

# 웹한글 기안기 API

## <목 차>

<b>1. 웹한글 기안기 구성/구조/지원포맷 .....</b>	<b>1</b>
1.1. 서버 구성 .....	1
1.2. 구조 .....	2
1.3. 호출 구조 .....	3
1.4. 지원 포맷 .....	4
<b>2. 한글 컨트롤(ActiveX) 대비 API 변경사항 .....</b>	<b>5</b>
2.1. 제외된 API .....	5
2.2. 사용법이 변경된 API .....	5
<b>3. 웹한글 기안기 API 기본 사용법 .....</b>	<b>8</b>
3.1. 초기 설정 .....	8
3.2. 컨트롤 객체 생성 .....	9
3.3. HwpCtrl Object(Action, ParameterSet) 사용 예 .....	9
<b>4. Action .....</b>	<b>10</b>
4.1. Overview .....	10
4.2. Properties .....	10
4.3. Method .....	11
<b>5. CtrlCode .....</b>	<b>17</b>
5.1. OverView .....	17
5.2. Properties .....	17
5.3. Method .....	22
<b>6. ParameterSet .....</b>	<b>24</b>
6.1. Overview .....	24
6.2. Properties .....	24
6.3. Method .....	26
<b>7. ParameterArray .....</b>	<b>34</b>
7.1. Overview .....	34
7.2. Properties .....	34
7.3. Method .....	35
<b>8. HwpCtrl .....</b>	<b>38</b>
8.1. Overview .....	38
8.2. Properties .....	38
8.3. Method .....	51
8.4. Event .....	115

# 1. 웹한글 기안기 구성/구조/지원포맷

## 1.1. 서버 구성

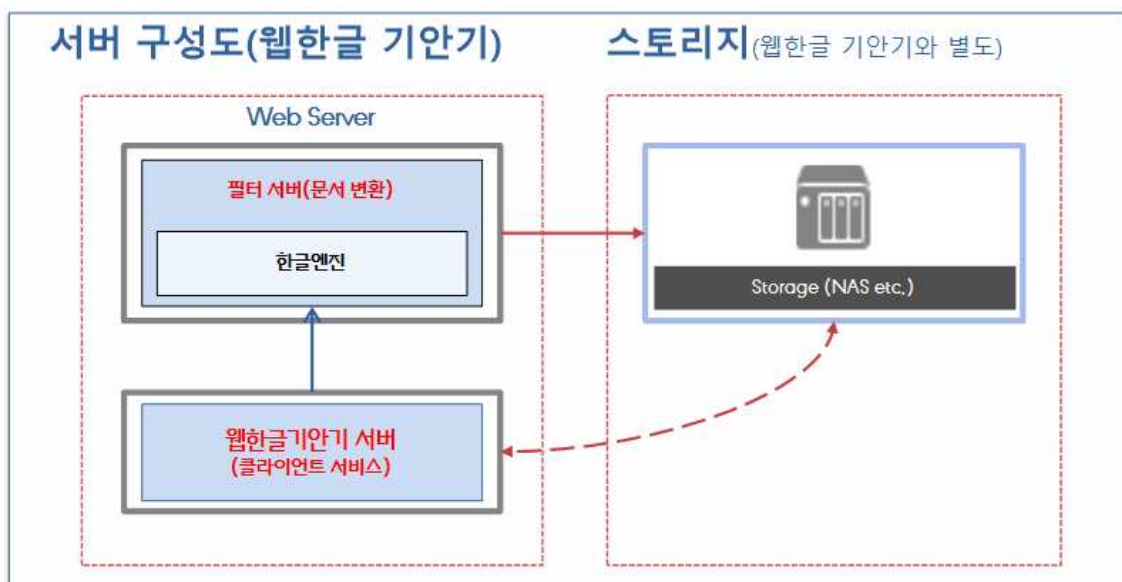
### Description

웹한글 기안기는 서버 기반의 제품으로 다양한 서버 환경을 제공하고자 한다. 서버 환경 지원을 통해 유관 시스템 등과의 연동을 지원하며, 데스크톱 오피스(한글)와의 복사/붙이기 호환성을 최대한 보장하기 위해 클라이언트의 한글을 2018 버전 이상, 최신 업데이트(2018년 7월 업데이트 이후)된 상태에서 이용하길 권장한다. 클라이언트에 한글이 설치되어 있지 않더라도 웹한글 기안기 자체적인 이용에는 아무런 제약이 없다.

### 서버 구성

웹한글 기안기는 다양한 사용자 환경에 빠른 적응을 위해 가볍고, 단순화된 구조를 갖추고 있다. 특히, Non DB 방식으로 웹서버, DB는 요구환경에 맞게 커스터마이징이 가능하다. 또한, 필터 서버를 통해 원활한 문서 변환을 지원하고 외부 서버와 통신이 가능한 구조를 제공한다.

구분	내용
필터 서버	· 문서 Open, Save 동작을 위한 문서 변환 필터 · 외부 서버와 통신 가능한 구조
웹한글 기안기 서버	· 웹한글 기안기 Client 서비스 관리
스토리지(별도)	· 문서 저장소



## 권장사항

구분	내용
서버구성	· 웹 서버 1식, 필터 서버 1식 (문서변환용)
웹 서버	· 지원운영체제 : 리눅스 계열 지원가능(Cent OS 최신버전에 최적화) * 타 운영체제 지원 필요시 별도 검토 필요
필터 서버	· RHEL 7.2 / CentOS 7.2 이상 지원
설치필요 프로그램/ 리소스	· 웹서버 : Tomcat 7 버전 이상 · Java : 1.7 버전 이상 · 웹한글 기안기 모듈

\* 한컴오피스 2018 이상 사용시, 최적의 호환성 제공 가능

## 1.2. 구조

### Description

기존 한글 컨트롤(ActiveX 기반) 대비 구조 변화

### 구조 비교

웹한글 기안기	한글 컨트롤(ActiveX)
1) IFrame 외부(JavaScript) - WebHwpCtrl API	1) 인터페이스 - HwpCtrl - HwpAction - HwpParameterSet - HwpParameterArray - HwpCtrlCode - HwpMenu
2) IFrame 내부(require js 기반 모듈화) - HwpCtrl Impl - HwpCtrl Object (HwpAction, HwpParameterSet, HwpCtrlCode...) - 웹한글 Engine - uiFramework	2) 구현부 - CHwpCtrlCtrl - CDHwpAction - CDHwpParameterSet - CDHwpParameterArray - CDHwpMenu

## 1.3. 호출 구조

### Description

기존 한글 컨트롤(ActiveX 기반) 대비 호출 구조 변화

### 호출 구조 - 웹한글 기안기

1) 외부에서 함수 호출

```
var testAction = HwpCtrl.CreateAction("CharShape");
```

//구현부 함수 호출

```
HwpCtrl.prototype.CreateAction = function(aidStr) {  
    return this.impl.CreateAction(aidStr);  
}
```

2) API 실행

```
HwpCtrlImpl.CreateAction= function(aidStr) {  
    return new HwpAction(aidStr);  
};
```

Action 객체

1) 외부에서 함수 호출(API 실행)

```
testAction.Run();
```

```
HwpAction.prototype.Run= function(){  
    .....  
};
```

## ※ 호출 구조 비교 – 한글 컨트롤(ActiveX)

- 1) 외부 인터페이스를 통해 호출

```
var testAction = HwpCtrl.CreateAction("CharShape");
```

- 2) 래퍼 클래스를 통해서 호출된 인터페이스의 ID로 Dispatch 메시지 발생

```
LPDISPATCH CreateAction(long bShow){....  
    InvokeHelper(0x11, DISPATCH_METHOD, VTVT_DISPATCH,  
(void*)&result, parms, actid);  
    return result;  
}
```

- 3) Dispatch Map에 해당 ID가 있다면 ID와 일치하는 API 함수 호출

```
BEGIN_DISPATCH_MAP(..  
    ....  
    DISP_FUNCTION_ID(CHwpCtrlCtrl, "CreateAction", 1 7 ,  
CreateAction, VT_DISPATCH, VTS_BSTR)  
    ....  
END_DISPATCH_MAP()
```

- 4) API가 실행

```
LPDISPATCH CHwpCtrlCtrl::CreateAction(LPCTSTR actidstr) {  
    .....  
}
```

## 1.4. 지원 포맷

### Description

웹한글 기안기는 기존 한글 포맷(HWP) 뿐만 아니라 KS표준 포맷 및 국제 표준 포맷인 ODF, 기록물 이관을 위한 PDF, 공문서 유통을 위한 포맷 등을 지원한다.

### 지원 포맷

HWP  
HWPX  
HWPML(HML)  
ODF(ODT)  
PDF  
XML(공문서 유통용)

## 2. 한글 컨트롤(ActiveX) 대비 API 변경사항

### 2.1. 제외된 API

#### Description

웹한글 기안기는 서버 상에서 동작하는 방식으로 변경됨에 따라 기술적 제약으로 몇가지 API 사용이 제한된다.

- 개인정보보호 관련 API
- 문서 비교, 버전 비교 관련 API
- Automation 관련 API
- 로컬 파일 접근 API

### 2.2. 사용법이 변경된 API

#### Description

웹한글 기안기는 기존 한글 컨트롤(ActiveX 기반)과는 구조가 변경되어 몇 가지 API들은 사용법이 변경되었다.

구분	내용
Point를 인자로 받는 API	<ul style="list-style-type: none"><li>· 함수 사용법 : Pointer를 인자로 받는 방식 → 객체 return</li><li>· 기존에 있던 대체 API는 유지<ul style="list-style-type: none"><li>- __BySet으로 Pointer를 사용할 수 없는 언어에서 set을 인자로 넣는 대체 API는 유지</li></ul></li><li>· 예) GetSelectedPos, GetSelectedPosSet</li></ul>
Server에 접근이 필요한 API	<ul style="list-style-type: none"><li>· Callback 함수를 등록해야 하는 API<ul style="list-style-type: none"><li>- 비동기 방식</li><li>- 예) GetTextFile, Open, OpenDocument, Insert, InsertDocument</li></ul></li><li>· Download를 사용하는 API<ul style="list-style-type: none"><li>- Download.js 사용</li><li>- 예) Save, SaveAs, SaveDocument, CreatePageImage</li></ul></li></ul>
파일에 접근하는 API	<ul style="list-style-type: none"><li>· JavaScript에서는 보안상의 이유로 Local File을 읽을 수 없음</li><li>· Upload한 파일 읽기<ul style="list-style-type: none"><li>- html form의 file input을 이용하여 upload한 파일을 읽음</li></ul></li><li>· 예) Open, OpenDocument, Insert, InsertDocument, InsertPicture, InsertBackgroundPicture</li></ul>
HwpCtrl Event	<ul style="list-style-type: none"><li>· 함수 사용법<ul style="list-style-type: none"><li>- HwpCtrl Event를 등록하기 위해 기존에 사용하던 문법은 Internet Explorer에서 ActiveX의 이벤트 등록을 위한 것으로 범용적으로 사용할 수 없음</li><li>- addEventListener 함수를 통해 콜백함수 등록하는 방식으로 변경</li></ul></li><li>· addEventListener API 추가</li></ul>

