

---

# TAPI

---

COM Interface

사용자 문서

v.5.3.0.4

---

비주얼리서치 개발팀

최종 수정: 03/24/2010

# 목차

<b>1 소개.....</b>	<b>5</b>
1.1 TAP & TAPI 란?.....	5
1.2 목표.....	5
1.3 기능 요약.....	5
<b>2 요약.....</b>	<b>6</b>
2.1 서비스 계층.....	6
2.2 통신 설정.....	7
2.2.1 Tornado3D(서버)에서 TAP 서비스 시작 방법.....	7
2.3 변수와 템플릿 제작.....	8
2.3.1 텍스트 객체에 변수명 지정하기.....	8
2.3.2 이미지 객체에 변수명 지정하기.....	9
2.4 템플릿 관리 팁.....	10
2.4.1 네트워크를 통한 코더용 템플릿 관리.....	10
<b>3 설치하기.....</b>	<b>11</b>
3.1 TAPI Setup 설치.....	11
3.1.1 TAPI 설치 시작.....	11
3.2 샘플 파일 실행.....	13
<b>4 런타임 배포 준비.....</b>	<b>14</b>
<b>5 API 사용법.....</b>	<b>15</b>
5.1 TAP & TAPI 응용 방법.....	15
5.1.1 ITAP 인터페이스 사용.....	15
5.1.2 프로토콜 규약 사용.....	15
5.2 인터페이스 객체 구조.....	15
5.3 프로토콜 시퀀스.....	16
5.3.1 접속하기.....	16
5.3.2 접속체크.....	16
5.3.3 페이지 로드/쿼리.....	17
5.3.4 변수값 설정/송출.....	17
<b>6 API 참조.....</b>	<b>18</b>
6.1 클래스.....	18
6.1.1 ITAP.....	18
ITAP::Connect.....	18
ITAP::Hello.....	19
ITAP::Destroy.....	19
ITAP::SetEventHandler.....	19
ITAP::Send.....	20
ITAP::Bye.....	20
ITAP::LoadPage.....	20
ITAP::UnloadPage.....	21
ITAP::UnloadAllPages.....	21
ITAP::LoadProject.....	22
ITAP::SaveProject.....	22
ITAP::PreparePage.....	22
ITAP::Play.....	23
ITAP::Stop.....	24
ITAP::PauseResume.....	24
ITAP::Out.....	25
ITAP::SetObjValue.....	25
ITAP::SetObjValue2.....	26
ITAP::SetText.....	27

ITAP::SetObjVisible.....	27
ITAP::SetObjAttr.....	28
ITAP::SetPageDefault.....	30
ITAP::SetBackground.....	30
ITAP::SetPageEffect.....	30
ITAP::SetPageEffectEx.....	32
ITAP::AddPause.....	32
ITAP::DeletePause.....	33
ITAP::SetScrollSpeed.....	34
ITAP::SetAirScrollSpeed.....	34
ITAP::SetAudioFile.....	34
ITAP::SetAudioEnable.....	35
ITAP::SetStyleText.....	35
ITAP::SetImageObject.....	36
ITAP::SetImageObject2.....	37
ITAP::AddStyleText.....	37
ITAP::AddImageObject.....	38
ITAP::AddScrollObject.....	38
ITAP::AddScrollObject2.....	39
ITAP::CreateImageFile.....	39
ITAP::SetObjLine.....	40
ITAP::QueryLoad.....	41
ITAP::QueryPageMode.....	41
ITAP::QueryPageAlias.....	41
ITAP::QueryAliasAll.....	42
ITAP::QueryObjValue.....	42
ITAP::QueryObjAttr.....	43
ITAP::QueryAudioFile.....	43
ITAP::QueryAudioEnable.....	44
ITAP::QueryScrollSpeed.....	44
ITAP::QueryAirScrollMargin.....	45
ITAP::QueryBackground.....	45
ITAP::QueryPageEffect.....	46
ITAP::QueryPageEffectEx.....	46
6.1.2ITAPEventHandler.....	48
ITAPEventHandler::OnConnect.....	48
ITAPEventHandler::OnReceive.....	48
ITAPEventHandler::OnClose.....	48
6.1.3ITAPParse.....	49
ITAPParse::Parse.....	49
ITAPParse::GetCommand.....	49
ITAPParse::GetNextCommand.....	49
ITAPParse::GetNextString.....	50
ITAPParse::GetNextQuotationString.....	50
ITAPParse::GetNextOptionString.....	51
ITAPParse::GetNextStringAll.....	51
6.2열거자(Enumerated Type).....	52
eTAPCommand.....	52
eLayer.....	53
eEffectType.....	54
eTAPAttr.....	54
eTAPBg.....	55
eTAPEffect.....	55
eTAPEffectWipe.....	55
eTAPEffectPush.....	56
eTAPEffectCurl.....	56
eTAPEffectDve.....	56
eTAPEffectParticle.....	57

eTAPEffectRipple.....	57
eTAPEffectOrganic.....	58
eTAPEffectDistortion.....	58
<b>7C++ 샘플 코드.....</b>	<b>59</b>
7.1기본 설정.....	59
7.2템플릿 로드/송출 ( TAPPlay ).....	59
<b>8C# 샘플 코드.....</b>	<b>62</b>
8.1기본 설정.....	62
8.2템플릿 로드/송출 ( TAPNPlay ).....	62

# 1 소개

## 1.1 TAP & TAPI 란?

1. TCP/IP Automation Protocol 의 약자로서, 자막기의 편집 및 송출제어를 네트워크를 통해 수행하기 위한 비주얼리서치의 명령규약이다.
2. TAP 프로토콜을 쉽게 사용하기 위해서, **TAPI** (COM 인터페이스)를 제공한다. 그러므로, 사용자는 통신 소켓 구현이나, 문자열 파싱(Parsing) 등에 크게 얽매이지 않고, 다양한 플랫폼에서 COM 인터페이스를 사용하여 프로그램을 쉽게 제작할 수 있다.

## 1.2 목표

1. 자막기에 대해 모르는 응용 프로그래머도 쉽고, 빠르게 고품질의 자막 송출이 가능한 어플리케이션 개발가능.
2. 실시간 데이터, 많은 양의 데이터, 자동화된 송출 등이 필요한 조건에서 최적화된 어플리케이션 개발가능.
3. 스포츠코더, 일기예보, 선거방송, 홈쇼핑, 보도정보, 주식거래, 로고송출 등의 다양한 어플리케이션 개발가능.

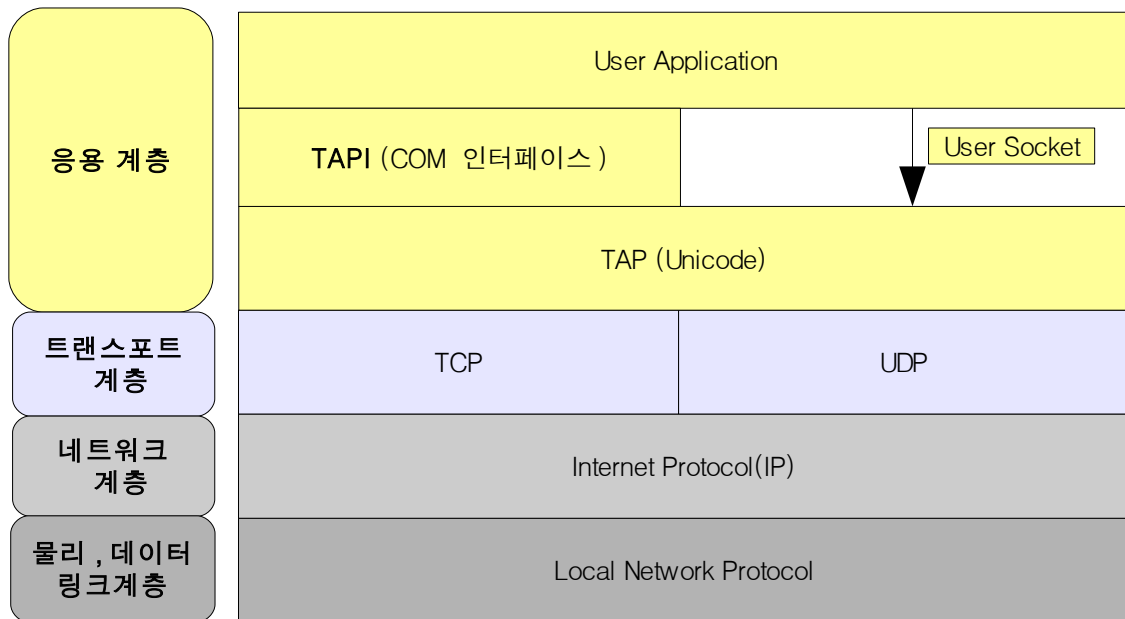
## 1.3 기능 요약

1. 자막기를 이용한 컷파일(\*tcf) 템플릿 제작.
2. 컷파일(\*.tcf) 템플릿 로드.
3. 객체의 변수화 및 검색.
4. 텍스트 객체 내용 바꾸기.
5. 도형 객체 텍스트 바꾸기.
6. 실시간 스크롤 내용 추가.
7. 실시간 Bar / Pie 애니메이션.
8. 멀티 레이어 송출.
9. 페이지 효과 설정.
10. 객체 효과 설정.
11. 다양한 송출 비디오 보드 지원.

## 2 요약

### 2.1 서비스 계층

#### ◆ TAPI 기능 계층



## 2.2 통신설정

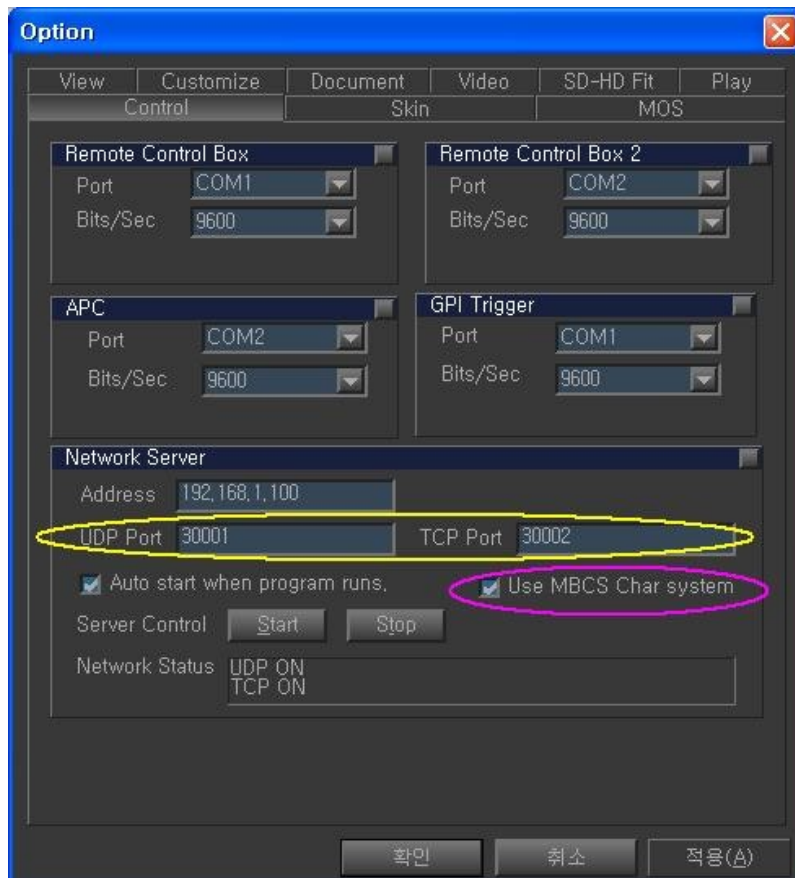
### 2.2.1 Tornado3D(서버)에서 TAP 서비스 시작 방법

#### ◆ Tools -> Option -> Control

을 선택하면, 아래와 같이 시스템에 할당되어 있는 IP가 나타나고, 사용할 Port번호를 아래와 같이 설정한다.

**UDP Port : 3001**

**TCP Port : 3002**



#### ◆ Auto start when program runs.

체크하면 프로그램 시작 시 자동으로 TAP 서비스를 시작한다. 이 기능을 해제하기 위해서는 체크 박스를 클릭하고 프로그램을 재시작 한다.

#### ◆ Use MBCS Char system :

TAP 는 유니코드를 사용하므로, 체크 해제해두어야 한다.

## 2.3 변수와 템플릿 제작

### ◆ 변수객체 (Variable Object / Alias Object)

페이지내의 객체들중에서, 변수명이 부여된 객체.

### ◆ 변수명 (Variable Name / Alias)

페이지내에서 여러 객체들을 구분하기 위해, 각각의 객체에 설정된 이름.

여러 페이지들을 구분하기 위해, 각각의 페이지에 설정된 이름.

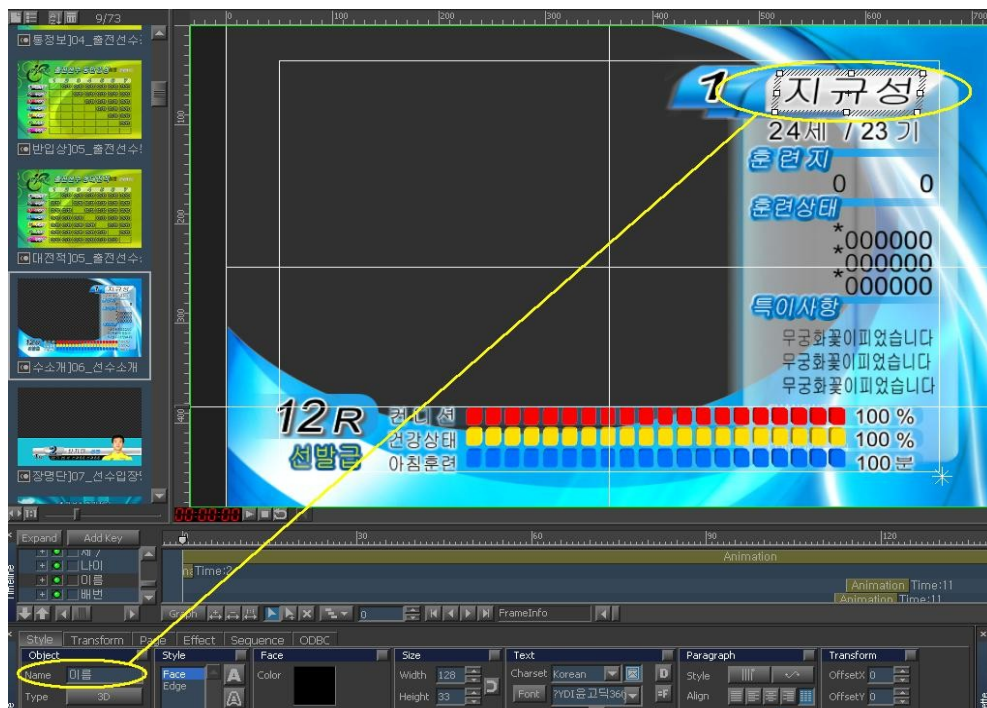
### ◆ 변수값 (Variable Value)

Text 객체일 경우, Text 내용.

