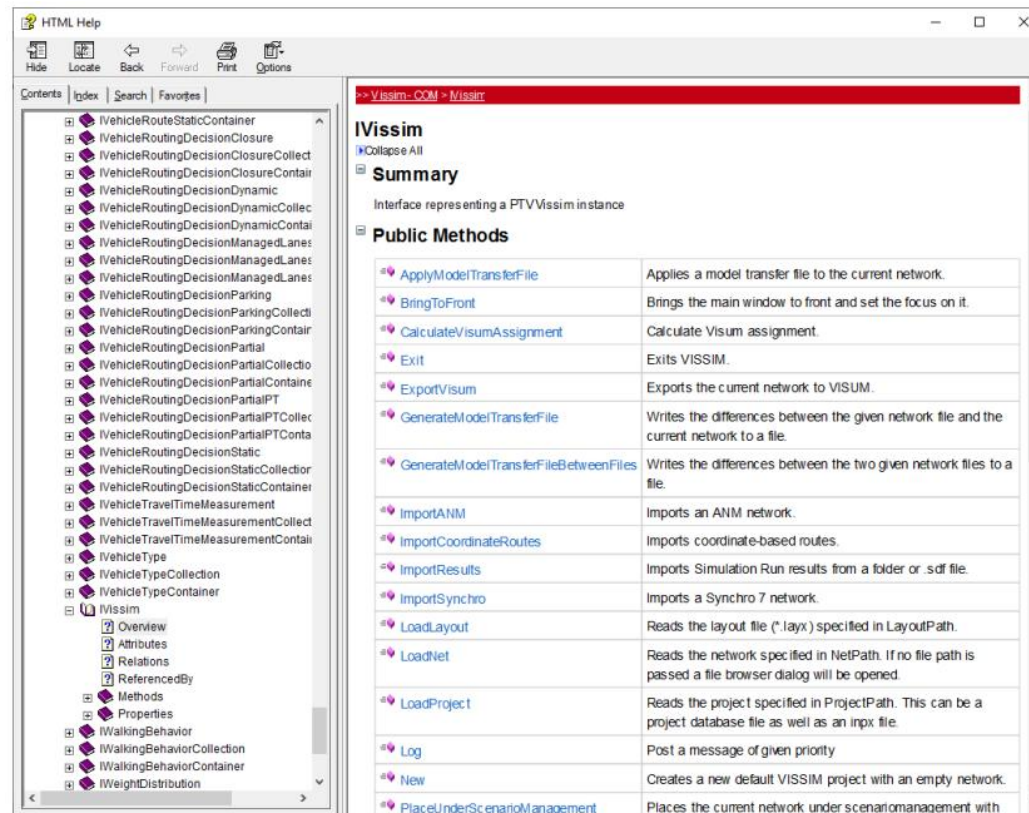
- 객체 계층 구조
  - Vissim의 COM 모델은 엄격한 계층 구조(object hierarchy)를 따릅니다. 최상위 객체는 'Ivissim'이며, 그 아래에 다양한 하위 객체들이 계층적으로 연결되어 있습니다. 예를 들어, 네트워크 상의 링크(link)에 접근하고자 할 경우, 반드시 상위 계층인 'Inet'을 거쳐야 합니다.
  - 객체에 접근하기 위해서는 이 계층 구조를 올바르게 따라야 하며, 이는 다음과 같이 구성됩니다.



### 3. VISSIM Indexing & Python 시뮬레이션 연동

- 다양한 객체와 해당 메서드 및 속성에 대한 자세한 내용은 Vissim의 COM 인터페이스 참조.

참고) C:\Users\Public\Documents\WPTV Vision\WPTV Vissim 2023\Examples Training\COM



### 3. VISSIM Indexing & Python 시뮬레이션 연동

- 객체 속성의 정의 및 구조 (Attributes)

Vissim에서 각 객체는 다양한 속성(attribute)을 갖고 있으며, COM Help의 각 속성 페이지에서 다음과 같은 정보가 제공됩니다.

속성 정보 항목 (COM에서 수정 가능한지 여부)	
식별자 (identifier)	COM 코드에서 사용되는 실제 이름
속성 이름	짧은 이름과 각 언어(German, English, French)로 된 이름
값의 자료형(value type)	숫자, 문자열 등

### 3. VISSIM Indexing & Python 시뮬레이션 연동

- 객체 속성의 정의 및 구조 (Attributes)

The screenshot shows the VISSIM HTML Help window. The left pane displays a tree view of the VISSIM object hierarchy, with 'ISignalGroup' selected. The right pane displays the 'ISignalGroup Attributes' page, which includes a 'Summary' table and an 'Attributes' section for the 'Amber' state.

**ISignalGroup Attributes**

Overview Collapse All

**Summary**

Identifier	Short name	Long name
Amber	Amber	Amber
ContrByCOM	ContrByCOM	Controlled by COM
GreenFish	GreenFish	Green flashing time
MinGreen	MinGreen	Minimum green time
MinRed	MinRed	Minimum red time
Name	Name	Name
No	No	Number
RedAmber	RedAmber	Red-amber time
SC	SC	Signal controller
SigState	SigState	Signal state
tSigState	tSigState	Signal state run time
Type	Type	Type

**Attributes**

**Amber**

Value type	preciseDurationInSec.onds
Editable	True
Simulation Behavior	ReadOnlyDuringSim
Type	Optional
Minimum	0

Names in other languages:

	Short name	Long name	Description
DEU	Gelb	Gelb	Dauer der Gelbzeit in einem Umlauf. Ist nur gültig für bestimmte LSA-Typen.
ENG	Amber	Amber	Duration of amber in one cycle. Only valid for specific SC types.
FRA	Jaune	Jaune	Durée du temps de jaune dans un cycle. Uniquement valable pour certains types d'LSL.

**ContrByCOM**



### 3. VISSIM Indexing & Python 시뮬레이션 연동

#### (3) VISSIM 시뮬레이션 Object 연동 실습

memo

### 3. VISSIM Indexing & Python 시뮬레이션 연동

#### (4) Vehicle Input & Static Route 연동 실습

memo

## 4. 시뮬레이션 실행 제어 및 데이터 전처리 보정



## 4. 시뮬레이션 실행 제어 및 데이터 전처리 보정

### (1) Simulation Object & Simulation Command Test

memo

## 4. 시뮬레이션 실행 제어 및 데이터 전처리 보정

(2) Result (Node, Travel Time)

memo

## 5. 데이터베이스 연동 및 결과 처리



## 4. 데이터베이스 연동 및 결과 처리

### (1) 데이터베이스 연동 및 결과 처리

memo

# Document

---

- C:\Users\Public\Documents\PTV Vision\PTV Vissim 2023\Examples Training\COM
- C:\Program Files\PTV Vision\PTV Vissim 2023\Doc



감사합니다.