## Pointer를 인자로 받는 API

### GetSelectedPos

```
var pos = HwpCtrl.GetSelectedPos();  
console.log(pos.slist + ' ' + pos.spara + ' ' + pos.spos);  
console.log(pos.elist + ' ' + pos.epara + ' ' + pos.epos);
```

### GetSelectedPosBySet

```
var sSet = HwpCtrl.CreateSet('ListParaPos');  
var eSet = HwpCtrl.CreateSet('ListParaPos');  
var posSet = HwpCtrl.GetSelectedPosBySet(sSet , eSet );
```

## Server에 접근이 필요한 API

### Open

CallBack 함수를 등록해야 하는 API

```
<input type="button" id="open" value="열기"/>
```

```
HwpCtrl.Open$("input[name=openFile"])[0].files[0], "", "", function(){  
    HwpCtrl.MoveToField('이름1', true, true, true);  
});
```

- \* input을 통해서 upload하거나 미리 서버에 올라와 있는 파일만 접근 가능
- \* Open이후 동작은 callback 함수를 통해 가능

### GetTextFile

CallBack 함수를 등록해야 하는 API

```
var text = "";  
HwpCtrl.GetTextFile("HWP", "", function(data){  
    text = data;  
});
```

- \* GetTextFile이후 동작은 callback 함수를 통해 가능

### SaveAs

Download를 사용하는 API

```
HwpCtrl.SaveAs("test.hwp", "Hwp", "");
```

- \* 파일 이름만 지정 가능
- \* 항상 Download 폴더에 저장



## 파일에 접근하는 API

### InsertPicture

```
<input type="button" id="open" value="열기"/>
var url = URL.createObjectURL($("#input[name=imageFile]")[0].files[0]);
```

```
HwpCtrl.InsertPicture(url, true, 0, false, false, 0, 0, 0);
```

\* input을 통해서 upload하거나 미리 서버에 올라와 있는 파일만 접근 가능

## HwpCtrl Event

### OnMouseDown Event 추가

```
function HwpCtrl_OnMouseDown(x,y){
    console.log(x + ' , ' + y);
}
```

```
HwpCtrl.addEventListener(0, HwpCtrl_OnMouseDown);
```

### addEventListener

```
addEventListener(eventType, callback)
```

EventType

- 1) HWPCTRLEVENT\_ONMOUSELBBOTTONDOWN = 0;
- 2) HWPCTRLEVENT\_ONMOUSELBBOTTONUP = 1;
- 3) HWPCTRLEVENT\_ONSCROLL = 2;
- 4) HWPCTRLEVENT\_NOTIFYMESSAGE = 3;

## 3. 웹한글 기안기 API 기본 사용법

### 3.1. 초기 설정

#### Description

웹한글 기안기 초기 설정 방법

```
<head>
...
    <script type="text/javascript" src="js/hwpctrlapp/hwpCtrlApp.js"> </script>
    <script type="text/javascript" src="js/webhwpctrl.js"> </script>
</head>
<body>
<div id="hwpctrl" style=" width:1000px; height:800px;" />
...
</body>
```

\* 참고 : 한글 컨트롤(ActiveX) 초기 설정 방법

```
<body>
..
<form name = "HwpControl">
<object id=HwpCtrl style="LEFT: 0px; TOP: 0px" height=90% width=90%
align=center
        classid=CLSID:BD9C32DE-3155-4691-8972-097D53B10052>
</object>
</form>
...
</body>
```

## 3.2. 컨트롤 객체 생성

### Description

웹한글 기안기 객체 생성 방법

```
...  
<script type="text/javascript" src="js/webhwpctrl.js"></script>  
...  
<script>  
/*  
@param id : div ID  
@param baseurl : 다른 서버에서 접근시 baseURL을 지정한다.  
생략하면 location.href가 지정된다.  
@param callback : HwpCtrl 객체가 생성된 후 실행될 동작.  
*/  
    var HwpCtrl = BuildWebHwpCtrl("hwpctrl", "", function() {});  
...  
</script>  
  
* webhwpctrl.js에 포함된 BuildWebHwpCtrl을 호출하여 객체 생성
```

\* 참고 : 한글 컨트롤(ActiveX) 객체 생성 방법

```
...  
<form name = "HwpControl">  
...  
<script>  
    var HwpCtrl = HwpControl.HwpCtrl;  
...  
</script>
```

## 3.3. HwpCtrl Object(Action, ParameterSet) 사용 예

### Description

웹한글 기안기와 기존 한글 컨트롤(ActiveX) Object 사용법은 동일하다.

## 4. Action

### 4.1. Overview

#### Description

##### 한글 액션 오브젝트

사용자가 하나의 단위 기능으로 인식하는 한글의 기능 각각을 액션이라고 부른다. 예를 들면 "파일열기", "파일저장", "표 삽입", "글자 속성 수정" 등이다. 가장 쉽게 이해할 수 있는 액션의 정의는, '메뉴, 툴바, 단축키를 통해 실행할 수 있는 하나의 기능'으로 생각하면 된다. 기존 윈도우즈 어플리케이션에서 WM\_COMMAND 핸들러에서 처리하는 개개의 단위 기능을 떠올리면 된다. 결국 사용자에게 어플리케이션은 이러한 단위 액션들의 집합으로 인식된다. HwpAction은 한글 컨트롤을 이용하는 사용자가 한글의 액션을 직접 다룰 수 있도록 하기 위한 오브젝트이다.

#### Example

Bold Action을 생성한 후 실행시킨다.

```
var act;  
act = HwpCtrl.CreateAction("CharShapeBold");// 액션 생성  
act.Run(); //액션 실행
```

### 4.2. Properties

#### 4.2.1. ActID

##### Syntax

```
act.ActID
```

##### Description

액션 ID를 나타낸다. 읽기 전용.

액션 ID의 종류 및 설명은 별도 문서 참조.

##### Return Values

ActionID를 String 형태로 return 한다.

#### Example

```
var act = HwpCtrl.CreateAction("CharShapeBold");  
var aID = act.ActID;  
console.log("Action ID : " + aID);
```

## See Also

[HwpCtrl.CreateAction](#)

### 4.2.2. SetID

#### Syntax

```
act.SetID
```

#### Description

액션이 사용하는 parameter set ID를 나타낸다. 읽기 전용.

#### Return Values

액션이 사용하는 SetID를 String 형태로 return 한다.

#### Example

```
var act = HwpCtrl.CreateAction("CharShapeBold");  
var setID = act.SetID;
```

## See Also

[HwpCtrl.CreateAction](#)

### 4.3. Method

#### 4.3.1. CreateSet

#### Syntax

```
act.CreateSet()
```

#### Description

액션과 대응하는 Parameter Set을 생성한다.

#### Parameters 0

#### Return Values

성공시 ParameterSet 객체를 return 한다.

Parameter Set을 사용하지 않는 Action은 null 객체를 리턴한다.

## Example

5행 5열의 표를 생성하여 본문에 삽입한다.

```
var tbact = HwpCtrl.CreateAction("TableCreate");
var tbset = tbact.CreateSet();
tbset.SetItem("Rows", 5);
tbset.SetItem("Cols", 5);
var table = HwpCtrl.InsertCtrl("tbl", tbset);
```

## Remarks

Action 객체의 CreateSet method의 수행결과는 HwpCtrl를 통해 다음과 같이 수행한 것과 동일하다.

```
var tbact = HwpCtrl.CreateAction("TableCreate");
var tbset = tbact.CreateSet();
tbact.GetDefault(tbset);
tbset.SetItem("Rows", 5);
tbset.SetItem("Cols", 5);
tbact.Execute(tbset);
```

## See Also

[HwpCtrl.CreateSet](#), [HwpCtrl.CreateAction](#), [ParameterSet.SetItem](#),  
[HwpCtrl.InsertCtrl](#)

## 4.3.2. GetDefault

### Syntax

```
act.GetDefault(param)
```

### Description

현재 상태에 따라 액션 실행에 필요한 인수를 구한다.

### Parameters 1

**param** : 인수를 저장할 parameter set

### Return Values

성공시 true, 실패시 false

## Example

[CreateSet 참조](#)

## Remarks

예를 들어 글자모양의 액션의 경우, 현재 선택션 상태에 따라 param의 아이템들이 채워진다.

서브셋을 만들 경우에는 서브셋을 만든 후에 GetDefault를 사용한다.

## 4.3.3. PopupDialog

### Syntax

```
act.PopupDialog(param, callback, callbackUserData)
```

### Description

액션의 대화상자를 띄운다

### Parameters 3

**param** : 여기에 지정한 아이템의 값에 따라 대화상자의 각 컨트롤의 초기값이 결정되고, 대화상자가 닫힌 후에는 사용자가 지정한 값들이 담겨 돌아온다.

**callback** : PopupDialog 동작을 마친 후 수행될 함수  
parameter 4

action: 콜백을 호출한 Action 객체

paramSet: 액션으로 전달된 ParameterSet 객체

result: 대화상자 버튼 클릭 결과 (true: 설정버튼, false: 취소버튼)

callbackUserData: 사용자가 콜백에 전달한 데이터  
(PopupDialog의 callbackUserData가 전달된다.)

**callbackUserData** : 사용자가 콜백에 전달한 데이터

### Example

글자 속성 실행

```

var dact = HwpCtrl.CreateAction("CharShape");// 액션 생성
var dset = dact.CreateSet();
dact.GetDefault(dset);
dact.PopupDialog(dset, function(action, paramSet, result,
callbackUserData){
    if(result == true){
        action.Execute(paramSet);
    }
});

```

## Remarks

Callback 함수로 변경되었다.

## 4.3.4. Execute

### Syntax

```
act.Execute(param, callback, callbackUserData)
```

### Description

지정한 인수로 액션을 실행한다.

### Parameters 3

**param** : 액션의 실행을 제어할 인수. parameterSet의 종류와 아이템의 의미는 액션이 정의한 바에 따라 다르다.

**callback** : Execute 동작 수행 후 실행될 함수

parameter 3

action: 콜백을 호출한 Action 객체

paramSet: 액션으로 전달된 ParameterSet 객체

result: 액션 실행 결과 (true: 액션 성공, false: 액션 실패)

callbackUserData: 사용자가 콜백에 전달한 데이터

**callbackUserData** : 사용자가 콜백에 전달한 데이터

### Example

PopupDialog 참고



## See Also

[HwpCtrl.Run](#)

## 4.3.5. Run

### Syntax

```
act.Run()
```

### Description

액션을 실행한다.

### Parameters 2

**callback** : Run 동작 수행 후 실행될 함수

parameter 3

action: 콜백을 호출한 Action 객체

result: 액션 실행 결과

callbackUserData: 사용자가 콜백에 전달한 데이터

**callbackUserData** : 사용자가 콜백에 전달한 데이터

### Example

Bold Action 실행(대화상자가 없는 액션은 동기 방식으로 실행한다.)

```
var act;  
act = HwpCtrl.CreateAction("CharShapeBold");// 액션 생성  
act.Run(); //액션 실행
```

### Remarks

CreateSet, GetDefault, PopupDialog, Execute를 차례로 부른 것과 같다.  
또, 다음 두 가지도 동일하다.

```
HwpCtrl.Run("action");  
HwpCtrl.CreateAction("action").Run();
```

대화상자가 없는 액션은 위와 같이 동기 방식으로 실행하고, 대화상자가 있는  
액션은 Callback 함수를 통해서 비동기 방식으로 실행해야 한다.