도형(사각형,원,다각형)의 경우, Texture 파일명.

### 2.3.1 텍스트 객체에 변수명 지정하기

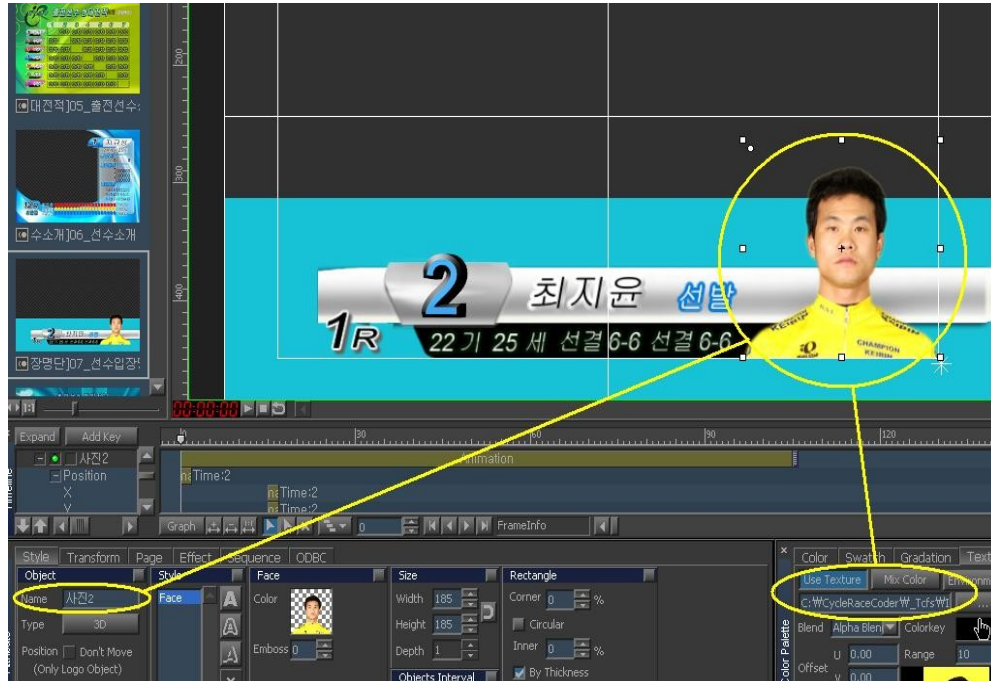
아래와 같이, 텍스트 객체를 생성한 후 선택하면, 하단 속성바 [Style]탭의 Name 란에 원하는 변수의 이름을 지정하고 템플릿을 저장한다.





### 2.3.2 이미지 객체에 변수명 지정하기

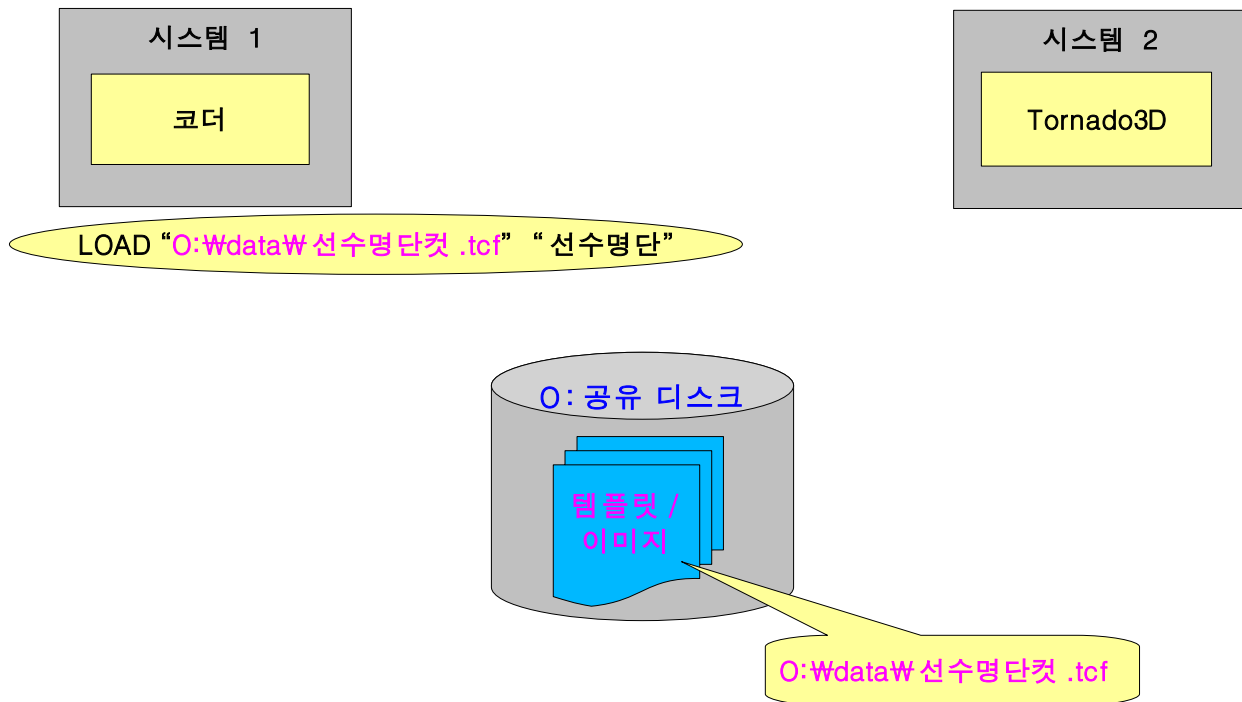
마찬가지로, 이미지 객체를 선택하여 변수명을 지정한 후, 하단우측 컬러팔레트바의 Texture 탭에서 “Use Texture” 버튼을 클릭하여, 텍스처링된 객체로 설정해준 뒤 저장한다.



## 2.4 템플릿 관리팁

### 2.4.1 네트워크를 통한 코더용 템플릿 관리.

네트워크를 통한 코더 운용시에는, 송출에 사용될 템플릿 파일(tcf, image) 들을 특정 디스크 드라이브에서 공유하여 사용하시면, 코더측에서 바로, 파일 갱신상황이나 썸네일 보기 등이 유용해집니다.



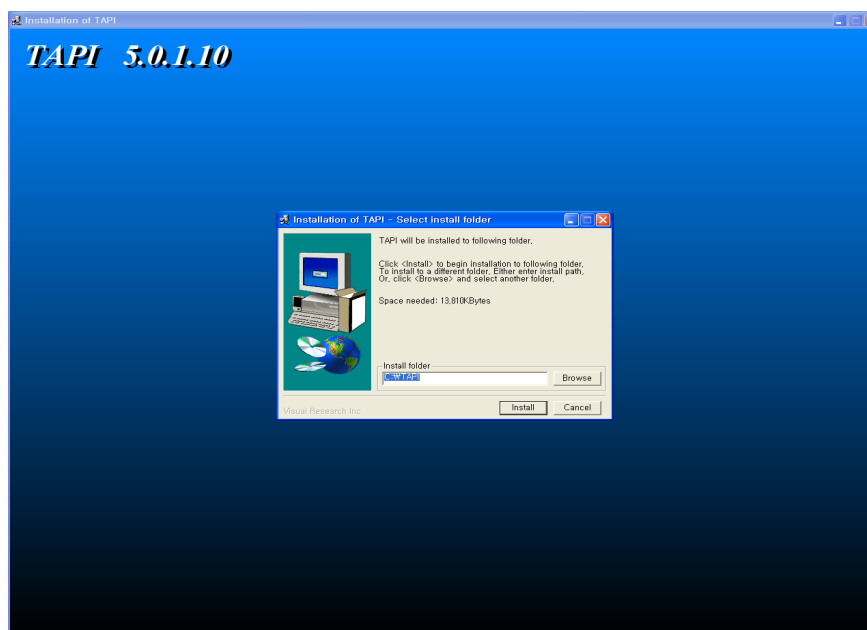
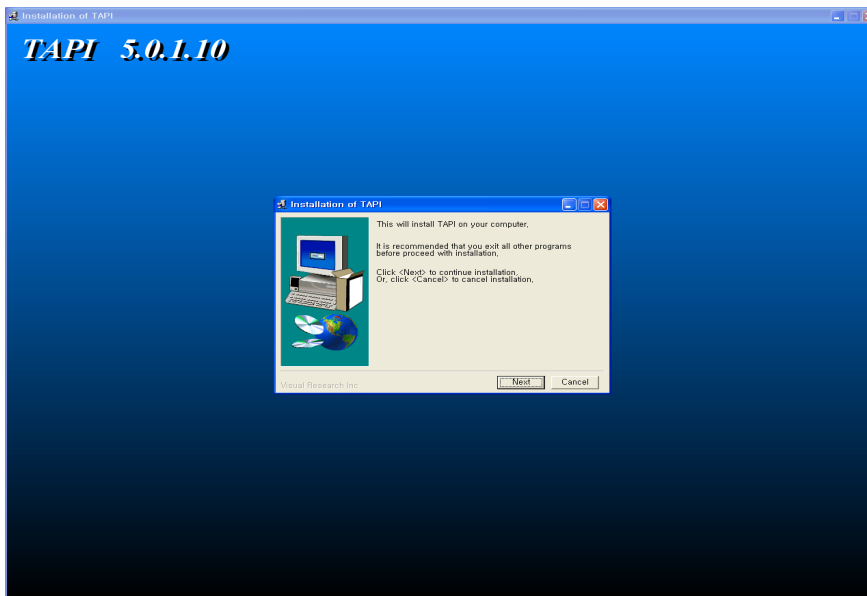
## 3 설치하기

### 3.1 TAPI Setup 설치

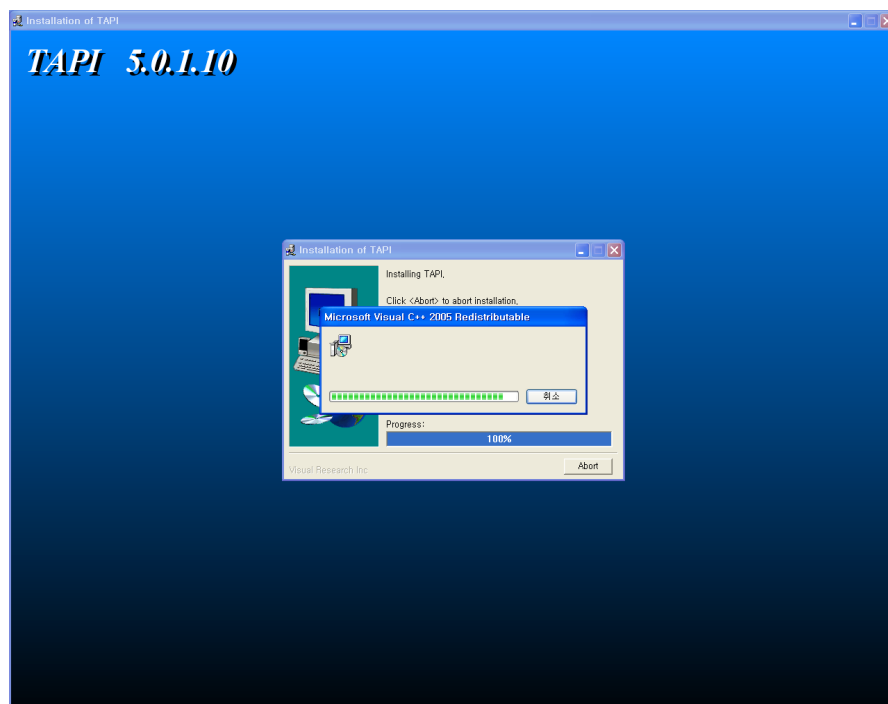
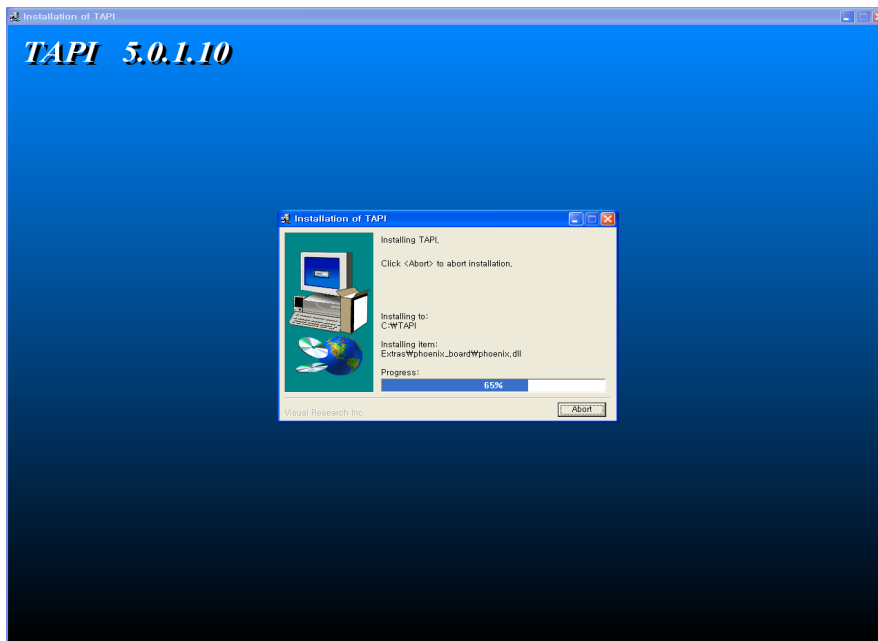
TAPI 설치 CD 의 Setup 을 실행하여 설치한다.

#### 3.1.1 TAPI 설치 시작

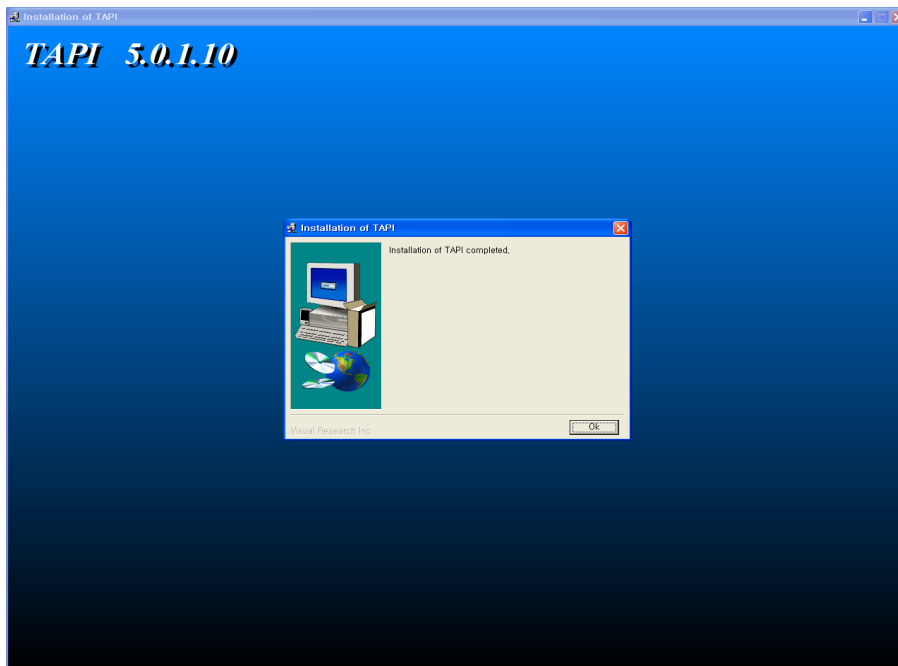
TAPI 5.x.x.x Kor Setup.exe 파일을 실행하여, 안내에 따라 다음 순서대로 설치한다.



설치 폴더를 지정한다. 디폴트는 C:/TAPI.



vcredist\_x86 을 자동으로 설치한다. 만약 이전에 한번이라도 설치했다면, 취소를 선택해도 된다.



설치 완료.

## 3.2 샘플 파일 실행

설치를 완료된 TAPI 폴더의 TAPI/Bin/C++/TAPPlay.exe 샘플 프로그램을 실행하여, 송출을 확인한다.

TAPPlay 샘플은 가장 간단히 컷파일을 로드하여 송출할 수 있는 샘플이며, 각 샘플에 대한 사용 설명은 "6.샘플코드"부분을 참고한다.

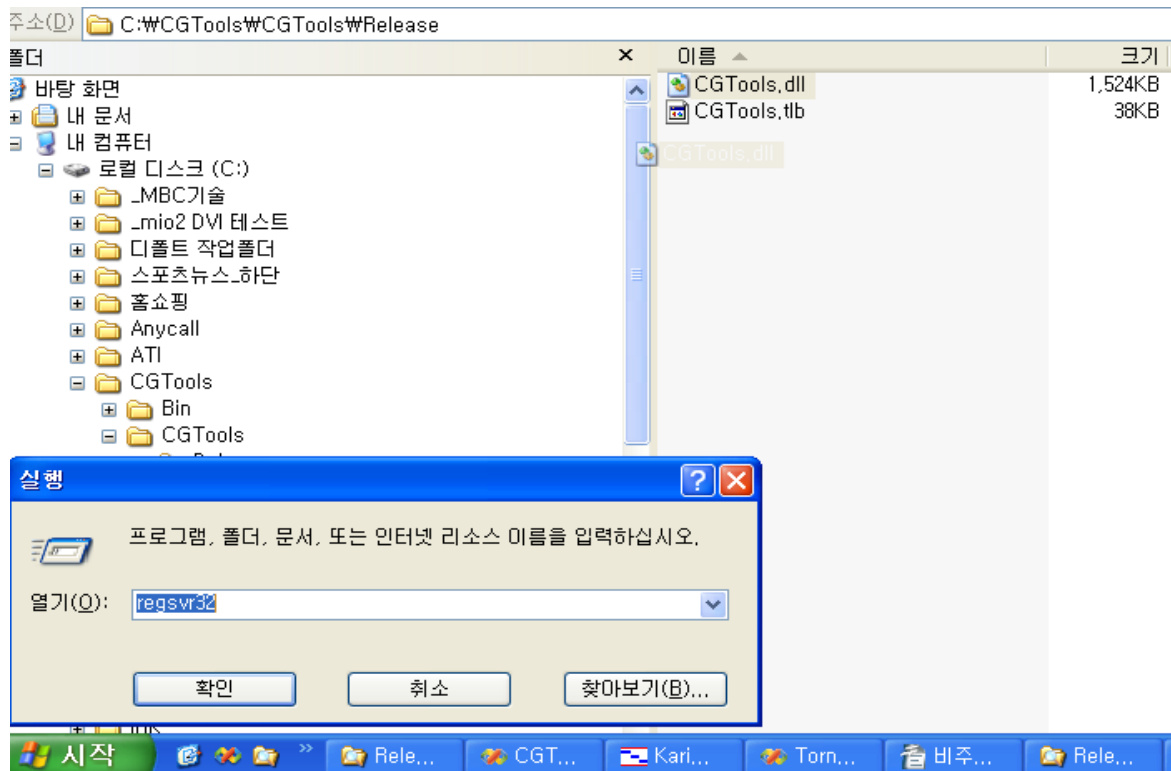
## 4 런타임 배포 준비

TAPI 를 사용하여 개발한 응용 프로그램을 배포할 경우, 새로운 시스템에서 제대로 실행되려면, 아래와 같은 작업을 해주면 된다.

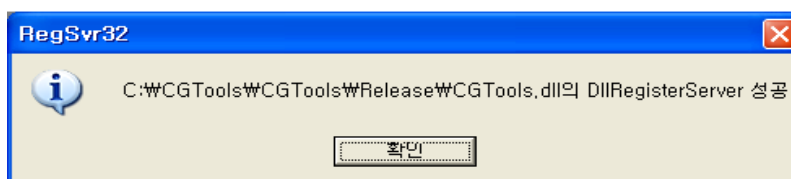
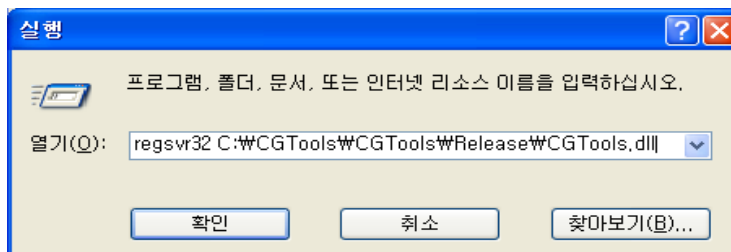
TAPI 설치 폴더내에 있는 vcredist\_x86 를 설치한 후,

시작 → 실행 → regsvr32 “TAPI.dll 의 파일 경로명”으로 COM 객체를 시스템에 등록하면 된다.

탐색기에서 “TAPI.dll” 파일을 마우스 오른쪽 클릭, 드래그 하여 [실행] 윈도우의 regsvr32 에 놓고 난 후, [확인] 버튼을 클릭한다.



성공시, 아래와 같은 창이 뜨게 되며, 응용 프로그램이 실행가능하게 된다.



## 5 API 사용법

### 5.1 TAP & TAPI 응용 방법

#### 5.1.1 ITAP 인터페이스 사용

기본적인 방법은 이 문서에서 소개하는 ITAP 인터페이스를 사용하여, CG와 통신하면 된다.

옵션이 많거나, 복잡한 명령을 한번에 보내고자 할 경우, **ITAP.Send(string)** 명령으로, **string** 을 직접 조합하여 CG에 메시지를 전달할 수 있다.

- 간단하게 함수와, 인자값을 사용하여 자막기로 명령을 보낼수 있다.

#### 5.1.2 프로토콜 규약 사용

사용자가 직접 소켓을 구성하고, "**TAP 사용자 설명서.pdf**" 문서를 참고하여 프로토콜 규약에 맞추어 유니코드로 된 메시지를 생성하여 명령을 보내는 방식이다.

- 이 경우, 변수값이나 속성값들을 **string**에 한번에 다중으로 설정할 수 있지만, 프로토콜을 규약에 맞게 조합하는 작업이 필요하다. ("**TAP 사용자 설명서.pdf**" 문서를 참고)
- 간단한 프로토콜 규칙을 다시 정리하면 다음과 같다.

- 명령과 인자, 인자와 인자 간에는 반드시 공백을 사용해야 한다. 공백간격에는 영향을 받지않는다.

예) **ITAP.Send(" LOAD "C:\Wdata\스코어컷.tcf" "스코어장면" )**

- 명령어는 대소문자 구분을 하지 않는다.

- 예약어, 정수를 제외한 모든 인자(페이지명, 파일명, 오브젝트명, 일반텍스트)는 " "기호로 묶어서 사용한다.

- ` ` 와 같은 특수 예약어를 문자열로 입력하기 위해서는 'W'와 같이 사용하면 입력이 가능하다.

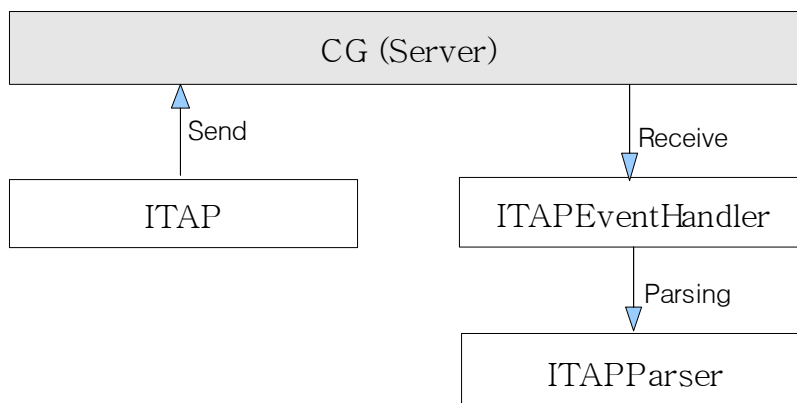
예) 국가1명 오브젝트를 "대한민국" 라는 문자열로 변경할 경우

**OBJVALUE "스코어장면" "국가1명" W"대한민국W"**

- '|' 기호를 사용하여 다중 명령문을 생성할 수 있다.

예) **OBJVALUE "스코어장면" "국가1명" "대한민국" | PREPARE "스코어장면"**

### 5.2 인터페이스 객체 구조



## 5.3 프로토콜 시퀀스

접속부터 송출하기까지의 시퀀스를 다이어그램으로 나타내었습니다.

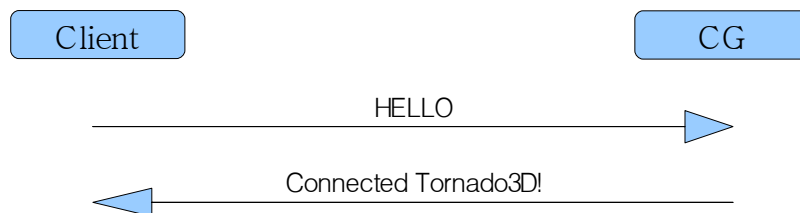
기본적인 순서를 요약하자면,

1. 접속 / 접속체크.
2. 페이지 로드 / 로드된 페이지리스트 쿼리해보기.
3. 원하는 페이지 변수값들 초기화.(PAGEDEFAULT)
4. 변수값 or 속성값 설정(OBJVALUE, OBJATTR) + 배경속성, 페이지효과 등등...
5. 송출준비(PREPARE)
6. 송출시작(PLAY)
7. 송출제어(PAUSE, RESUME, OUT, STOP)

이후, 다른 페이지를 송출할때에는, (3) ~ (7) 를 반복하면 됩니다.

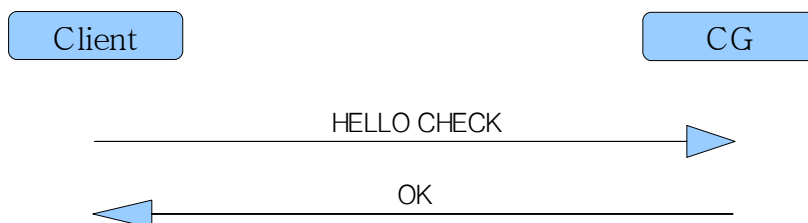
### 5.3.1 접속하기

TCP 모드의 경우, 소켓 연결 성공후, HELLO 메시지를 보내면, Connected to CG! 메시지가 리턴된다.



### 5.3.2 접속체크

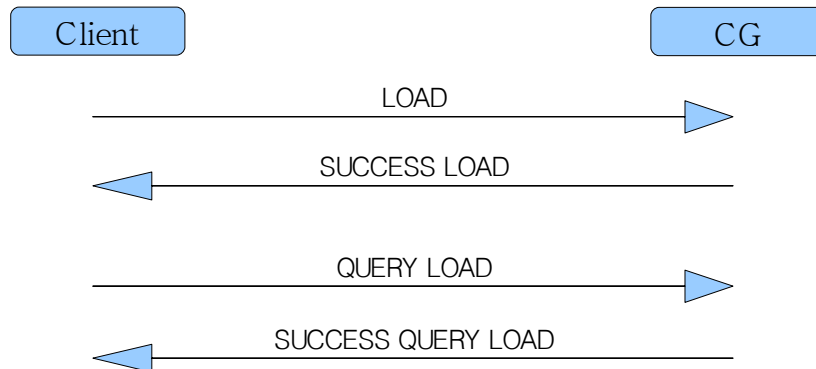
UDP 모드의 경우, 전송 네트워크 상황의 이상유무를 판단하기 위해, 접속체크를 할 수 있다.





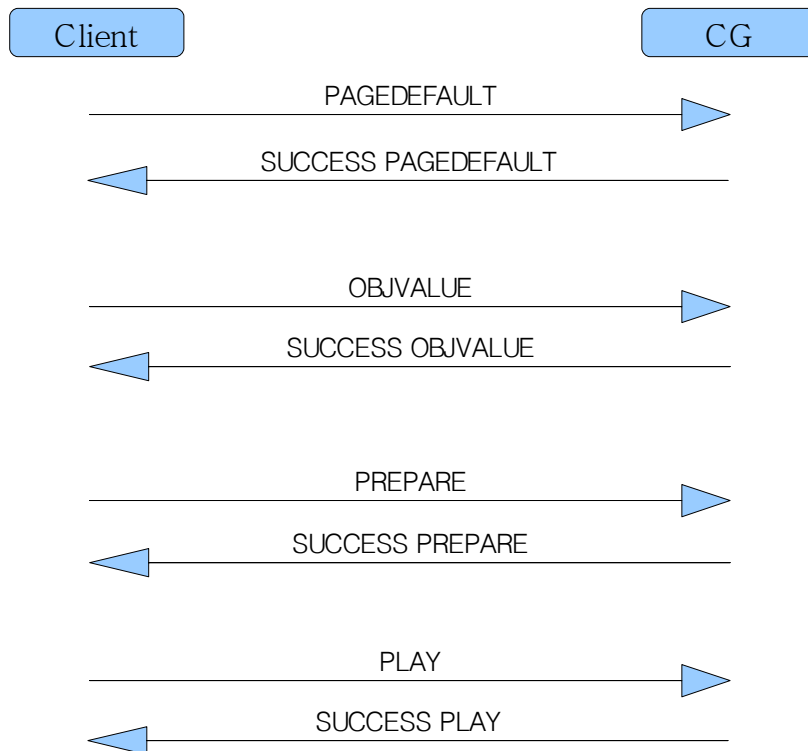
### 5.3.3 페이지 로드/쿼리

자막기에 페이지를 로드시키고, 로드된 페이지의 정보를 가져온다.