## See Also

[HwpCtrl.CreateAction](#), [HwpCtrl.Run](#)

## 5. CtrlCode

### 5.1. OverView

#### Description

문서 내부의 표, 각주 등의 컨트롤(특수 문자)를 나타내는 오브젝트이다.

### 5.2. Properties

#### 5.2.1. CtrlCh

##### Syntax

```
ctrlCode.CtrlCh
```

##### Description

컨트롤 문자. 읽기 전용.

##### Return Values

현재 컨트롤의 컨트롤 문자를 Number 타입으로 리턴한다.

##### Remarks

일반적으로 컨트롤 ID를 사용해 컨트롤의 종류를 판별하지만, 이보다 더 포괄적인 범주를 나타내는 컨트롤 문자로 판별할 수도 있다. 예를 들어 각주와 미주는 ID는 다르지만, 컨트롤 문자는 17로 동일하다. 컨트롤 문자는 1-31 사이의 값을 사용한다. 그 중 CtrlCode 객체로 리턴 받을 수 있는 문자는 다음과 같다.

Ch	설명
2	구역/단 정의
3	필드 시작
11	그리기 개체 / 표
15	주석
16	머리말 / 꼬리말
17	각주 / 미주
18	자동 번호
21	새번호/ 번호 감주기/ 번호 위치
22	책갈피 / 찾아보기 표시
23	덧말 / 글짜 겹침,,,

##### Example

```

var ctrl = HwpCtrl.HeadCtrl;
while(ctrl) {
    if (ctrl.CtrlCh == 16) {
        alert("머리말/꼬리말이 존재합니다.");
        return;
    }
    ctrl = ctrl.Next;
}
alert("머리말/꼬리말이 없습니다.");

```

## 5.2.2. CtrlID

### Syntax

```
ctrlCode.CtrlID
```

### Description

컨트롤 ID. 읽기 전용.

### Return Values

현재 컨트롤의 ID를 String 타입으로 리턴한다.

### Remarks

컨트롤 ID는 컨트롤의 종류를 나타내기 위해 할당된 ID로서, 최대 4개의 문자로 구성된 문자열이다. 예를 들어 표는 "tbl", 각주는 "fn"이다. 한글에서 현재까지 지원되는 모든 컨트롤의 ID는 다음 표 참조.

ID	Property Set	Initialization Set	설명
cold	ColDef	ColDef	단
secd	SecDef	SecDef	구역
fn	FootnoteShape	FootnoteShape	각주
en	FootnoteShape	FootnoteShape	미주
tbl	Table	TableCreation	표
eqed	EqEdit	EqEdit	수식
gso	ShapeObject	ShapeObject	그리기 개체
atno	AutoNum	AutoNum	번호넣기
nwno	AutoNum	AutoNum	새번호로

ID	Property Set	Initialization Set	설명
pgct	PageNumCtrl	PageNumCtrl	페이지 번호 제어 (97의 홀수쪽에서 시작)
pghd	PageHiding	PageHiding	감추기
pgnp	PageNumPos	PageNumPos	쪽번호 위치
head	HeaderFooter	HeaderFooter	머리말
foot	HeaderFooter	HeaderFooter	꼬리말
%dte	FieldCtrl	FieldCtrl	현재의 날짜/시간 필드
%ddt	FieldCtrl	FieldCtrl	파일 작성 날짜/시간 필드
%pat	FieldCtrl	FieldCtrl	문서 경로 필드
%bmk	FieldCtrl	FieldCtrl	블록 책갈피
%mmg	FieldCtrl	FieldCtrl	메일 머지
%xrf	FieldCtrl	FieldCtrl	상호 참조
%fmu	FieldCtrl	FieldCtrl	계산식
%clk	FieldCtrl	FieldCtrl	누름틀
%smr	FieldCtrl	FieldCtrl	문서 요약 정보 필드
%usr	FieldCtrl	FieldCtrl	사용자 정보 필드
%hlk	FieldCtrl	FieldCtrl	하이퍼링크
bokm	TextCtrl	TextCtrl	책갈피
idxm	IndexMark	IndexMark	찾아보기
tdut	Dutmal	Dutmal	덧말
tcmt	없음	없음	주석

◆ Property Set : Ctrl.Properties를 통해 액세스할 수 있는 속성 parameter set ID

◆ Initialization Set : HwpCtrl.InsertCtrl에 지정할 수 있는 initparam의 parameter set ID

## Example

현재 문서에 있는 모든표의 셀간격을 1pt로 설정한다.

```
var ctrl = HwpCtrl.HeadCtrl;
var tbset = HwpCtrl.CreateSet("Table");
tbset.SetItem("CellSpacing", 100);

while(ctrl) {
    if (ctrl.CtrlID == "tbl") {
        ctrl.Properties = tbset;
    }
}
```

```
ctrl = ctrl.Next;  
}
```

### See Also

[HwpCtrl.HeadCtrl](#), [HwpCtrl.CreateSet](#), [ParameterSet.SetItem](#), [Properties](#), [Next](#)

## 5.2.3. Next

### Syntax

```
ctrlCode.Next
```

### Description

다음 컨트롤. 읽기 전용.

### Return Values

현재 컨트롤의 뒤 컨트롤을 CtrlCode 객체 타입으로 리턴한다.

### Example

[CtrlID 참고](#)

### Remarks

문서 중의 모든 컨트롤(표, 그림 등의 특수 문자들)은 linked list로 서로 연결되어 있는데, list 중 다음 컨트롤을 나타낸다.

### See Also

[HwpCtrl.HeadCtrl](#), [Prev](#)

## 5.2.4. Prev

### Syntax

```
ctrlCode.Prev
```

### Description

앞 컨트롤. 읽기 전용.

### Return Values

현재 컨트롤의 앞 컨트롤을 CtrlCode 객체 타입으로 리턴한다.

## Example

현재 문서에 있는 모든표의 셀간격을 1pt로 설정한다.

```
var ctrl = HwpCtrl.LastCtrl;
var tbset = HwpCtrl.CreateSet("Table");
tbset.SetItem("CellSpacing", 100);

while(ctrl) {
    if (ctrl.CtrlID == "tbl") {
        ctrl.Properties = tbset;
    }
    ctrl = ctrl.Prev;
}
```

## Remarks

문서 중의 모든 컨트롤(표, 그림 등의 특수 문자들)은 linked list로 서로 연결되어 있는데, list 중 앞 컨트롤을 나타낸다.

## See Also

[HwpCtrl.LastCtrl](#), [Next](#)

## 5.2.5. Properties

### Syntax

```
ctrlCode.Properties
```

### Description

컨트롤의 속성을 나타낸다.

### Return Values

컨트롤의 속성을 ParameterSet 객체 타입으로 리턴한다.

## Example

[Prev](#) 참고

## Remarks

모든 컨트롤은 대응하는 parameter set으로 속성을 읽고 쓸 수 있다.

## 5.2.6. UserDesc

## Syntax

```
ctrlCode.UserDesc()
```

## Description

컨트롤의 종류를 사용자에게 보여줄 수 있는 localize된 문자열로 나타낸다. 읽기 전용.

## Example

```
console.log(HwpCtrl.CurSelectedCtrl.UserDesc);
```

## See Also

[CurSelectedCtrl](#)

## 5.3. Method

### 5.3.1. GetAnchorPos

## Syntax

```
ctrlCode.GetAnchorPos(type)
```

## Description

컨트롤의 anchor의 위치를 리턴한다.

## Parameters 1

**type** : 기준 위치

값	설명	비고
0	바로 상위 리스트에서의 anchor position	default
1	탭레벨 리스트에서의 anchor position	
2	루트 리스트에서의 anchor position	

## Return Values

성공했을 경우 ListParaPos ParameterSet이 리턴 된다.  
실패했을 경우 null이 리턴된다.

## Example

현재 문서의 첫 번째 표로 커서를 이동한다.

```
var SetPosToTable = function(){
```

```

var code = HwpCtrl.HeadCtrl;
var paramSet = null;
var list = 0;
var para = 0;
var pos = 0;

while (code && code != HwpCtrl.LastCtrl) {
    var strID = code.CtrlID;
    if (strID == "tbl") {
        paramSet = code.GetAnchorPos(0);
        list = paramSet.Item("List");
        para = paramSet.Item("Para");
        pos = paramSet.Item("Pos");
        HwpCtrl.SetPos(list, para, pos);
        return true;
    }
    code = code.Next;
}
return false;
}

```

## See Also

[HwpCtrl.HeadCtrl](#), [HwpCtrl.SetPos](#), [ParameterSet.Item](#)



## 6. ParameterSet

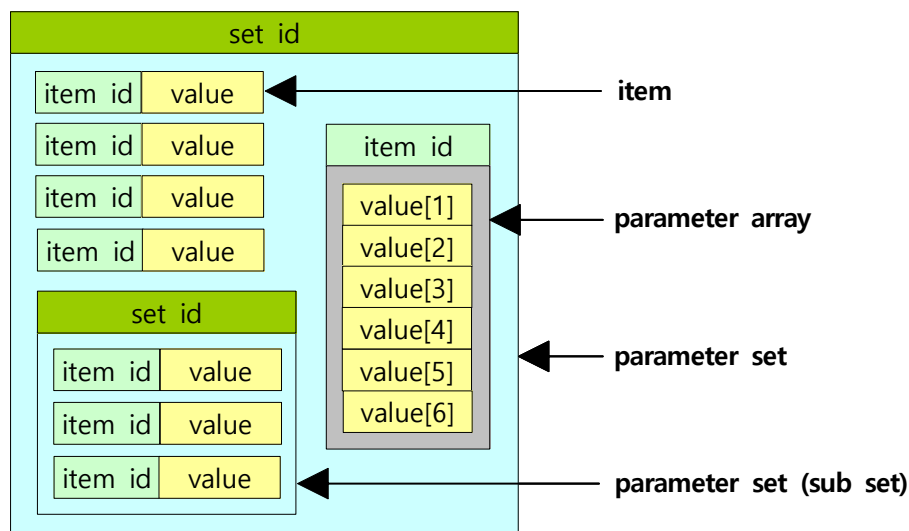
### 6.1. Overview

#### Description

오브젝트간 또는 액션의 실행에 필요한 정보를 주고 받을 수 있도록 하기 위한 오브젝트이다.

#### Remarks

어떤 액션을 수행하기 위해서는 액션이 필요로 하는 정보를 전달해주어야 하는 경우가 발생한다. 예를 들어, "표만들기"를 할때 행과 열은 각각 얼마로 할 것인지 행의 디폴트 높이와 열의 디폴트 폭은 얼마로 할 것인지 등을 한글에게 알려주어야 한다. 이와 같이 두 객체간에 전달해야 할 정보들의 묶음을 parameter set이라 하고, 묶음 내 각각의 정보에 해당하는 행의 수, 열의 수 등을 set item이라 한다. 각 item은 item 아이디와 item 값의 쌍으로 구성된다. parameter set의 item으로 parameter set이 올 수 있어 서브셋을 갖는 구조가 가능하며, 배열을 표현하기 위한 parameter array라는 구조가 item으로 올 수도 있다.