### 5.3.4 변수값 설정/송출

로드된 페이지의 변수값을 설정하고, 송출준비한 후, 송출한다.



## 6 API 참조

TAPI 에서 사용되는 COM(Component Object Model) 인터페이스에 대한 설명이다.

### 6.1 클래스

#### 6.1.1 ITAP

TAPI 를 이용하기 위한 메인 객체이며, 자막기에 명령을 보내기 위해, 소켓을 통해 연결하는 초기화 작업과 함께, 특정 메시지를 보내기 위한 인터페이스를 가지고 있다.

---

##### ITAP::Connect

---

자막기의 IP, Port 를 알고 있다면, 초기화 및 접속을 실행한다.

###### 구문

```
HRESULT Connect(  
    BOOL bTCP,  
    BSTR ServerIPAddress,  
    LONG ServerPort,  
    LONG ClientPort,  
    ITAPEventHandler* pHandler  
);
```

###### 인자

bTCP

TCP 또는 UDP 인가를 선택한다. TCP 모드라면 TRUE

ServerIPAddress

자막기의 IP 주소.

ServerPort

자막기의 열린 Port 번호.

ClientPort

사용자 클라이언트의 특정 Port 번호. 현재 사용되지 않음. 0 을 지정한다.

pHandler

ITAPEventHandler 를 상속받아 생성된 사용자측의 객체. 이 객체를 통하여 자막기가 보내오는 콜백 메시지를 수신한다.

###### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

###### 비고

연결 후, ITAP::Hello 함수로 자막기와의 연결상태를 체크하고, 페이지 리스트를 초기화할 수 있다.

###### 참조

ITAP::Destroy , ITAP::Hello

---

## ITAP::Hello

---

사용자 클라이언트와 자막기와의 연결 상태를 하기 위해 보낸다.

### 구문

```
HRESULT Hello(  
    BOOL bCheck  
);
```

### 인자

bCheck

처음 연결 후, 페이지 리스트를 초기화 하기 위해서는 FALSE 를 사용하고, 사용중에 자막기와의 연결상태만을 체크하고자 할 경우 TRUE 를 사용한다.

### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

### 비고

### 예제

연결 후, 페이지 리스트 초기화.

```
<ITAP>      Hello(FALSE);  
<OnReceive> Connected CG!
```

연결상태만 체크

```
<ITAP>      Hello(TRUE);  
<OnReceive> OK
```

### 참조

ITAP::Connect

---

## ITAP::Destroy

---

송출 하드웨어를 중지하고, 소켓연결을 해제한다.

### 구문

```
HRESULT Destroy();
```

### 인자

### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

### 비고

### 참조

ITAP::Connect

---

## ITAP::SetEventHandler

---

자막기로부터 메시지를 수신하기 위한 핸들러를 설정한다.

### 구문

```
HRESULT SetEventHandler(  
    ITAPEventHandler* pHandler  
);
```

### 인자

pHandler

ITAPEventHandler 를 상속받아 생성한 사용자용 인터페이스. ITAPEventHandler::OnReceive 함수를 오버라이드하여 자막기로부터의 메시지를 수신하여 처리할 수 있다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 참조

ITAPEventHandler 인터페이스

---

### ITAP::Send

---

사용자가 자막기로 메시지를 보낼때 사용한다. TAP 프로토콜에 맞춰서 어떤 명령이든 보낼 수 있다.

#### 구문

```
HRESULT Send(  
    BSTR string  
);
```

#### 인자

string

유니코드 문자열로 구성된 TAP 프로토콜.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

ITAP 에서 제공되는 모든 함수들은 TAP 프로토콜을 이용할 경우, 이 함수만으로도 모두 대체가 가능하다.

#### 예제

```
<ITAP> Send(" LOAD "C:\WdataW선수명단컷.tcf" "선수명단" ");  
<OnReceive> SUCCESS LOAD
```

#### 참조

---

### ITAP::Bye

---

UDP 로 자막기에 프로토콜을 주고 받을 경우, 종료하기 전에, 자막기에 로드되어 있는 페이지 리스트를 제거하기 위해 사용한다.

#### 구문

```
HRESULT Bye();
```

#### 인자

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP> Bye();  
<OnReceive> Bye!
```

#### 참조

---

### ITAP::LoadPage

---

Tcf 파일을 자막기에 로드시킨다.

#### 구문

```
HRESULT LoadPage(  
    BSTR FileName,  
    BSTR PageName  
);
```

#### 인자

FileName  
Tcf 파일의 전체 경로명  
PageName  
사용자가 지정하는 페이지명

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP> LoadPage("컷파일명", "페이지명");  
<OnReceive> SUCCESS LOAD 페이지명
```

#### 참조

ITAP::QueryLoad

---

## ITAP::UnloadPage

---

자막기에 로드되어 있는 페이지를 제거하도록 한다.

#### 구문

```
HRESULT UnloadPage(  
    BSTR PageName  
);
```

#### 인자

PageName  
LoadPage 로 로드할 당시의 페이지에 사용자가 지정한 페이지명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP> UnloadPage("페이지명");  
<OnReceive> SUCCESS UNLOAD
```

#### 참조

ITAP::DeleteAllPages

---

## ITAP::UnloadAllPages

---

자막기에 로드되어 있는 모든 페이지들을 제거한다.

#### 구문

```
HRESULT UnloadAllPages();
```

#### 인자

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP>      UnloadAllPages();  
<OnReceive> SUCCESS UNLOADALL
```

#### 참조

UnloadPage

---

### ITAP::LoadProject

---

T3D 프로젝트 파일을 자막기에 로드시킨다.

#### 구문

```
HRESULT LoadProject(  
    BSTR T3DFileName  
);
```

#### 인자

T3DFileName  
T3D 프로젝트 파일의 전체 경로명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP>      LoadProject("C:\\Data\\ProjectFile.t3d");  
<OnReceive> SUCCESS LOADPROJECT
```

#### 참조

ITAP::SaveProject

---

### ITAP::SaveProject

---

자막기에 로드되어 있는 모든 페이지들을 T3D 프로젝트 파일로 저장하도록 한다.

#### 구문

```
HRESULT SaveProject(  
    BSTR T3DFileName  
);
```

#### 인자

T3DFileName  
사용자가 원하는 T3D 파일명의 전체경로명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP>      SaveProject("C:\\Save\\ProjectFile.t3d");  
<OnReceive> SUCCESS SAVEPROJECT
```

#### 참조

ITAP::LoadProject

---

### ITAP::PreparePage

---

자막기에 로드되어 있는 페이지에 모든 변수나 속성들을 설정한 후, 송출 직전에 준비하도록 한다.

## 구문

```
HRESULT PreparePage(  
    BSTR PageName,  
    eLayer Layer,  
    BOOL bAutoPreview  
);
```

## 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

Layer

송출하게 될 레이어.(주의: 현재, LAYER\_TOP 은 안됨) - eLayer 참조.

bAutoPreview

TRUE 로 지정되어 있을 경우, Next 프리뷰 모니터로 프리뷰를 나타나도록 한다.

## 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

## 비고

### 예제

ex) 만약, 기본 레이어를 준비하게 되면, 응답은 "PREPARE".

```
<ITAP> PreparePage("PageName", TAPI::LAYER_BASE );
```

```
<OnReceive> SUCCESS PREPARE PageName
```

ex) 만약, 섹션 레이어를 준비하게 되면, 응답은 "SECTIONPREPARE" + "Layer Index".

```
<ITAP> PreparePage("PageName", TAPI::LAYER_SECTION_3 );
```

```
<OnReceive> SUCCESS SECTIONPREPARE 3 PageName
```

- 3 : Index corresponding to the layer.
  - 0 : LAYER\_SECTION\_0
  - 1 : LAYER\_SECTION\_1
  - 2 : LAYER\_SECTION\_2
  - 3 : LAYER\_SECTION\_3

## 참조

ITAP::Play

---

## ITAP::Play

---

Prepare 가 완료된 페이지를 송출하도록 한다.

## 구문

```
HRESULT Play(  
    eLayer Layer  
);
```

## 인자

Layer

송출하게 될 레이어.(주의: 현재, LAYER\_TOP 은 안됨) - eLayer 참조.

## 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

## 비고

### 예제

ex) 만약 기본 레이어라면,

```
<ITAP> Play(TAPI::LAYER_BASE );
```

```
<OnReceive> SUCCESS PLAY START
```

```
... At the end of playing...
```

```
<OnReceive> SUCCESS PLAY STOP
```

ex) 만약, 섹션 레이어라면,

```
<ITAP>      Play(TAPI::LAYER_SECTION_2 );  
<OnReceive> SUCCESS  SECTIONIN  2  
... At the end of playing...  
<OnReceive> SUCCESS  SECTIONPLAY  STOP  2
```

#### 참조

ITAP::Prepare

---

## ITAP::Stop

---

송출중인 전체 화면을 지우고 정지하도록 한다.

#### 구문

```
HRESULT Stop();
```

#### 인자

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

송출에 관련된 동작을 모두 중지한다.

#### 예제

```
<ITAP>      Stop();  
<OnReceive> SUCCESS  STOP
```

#### 참조

ITAP::Play , ITAP::Out

---

## ITAP::PauseResume

---

송출중인 화면을 일시정지/재시작 을 수행한다.

#### 구문

```
HRESULT PauseResume(  
    eLayer Layer,  
    BOOL bBackground  
);
```

#### 인자

Layer

송출하게 될 레이어.(주의: 현재, LAYER\_TOP 은 안됨) – eLayer 참조.

bBackground

배경만을 일시정지/재시작 하고자 할때, TRUE 를 지정한다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

토글 방식이므로, 송출중이었다면 일시정지하게 되고, 일시정지 상태였다면 재시작하게 된다.

#### 예제

ex) 만약 기본 레이어라면,

```
<ITAP>      PauseResume(TAPI::LAYER_BASE , FALSE);  
<OnReceive> SUCCESS  PAUSE  PAUSED  
<OnReceive> SUCCESS  PAUSE  RESUMED
```

ex) 만약, 섹션 레이어라면,

```
<ITAP>      PauseResume(TAPI::LAYER_SECTION_2 , FALSE);
```



```

<OnReceive>    SUCCESS    PAUSE    PAUSED    2
<OnReceive>    SUCCESS    PAUSE    RESUMED    2
ex) Background Pause/Resume
<ITAP>         PauseResume(TAPI::LAYER_BASE , TRUE);
<OnReceive>    SUCCESS    PAUSE    PAUSED

```

#### 참조

---

## ITAP::Out

---

송출중인 장면을 아웃시킨다. 효과가 지정되어 있다면, 효과가 적용되어 아웃되게 된다.

#### 구문

```

HRESULT Out(
    eLayer Layer
);

```

#### 인자

Layer

송출하게 될 레이어.(주의: 현재, LAYER\_TOP 은 안됨) - eLayer 참조.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

ex) 만약 기본 레이어라면,

```

<ITAP>         Out(TAPI::LAYER_BASE);
<OnReceive>    SUCCESS    OUT

```

ex) 만약, 섹션 레이어라면,

```

<ITAP>         Out(TAPI::LAYER_SECTION_0);
<OnReceive>    SUCCESS    SECTIONOUT    0

```

#### 참조

ITAP::Stop

---

## ITAP::SetObjValue

---

로드된 페이지 내부의 변수객체에 변수값을 설정한다.

#### 구문

```

HRESULT SetObjValue(
    BSTR PageName,
    BSTR ObjName,
    BSTR Value,
    BOOL bTextImage
);

```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명.

Value

사용자가 지정할 변수값.

텍스트 변수객체라면, 텍스트 내용을 설정해야 하며, 도형 변수객체라면, "이미지 파일 경로명"을

넣어주어야 하며, 텍스트로 지정되어 송출된다.

bTextImage

텍스트 변수객체일 경우, Value 인자에 "이미지 파일명"을 지정한 후, TRUE 를 설정한다면, Value 에 지정된 텍스트 내용이 아니라, 이미지를 텍스트내용 사이에 삽입하게 된다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

프로토콜 패턴

1. OBJVALUE “페이지명” “오브젝트명” “텍스트”
2. OBJVALUE “페이지명” “오브젝트명” SEQTGA “텍스트”
3. OBJVALUE “페이지명” “오브젝트명” SHOW/HIDE
4. OBJVALUE “페이지명” “오브젝트명” TEXTIMAGE “텍스트”

#### 예제

하나의 변수에 대해 값 설정

```
<ITAP> SetObjValue("PageName", "ObjName", "Hello World", FALSE);  
<OnReceive> SUCCESS OBJVALUE
```

여러개의 변수에 대해 한꺼번에 설정할 경우, ITAP::Send 를 이용

```
string = OBJVALUE “스코어장면” “팀1이름” “수원삼성” “팀2이름” “서울FC” “팀1로고” “C:\WdataW수  
원 로고.jpg” “팀2로고” “C:\WdataW서울FC로고.jpg”  
  
<ITAP> Send(string);  
<OnReceive> SUCCESS OBJVALUE
```

#### 참조

---

## ITAP::SetObjValue2

---

로드된 페이지 내부의 변수객체에 변수값을 설정한다.

#### 구문

```
HRESULT SetObjValue2(  
    BSTR PageName,  
    BSTR ObjName,  
    BSTR Value,  
    BOOL bSeqTga  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명.

Value

사용자가 지정할 변수값.

텍스트 변수객체라면, 텍스트 내용을 설정해야 하며, 도형 변수객체라면, "이미지 파일 경로명"을 넣어주어야 하며, 텍스트로 지정되어 송출된다.

bSeqTga

Sequence Tga 일 경우, 변수객체에 해당하는 도형에 연속된 이미지를 삽입하게 된다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

프로토콜 패턴

1. OBJVALUE “페이지명” “오브젝트명” “텍스트”

2. OBJVALUE “페이지명” “오브젝트명” SEQTGA “텍스트”
3. OBJVALUE “페이지명” “오브젝트명” SHOW/HIDE

#### 예제

하나의 변수에 대해 값 설정

```
<ITAP>      SetObjValue2("PageName", "ObjName", "Hello World", FALSE);
<OnReceive> SUCCESS OBJVALUE2
```

여러개의 변수에 대해 한꺼번에 설정할 경우, ITAP::Send 를 이용

```
string = OBJVALUE2 “스코어장면” “팀1이름” “수원삼성” “팀2이름” “서울FC” “팀1로고”
“C:\WdataW수원 로고.jpg” “팀2로고” “C:\WdataW서울FC로고.jpg”
```

```
<ITAP>      Send(string);
<OnReceive> SUCCESS OBJVALUE2
```

#### 참조

---

### ITAP::SetText

---

텍스트 변수객체의 내용을 설정한다. SetObjValue 로도 가능하다.

#### 구문

```
HRESULT SetText(
    BSTR PageName,
    BSTR ObjName,
    BSTR TextValue
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명. 텍스트 변수객체이어야 한다.

TextValue

사용자가 지정할 변수값.

텍스트 내용을 설정해야 한다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

텍스트 변수에 대해서는 기능상, ITAP::SetObjValue 와 동일하다.

#### 예제

```
<ITAP>      SetText("PageName", "ObjName", "Hello World");
<OnReceive> SUCCESS OBJVALUE
```

#### 참조

ITAP::SetObjValue

---

### ITAP::SetObjVisible

---

송출하기 전, 특정 변수객체를 보여줄것인지, 안보여줄 것인지를 설정한다.

#### 구문

```
HRESULT SetObjVisible(
    BSTR PageName,
    BSTR ObjName,
    BOOL bVisible
);
```

## 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명. 텍스트 변수객체이어야 한다.

bVisible

TRUE 일 경우, 송출시 보여지게 된다.

## 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

## 비고

## 예제

```
<ITAP>      SetObjVisible("페이지명", "변수명", FALSE);  
<OnReceive> SUCCESS OBJVALUE
```

## 참조

## ITAP::SetObjAttr

변수 객체의 속성값들을 설정한다. 속성값들이 다양한 관계로, Options 인자에 다양한 속성들을 한번에 설정할 수 있다.

## 구문

```
HRESULT SetObjAttr(  
    BSTR PageName,  
    BSTR ObjName,  
    BSTR Options  
);
```

## 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명. 텍스트 변수객체이어야 한다.

Options

아래에 있는 옵션어와 옵션값을 이용한 속성을 설정한 내용. 반드시, 옵션어와 옵션값은 짝을 이루어야 한다.

옵션어	옵션값	내용
Type	2D, 3D	예약어
Color	정수,정수,정수,정수	A, R, G, B ( 0 ~ 255 범위의 색상 및 투명도값 )
Width	1 ~ 9999	
Height	1 ~ 9999	
Depth	1 ~ 9999	
Position	정수,정수,정수	X, Y, Z
Pivot	정수,정수,정수	X, Y, Z
Rotation	정수,정수,정수	X, Y, Z
Scale	정수,정수,정수	X, Y, Z
ScaleStart	정수,정수,정수	X, Y, Z (애니메이션 객체일 경우, 시작프레임때의 스케일값을 지정)

Slant	정수,정수	X, Y
Start	1 ~ 정수	SeqTga 재생 시작 프레임
Finish	1 ~ 정수	SeqTga 재생 끝 프레임
Loop	-1 ~ 정수	SeqTga 재생 반복횟수 (-1 = 무한반복)
BarWidth	정수,정수	Ratio, Type, (Ratio: 원래 사이즈의 % 지정) (Type: 1:좌->우, 2:하->상)
CircleAngle	정수,정수,정수,정수	시작프레임,시작각도,종료프레임,종료각도
Effect	IN/OUT 효과명 IN/OUT 효과명 옵션어 옵션값	(효과명,옵션어,옵션값) 은 PAGEEFFECT 내용과 동일.
이하, Clock/Timer/Counter 오브젝트일 경우의 옵션		
TextType	Clock, Timer, Counter	Clock/Timer/Counter 객체의 구분
Format	1 ~ 정수 (따옴표로 감싸주어야함)	Clock 의 경우와 Timer/Counter 의 경우. 표시되는 형식의 인덱스 (1부터). (Tornado3D의 하단 Style탭-> Paragraph 에서, "Clock/Counter" 버튼 참조) [Type] 콤보에서 Clock 을 선택했을때와 Timer/Counter를 선택했을때, [Format] 콤보박스에 나타나는 포맷 순서를 인덱스화.
TimerBegin	정수 00:00:00 (타입)	-> 정수 (타입)과 -> 분;초;프레임 (타입)
TimerEnd	정수 00:00:00 (타입)	-> 정수 (타입)과 -> 분;초;프레임 (타입)
TimerUpdate	정수	업데이트 간격(초)
TimerDuration	정수	전체송출시간(초)

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

(ex1) <스코어장면>페이지의 <팀1이름>오브젝트의 타입을 3D 객체로 설정하기.

```
<ITAP>      ObjAttr("스코어장면", "팀1이름", "Type 3D");
<OnReceive> SUCCESS OBJATTR
```

(ex2) 하나의 오브젝트에서 여러속성을 한꺼번에 설정하기.

```
<ITAP>      ObjAttr("스코어장면", "팀1이름", "Type 3D Width 200 Slant 10,20");
<OnReceive> SUCCESS OBJATTR
```

(ex3) 바 애니메이션 설정. (x축으로 0 -> 80 까지 증가) (x축으로 80 -> 20 으로 감소)

```
<ITAP>      ObjAttr("페이지명", "하단바", "Scale 80,100,100");
<ITAP>      ObjAttr("페이지명", "하단바", "ScaleStart 80,100,100 Scale 20,100,100");
```

(ex4) 객체에 WIPE 효과 설정.

```
<ITAP>      ObjAttr("페이지명", "박스1", "Effect IN WIPE");
<OnReceive> SUCCESS OBJATTR
```

#### 참조

---

## ITAP::SetPageDefault

---

로드된 페이지에 변수값등을 설정하여 송출한 후, 다시 원래 템플릿 상태의 변수내용으로 돌려놓는다. 이후, 필요에 따라 특정 변수값만을 설정하여 송출할 수 있다.

### 구문

```
HRESULT SetPageDefault(  
    BSTR PageName  
);
```

### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

### 비고

### 예제

```
<ITAP> SetPageDefault("페이지명");  
<OnReceive> SUCCESS PAGEDEFAULT
```

### 참조

---

## ITAP::SetBackground

---

로드된 페이지의 배경을 지정한다.

### 구문

```
HRESULT SetBackground(  
    BSTR PageName,  
    BSTR BGFileName  
);
```

### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

BGFileName

배경으로 사용될 텍스처 파일명. AVI 동영상도 가능하다.

### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

### 비고

### 예제

```
<ITAP> SetBackground("페이지명", "C:\WimageWback01.tga");  
<OnReceive> SUCCESS BACKGRND
```

### 참조

---

## ITAP::SetPageEffect

---

로드된 페이지의 페이지효과를 지정한다. 효과설정이 다양한 관계로, Options 인자에 직접 TAP 프로토콜을 이용하여 입력해야 한다.

### 구문

```
HRESULT SetPageEffect(  

```

```

BSTR PageName,
BOOL bIn,
eEffectType EffectType,
BSTR Options

```

```
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정 한 페이지명.

bIn

TRUE 일 경우, 장면 IN 송출시 사용되는 효과를 지정한다.

EffectType

지정할 효과의 종류. eEffectType 참고.

Options

TAP 프로토콜로 구성된 효과의 하위 속성 설정.

(굵은 글씨는 키워드이며, 프로토콜 조합시, 그대로 써주면 된다)

효과명	옵션어	옵션값
Cut		없음.
Wipe	Shape	Divide, Rectangle, Circle, Fan
	Position	Up, Down, Left, Right, UpLeft, UpRight, DownLeft, DownRight, Center
Push	Direction	Up, Down, Left, Right, UpLeft, UpRight, DownLeft, DownRight
Dve	Position	정수, 정수, 정수
	Center	정수, 정수, 정수
	Scale	정수, 정수, 정수
	Rotation	정수, 정수, 정수
Curl	Radius	정수 : 4 ~ 50
	Angle	정수 : 0 ~ 359
Particle	Shape	Tile, Random, Radial
	Order	AtOnce, Random, Top, Bottom, Left, Right, TopLeft, TopRight, BottomLeft, BottomRight, PointCenter, PointSide, SnakeTop, SnakeBottom, SnakeLeft, SnakeRight, SnakeCenter, SnakeSide
	Motion	Scatter2D, Scatter3D
	DivideX	정수 : 1 ~ 50
	DivideY	정수 : 1 ~ 50
	Torque	정수 : -30 ~ 30
	OrderX	정수 : 0 ~ 100
	OrderY	정수 : 0 ~ 100
	ConvergeX	정수 : 0 ~ 100
	ConvergeY	정수 : 0 ~ 100
	Direction	정수 : 0 ~ 360
Ripple	Shape	Water, Flag
	Frequency	정수 : 2 ~ 5
	Amplitude	정수 : 1 ~ 50
	OriginX	정수 : 0 ~ 100
	OriginY	정수 : 0 ~ 100
	Slant	정수 : 0 ~ 90
Organic	bUseMatte	On, Off
	FileName	“문자열” : 파일명

Distortion	Height	정수 : 0 ~ 999
	FileName	“문자열” : 파일명

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

Fade 설정은 없지만, Organic 으로 Fade 효과가 가능하다. 단, bUseMatte = Off 여야 한다.

#### 예제

(ex1) <스코어장면>페이지의 IN 효과를 Push 로 설정하기.

```
<ITAP> SetPageEffect(“스코어장면”, IN, EFFECT_PUSH, NULL);
<OnReceive> SUCCESS PAGEEFFECT
```

(ex1) <스코어장면>페이지의 IN 효과를 Push 로 설정하고, 하위 속성 설정.

```
<ITAP> SetPageEffect(“스코어장면”, IN, EFFECT_PUSH, "Direction Up");
<OnReceive> SUCCESS PAGEEFFECT
```

#### 참조

ITAP::SetPageEffectEx

## ITAP::SetPageEffectEx

로드된 페이지의 효과를 간단히 종류와 프레임수만 바꾸도록 설정한다.

#### 구문

```
HRESULT SetPageEffectEx(
    BSTR PageName,
    BOOL bIn,
    eEffectType EffectType,
    LONG FrameCount
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

bIn

TRUE 일 경우, 장면 IN 송출시 사용되는 효과를 지정한다.

EffectType

지정할 효과의 종류. eEffectType 참고.

FrameCount

지정된 효과가 보여질 프레임수. Duration 개념이다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

Fade 설정은 없지만, Organic 으로 Fade 효과가 가능하다.

#### 예제

(ex1) <스코어장면>페이지의 IN 효과를 Push로 지정 후, 60프레임동안 송출하도록 지정.

```
<ITAP> SetPageEffectEx(“스코어장면”, IN, EFFECT_PUSH, 60);
<OnReceive> SUCCESS PAGEEFFECTEX
```

#### 참조

ITAP::SetPageEffect

## ITAP::AddPause

로드된 페이지에 송출중 일시정지하게 될 일시정지점을 지정한다. 일시정지후, 다시 자동으로 재개되도



록 할 수도 있다.

#### 구문

```
HRESULT AddPause(  
    BSTR PageName,  
    LONG PauseFrame,  
    LONG DelayFrame,  
    BOOL bAutoResume  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

PauseFrame

송출중, 일시정지하게 될 시점을 프레임수로 입력.

DelayFrame

일시정지후, 지속되는 프레임수.

bAutoResume

일시정지후, 자동으로 재시작될것인지를 설정.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

"뉴스하단" 라는 스크롤 페이지에서, 송출후 15 프레임에서, 20프레임간 멈춘후, 자동 Resume 된다.

```
<ITAP> AddPause("뉴스하단", 15, 20, TRUE);  
<OnReceive> SUCCESS ADDPAUSE
```

#### 참조

ITAP::DeletePause

---

## ITAP::DeletePause

---

로드된 페이지에 지정되어 있는 일시정지점을 제거한다.

#### 구문

```
HRESULT DeletePause(  
    BSTR PageName,  
    LONG PauseFrame  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

PauseFrame

지정되어 있는 일시정지점의 위치. 프레임수로 입력.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

"뉴스하단" 이라는 스크롤 페이지에서, 15 프레임에 지정된 일시정지점을 삭제하기.

```
<ITAP> DeletePause("뉴스하단", 15);  
<OnReceive> SUCCESS DELETEPAUSE
```

## 참조

ITAP::AddPause

---

## ITAP::SetScrollSpeed

---

롤/크롤 등의 스크롤 페이지의 스크롤 속도를 설정한다. 송출중이 아니라, Prepare 하기 전에 사용한다.

### 구문

```
HRESULT SetScrollSpeed(  
    BSTR PageName,  
    LONG Speed  
);
```

### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

Speed

스크롤 속도. 초당 진행 픽셀수.

### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

### 비고

### 예제

```
<ITAP> SetScrollSpeed("뉴스하단", 10);  
<OnReceive> SUCCESS SCROLLSPEED
```

### 참조

SetAirScrollSpeed

---

## ITAP::SetAirScrollSpeed

---

송출중인 크롤/롤 등의 스크롤 페이지의 스크롤 속도를 실시간으로 조정한다.

### 구문

```
HRESULT SetAirScrollSpeed(  
    LONG Speed  
);
```

### 인자

Speed

스크롤 속도. 초당 진행 픽셀수.

### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

### 비고

### 예제

```
<ITAP> SetAirScrollSpeed("뉴스하단", 20);  
<OnReceive> SUCCESS AIRSCROLLSPEED
```

### 참조

SetScrollSpeed

---

## ITAP::SetAudioFile

---

로드된 페이지의 오디오/사운드 파일을 설정한다. 단, 송출보드가 오디오/사운드 송출을 지원해야 한다.  
(ex: DSX.mio)

#### 구문

```
HRESULT SetAudioFile(  
    BSTR PageName,  
    BOOL bIn,  
    BSTR AudioFileName  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

bIn

TRUE 일 경우, 장면 IN 송출시에 사용될 오디오 파일이 설정된다.

AudioFileName

오디오 파일의 전체경로명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP> SetAudioFile("페이지명", TRUE, "C:\\wave\\music.wav");  
<OnReceive> SUCCESS AUDIOFILE
```

#### 참조

ITAP::SetAudioEnable

---

## ITAP::SetAudioEnable

---

로드된 페이지에 설정된 오디오를 송출시 끄거나 재생할 수 있다.

#### 구문

```
HRESULT SetAudioEnable(  
    BSTR PageName,  
    BOOL bIn,  
    BOOL bEnable  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

bIn

TRUE 일 경우, 장면 IN 송출시에 설정된 오디오의 재생여부를 결정하게 된다.

bEnable

TRUE 일 경우, 송출시 오디오를 재생하게 된다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP> SetAudioEnable("페이지명", TRUE, TRUE);  
<OnReceive> SUCCESS AUDIOENABLE
```

#### 참조

ITAP::SetAudioFile

---

## ITAP::SetStyleText

---

텍스트 변수객체의 내용과 스타일을 지정한다. 주로, 실시간 크롤을 하기위해 사용한다.