### 6.2. Properties

#### 6.2.1. Count

##### Syntax

```
set.Count
```

##### Description

현재 존재하는 아이템의 개수를 나타낸다. 읽기 전용.

## Return Values

아이템 개수를 Number 타입으로 리턴한다

## Example

2줄 2칸짜리 표를 가로, 세로 길이 설정한 후 만드는 예제

```
var act = HwpCtrl.CreateAction("TableCreate");
var set = act.CreateSet();
console.log(set.SetID);
if(set.IsSet){
    var colset = set.CreateItemArray("ColWidth", 2);
    var rowset = set.CreateItemArray("RowHeight", 2);
    console.log(colset.Count);
    if(colset.IsSet){
        alert("ColWidth는 ParameterArray 타입이어야 합니다.");
        return;
    }

    act.GetDefault(set);
    set.SetItem("Rows", 2);
    set.SetItem("Cols", 2);
    console.log(set.Count);

    colset.SetItem(0, 14400);
    colset.SetItem(1, 7200);
    rowset.SetItem(0, 3600);
    rowset.SetItem(1, 7200);

    act.Execute(set);
}
```

## See Also

[IsSet](#), [SetID](#), [ParameterArray.Count](#), [ParameterArray.IsSet](#)

## 6.2.2. IsSet

### Syntax

```
set.IsSet
```

## Description

parameter set인지 여부를 나타낸다. 읽기 전용.

## Return Values

parameterSet 객체인지 boolean 타입으로 리턴한다.

parameterSet인 경우 true, parameterArray인 경우 false를 리턴한다.

## Example

[Count](#) 참고

## 6.2.3. SetID

### Syntax

```
set.SetID
```

## Description

parameter set의 ID를 나타낸다. 읽기 전용.

## Return Values

SetID를 String 타입으로 리턴한다.

## Example

[Count](#) 참고

## 6.3. Method

### 6.3.1. Clone

### Syntax

```
set.Clone()
```

## Description

parameter set을 복사한다.

## Parameters 0

## Return Values

동일한 데이터를 가진 parameter set을 복사하여 ParameterSet 객체 타입으로 리턴한다.

## Example

```
var charShapeSet = HwpCtrl.CharShape;  
var cloneSet = charShapeSet.Clone();  
  
console.log(cloneSet.IsEquivalent(charShapeSet));
```

### 6.3.2. CreateItemArray

#### Syntax

```
set.CreateItemArray(itemid, count)
```

#### Description

아이템으로 parameter array 타입의 배열을 생성한다.

#### Parameters

2

**itemid** : 아이템 ID

**count** : 생성할 배열의 초기 크기

#### Return Values

인자에 따라 parameter array를 생성하고, ParameterArray 객체 타입으로 리턴한다.

#### Remarks

동일한 ID를 가진 기존의 아이템은 삭제된다.

## Example

[Count](#) 참고

## See Also

[HwpCtrl.CreateAction](#), [Action.CreateSet](#), [Action.GetDefault](#), [Action.Execute](#)

### 6.3.3. CreateItemSet

#### Syntax

```
set.CreateItemSet(itemid, setid)
```

#### Description

아이템으로 parameter set을 생성한다.

## Parameters 2

**itemid** : 아이템 ID (별도 문서 참조)

**setid** : 생성할 parameter set ID (별도 문서 참조)

## Return Values

생성된 서브 셋을 ParameterSet 객체 타입으로 리턴한다.

## Remarks

parameter set 내부에 아이템으로 또 다른 parameter set을 가지는 서브 셋의 개념이다.

## Example

문서 전체 용지의 속성을 변경한다.

```
var act = HwpCtrl.CreateAction("PageSetup");
var set = act.CreateSet();

act.GetDefault(set);
set.SetItem("ApplyTo", 3); //적용범위 : 문서전체

var pset = set.CreateItemSet("PageDef","PageDef");
pset.SetItem("TopMargin", 3401);
pset.SetItem("BottomMargin", 5669);
pset.SetItem("LeftMargin", 4251);
pset.SetItem("RightMargin", 4251);
pset.SetItem("HeaderLen", 0);
pset.SetItem("FooterLen", 0);
pset.SetItem("GutterLen", 0);
act.Execute(set);
```

## See Also

[HwpCtrl.CreateAction](#), [Action.CreateSet](#), [Action.Execute](#), [SetItem](#)

## 6.3.4. GetInterSection

### Syntax

```
set.GetInterSection(srcset)
```

## Description

두 set에 공통적으로 존재하고, 값도 동일한 아이템만으로 구성된 intersection set을 구한다.

## Parameters

1

**srcset** : this와 srcset의 intersection이 this에 저장된다.

## Example

```
var charShapeSet = HwpCtrl.CharShape;
var cloneSet = charShapeSet.Clone();

cloneSet.RemoveItem('Height'); //cloneSet에서 글자크기 속성 제거
charShapeSet.GetIntersection(cloneSet);
// charShapeSet에서 charShapeSet 과 cloneSet의 공통적으로 존재하는 아이템만 남긴다.

if(charShapeSet.ItemExist('Height') == false){
    alert('글자크기 속성이 삭제되었습니다.')
}
```

## 6.3.5. IsEquivalent

### Syntax

```
set.IsEquivalent(srcset)
```

### Description

두 set의 내용이 동일한 값을 가지고 있는지 검사한다.

## Parameters

1

**srcset** : this와 srcset을 비교한 결과를 리턴한다.

## Return Values

동일하면 True, 다르면 False

## Example

[clone](#) 참고

## 6.3.6. Item

## Syntax

```
set.Item(itemid)
```

## Description

지정한 아이템의 값을 리턴한다.

## Parameters 1

**itemid** : 아이템 ID (별도 문서 참조)

## Return Values

아이템의 값, 만약 지정한 아이템이 존재하지 않으면 아이템의 포맷에 따라 0 또는 빈 문자열을 리턴한다.

## Example

현재 캐럿의 위치를 저장한다.

```
var posSet = HwpCtrl.GetPosBySet();  
var curPos = {};  
curPos.List = posSet.Item("List");  
curPos.Para = posSet.Item("Para");  
curPos.Pos = posSet.Item("Pos");
```

## See Also

[HwpCtrl.GetPosBySet](#)

## 6.3.7. ItemExist

### Syntax

```
set.ItemExist(itemid)
```

### Description

지정한 아이템이 존재하는지 검사한다.

### Parameters 1

**itemid** : 아이템 ID (별도 문서 참조)

### Return Values

존재하면 True, 존재하지 않으면 False

### Example

```
var charShapeSet = HwpCtrl.CharShape;  
if(charShapeSet.ItemExist('Height') == true){  
    charShapeSet.SetItem('Height', 2000);  
    HwpCtrl.CharShape = charShapeSet;  
}
```

### See Also

[HwpCtrl.CharShape](#), [SetItem](#)

## 6.3.8. Merge

### Syntax

```
set.Merge(srcset)
```

### Description

두 set의 내용을 병합한다.

결과는 "this의 모든 아이템 + srcset에만 존재하는 아이템"이다.

### Parameters 1

**srcset** : this와 srcset이 병합되어 this에 저장된다.

## 6.3.9. RemoveAll

### Syntax

```
set.RemoveAll(setID)
```

### Description

parameter set을 초기화한다.

### Parameters 1

**setid** : 새로 적용할 Set ID (별도 문서 참조)

### Remarks



이미 존재하는 parameter set 오브젝트를 이용해 새로운 타입의 parameter set 으로 초기화하여 재사용하는 목적에 사용된다.

## 6.3.10. RemoveItem

### Syntax

```
set.RemoveItem(itemid)
```

### Description

지정한 아이템을 삭제한다.

### Parameters 1

**itemid** : 아이템 ID (별도 문서 참조)

### Example

[GetIntersection](#)참고

## 6.3.11. SetItem

### Syntax

```
set.SetItem(itemid, value)
```

### Description

지정한 아이템의 값을 설정한다.

### Parameters 2

**itemid** : 아이템 ID (별도 문서 참조)

**value** : 설정할 값

### Remarks

이미 동일한 ID의 아이템이 존재하면 지정한 값으로 바뀌고, 존재하지 않으면 아이템이 생성된다.

### Example

[ItemExist](#), [Action.CreateSet](#) 참고

## 7. ParameterArray

### 7.1. Overview

#### Description

parameter set의 아이템으로 배열을 표현하는 데 사용된다. 일반적인 method의 독립적인 인수로 사용되는 일은 없고, parameter set의 아이템으로만 사용된다.

### 7.2. Properties

#### 7.2.1. Count

##### Syntax

```
itemArray.Count
```

##### Description

배열의 크기를 나타낸다.

##### Return Values

배열의 크기를 Number 타입으로 리턴한다.

##### Remarks

배열의 크기는 runtime에 dynamic하게 조절할 수 있다.

##### Example

[ParameterSet.Count](#) 참고

#### 7.2.2. IsSet

##### Syntax

```
itemArray.IsSet
```

##### Description

parameter set인지 여부를 나타낸다. 읽기 전용.

##### Return Values

parameterSet인 경우 true, parameterArray인 경우 false를 리턴한다.

##### Example

[ParameterSet.Count](#) 참고

## 7.3. Method

### 7.3.1. Clone

#### Syntax

```
itemArray.Clone()
```

#### Description

동일한 크기와 데이터를 갖는 ParameterArray 개체를 복사하여 돌려준다.

**Parameters**        **0**

#### Return Values

성공시 복사된 ParameterArray 객체를 리턴한다.

실패시 null을 리턴한다.

#### Example

ParameterArray Method 동작 테스트 코드

```
var set = HwpCtrl.CreateSet("TableCreation");
var colset = set.CreateItemArray("ColWidth", 2);
var colWidth = 14400;
colset.SetItem(0, colWidth);
colset.SetItem(1, colWidth);
if(colset.Item(0) == colWidth){
    alert("ParameterArray SetItem/Item 성공");
} else {
    alert("ParameterArray SetItem/Item 실패");
}
var cloneColset = colset.Clone();
if(cloneColset.Item(0) == colWidth && cloneColset.Item(1) == colWidth ) {
    alert("ParameterArray Clone 성공");
} else {
    alert("ParameterArray Clone 실패");
}
cloneColset.SetItem(0, 0);
if(cloneColset.Item(0) == 0) {
    cloneColset.Copy(colset);
    if(cloneColset.Item(0) == colWidth && cloneColset.Item(1) ==
colWidth) {
```

```

        alert("ParameterArray Copy 성공");
    } else {
        alert("ParameterArray Copy 실패");
    }
} else {
    alert("ParameterArray SetItem/Item 실패");
}

```

### 7.3.2. Copy

#### Syntax

```
itemArray.Copy(srcarray)
```

#### Parameters 1

**srcarray** : srcarray의 내용이 그대로 this로 복사된다.

#### Example

[Clone](#) 참고

### 7.3.3. Item

#### Syntax

```
itemArray.Item(index)
```

#### Description

지정한 원소의 값을 리턴한다.

#### Parameters 1

**index** : 원소의 인덱스. 1부터 시작한다.

#### Return Values

index에 해당하는 원소의 값을 리턴한다.