#### 구문

```
HRESULT SetStyleText(  
    BSTR PageName,  
    BSTR ObjName,  
    LONG StyleID,  
    BSTR TextValue  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명. 텍스트 변수객체이어야 한다.

StyleID

자막기의 템플릿 패널에서 스타일탭에 지정된 스타일 번호(1~9).

만약 -1 을 지정하면, 스타일은 바뀌지 않고, 텍스트내용만 설정된다.

TextValue

설정하게 될 텍스트의 내용.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

“크롤1” 텍스트를 스타일 5 로 지정하고, 내용을 설정

```
<ITAP> SetStyleText(“페이지명”, “크롤1”, 5, “뉴스속보”);  
<OnReceive> SUCCESS SETSTYLETEXT “페이지명”  
<ITAP> AddScrollObject(“페이지명”, “크롤1”);  
<OnReceive> SUCCESS ADDSCROLLOBJECT
```

#### 참조

ITAP::AddStyleText , ITAP::AddScrollObject

---

## ITAP::SetImageObject

---

텍스트 변수객체에 이미지를 지정한다. 주로, 실시간 크롤을 하기위해 사용한다.

#### 구문

```
HRESULT SetImageObject(  
    BSTR PageName,  
    BSTR ObjName,  
    LONG ObjectID  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명. 텍스트 변수객체이어야 한다.

ObjectID

자막기의 템플릿 패널에서 오브젝트탭에 지정된 번호(1~9)

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

## 예제

```
<ITAP>      SetImageObject("페이지명", "변수명", 2);  
<OnReceive> SUCCESS SETIMAGEOBJECT
```

## 참조

ITAP::AddImageObject , ITAP::AddScrollObject

---

## ITAP::SetImageObject2

---

텍스트 변수객체에 이미지를 지정한다. 주로, 실시간 크롤을 하기위해 사용한다.

### 구문

```
HRESULT SetImageObject2(  
    BSTR PageName,  
    BSTR ObjName,  
    BSTR FileName  
);
```

### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명. 텍스트 변수객체이어야 한다.

FileName

이미지 파일명.

### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

### 비고

ITAP::SetImageObject 와는 다르게 사용자가 원하는 임의의 이미지 파일명을 지정할 수 있다.

## 예제

```
<ITAP>      SetImageObject2("페이지명", "변수명", "D:/Image Data/Image001.tga");  
<OnReceive> SUCCESS SETIMAGEOBJECT
```

## 참조

ITAP::SetImageObject, ITAP::AddImageObject , ITAP::AddScrollObject

---

## ITAP::AddStyleText

---

텍스트 변수객체의 내용 뒷부분에 새로운 텍스트 내용을 추가한다.( 동시에 스타일도 지정). 주로, 실시간 크롤을 하기위해 사용한다.

### 구문

```
HRESULT AddStyleText(  
    BSTR PageName,  
    BSTR ObjName,  
    LONG StyleID,  
    BSTR TextValue  
);
```

### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명. 텍스트 변수객체이어야 한다.

StyleID

자막기의 템플릿 패널에서 스타일탭에 지정된 스타일 번호(1~9).  
만약 -1 을 지정하면, 스타일은 바뀌지 않고, 텍스트내용만 설정된다.

TextValue  
설정하게 될 텍스트의 내용.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP>      AddStyleText("페이지명", "변수명", 3, "Hello World");  
<OnReceive> SUCCESS ADDSTYLETEXT
```

#### 참조

ITAP::SetStyleText , ITAP::AddScrollObject

---

## ITAP::AddImageObject

---

텍스트 변수객체의 내용 뒷부분에 이미지를 추가한다. 주로, 실시간 크롤을 하기위해 사용한다.

#### 구문

```
HRESULT AddImageObject(  
    BSTR PageName,  
    BSTR ObjName,  
    LONG ObjectID  
);
```

#### 인자

PageName  
로드할 당시, 사용자가 지정한 페이지명.  
ObjName  
페이지 내부의 변수객체명. 텍스트 변수객체이어야 한다.  
ObjectID  
자막기의 템플릿 패널에서 오브젝트탭에 지정된 번호(1~9)

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP>      AddImageObject("페이지명", "변수명", 7);  
<OnReceive> SUCCESS ADDIMAGEOBJECT
```

#### 참조

ITAP::SetImageObject , ITAP::AddScrollObject

---

## ITAP::AddScrollObject

---

현재 송출중인 스크롤 페이지에 존재하는 텍스트 변수객체를 복사생성하여 바로 송출시킨다. 주로, 실시간 크롤을 하기위해 사용한다. AddScrollObject2 와는 다르게, 컷파일에 저장된 레이어로만 추가된다.

#### 구문

```
HRESULT AddScrollObject(  
    BSTR PageName,  
    BSTR ObjName  
);
```

## 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명. 텍스트 변수객체이어야 한다.

## 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

## 비고

## 예제

```
<ITAP> AddScrollObject("페이지명", "변수명");  
<OnReceive> SUCCESS ADDSCROLLOBJECT "페이지명" "변수명" 0 // 0 일경우, Section0 의미함.
```

## 참조

ITAP::AddScrollObject2, ITAP::SetStyleText , ITAP::SetImageObject , ITAP::AddStyleText ,  
ITAP::AddImageObject

---

## ITAP::AddScrollObject2

현재 송출중인 스크롤 페이지에 존재하는 텍스트 변수객체를 복사생성하여 바로 송출시킨다. 주로, 실시간 크롤을 하기위해 사용한다. AddScrollObject 와는 다르게, 실시간으로 원하는 레이어에 추가할 수 있다.

## 구문

```
HRESULT AddScrollObject2(  
    BSTR PageName,  
    BSTR ObjName,  
    eLayer LayerID  
);
```

## 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명. 텍스트 변수객체이어야 한다.

LayerID

송출할 레이어 인덱스.( LAYER\_SECTION\_0 ~ LAYER\_SECTION\_3 )

## 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

## 비고

## 예제

```
<ITAP> AddScrollObject2("페이지명", "변수명", ITAP::LAYER_SECTION_0);  
<OnReceive> SUCCESS ADDSCROLLOBJECT "페이지명" "변수명" 0 // 0 일경우, Section0 의미함.
```

## 참조

ITAP::AddScrollObject, ITAP::SetStyleText , ITAP::SetImageObject , ITAP::AddStyleText ,  
ITAP::AddImageObject

---

## ITAP::CreateImageFile

현재 송출중인 페이지를 캡처하여, 이미지파일로 만들어 사용자가 원하는 위치에 저장한다.

## 구문

```
HRESULT CreateImageFile(  
    BSTR PageName,
```

```

    LONG ImageWidth,
    LONG ImageHeight,
    BSTR ImagePathName
);

```

#### 인자

PageName  
 로드할 당시, 사용자가 지정한 페이지명.  
 ImageWidth / ImageHeight  
 생성될 이미지파일의 크기 설정값. 픽셀 단위.  
 ImagePathName  
 생성될 이미지 파일의 전체경로 파일명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

현재 페이지를 cpature.jpg 파일로 저장하기.

```

<ITAP>      CreateImageFile("페이지명", 640, 480, "C:WimageWcapture.jpg");
<OnReceive> SUCCESS CREATEIMAGE "페이지명" "C:WimageWcapture.jpg");

```

#### 참조

---

## ITAP::SetObjLine

---

Line 오브젝트를 지정한 위치에 생성한다.

#### 구문

```

HRESULT SetObjLine(
    LPCTSTR PageName,
    LPCTSTR ObjName,
    bool b3D,
    int Thick,
    Int Depth,
    LPCTSTR strCoordinates
);

```

#### 인자

PageName  
 로드할 당시, 사용자가 지정한 페이지명.  
 ObjName  
 페이지 내부의 변수객체명.( 이 객체는 'Path' 객체여야 한다)  
 B3D  
 2D or 3D 택1.  
 Thick  
 라인의 두께.  
 Depth  
 3D일 경우, 라인의 Depth 두께.  
 strCoordinates  
 라인의 제어점 좌표. 원하는 만큼 생성.

#### 반환값

SUCCESS(FAILURE) OBJLINE "변수명": 명령수행 성공(실패).

#### 비고

쿼리없음



#### 예제

```
<ITAP> SetObjLine("PageName", "ObjName", true, 4, 4, "0,0 100,100 200,200 300,300");  
<Tornado3D> SUCCESS OBJLINE "라인1"
```

---

### ITAP::QueryLoad

---

PageName 인자가 NULL 일 경우, 현재 자막기에 로드해 둔 페이지들의 페이지명들 전체를 요청한다.  
PageName 이 있다면, 해당 페이지의 로드 상태를 요청한다.

#### 구문

```
HRESULT QueryLoad(  
    BSTR PageName  
);
```

#### 인자

PageName  
로드할 당시, 사용자가 지정한 페이지명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
로드된 전체 페이지들을 요청.  
<ITAP> QueryLoad(NULL);  
<OnReceive> SUCCESS QUERY LOAD 페이지명1 페이지명2 ...  
특정 페이지의 로드상태를 요청.  
<ITAP> QueryLoad("페이지명");  
<OnReceive> SUCCESS QUERY LOAD LOADED 페이지명
```

#### 참조

---

### ITAP::QueryPageMode

---

현재 페이지 모드를 요청한다.  
페이지 모드는 STILL, CRAWL, ROLL, ANIMATION 이 있다.

#### 구문

```
HRESULT QueryPageMode(  
    BSTR PageName  
);
```

#### 인자

PageName  
로드할 당시, 사용자가 지정한 페이지명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP> QueryPageMode("페이지명");  
<OnReceive> SUCCESS QUERY PAGEMODE "페이지명" ANIMATION
```

#### 참조

---

### ITAP::QueryPageAlias

---

현재 페이지의 페이지명을 포함하여 모든 객체명들을 구한다.

#### 구문

```
HRESULT QueryPageAlias(  
    BSTR PageName  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP>      QueryPageAlias("페이지명");  
<OnReceive> SUCCESS QUERY PAGEALIAS "페이지명" 오브젝트명1 오브젝트명2 ...
```

#### 참조

---

## ITAP::QueryAliasAll

---

현재 페이지의 페이지명을 포함하여 모든 객체명들을 구한다.

#### 구문

```
HRESULT QueryAliasAll(  
    BSTR PageName  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP>      QueryAliasAll("페이지명");  
<OnReceive> SUCCESS QUERY ALIASALL "페이지명" 변수명 종류 "변수값" 변수명 종류 "변수값"  
" ...  
- [ 변수명 종류 "변수값" ] 의 형태로 반복.  
- 종류 : T 혹은 I 문자가 들어가며, Text 인지 Image 인지를 식별할수 있도록 한다.
```

#### 참조

---

## ITAP::QueryObjValue

---

현재 페이지내의 특정 변수객체의 값을 구한다.

텍스트 변수객체의 경우, 텍스트의 내용을 응답하며, 도형 변수객체의 경우, 텍스처 파일명을 응답한다.

#### 구문

```
HRESULT QueryObjValue(  
    BSTR PageName,  
    BSTR ObjName  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

OBJVALUEEX 명령어로 응답됨에 주의.

#### 예제

```
<ITAP> QueryObjValue("페이지명", "변수명");  
<OnReceive> SUCCESS QUERY OBJVALUEEX "페이지명" "변수명" 변수값
```

#### 참조

---

### ITAP::QueryObjAttr

---

현재 페이지내의 특정 변수객체의 속성값 전체를 가져온다.

#### 구문

```
HRESULT QueryObjAttr(  
    BSTR PageName,  
    BSTR ObjName  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

ObjName

페이지 내부의 변수객체명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP> QueryObjAttr("페이지명", "변수명");  
<OnReceive> SUCCESS QUERY OBJATTR Type 2D Width 250 Height 50 Depth 0 Position  
996,790,0 Pivot 0,0,0 Rotation 0,0,0 Slant 0,0 Scale 100,100,100
```

#### 참조

---

### ITAP::QueryAudioFile

---

현재 페이지에 설정된 오디오/사운드 파일명을 요청한다.

#### 구문

```
HRESULT QueryAudioFile(  
    BSTR PageName,  
    BOOL bln  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

bln

TRUE 일 경우, 장면 IN 송출시에 설정된 오디오 파일명을 응답한다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP> QueryAudioFile("페이지명", TRUE);  
<OnReceive> SUCCESS QUERY AUDIOFILE "페이지명" IN "오디오파일명"
```

#### 참조

---

## ITAP::QueryAudioEnable

---

현재 페이지에 설정된 오디오/사운드 파일의 송출시, 재생여부를 요청한다.

#### 구문

```
HRESULT QueryAudioEnable(  
    BSTR PageName,  
    BOOL bln  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

bln

TRUE 일 경우, 장면 IN 송출시에 설정된 오디오의 재생여부를 요청한다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP> QueryAudioEnable("페이지명", TRUE);  
<OnReceive> SUCCESS QUERY AUDIOENABLE "페이지명" IN ON
```

#### 참조

---

## ITAP::QueryScrollSpeed

---

로드되어 있는 특정 스크롤 페이지의 스크롤 속도를 찾는다.

스크롤 속도는 정수이며, 초당 스크롤진행 픽셀수를 의미한다.

#### 구문

```
HRESULT QueryScrollSpeed(  
    BSTR PageName  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

```
<ITAP> QueryScrollSpeed("페이지명");
```

```
<OnReceive> SUCCESS QUERY SCROLLSPEED “페이지명” 10
```

## 참조

---

### ITAP::QueryAirScrollMargin

---

현재 송출중인 스크롤 페이지(크롤/롤)에서 최종스크롤 객체가 화면가장자리에서부터 얼마나 떨어져 있는지를 구한다.

주로, 뉴스크롤등에서 실시간 크롤객체를 추가하기 위한 타이밍을 찾기위해 사용.

#### 구문

```
HRESULT QueryAirScrollMargin(  
    eLayer Layer  
);
```

#### 인자

Layer

스크롤페이지가 송출중인 레이어.

(가능범위: LAYER\_SECTION\_0 ~ LAYER\_SECTION\_3) - eLayer 참조.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

스크롤중인 [레이어 2] 에서 스크롤 마진을 요청하여 가져온다.

```
<ITAP> QueryAirScrollMargin(TAPI::LAYER_SECTION_2)  
<OnReceive> SUCCESS QUERY SCROLLMARGIN 2 567
```

- 2 : 레이어에 해당하는 인덱스.
  - 0 : LAYER\_SECTION\_0
  - 1 : LAYER\_SECTION\_1
  - 2 : LAYER\_SECTION\_2
  - 3 : LAYER\_SECTION\_3
- 567 : 최종 스크롤객체의 송출영역 가장자리로부터의 거리.  
만약, 스크롤이 끝난상태라면, -123456789 가 리턴되며,  
에러발생시, -987654321 이 리턴된다.  
스크롤이 끝났을 경우, 다시 Play 를 해주어야 한다.