#### Example

[Clone](#) 참고

### 7.3.4. SetItem

#### Syntax

```
itemArray.SetItem(index, value)
```

## Description

지정한 원소의 값을 설정한다.

## Parameters 2

**index** : 원소의 인덱스. 0부터 시작.

**value** : 원소의 값

## Example

[ParameterSet.Count](#), [Clone](#) 참고

## 8. HwpCtrl

### 8.1. Overview

#### Description

HwpCtrl 오브젝트는 한글 컨트롤의 메인 오브젝트로서 다른 오브젝트의 생성, 툴바 제어 등 컨트롤 전반적인 기능 외에 사용 빈도수가 높고 중요한 대부분의 메소드들을 사용자에게 직접 제공한다.

### 8.2. Properties

#### 8.2.1. CellShape

##### Syntax

`HwpCtrl.CellShape`

##### Description

현재 선택되어 있는 표와 셀의 모양 정보를 나타낸다.

##### Return Values

블록된 셀 혹은, 현재 캐럿이 위치한 셀의 속성을 ParameterSet 객체 타입으로 리턴한다.

##### Remarks

ParameterSet/Table로 표의 속성에 대한 기본 정보를 나타내며, 이 가운데 "Cell" 아이템이 ParameterSet/Cell로 셀의 속성을 나타낸다. 셀 블록이 잡혀있지 않은 상태이면 현재 캐럿이 위치한 셀 하나만을 대상으로 한다. 현재 표 내부에 캐럿이 위치하지 않으면 에러가 발생한다.

##### Example

셀간격을 2pt로, 선택된 모든 셀의 제목셀 속성을 on으로

```
var tp = HwpCtrl.CreateSet("Table");
var cp = tp.CreateItemSet("Cell", "Cell");
tp.SetItem("CellSpacing", 200);
cp.SetItem("Header", 1);
HwpCtrl.CellShape = tp;
```

##### See Also

[HwpCtrl.CreateSet](#), [ParameterSet.CreateItemSet](#), [ParameterSet.SetItem](#)

## 8.2.2. CharShape

### Syntax

```
HwpCtrl.CharShape
```

### Description

현재 selection의 글자 모양을 나타낸다.

### Return Values

블록된 구역 혹은, 현재 캐럿이 위치한 곳의 글자 모양을 ParameterSet 객체 타입으로 리턴한다.

### Remarks

property get을 수행하면 현재 selection 내의 글자 모양을 구할 수 있다.  
selection이 존재하지 않으면 현재 캐럿이 위치한 곳의 글자 모양을 돌려준다.  
글자 모양 중 특정 항목이 selection 내에서 서로 다른 속성을 가지고 있으면 아예  
아이템 자체가 존재하지 않는다.  
property set을 수행하면 아이템이 존재하는 항목에 대해서만 속성을 설정한다.  
ParameterSet의 형식은 ParameterSet/CharShape 참조.

### Example

selection 내의 글자 크기가 모두 동일할 때만 20pt로 설정한다.

```
var cs = HwpCtrl.CharShape;  
if(cs.ItemExist("Height")){  
    cs.SetItem("Height", 2000);  
    HwpCtrl.CharShape = cs;  
}
```

### See Also

[ParameterSet.ItemExist](#), [ParameterSet.SetItem](#)

## 8.2.3. CurFieldState

### Syntax

```
HwpCtrl.CurFieldState
```

### Description

캐럿이 위치한 필드의 상태 정보를 구한다. 읽기 전용.

## Remarks

다음과 같은 의미를 가진다.

bit 4               = 필드명의 존재 여부 (1 = 있음, 0 = 없음)  
bit 0 - 3           = 필드의 종류 (0 = 없음, 1 = 셀, 2 = 누름틀, 4 = 글상자)

필드명의 존재 여부/ 필드의 종류에 따라서 0~20까지의 값을 갖는다.

## Example

현재 필드의 상태를 나타낸다.

```
var fieldState = HwpCtrl.CurFieldState;
var msg = "";
switch (fieldState){
    case 0:
        msg = "필드 없음";
        break;
    case 1:
        msg = "셀";
        break;
    case 2:
        msg = "누름틀";
        break;
    case 4:
        msg = "글상자";
        break;
}
alert("현재선택된 필드 : " + msg);
```

## See Also

[GetCurFieldName](#), [SetCurFieldName](#)

## 8.2.4. EditMode

### Syntax

```
HwpCtrl.EditMode
```

### Description



현재 편집 모드

## Return Values

현재 편집 모드를 Number 타입으로 리턴한다.

## Remarks

현재 편집 모드를 나타낸다.

0 : 읽기 전용

1 : 일반 편집모드

2 : 양식 모드(양식 사용자 모드) : Cell과 누름틀 중 양식 모드에서 편집 가능 속성을 가진 것만 편집 가능하다.

16 : 배포용 문서 (SetEditMode로 지정 불가능)

## Example

읽기 전용 모드로 전환한다.

```
HwpCtrl.EditMode = 0;
```

현재 EditMode가 무엇인지 나타낸다.

```
var editmode = HwpCtrl.EditMode;
var msg = "";
switch (editmode){
    case 0:
        msg = "읽기 전용 모드입니다."
        break;
    case 1:
        msg = "일반 편집 모드입니다."
        break;
    case 2:
        msg = "양식 모드입니다."
        break;
    case 16:
        msg = "배포용 문서 모드입니다."
        break;
    default:
        msg = "잘못된 EditMode 입니다."
}
alert(msg);
```

## 8.2.5. EngineProperties

### Syntax

```
HwpCtrl.EngineProperties
```

### Description

환경 설정 정보

### Return Value

환경 설정 정보를 ParameterSet 객체 타입으로 리턴한다.

### Remarks

환경 설정에서 지정할 수 있는 옵션 값을 설정할 수 있다.  
ParameterSet의 형식은 ParameterSet/EngineProperties 참조.  
\*Engine에 옵션 값이 적용은 되지 않는다.

### Example

문서 중 모든 표를 삭제한다.

```
var set = HwpCtrl.CreateSet("EngineProperties");  
set.SetItem("EnableAutoSpell", 1);  
HwpCtrl.EngineProperties = set;
```

## 8.2.6. HeadCtrl

### Syntax

```
HwpCtrl.CellShape
```

### Description

문서 중 첫번째 컨트롤. 읽기 전용.

### Return Values

문서 중 첫번째 컨트롤을 HwpCtrlCode 객체 타입으로 리턴한다.

### Remarks

문서 중의 모든 컨트롤(표, 그림 등의 특수 문자들)은 linked list로 서로 연결되어 있는데, 그 list의 시작 컨트롤을 나타낸다. 이 컨트롤로부터 시작, Ctrl.Next를 이용해 forward iteration을 수행할 수 있다.

### Example

문서 중 모든 표를 삭제한다.

```
var ctrl = HwpCtrl.HeadCtrl;

while(ctrl != null) {
    if(ctrl.CtrlID == "tbl"){
        HwpCtrl.DeleteCtrl(ctrl);
    }
    ctrl = ctrl.Next;
}
```

## 8.2.7. IsEmpty

### Syntax

```
HwpCtrl.IsEmpty
```

### Description

아무 내용도 들어있지 않은 빈 문서인지 여부를 나타낸다. 읽기 전용.

### Return Values

빈 문서인지 여부를 Boolean 타입으로 리턴한다.

## 8.2.8. IsModified

### Syntax

```
HwpCtrl.IsModified
```

### Description

문서가 변경되었는지 나타낸다.

### Return Values

변경 상태를 Boolean 타입으로 리턴한다.

## 8.2.9. LastCtrl

## Syntax

```
HwpCtrl.LastCtrl
```

## Description

문서 중 마지막 컨트롤. 읽기 전용.

## Return Values

문서의 마지막 컨트롤을 CtrlCode 객체 타입으로 리턴한다.

## Remarks

문서 중의 모든 컨트롤(표, 그림 등의 특수 문자들)은 linked list로 서로 연결되어 있는데, 그 list의 마지막 컨트롤을 나타낸다. 이 컨트롤로부터 시작, Ctrl.Prev를 이용해 backward iteration을 수행할 수 있다.

## Example

[CtrlCode.Prev](#) 참고

## See Also

[HeadCtrl](#)

## 8.2.10. PageCount

### Syntax

```
HwpCtrl.PageCount
```

### Description

문서의 페이지 수. 읽기 전용.

### Description

문서의 페이지 수를 Number 타입으로 리턴한다

### Remarks

문서의 전체 페이지 수를 나타낸다. 문서 전체에 대한 pagination이 수행되지 않은 상태에서 이 property를 참조하면 먼저 문서 전체의 pagination을 먼저 수행하므로 긴 문서에 대해 문서 내용 변경과 참조를 반복하면 속도가 심각하게 느려질 수 있다.

## 8.2.11. ParaShape

### Syntax

```
HwpCtrl.ParaShape
```

### Description

현재 selection의 문단 모양을 나타낸다.

### Return Values

블록된 구역 혹은, 현재 캐럿이 위치한 곳의 문단 모양을 ParameterSet 객체 타입으로 리턴한다.

### Remarks

property get을 수행하면 현재 selection 내의 문단 모양을 구할 수 있다.  
selection이 존재하지 않으면 현재 캐럿이 위치한 곳의 문단 모양을 돌려준다.  
문단 모양 중 특정 항목이 selection 내에서 서로 다른 속성을 가지고 있으면 아예  
아이템 자체가 존재하지 않는다.  
property set을 수행하면 아이템이 존재하는 항목에 대해서만 속성을 설정한다.  
ParameterSet의 형식은 ParameterSet/ParaShape 참조.

### Example

현재 문단의 왼쪽 여백을 "10pt"로 설정한다.

```
var paraShape = HwpCtrl.ParaShape;  
paraShape.SetItem('LeftMargin', 2000);  
HwpCtrl.ParaShape = paraShape;
```

### See Also

[CharShape](#)

## 8.2.12. ParentCtrl

### Syntax

```
HwpCtrl.ParentCtrl
```

### Description

현재 캐럿의 상위 컨트롤. 읽기 전용.

## Return Values

현재 캐럿의 상위 컨트롤을 CtrlCode 객체 타입으로 리턴한다.

## Remarks

상위 컨트롤은 현재 캐럿이 위치한 리스트를 보유한 컨트롤이다. 예를 들어 셀 내부에 위치하면 표, 각주 내용에 위치하면 각주, 바탕쪽이면 구역 컨트롤이 상위 컨트롤이다.

현재 캐럿이 본문 레벨에 위치해 상위 컨트롤이 없을 때는 NULL이 리턴된다.

## Example

현재 캐럿의 위치가 캐럿이라면 부모 표의 셀간격을 1pt로 설정한다.

```
var ctrl = HwpCtrl.ParentCtrl;
if (ctrl.CtrlID == "tbl") {
    var tbset = HwpCtrl.CreateSet("Table");
    tbset.SetItem("CellSpacing", 100);
    ctrl.Properties = tbset;
}
```

## See Also

[CreateSet](#), [ParameterSet.SetItem](#), [CtrlCode.Properties](#)

## 8.2.13. SelectionMode

### Syntax

HwpCtrl.SelectionMode

### Description

문서의 내용이 어떤 selection 상태인가를 알려준다. 읽기 전용.

### Remarks

일반 블록이 아닌 F3키나 F4키에 의해 블록이 지정된 경우, HWPSEL\_STRICT\_MODE ( = 0x10, 십진수 16)으로 OR 마스크되어 오기때문에, 항상 0x0F(십진수15)로 AND 마스크한 결과로 판단하도록 한다.

값	설명
0	블록지정이 되지 않은 상태
1	일반 블록이거나 F3키에 의한 블록 상태
2	F4키에 의한 블록 상태
3	테이블 셀 블록 상태

아래의 예제 참조.

## Example

```
var HWPSEL_NONE = 0;
var HWPSEL_NORMAL = 1;
var HWPSEL_COLUMN = 2;
var HWPSEL_TABLE = 3;
var HWPSEL_CTRL = 4;
var HWPSEL_MODE_MASK = 15;
var HWPSEL_STRICT_MODE = 16;

mode = HwpCtrl.SelectionMode & HWPSEL_MODE_MASK;

if(mode == 0){
    //블록지정이 되지 않은 경우
    console.log("No Selection ...");
} else if(mode == HWPSEL_NORMAL) {
    //일반 블록이거나 F3키에 의한 블록인 경우 (TEXT)
    console.log("Normal Selection ...");
} else if(mode == HWPSEL_COLUMN) {
    //F4키에 의한 블록인 경우 (TEXT)
    console.log("Column Selection ...");
} else if(mode == HWPSEL_TABLE) {
    //테이블 셀 블록 상태
    console.log("Table Selection ...");
} else if(mode == HWPSEL_CTRL) {
    //그리기 개체나 표, 그림 등 ShapeObject 컨트롤이 선택된 상태
    console.log("Ctrl Selection ...");
}
```

## 8.2.14. Version

### Syntax

HwpCtrl.Version
-----------------

### Description

웹한글의 버전을 구한다. 읽기 전용.

## Remarks

기존 컨트롤에서의 Version은 한글 과 한글 OCX의 버전 정보를 구한다.  
그러나 웹한글 기안기는 한글설치과 관계없이 사용되므로 웹한글의 버전을 리턴한다.

## 8.2.15. ViewProperties

### Syntax

HwpCtrl.ViewProperties

### Description

View의 상태 정보

## Remarks

조판 부호, 화면 확대 비율과 같은 view에 관련된 정보를 나타낸다.  
ParameterSet의 형식은 ParameterSet/ViewProperties 참조.

현재는 ZoonType, ZoomRatio에 한해서 제공한다.

ZoomType	화면 확대 종류. 0 = 사용자 정의 1 = 쪽 맞춤 2 = 폭 맞춤 3 = 여러 쪽
ZoomRatio	화면 확대 종류가 "사용자 정의"인 경우 화면 확대 비율. 10 ~ 500 (단위 %)

## Example

화면 확대 비율을 100%로 맞춘다.

```
var vp = HwpCtrl.CreateSet("ViewProperties");
vp.SetItem("ZoomType", 0); //HwpZoomCustom
vp.SetItem("ZoomRatio", 100);
HwpCtrl.ViewProperties = vp;
```

## See Also

[ParameterSet.SetItem](#)

## 8.2.16. ScrollPosInfo

### Syntax



HwpCtrl.ScrollPosInfo

## Description

View 스크롤바 위치 정보

## Remarks

현재 스크롤바(수직, 수평)의 위치 정보를 "ScrollPosInfo" 파라미터 셋으로 얻어 오거나 수정할 수 있다.

· [Set ID] ScrollPosInfo : 스크롤바 위치 정보

Item ID	Description
HorzPos	수평 스크롤바 Position
HorzLimitPos	수평 스크롤바 Limit Position (쓰기 X)
VertPos	수직 스크롤바 Position
VertLimitPos	수직 스크롤바 Limit Position (쓰기 X)

## Example

ScrollPos 수정하기

```
var ScrollPosSet;  
ScrollPosSet = HwpCtrl.ScrollPosInfo;  
ScrollPosSet.SetItem("HorzPos", 100);  
ScrollPosSet.SetItem("VertPos", 100);  
HwpCtrl.ScrollPosInfo = ScrollPosSet;
```

## See Also

[ParameterSet.SetItem](#)

## 8.2.17. ReadOnlyMode

### Syntax

HwpCtrl.ReadOnlyMode

## Description

현재 문서가 읽기 전용인지 여부를 판단하고, 읽기 전용 모드를 설정한다.  
설정값 true, false

## Return Values

읽기 전용인지 Boolean타입으로 리턴한다.

## 8.2.18. CurSelectedCtrl

### Syntax

```
HwpCtrl.CurSelectedCtrl
```

### Description

현재 선택된 컨트롤. 읽기 전용.

### Return Values

현재 선택된 컨트롤을 CtrlCode 객체 타입으로 리턴한다.

### Example

현재 선택된 컨트롤이 표라면 제거한다.

```
var ctrl = HwpCtrl.CurSelectedCtrl;

if(ctrl.CtrlID == 'tbl'){
    HwpCtrl.DeleteCtrl(ctrl);
}
```

### See Also

[DeleteCtrl](#), [CtrlCode.CtrlID](#)

## 8.3. Method

### 8.3.1. Clear

#### Syntax

```
HwpCtrl.Clear(option)
```

#### Description

현재 편집중인 문서의 내용을 닫고 빈문서 편집 상태로 돌아간다.

#### Parameters 1

**option** : 편집중인 문서의 내용에 대한 처리 방법. 생략하면 hwpAskSave가 선택된다.

ID	값	설명
hwpAskSave	0	문서의 내용이 변경되었을 때 사용자에게 저장할지 묻는 대화상자를 띄운다.
hwpDiscard	1	문서 내용을 버린다.
hwpSaveIfDirty	2	문서가 변경된 경우 저장 한다.
hwpSave	3	무조건 저장한다.

## Remarks

format, arg에 대해서는 Open 참조. hwpSaveIfDirty, hwpSave가 지정된 경우 현재 문서 경로가 지정되어 있지 않으면 "새이름으로 저장" 대화상자를 띄워 사용자에게 경로를 묻는다.

현재는 옵션 1에 한해서만 사용할 수 있다.  
다른 옵션 적용시 옵션 1로 동작한다.

## Example

빈문서 편집 상태로 되돌린다.

```
HwpCtrl.Clear(1);
```

## 8.3.2. CreateAction

### Syntax

```
HwpCtrl.CreateAction(actionID)
```

### Description

Action 객체를 생성한다.

### Parameters 1

**actionID** : 액션 ID (별도 문서 참조)

### Return Values

성공시 Action 객체를 리턴한다.  
존재하지 않는 액션 ID를 입력하면 null을 리턴한다.

## Remarks

액션에 대한 세부적인 제어가 필요할 때 사용한다. 예를 들어 기능을 수행하지 않고 대화상자만을 띄운다든지, 대화상자 없이 지정한 옵션에 따라 기능을 수행하는 등에 사용할 수 있다.

## Example

[Action.Run](#) 참고

### 8.3.3. CreateField

#### Syntax

```
HwpCtrl.CreateField(direction, memo, name)
```

#### Description

캐럿의 현재 위치에 누름틀을 생성한다.

#### Parameters 3

**direction** : 누름틀에 입력이 안된 상태에서 보여지는 안내문/지시문.

**memo** : 누름틀에 대한 설명/도움말

**name** : 누름틀 필드에 대한 필드 이름

#### Return Values

성공이면 True, 실패면 False

#### Remark

memo Parameter는 ActiveX 한/글 컨트롤과 호환용으로 사용되는 Parameter이다. 웹한글에서는 내용을 확인할 수 없으나 누름틀 저장 후 데스크탑 한/글에서 메모 내용을 확인할 수 있다.

## Example

현재 위치에 본문 필드를 생성한다.

```
HwpCtrl.CreateField('본문을 입력하세요', ", '본문');
```

### 8.3.4. CreatePageImage

#### Syntax

```
HwpCtrl.CreatePageImage(fileName, pgno, format)
```

#### Description

지정한 페이지의 이미지를 파일로 생성한다.

#### Parameters 5

**fileName** : 생성할 이미지 파일의 경로

**pgno** : 페이지 번호. 0부터 PageCount - 1까지. 생략하면 0이 사용된다.

**format** : 이미지 파일의 포맷. "jpg", "jpeg", "png" 중의 하나. 생략하면 "png"가 사용된다.

### Return Values

성공하면 true, 실패하면 false

### Example

첫 번째 페이지를 이미지로 다운로드한다.

```
HwpCtrl.CreatePageImage("pageimage.png", 0, "png");
```

## 8.3.5. CreateSet

### Syntax

```
HwpCtrl.CreateSet(setid)
```

### Description

parameter set을 생성한다.

### Parameters 1

**setid** : 생성할 parameter set의 ID

### Return Values

생성된 ParameterSet 객체를 리턴한다.

### Example

[CtrlCode.CtrlID](#) 참고

## 8.3.6. DeleteCtrl

### Syntax

```
HwpCtrl.DeleteCtrl(ctrl)
```

### Description

문서내 컨트롤을 삭제한다.

## Parameters 1

**ctrl** : 삭제할 문서내 컨트롤(CtrlCode 객체)

## Return Values

성공하면 true, 실패하면 false

## Example

[CurSelectedCtrl](#) 참고

### 8.3.7. FieldExist

#### Syntax

```
HwpCtrl.FieldExist(field)
```

#### Description

문서 중에 지정한 데이터 필드가 존재하는지 검사한다.

## Parameters 1

**field** : 필드 이름

## Return Values

성공하면 true, 실패하면 false

## Example

문서내에 '이름1' 필드가 있다면 해당 필드의 정렬을 가운데 정렬로 변경하고, '홍길동'이라고 텍스트를 입력한다.

```
if(HwpCtrl.FieldExist('이름1')){  
    HwpCtrl.MoveToField('이름1', true, true, true);  
    HwpCtrl.Run('ParagraphShapeAlignCenter');  
    HwpCtrl.PutFieldText('이름1', '홍길동');  
}
```

### 8.3.8. GetCurFieldName

#### Syntax

```
HwpCtrl.GetCurFieldName(option)
```

## Description

현재 캐럿 위치의 데이터 필드 이름을 구한다.

## Parameters 1

**option** : 다음과 같은 옵션을 지정할 수 있다. 0을 지정하면 모두 off이다.  
생략하면 0이 지정된다.

ID	값	설명
hwpFieldCell	1	셀에 부여된 필드 리스트만을 구한다. hwpFieldClickHere와는 함께 지정할 수 없다.
hwpFieldClickHere	2	누름틀에 부여된 필드 리스트만을 구한다. hwpFieldCell과는 함께 지정할 수 없다.

## Return Values

필드 이름이 돌아온다. 필드 이름이 없는 경우 빈 문자열이 돌아온다.

## Remarks

GetFieldList()의 option 중에 hwpFieldSelection (= 4) 옵션은 사용하지 않는다.

## Example

현재 필드의 이름이 셀1이라면 셀2로 변경해준다.

```
if(HwpCtrl.GetCurFieldName(0) == '셀1'){  
    HwpCtrl.SetCurFieldName('셀2');  
}
```

## 8.3.9. GetFieldList

### Syntax

```
HwpCtrl.GetFieldList(number, option)
```

## Description

문서 중의 필드 리스트를 구한다.