## 참조

---

### ITAP::QueryBackground

---

현재 선택된 페이지의 배경정보를 요청한다.

#### 구문

```
HRESULT QueryBackground();
```

#### 인자

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

<스코어장면>페이지의 배경에 avi 동영상 지정하기.

```
<ITAP> BACKGRND “스코어장면” “C:\Wdata\배경동영상.avi”  
<OnReceive> SUCCESS QUERY BACKGRND BG_Begin 0 BG_End 9999 BG_FileName  
"C:\Wdata\배경동영상.avi"
```

쿼리되어 오는 값은 아래와같은 키워드와 값이 짝을 이루어서 날라온다.

BG\_Begin : 동영상 첫프레임 ( 0 ~ 정수)  
BG\_End : 동영상 끝프레임 ( 0 ~ 정수)  
BG\_FileName : 파일명

#### 참조

---

### ITAP::QueryPageEffect

---

현재 페이지에 설정된 페이지효과에 대한 정보를 요청한다.

정보는 프레임수, 효과종류, 효과의 하부옵션, 순서로 응답한다. 하부옵션의 경우는 키워드와 함께 쌍을 이루어서 구성하게 된다.

#### 구문

```
HRESULT QueryPageEffect(  
    BSTR PageName,  
    BOOL bln  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

bln

TRUE 일 경우, 장면 IN 송출시에 설정된 페이지효과에 대한 정보를 요청한다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

**ex1)** <스코어장면> 페이지의 IN 효과를 알아본다. (IN 효과가 없는(컷전환)일 경우)

```
<ITAP> QueryPageEffect("스코어장면", IN);  
<OnReceive> SUCCESS QUERY PAGEEFFECT Frame 30
```

**ex2)** IN 효과가 WIPE 일 경우.

```
<ITAP> QueryPageEffect("스코어장면", IN);  
<OnReceive> SUCCESS QUERY PAGEEFFECT Frame 30 WIPE Shape Circle Position Up
```

#### 참조

---

### ITAP::QueryPageEffectEx

---

현재 페이지에 설정된 페이지효과에 대한 정보를 요청한다.

정보는 "페이지명", IN/OUT 종류, 효과종류번호, 프레임수, 순서로 응답한다.

#### 구문

```
HRESULT QueryPageEffectEx(  
    BSTR PageName,  
    BOOL bln  
);
```

#### 인자

PageName

로드할 당시, 사용자가 지정한 페이지명.

bln

TRUE 일 경우, 장면 IN 송출시에 설정된 페이지효과에 대한 정보를 요청한다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

**ex1)** <스코어장면>페이지의 IN 효과를 알아본다.

```
<ITAP> QueryPageEffectEx("스코어장면", TRUE);
```

```
<OnReceive> SUCCESS QUERY PAGEEFFECTEX "스코어장면" IN 2 60
```

효과번호: 효과명을 번호로 인덱스화한 값으로서 아래와 같다.

- 0 : CUT
- 1 : WIPE
- 2 : PUSH
- 3 : DVE
- 4 : CURL
- 5 : PARTICLE
- 6 : RIPPLE
- 7 : ORGANIC
- 8 : DISTORTION
- 9 : BLINK
- 10 : MASK

## 6.1.2 ITAPEventHandler

자막기측에서 클라이언트로 응답하는 메시지를 받아서 처리하기 위한 핸들러.

예를 들어, 클라이언트가 "PLAY" 라는 명령을 자막기로 보냈을 경우, 자막기는 송출을 시작함과 동시에, "SUCCESS PLAY" 라고 클라이언트로 응답하게 된다.

---

### ITAPEventHandler::OnConnect

---

자막기와 네트워크 연결 성공시, 호출된다.

구문

```
HRESULT OnConnect();
```

인자

반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

비고

예제

참조

---

### ITAPEventHandler::OnReceive

---

사용자 클라이언트가 자막기로부터 메시지를 수신할 경우, 호출된다. 유니코드의 문자열 데이터가 수신된다.

구문

```
HRESULT OnReceive(  
    BSTR pData  
);
```

인자

pData

유니코드 문자열.

반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

비고

예제

참조

샘플 코드 참조.

---

### ITAPEventHandler::OnClose

---

자막기측과의 네트워크 연결이 종료될때, 호출된다.

구문

```
HRESULT OnClose();
```

인자

반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

비고

예제



### 6.1.3 ITAParser

자막기측에서 보내온 메시지를 TAP 프로토콜 규칙에 맞춰서 쉽게 파싱하기 위해 제공되는 인터페이스이다.

---

#### ITAParser::Parse

---

자막기측에서 보내온 메시지를 파싱하기 위해 가장 먼저 한번 호출해야 한다.

##### 구문

```
HRESULT Parse(  
    BSTR Text,  
    BOOL* ret  
);
```

##### 인자

Text

ITAPEventHandler::OnReceive 에서 수신된 유니코드 문자열 데이터.

ret

파싱 성공여부를 반환한다.

##### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

##### 비고

##### 예제

TAP 샘플 프로그램 참고.

##### 참조

---

#### ITAParser::GetCommand

---

ITAParser::Parse 함수로 파싱 후, 주 명령어를 찾는다.

##### 구문

```
HRESULT GetCommand(  
    eTAPCommand* Command  
);
```

##### 인자

Command

TAPI::eTAPCommand 이며, 명령어를 판단하여 분기한다.

##### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

##### 비고

##### 예제

TAP 샘플 프로그램 참고.

##### 참조

---

#### ITAParser::GetNextCommand

---

ITAParser::GetCommand 함수로 주 명령어에 따라 분기한 후, 주 명령어가 QUERY 일 경우, 하위 명

령어를 찾는다.

#### 구문

```
HRESULT GetNextCommand(  
    eTAPCommand* Command  
);
```

#### 인자

Command

TAPI::eTAPCommand 이며, 명령어를 판단하여 분기한다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

TAP 샘플 프로그램 참고.

#### 참조

---

## ITAPParser::GetNextString

---

ITAPParser::GetNextCommand 함수로 하위 명령어에 따라 분기한 후, 프로토콜 규칙에 따라 필요한 문자열을 찾는다.

비교할 것은, 따옴표(" ") 로 묶이지 않은 문자열만 찾기가 가능하다.

따옴표(" ") 로 묶여진 문자열을 찾기 위해서는 ITAPParser::GetQuotationString 으로 찾기가 가능하다.

#### 구문

```
HRESULT GetNextString(  
    BSTR* str  
);
```

#### 인자

str

찾고자 하는 문자열이 반환된다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

TAP 샘플 프로그램 참고.

#### 참조

ITAPParser::GetQuotationString

---

## ITAPParser::GetNextQuotationString

---

ITAPParser::GetNextCommand 함수로 하위 명령어에 따라 분기한 후, 프로토콜 규칙에 따라 필요한 문자열을 찾는다.

비교할 것은, 따옴표(" ")로 묶여진 문자열만 찾기가 가능하다.

cf) ITAPParser::GetNextString

#### 구문

```
HRESULT GetNextQuotationString(  
    BSTR* str  
);
```

#### 인자

str

찾고자 하는 문자열이 반환된다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

TAP 샘플 프로그램 참고.

#### 참조

ITAPParser::GetNextString

---

## ITAPParser::GetNextOptionString

---

ITAPParser::GetNextCommand 함수로 하위 명령어에 따라 분기한 후, 프로토콜 규칙에 따라 필요한 문자열을 찾는다.

수신된 문자열 데이터에서 다음에 와야할 정보가 각종 옵션정보라면, 하나의 옵션 정보를 찾는다.

#### 구문

```
HRESULT GetNextOptionString(  
    BSTR* str  
);
```

#### 인자

str

찾고자 하는 문자열이 반환된다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

TAP 샘플 프로그램 참고.

#### 참조

---

## ITAPParser::GetNextStringAll

---

수신된 문자열 데이터에서 다음에 오는 문자열을 모두 한번에 찾는다.

다음에 올 문자열이 명령어나 키워드가 아니고, 더이상 뒤에 붙을 정보가 없다고 판단될 경우, 주로 사용한다.

#### 구문

```
HRESULT GetNextStringAll(  
    BSTR* str  
);
```

#### 인자

str

찾고자 하는 문자열이 반환된다.

#### 반환값

성공할 경우, T3D\_TRUE 를 리턴하고, 실패할 경우, T3D\_FALSE 를 반환한다.

#### 비고

#### 예제

TAP 샘플 프로그램 참고.

#### 참조

## 6.2 열거자(Enumerated Type)

TAPI 에서 사용되는 Enumerated Type 에 대한 Information.

---

### eTAPCommand

---

TAP 의 명령어.

```
enum eTAPCommand{ // 커맨드
    TAP_COMMAND_Play,
    TAP_COMMAND_Pause,
    TAP_COMMAND_Out,
    TAP_COMMAND_Stop,
    TAP_COMMAND_Prepair,

    TAP_COMMAND_StopPreview,

    TAP_COMMAND_Section_Prepair,
    TAP_COMMAND_Section_In,
    TAP_COMMAND_Section_Out,
    TAP_COMMAND_Stop_All_Sections,

    TAP_COMMAND_TopLayer_In,
    TAP_COMMAND_TopLayer_Out,
    TAP_COMMAND_Default_Out,

    TAP_COMMAND_Load_Project,
    TAP_COMMAND_Save_Project,
    TAP_COMMAND_UnloadAll,
    TAP_COMMAND_Load,
    TAP_COMMAND_LoadEx,
    TAP_COMMAND_LoadEx_2,           // 퀴리시, "페이지명", Tcf파일명" ... 반복되는 형태로
리턴.
    TAP_COMMAND_Insert,           // Index 로 페이지 로드.
    TAP_COMMAND_Unload,           // UNLOAD "페이지명"   UNLOAD ALL 하면 모든 페
이지를 UNLOAD 함

    TAP_COMMAND_CreateFile, // CREATEFILE "페이지명" "생성될 파일의 FullPathName"
    TAP_COMMAND_PageDefault, // PAGEDEFAULT "페이지명"
    TAP_COMMAND_BackGrnd,    // BACKGRND "페이지명" "파일명"
    TAP_COMMAND_ObjValue,    // OBJVALUE "오브젝트명" SHOW | OBJVALUE "오
브젝트명" HIDE ...
    TAP_COMMAND_SetStyleText, // SETSTYLETEXT "페이지명" "Text오브젝트명" "StyleID"
"Text내용"
    TAP_COMMAND_AddStyleText, // ADDSTYLETEXT "페이지명" "Text오브젝트명"
"StyleID" "Text내용"
    TAP_COMMAND_SetImageObject, // 이미지텍스트 시작.
    TAP_COMMAND_AddImageObject, // 이미지삽입
    TAP_COMMAND_AddScrollObject, // 연속스크롤 실시간 객체 삽입.
    TAP_COMMAND_ObjAttr,
    TAP_COMMAND_PageEffect,
```

```

TAP_COMMAND_PageEffectEx,
TAP_COMMAND_Query,

TAP_COMMAND_Hello,
TAP_COMMAND_HeartBeat,
TAP_COMMAND_Bye,
TAP_COMMAND_Hello_Connected, // 서버가 보내는 "HELLO"에 대한 무조건 반응
TAP_COMMAND_Hello_Check_Ok, // 서버가 보내는 "HELLO CHECK"에 대한 무조건 반응

TAP_COMMAND_AudioEnable,
TAP_COMMAND_AudioFile,

TAP_COMMAND_ScrollSpeed,

TAP_COMMAND_AirScrollSpeed,
TAP_COMMAND_AirSectionScrollSpeed,

TAP_COMMAND_Pause_Add, // PAUSEADD "페이지명" Frame Delay (Auto / Manual)
TAP_COMMAND_Pause_Delete, // PAUSEDELETE "페이지명" Frame

// 쿼리 전용. -----
TAP_COMMAND_Page, // 쿼리시, 변수명 변수명 ...
TAP_COMMAND_PageAlias, // 쿼리시, "페이지명" 변수명 변수명 ...
TAP_COMMAND_ObjValueEx, // 쿼리시 "페이지명" "변수명" "변수값" 형태로 전송.
TAP_COMMAND_PageMode,
TAP_COMMAND_ScrollMargin,
TAP_COMMAND_Section_Play,
TAP_COMMAND_TextAll,
TAP_COMMAND_Error,
//-----

TAP_COMMAND_Continue, // '|'
TAP_COMMAND_End, // '^'

TAP_COMMAND_NoMatch,
}

```

---

## eLayer

---

페이지의 레이어 정보를 나타낸다.

송출시, Base → Top → Section0 → Section1 → Section2 → Section3 순서로 아래에 깔린다.

(그러므로, Top 이 아니라, Section3 레이어가 최상위에 표출됨에 주의.)

```

enum eLayer{
    LAYER_BASE,
    LAYER_TOP,
    LAYER_SECTION_0,
    LAYER_SECTION_1,
    LAYER_SECTION_2,
    LAYER_SECTION_3,
}

```

---

## eEffectType

---

페이지 효과의 종류를 나타낸다.

```
enum eEffectType{
    EFFECT_CUT,
    EFFECT_WIPE,
    EFFECT_PUSH,
    EFFECT_DVE,
    EFFECT_CURL,
    EFFECT_PARTICLE,
    EFFECT_RIPPLE,
    EFFECT_ORGANIC,
    EFFECT_DISTORTION,
    EFFECT_BLINK,
    EFFECT_MASK,

    EFFECT_COUNT,
}
```

---

## eTAPAttr

---

```
enum eTAPAttr{
    TAP_ATTR_Value = TAP_COMMAND_NoMatch + 1,
    TAP_ATTR_Type,
    TAP_ATTR_Color,
    TAP_ATTR_Color_Start, // 애니메이션 객체용.(시작키프레임값 설정) 2008.2.13 추가. <===
    현재, 컬러는 자막기 구현안된 기능임(3D객체경우). 사용자문서에서 제외.
    TAP_ATTR_Width,
    TAP_ATTR_Height,
    TAP_ATTR_Depth,
    TAP_ATTR_Position,
    TAP_ATTR_Pivot,
    TAP_ATTR_BarWidth, // 바 Wipe 효과용
    TAP_ATTR_Rotation,
    TAP_ATTR_Scale,
    TAP_ATTR_Scale_Start, // 애니메이션 객체용.(시작키프레임값 설정) 2008.2.13 추가.
    TAP_ATTR_Slant,

    TAP_ATTR_Text_Format,

    TAP_ATTR_TextType,
    TAP_ATTR_TextType_Text,
    TAP_ATTR_TextType_Clock,
    TAP_ATTR_TextType_Timer,
    TAP_ATTR_TextType_Counter,

    TAP_ATTR_TimerBegin,
    TAP_ATTR_TimerEnd,
    TAP_ATTR_TimerDuration,
    TAP_ATTR_TimerUpdate, // 업데이트간격
}
```

```

// SeqTga 파일정보
TAP_ATTR_Seq_Start, // 송출시작 파일 번호.
TAP_ATTR_Seq_Finish,
TAP_ATTR_Seq_Loop,

TAP_ATTR_Circle_Angle, // 애니메이션 : 파이그래프 지원.

TAP_ATTR_Effect, // 객체의 Effect

TAP_ATTR_NoMatch,
}

```

---

## eTAPBg

---

```

enum eTAPBg{ // TAP_QUERY_BACKGRND 에 대한 무조건 리턴정보.
    TAP_BG_FileName = TAP_ATTR_NoMatch + 1, // 배경동영상 정보 쿼리.
    TAP_BG_Begin,
    TAP_BG_End,
    TAP_BG_Loop,
    TAP_BG_Start,
    TAP_BG_Finish,
    TAP_BG_NoMatch,
}

```

---

## eTAPEffect

---

Fade 설정은 없지만, Organic 으로 Fade 효과가 가능하다. 단, bUseMatte = Off 여야 한다.

```

enum eTAPEffect{
    TAP_EFFECT_Cut = TAP_BG_NoMatch + 1,
    TAP_EFFECT_Wipe,
    TAP_EFFECT_Push,
    TAP_EFFECT_Dve,
    TAP_EFFECT_Curl,
    TAP_EFFECT_Particle,
    TAP_EFFECT_Ripple,
    TAP_EFFECT_Organic,
    TAP_EFFECT_Distortion,
    TAP_EFFECT_Blink,
    TAP_EFFECT_Mask,

    TAP_EFFECT_Frame, // 효과 프레임수

    TAP_EFFECT_NoMatch,
}

```

---

## eTAPEffectWipe

---

```

typedef enum eTAPEffectWipe{

```

```

    TAP_EFFECT_Wipe_Shape = TAP_EFFECT_NoMatch + 1,
    TAP_EFFECT_Wipe_Shape_Divide,
    TAP_EFFECT_Wipe_Shape_Rectangle,
    TAP_EFFECT_Wipe_Shape_Circle,
    TAP_EFFECT_Wipe_Shape_Fan,
    TAP_EFFECT_Wipe_Position,
    TAP_EFFECT_Wipe_Position_Up,
    TAP_EFFECT_Wipe_Position_Down,
    TAP_EFFECT_Wipe_Position_Left,
    TAP_EFFECT_Wipe_Position_Right,
    TAP_EFFECT_Wipe_Position_UpLeft,
    TAP_EFFECT_Wipe_Position_UpRight,
    TAP_EFFECT_Wipe_Position_DownLeft,
    TAP_EFFECT_Wipe_Position_DownRight,
    TAP_EFFECT_Wipe_Position_Center,

    TAP_EFFECT_Wipe_NoMatch,
}

```

---

### **eTAPEffectPush**

---

```

enum eTAPEffectPush{
    TAP_EFFECT_Push_Direction = TAP_EFFECT_Wipe_NoMatch + 1,
    TAP_EFFECT_Push_Direction_Up,
    TAP_EFFECT_Push_Direction_Down,
    TAP_EFFECT_Push_Direction_Left,
    TAP_EFFECT_Push_Direction_Right,
    TAP_EFFECT_Push_Direction_UpLeft,
    TAP_EFFECT_Push_Direction_UpRight,
    TAP_EFFECT_Push_Direction_DownLeft,
    TAP_EFFECT_Push_Direction_DownRight,

    TAP_EFFECT_Push_NoMatch,
};

```

---

### **eTAPEffectCurl**

---

```

enum eTAPEffectCurl{
    TAP_EFFECT_Curl_Radius = TAP_EFFECT_Push_NoMatch + 1,
    TAP_EFFECT_Curl_Angle,

    TAP_EFFECT_Curl_NoMatch,
}

```

---

### **eTAPEffectDve**

---

```

enum eTAPEffectDve{
    TAP_EFFECT_Dve_Position = TAP_EFFECT_Curl_NoMatch + 1,
    TAP_EFFECT_Dve_Center,
    TAP_EFFECT_Dve_Scale,
}

```



```

        TAP_EFFECT_Dve_Rotation,

        TAP_EFFECT_Dve_NoMatch,
    }

```

---

## eTAPEffectParticle

---

```

enum eTAPEffectParticle{ // Particle
    TAP_EFFECT_Particle_Shape = TAP_EFFECT_Dve_NoMatch + 1,
    TAP_EFFECT_Particle_Shape_Tile,
    TAP_EFFECT_Particle_Shape_Random,
    TAP_EFFECT_Particle_Shape_Radial,
    TAP_EFFECT_Particle_Order,
    TAP_EFFECT_Particle_Order_AtOnce,
    TAP_EFFECT_Particle_Order_Random,
    TAP_EFFECT_Particle_Order_Top,
    TAP_EFFECT_Particle_Order_Bottom,
    TAP_EFFECT_Particle_Order_Left,
    TAP_EFFECT_Particle_Order_Right,
    TAP_EFFECT_Particle_Order_TopLeft,
    TAP_EFFECT_Particle_Order_TopRight,
    TAP_EFFECT_Particle_Order_BottomLeft,
    TAP_EFFECT_Particle_Order_BottomRight,
    TAP_EFFECT_Particle_Order_PointCenter,
    TAP_EFFECT_Particle_Order_PointSide,
    TAP_EFFECT_Particle_Order_SnakeTop,
    TAP_EFFECT_Particle_Order_SnakeBottom,
    TAP_EFFECT_Particle_Order_SnakeLeft,
    TAP_EFFECT_Particle_Order_SnakeRight,
    TAP_EFFECT_Particle_Order_SnakeCenter,
    TAP_EFFECT_Particle_Order_SnakeSide,
    TAP_EFFECT_Particle_Motion,
    TAP_EFFECT_Particle_Motion_Scatter2D,
    TAP_EFFECT_Particle_Motion_Scatter3D,
    TAP_EFFECT_Particle_Motion_Converge,
    TAP_EFFECT_Particle_Motion_Directional,
    TAP_EFFECT_Particle_DIVIDEY,
    TAP_EFFECT_Particle_OrderX,
    TAP_EFFECT_Particle_OrderY,
    TAP_EFFECT_Particle_ConvergeX,
    TAP_EFFECT_Particle_ConvergeY,
    TAP_EFFECT_Particle_Direction,

    TAP_EFFECT_Particle_NoMatch,
}

```

---

## eTAPEffectRipple

---

```

enum eTAPEffectRipple{ // Ripple
    TAP_EFFECT_Ripple_Shape = TAP_EFFECT_Particle_NoMatch + 1,
    TAP_EFFECT_Ripple_Shape_Water,
}

```

```
TAP_EFFECT_Ripple_Shape_Flag,  
TAP_EFFECT_Ripple_Frequency,  
TAP_EFFECT_Ripple_Amplitude,  
TAP_EFFECT_Ripple-OriginX,  
TAP_EFFECT_Ripple-OriginY,  
TAP_EFFECT_Ripple_Slant,  
  
TAP_EFFECT_Ripple_NoMatch,  
}
```

---

### **eTAPEffectOrganic**

---

```
enum eTAPEffectOrganic{    // Organic  
    TAP_EFFECT_Organic_UseMatte = TAP_EFFECT_Ripple_NoMatch + 1,  
    TAP_EFFECT_Organic_FILE,  
  
    TAP_EFFECT_Organic_NoMatch,  
}
```

---

### **eTAPEffectDistortion**

---

```
enum eTAPEffectDistortion{    // Distortion  
    TAP_EFFECT_Distortion_Height = TAP_EFFECT_Organic_NoMatch + 1,  
    TAP_EFFECT_Distortion_File,  
  
    TAP_EFFECT_Distortion_NoMatch,  
}
```

## 7 C++ 샘플 코드

### 7.1 기본 설정

프로그램 초기 작업.

```
#include "TAPEventHandler.h" // 제공되는 헤더 파일.
```

전역 선언.

```
TAPI::ITAPPtr      pTAP;  
CMyTAPHandler*    pMyHandler;  
TAPI::ITAPPParserPtr pTAPPParser;
```

초기화.

```
Colnitalize(0);  
  
pTAP.CreateInstance(__uuidof(TAPI::TAP));  
pTAPPParser.CreateInstance(__uuidof(TAPI::TAPPParser));  
  
pMyHandler = new CMyTAPHandler;
```

프로그램 종료 작업.

```
pTAP->Destroy();  
pMyHandler->Release();
```

### 7.2 템플릿 로드/송출 (TAPPlay)

자막기에 연결.

```
void CTAPPlayDlg::OnBnClickedConnect()  
{  
    CString strIPAddress, strPort;  
    GetDlgItem(IDC_IP)->GetWindowText(strIPAddress);  
    GetDlgItem(IDC_PORT)->GetWindowText(strPort);  
    int Port = _ttoi(strPort);  
    theApp.pTAP->Connect(TRUE, strIPAddress.AllocSysString(), Port, 1000,  
theApp.pMyHandler);  
}
```

템플릿 로드.

```
void CTAPPlayDlg::OnBnClickedLoad()  
{  
    CString TCFFFileName, PageName;  
    GetDlgItem(IDC_TCF_FILENAME)->GetWindowText(TCFFFileName);  
    GetDlgItem(IDC_PAGE_NAME)->GetWindowText(PageName);  
    theApp.pTAP->LoadPage(TCFFFileName.AllocSysString(), PageName.AllocSysString());  
}
```

```
}
```

#### 템플릿 송출준비.

```
void CTAPPlayDlg::OnBnClickedPrepare()
{
    CString PageName;
    GetDlgItem(IDC_PAGE_NAME)->GetWindowText(PageName);
    theApp.pTAP->PreparePage(PageName.AllocSysString(), TAPI.LAYER_BASE, TRUE);
}

```

#### 템플릿 송출시작.

```
void CTAPPlayDlg::OnBnClickedPlay()
{
    theApp.pTAP->Play(TAPI.LAYER_BASE);
}

```

#### 송출화면 지우기.

```
void CTAPPlayDlg::OnBnClickedClear()
{
    theApp.pTAP->Stop();
}

```

#### 네트워크 수신부.

```
HRESULT STDMETHODCALLTYPE CMyTAPHandler::raw_OnReceive(const BSTR pData){
    CString strReceive = LPCTSTR(pData);

    if(!theApp.m_pMainWnd)
        return S_FALSE;

    theApp.OnReceive(strReceive);

    return S_OK;
}

```

```
void CTAPPlayDlg::OnReceive(CString& strRecv){ // Receive Network Data.

    if(!theApp.pTAPParser->Parse(strRecv.AllocSysString()))
        return;

    TAPI::eTAPCommand Command = theApp.pTAPParser->GetCommand();
    switch(Command){
    case TAPI::TAP_COMMAND_Hello_Connected:
        {
        }
        break;
    case TAPI::TAP_COMMAND_Load:
        {
        }
        break;
    case TAPI::TAP_COMMAND_Query:
        {
            CString strWord;
            TAPI::eTAPCommand Command2 = theApp.pTAPParser->GetNextCommand();
            switch(Command2){
            case TAPI::TAP_COMMAND_Load: // SUCCESS QUERY LOAD
                {
                }
                break;
            case TAPI::TAP_COMMAND_Page: // SUCCESS QUERY PAGE
                {
                }
            }
        }
    }
}

```

```

        }
        break;
    case TAPI::TAP_COMMAND_ObjValue: // SUCCESS QUERY OBJVALUE 발견
    {
    }
    break;
    case TAPI::TAP_COMMAND_ObjAttr: // SUCCESS QUERY OBJATTR 발견
    {

    }
    break;
    case TAPI::TAP_COMMAND_ScrollMargin:
    {
    }
    break;
    default:
        break;
    }
}
break;
default:
    break;
}
}

```

## 8 C# 샘플 코드

### 8.1 기본 설정

프로그램 초기 작업.

Project -> 참조 추가 -> COM -> TAPI Type Library

전역 선언

```
protected TAPI.TAP TAP;
protected TAPI.TAPParser Parser;
protected cMyHandler MyHandler;

protected string TcfFileName;
```

초기화

```
InitializeComponent();

TAP = new TAPI.TAP();
Parser = new TAPI.TAPParser();

try
{
    MyHandler = new cMyHandler();
}
catch (Exception e)
{
    System.Windows.Forms.MessageBox.Show(e.Message);
}
```

프로그램 종료 작업

```
TAP.Destroy();
```

### 8.2 템플릿 로드/송출 ( TAPNPlay )

자막기에 연결.

```
TAP.Connect(1, ServerIP.Text, Convert.ToInt32(ServerPort.Text), 0, MyHandler);
```

템플릿 로드.

```
private void Load_Click(object sender, EventArgs e)
{
    TAP.LoadPage(TcfFileNameEdit.Text, PageNameEdit.Text);
}
```

템플릿 송출준비.

```
private void Prepare_Click(object sender, EventArgs e)
{
    TAP.PreparePage(PageNameEdit.Text, TAPI.eLayer.LAYER_BASE, 1);
}
```

```
}
```

#### 템플릿 송출시작.

```
private void Play_Click(object sender, EventArgs e)
{
    TAP.Play(TAPI.eLayer.LAYER_BASE);
}
```

#### 송출화면 지우기.

```
private void Clear_Click(object sender, EventArgs e)
{
    TAP.Stop();
}
```

#### 네트워크 수신부.

```
public void OnReceive(string pData)
{
    TAPNPlay theApp = (TAPNPlay)System.Windows.Forms.Form.ActiveForm;

    theApp.OnReceive(pData);
}
```

```
public void OnReceive(string text)
{
    string Message = text + "WrWn";
    MessageView.AppendText(Message);

    if(Parser.Parse(text) == 0)
        return;

    TAPI.eTAPCommand command = Parser.GetCommand();
    switch(command){
        case TAPI.eTAPCommand.TAP_COMMAND_Query:
        {
            TAPI.eTAPCommand command2 = Parser.GetNextCommand();
            switch (command2){
                case TAPI.eTAPCommand.TAP_COMMAND_Load:
                    // SUCCESS QUERY LOAD
                    break;
                case TAPI.eTAPCommand.TAP_COMMAND_ObjValue:
                    // SUCCESS QUERY OBJVALUE
                    break;
                case TAPI.eTAPCommand.TAP_COMMAND_ScrollMargin:
                    // SUCCESS QUERY SCROLLMARGIN
                    break;
            }
        }
        break;
    }
}
```