## Parameters 2

**number** : 문서 중에서 동일한 이름의 필드가 여러 개 존재할 때 이를 구별하기 위한 식별 방법을 지정한다.

생략하면 hwpFieldPlain이 지정된다.

ID	값	설명
hwpFieldPlain	0	아무 기호 없이 순서대로 필드 이름이 나열된다.
hwpFieldNumber	1	필드 이름 뒤에 일련번호가 {{#}}와 같은 형식으로 붙는다.
hwpFieldCount	2	필드 이름뒤에 그 이름의 필드가 몇 개 있는지 {{#}}와 같은 형식으로 붙는다.

**option** : 다음과 같은 옵션을 조합할 수 있다. 0을 지정하면 모두 off이다. 생략하면 0이 지정된다.

ID	값	설명
hwpFieldCell	1	셀에 부여된 필드 리스트만을 구한다. hwpFieldClickHere와는 함께 지정할 수 없다.
hwpFieldClickHere	2	누름틀에 부여된 필드 리스트만을 구한다. hwpFieldCell과는 함께 지정할 수 없다.
hwpFieldSelection	4	셀렉션 내에 존재하는 필드 리스트를 구한다.

## Return Values

각 필드 사이를 문자코드 0x02로 구분하여 다음과 같은 형식으로 리턴 한다. (가장 마지막 필드에는 0x02가 붙지 않는다.)

"필드이름#1Wx02필드이름#2Wx02...필드이름#n"

## Remarks

문서 중에 동일한 이름의 필드가 여러 개 존재할 때는 number에 지정한 타입에 따라 3 가지의 서로 다른 방식을 중에서 선택할 수 있다. 예를 들어 문서 중 title, body, title, body, footer 순으로 5 개의 필드가 존재할 때, hwpFieldPlain, hwpFieldNumber, HwpFieldCount 세 가지 형식에 따라 다음과 같은 내용이 돌아온다.

hwpFieldPlain	"titleWx02bodyWx02titleWx02bodyWx02footer"
hwpFieldNumber	"title{{0}}Wx02body{{0}}Wx02title{{1}}Wx02body{{1}}Wx02footer{{0}}"
hwpFieldCount	"title{{2}}Wx02body{{2}}Wx02footer{{1}}"

## Example

```
alert(HwpCtrl.GetFieldList(0));
```

## See Also

[FieldExist](#)



## 8.3.10. GetFieldText

### Syntax

```
HwpCtrl.GetFieldText(fieldlist)
```

### Description

지정한 필드에서 문자열을 구한다.

### Parameters 1

**fieldlist** : 텍스트를 구할 필드 이름의 리스트. 다음과 같이 필드 사이를 문자 코드 0x02로 구분하여 한 번에 여러 개의 필드를 지정할 수 있다.

"필드이름#1Wx02필드이름#2Wx02...필드이름#n"

지정한 필드 이름이 문서 중에 두 개 이상 존재할 때의 표현 방식은 다음과 같다.

필드이름            이름의 필드 중 첫 번째

필드이름{{n}}    지정한 이름의 필드 중 n 번째

예를 들어 "제목{{1}}Wx02본문Wx02이름{{0}}" 과 같이 지정하면 '제목'이라는 이름의 필드 중 두 번째, '본문'이라는 이름의 필드 중 첫 번째, '이름'이라는 이름의 필드 중 첫 번째를 각각 지정한다. 즉, '필드이름'과 '필드이름0'은 동일한 의미로 해석된다.

### Return Values

텍스트 데이터가 돌아온다. 텍스트에서 탭은 'Wt'(0x9), 문단 바꿈은 CR/LF(0x0D/0x0A)로 표현되며, 이외의 특수 코드는 포함되지 않는다. 필드 텍스트의 끝은 0x02로 표현되며, 그 이후 다음 필드의 텍스트가 연속해서 지정한 필드 리스트의 개수만큼 위치한다. 지정한 이름의 필드가 없거나 사용자가 해당 필드에 아무 텍스트도 입력하지 않았으면 해당 텍스트에는 빈 문자열이 돌아온다.

### Example

'본문'이라는 이름의 필드 중 첫 번째 필드의 Text를 얻어온다.

```
alert(HwpCtrl.GetFieldText('본문{{0}}'));
```

## 8.3.11. GetMousePos

### Syntax

```
HwpCtrl.GetMousePos(Xrelto, Yrelto)
```

## Description

마우스의 현재 위치를 구한다<sup>1</sup>

## Parameters

2

**Xrelto** : X좌표계의 기준 위치

value	설명	비고
0	종이 기준	종이 기준으로 좌표를 가져온다.
1	쪽 기준	쪽 기준으로 좌표를 가져온다.

**Yrelto** : Y좌표계의 기준 위치

value	설명	비고
0	종이 기준	종이 기준으로 좌표를 가져온다.
1	쪽 기준	쪽 기준으로 좌표를 가져온다.

## Return Values

"MousePos" ParameterSet이 리턴된다.

· [Set ID] MousePos

Item ID	Description
XRelTo	가로 상대적 기준 0 : 종이 1 : 쪽
YRelTo	세로 상대적 기준 0 : 종이 1 : 쪽
Page	페이지 번호 ( 0 based)
X	가로 클릭한 위치 (HWPUNIT)
Y	세로 클릭한 위치 (HWPUNIT)

## Remarks

단위가 HWPUNIT임을 주의하십시오.

## Example

MouseButtonDown Event가 발생할 때 Mouse의 위치 정보를 log로 찍는다.

```
var OnMouseButtonDown = function(x, y){  
    var MousePosSet = pHwpCtrl.GetMousePos(0, 0);  
    var xrelto = MousePosSet.Item("XRelTo");  
    var yrelto = MousePosSet.Item("YRelTo");
```

```

var page = MousePosSet.Item("Page");
var pagex = MousePosSet.Item("X");
var pagey = MousePosSet.Item("Y");

console.log("xrelto : " + xrelto);
console.log("yrelto : " + yrelto);
console.log("page : " + page);
console.log("pagex : " + pagex);
console.log("pagey : " + pagey);
}

HwpCtrl.addEventListener(0, OnMouseLButtonDown);

```

## See Also

HwpCtrl [Event] [OnMouseLButtonDown](#)

## 8.3.12. GetPos

### Syntax

```
HwpCtrl.GetPos()
```

### Description

현재 캐럿의 위치 정보를 얻어온다.

### Return Values

캐럿 위치 정보를 Object 타입으로 리턴한다. Object의 Property는 다음과 같다.

**list** : 캐럿이 위치한 문서 내 리스트 아이디.

**para** : 캐럿이 위치한 문단 아이디.

**pos** : 캐럿이 위치한 문단 내 글자 단위 위치.

### Remarks

기존 ActiveX 한글 컨트롤에서는 포인터를 인자로 받아서 처리하였다. web 환경에 맞춰서 객체를 리턴하는 방식으로 변경되었다.

### Example

```

var pos = HwpCtrl.GetPos();
alert('GetPos : ' + pos.list + ' ' + pos.para + ' ' + pos.pos);

```

## See Also

[SetPos](#), [SelectText](#)

### 8.3.13. GetPosBySet

#### Syntax

```
HwpCtrl.GetPosBySet()
```

#### Description

현재 캐럿의 위치 정보를 얻어온다. (GetPos의 ParameterSet 버전)

**Parameters**      0

#### Return Values

캐럿의 위치에 대한 ParameterSet 객체를 리턴한다.

#### Example

현재 위치를 ParameterSet 객체 형태로 저장, 이후 저장된 ParameterSet으로 위치를 변경한다(이동한다).

```
var posSet = null;
function GetPosTest() {
    posSet = HwpCtrl.GetPosBySet();

    alert(posSet.Item("List") + " " + posSet.Item("Para") + " " +
posSet.Item("Pos"));
};

function SetPosTest(){
    if ( posSet != null) {
        if (!HwpCtrl.SetPosBySet(posSet)) {
            alert ("GetPos err");
        }
    } else {
        alert("GetPos를 먼저 실행하세요.");
    }
};

GetPosTest();
.....
SetPosTest();
```

## See Also

[GetPos](#), [ParameterSet.Item](#)

## 8.3.14. GetSelectedPos

### Syntax

```
HwpCtrl.GetSelectedPos()
```

### Description

현재 설정된 블록의 위치 정보를 얻어온다.

### Return Values

캐럿 위치 정보를 Object 타입으로 리턴한다. Object의 Property는 다음과 같다.

**slist** : 설정된 블록의 시작 리스트 아이디.

**spara** : 설정된 블록의 시작 문단 아이디.

**spos** : 설정된 블록의 문단 내 시작 글자 단위 위치.

**elist** : 설정된 블록의 끝 리스트 아이디.

**epara** : 설정된 블록의 끝 문단 아이디.

**epos** : 설정된 블록의 문단 내 끝 글자 단위 위치.

### Remarks

기존 ActiveX 한글 컨트롤에서는 포인터를 인자로 받아서 처리하였다. web 환경에 맞춰서 객체를 리턴하는 방식으로 변경되었다.

위의 리스트란, 문단과 컨트롤들이 연결된 한/글 문서 내 구조를 뜻한다.

리스트 아이디는 문서 내 위치 정보 중 하나로서 SelectText에 넘겨줄 때 사용한다.

### Example

현재 Select된 위치를 저장하고, 이후에 그 위치의 Text를 블록한다.

```
var selectedPos = null;
function GetSelectedPosTest() {
    selectedPos = HwpCtrl.GetSelectedPos();
};

function SelectTextTest() {
```

```

        if(selectedPos != null){
            HwpCtrl.SelectText(selectedPos.spara, selectedPos.spos,
selectedPos.epara, selectedPos.epos);
        } else {
            alert("GetSelectedPos를 먼저 실행하세요.");
        }
    };

    GetSelectedPosTest();
    .....
    SelectTextTest();

```

## See Also

[GetSeletedPosBySet](#), [SelectText](#)

## 8.3.15. GetSelectedPosBySet

### Syntax

```
HwpCtrl.GetSelectedPosBySet(sset, eset)
```

### Description

현재 설정된 블록의 위치 정보를 얻어온다. (GetSelectedPos의 ParameterSet 버전)

### Parameters 2

**sset** : 설정된 블록의 시작 ParameterSet (ListParaPos)

**eset** : 설정된 블록의 끝 ParameterSet (ListParaPos)

### Return Values

성공하면 true, 실패하면 false

### Example

현재 블록의 시작 PosSet과 끝 PosSet을 얻어낸다.

```

var sset = HwpCtrl.CreateSet('ListParaPos');
var eset = HwpCtrl.CreateSet('ListParaPos');
HwpCtrl.GetSelectedPosBySet(sset, eset);
alert(
    sset.Item('List') + ';' + sset.Item('Para') + ';' + sset.Item('Pos') + 'Wn' +
    eset.Item('List') + ';' + eset.Item('Para') + ';' + eset.Item('Pos')
);

```

## See Also

[GetSelectedPos](#)

## 8.3.16. GetTableCellAddr

### Syntax

```
HwpCtrl.GetTableCellAddr(type)
```

### Description

현재 커서가 위치한 셀의 주소 정보를 얻어온다.

### Parameters 1

**type** : 얻어올 주소의 type

0 : 행

1 : 열

### Return Values

0~n까지의 행, 열을 얻어온다.

커서가 셀에 있지 않거나, 실패하면 -1을 리턴한다.

### Remarks

아래 표와 같이 셀이 합쳐져 있을 경우에는 셀이 합쳐지기 전 상태에서 좌측상단의 셀의 주소를 얻어온다.

(0, 0)			(3, 0)
	(1, 1)		
(0, 2)			
	(1, 3)		
(0, 4)			(3, 4)

	(1, 0)			
(0, 1)	(1, 1)	(3, 1)		(5, 1)
	(1, 2)	(2, 2)	(4, 2)	
(0, 3)				(6, 3)

## Example

```
var col = HwpCtrl.GetTableCellAddr(0);
var row = HwpCtrl.GetTableCellAddr(1);
console.log("col : " + col + ", row : " + row);
```

## 8.3.17. GetText

### Syntax

```
HwpCtrl.GetText()
```

### Description

문서 중에서 텍스트를 얻는다.

### Return Values

Object 타입으로 리턴한다. Property는 다음과 같다.

**text** : 텍스트 데이터가 돌아온다. 텍스트에서 탭은 'Wt'(0x9), 문단 바꿈은 CR/LF(0x0D/0x0A)로 표현되며, 이외의 특수 코드는 포함되지 않는다.

**result** : 결과값을 Number 타입으로 리턴한다.

다음과 같은 결과값을 리턴한다.

- 0 = 텍스트 정보 없음.
- 1 = 리스트의 끝.
- 2 = 일반 텍스트.
- 3 = 다음 문단.
- 4 = 제어문자 내부로 들어감.
- 5 = 제어 문자를 빠져 나옴.
- 101 = 초기화 안됨. (InitScan() 실패 또는 InitScan()를 실행하지 않은 경우.)
- 102 = 텍스트 변환 실패.

### Remarks

기존 ActiveX 한글 컨트롤에서는 포인터를 인자로 받아서 처리하였다. web 환경에 맞춰서 객체를 리턴하는 방식으로 변경되었다.



GetText()의 사용이 끝나면 ReleaseScan()을 반드시 호출하여 관련 정보를 초기화 해주어야 한다.

텍스트가 있는 문단으로 캐럿을 이동 시키려면 moveScanPos를 주고 MovePos()를 호출하면 된다.

### Example

현재 캐럿부터 줄의 끝까지의 Text를 얻어낸다.

```
HwpCtrl.InitScan(0, 2);  
alert(HwpCtrl.GetText());  
HwpCtrl.ReleaseScan();
```

### See Also

[InitScan](#), [ReleaseScan](#), [MovePos](#)

## 8.3.18. GetTextBySet

### Syntax

```
HwpCtrl.GetTextBySet(text)
```

### Description

문서 중에서 텍스트를 얻는다.(GetText의 ParameterSet 버전)

### Parameters 1

**text** : Item의 "Text"에 텍스트 데이터가 돌아온다. 텍스트에서 탭은 'wt'(0x9), 문단 바꿈은 CR/LF(0x0D/0x0A)로 표현되며, 이외의 특수 코드는 포함되지 않는다.

### Return Values

다음과 같은 결과값을 리턴한다.

- |     |  |
|-----|--|
| 0   | = 텍스트 정보 없음.   |
| 1   | = 리스트의 끝.  |
| 2   | = 일반 텍스트.  |
| 3   | = 다음 문단.   |
| 4   | = 제어문자 내부로 들어감.                                      |
| 5   | = 제어 문자를 빠져 나옴.                                      |
| 101 | = 초기화 안됨. (InitScan() 실패 또는 InitScan()를 실행하지 않은 경우.) |
| 102 | = 텍스트 변환 실패.   |

## Remarks

GetTextBySet()의 사용이 끝나면 ReleaseScan()을 반드시 호출하여 관련 정보를 초기화 해주어야 한다.

텍스트가 있는 문단으로 캐럿을 이동 시키려면 moveScanPos를 주고 MovePos()를 호출하면 된다.

## Example

현재 캐럿부터 줄의 끝까지의 Text를 얻어낸다.

```
HwpCtrl.InitScan(0, 2);  
var textSet = HwpCtrl.CreateSet('GetText');  
HwpCtrl.GetTextBySet(textSet);  
alert(textSet.Item("Text"));  
HwpCtrl.ReleaseScan();
```

## See Also

[InitScan](#), [ReleaseScan](#), [MovePos](#), [GetText](#)

## 8.3.19. GetTextFile

### Syntax

```
HwpCtrl.GetTextFile(format, option, callback)
```

### Description

현재 열린 문서를 문자열로 넘겨준다.

### Parameters 3

**format** : 파일의 형식

format	설명	비고
HWP	HWP native format	BASE64로 인코딩되어 있다. 저장된 내용을 다른 곳에서 보여줄 필요가 없다면 이 포맷을 사용하기를 권장합니다.
HWPML2X	HWP 형식과 호환	문서의 모든 정보를 유지
HTML	인터넷 문서 HTML 형식	한글 고유의 서식은 손실된다.
UNICODE	유니코드 텍스트	서식정보가 없는 텍스트만 저장 Text 옵션과 동일하게 동작한다. ActiveX 한/글 Ctrl과 호환을 위해 존재하는

		옵션으로 사용을 권장하지 않는다.
TEXT	유니코드 텍스트	서식정보가 없는 텍스트만 저장
JSON	json format	문서의 정보를 Json 형태로 만든다. 문서의 모든 정보를 유지한다.

#### option

option	설명	비고
saveblock	선택된 블록만 저장	개체 선택 상태에서는 동작하지 않는다.

**callback** : 콜백함수

서버를 통하여 Text를 얻어내므로 반드시 이후 동작을 콜백함수에 넣어주어야 한다.

### Return Values

저장된 텍스트로 파일을 String 타입으로 리턴한다.

### Example

```
var text = "";
var OnSetTextFile = function () {
    HwpCtrl.GetTextFile("HWP", "", function(data) {
        text = data;
        alert(text);
    });
};

var OnSetTextFile = function () {
    if(text)
        HwpCtrl.SetTextFile(text, "HWP", "");
    else
        alert('GetTextFile을 먼저 실행해주세요.')
};
```

## 8.3.20. GetViewStatus

### Syntax

```
HwpCtrl.GetViewStatus()
```

### Description

현재 보이고 있는 View의 절대 Pos값을 알려준다.

## Parameters

**nType** : 뷰의 정보를 얻기위한 Flag (0일 경우 View의 절대 Pos값을 알려준다.)

## Remarks

· [Set ID] ViewStatus : 뷰 상태 정보

Item ID	Description
Type	0 (현재 View의 절대값 Pos를 알려준다.)
ViewPosX	현재 뷰의 X값
ViewPosY	현재 뷰의 Y값

## Example

```
//한글 전체 페이지에 대하여 현재 보여지는 뷰의Left 좌표
var viewLeft = 0;

//한글 전체 페이지에 대하여 현재 보여지는 뷰의Top 좌표
var viewTop = 0;
var viewSet = HwpCtrl.GetViewStatus(0);

viewLeft = viewSet.Item("ViewPosX");
viewTop = viewSet.Item("ViewPosY");
console.log(" Left = " + viewLeft + ", Top = " + viewTop)
```

## 8.3.21. InitScan

### Syntax

```
HwpCtrl.InitScan(option, rang, spara, spos, epara, epos)
```

### Description

문서 검색을 위한 준비 작업을 한다.

### Parameters 6

**option** : 찾을 대상을 다음과 같은 옵션을 조합하여 지정할 수 있다. 생략하면 모든 컨트롤을 찾을 대상으로 한다.

ID	값	설명
maskNormal	0x00	본문을 대상으로 검색한다. (서브리스트를 검색하지 않는다.)
maskChar	0x01	char 타입 컨트롤 마스크를 대상으로 한다. (강제줄나눔, 문단끝, 하이픈, 묶음빈칸, 고정폭빈칸, 등...)
maskInline	0x02	inline 타입 컨트롤 마스크를 대상으로 한다. (누름틀 필드 끝, 등...)
maskCtrl	0x04	extende 타입 컨트롤 마스크를 대상으로 한다. (바탕쪽, 프리젠테이션, 다단, 누름틀 필드 시작, Shape Object, 머리말, 꼬리말, 각주, 미주, 번호관련 컨트롤, 새번호 관련 컨트롤, 감추기, 찾아보기, 글자겹침, 등...)

**range** : 검색의 범위를 다음과 같은 옵션을 조합하여 지정할 수 있다. 생략하면 “문서 시작부터 - 문서의 끝까지” 검색 범위가 지정된다.

ID	값	설명
scanSposCurrent	0x0000	캐럿 위치부터. (시작 위치)
scanSposSpecified	0x0010	특정 위치부터. (시작 위치)
scanSposLine	0x0020	줄의 시작부터. (시작 위치)
scanSposParagraph	0x0030	문단의 시작부터. (시작 위치)
scanSposSection	0x0040	구역의 시작부터. (시작 위치)
scanSposList	0x0050	리스트의 시작부터. (시작 위치)
scanSposControl	0x0060	컨트롤의 시작부터. (시작 위치)
scanSposDocument	0x0070	문서의 시작부터. (시작 위치)
scanEposCurrent	0x0000	캐럿 위치까지. (끝 위치)
scanEposSpecified	0x0001	특정 위치까지. (끝 위치)
scanEposLine	0x0002	줄의 끝까지. (끝 위치)
scanEposParagraph	0x0003	문단의 끝까지. (끝 위치)
scanEposSection	0x0004	구역의 끝까지. (끝 위치)
scanEposList	0x0005	리스트의 끝까지. (끝 위치)
scanEposControl	0x0006	컨트롤의 끝까지. (끝 위치)
scanEposDocument	0x0007	문서의 끝까지. (끝 위치)
scanWithinSelection	0x00ff	검색의 범위를 블록으로 제한.
scanForward	0x0000	정방향. (검색 방향)
scanBackward	0x0100	역방향. (검색 방향)

**spara** : 검색 시작 위치의 문단 번호. scanSposSpecified 옵션이 지정되었을 때만 유효하다.

**spos** : 검색 시작 위치의 문단 중에서 문자의 위치. scanSposSpecified 옵션이 지정되었을 때만 유효하다.

**epara** : 검색 끝 위치의 문단 번호. scanEposSpecified 옵션이 지정되었을 때만 유효하다.

**epos** : 검색 끝 위치의 문단 중에서 문자의 위치. scanEposSpecified 옵션이 지정되었을 때만 유효하다.

## Return Values

성공하면 true, 실패하면 false

## Remarks

문서의 검색 과정은 InitScan()으로 검색위한 준비 작업을 하고 GetText()를 호출하여 본문의 텍스트를 얻어온다. GetText()를 반복호출하면 연속하여 본문의 텍스트를 얻어올 수 있다. 검색이 끝나면 ReleaseScan()을 호출하여 관련 정보를 초기화해 주면 된다.

## Example

[GetText](#) 참고

## See Also

[GetText](#), [ReleaseScan](#)

## 8.3.22. Insert

### Syntax

```
HwpCtrl.Insert(path, format, arg, callback, callbackUserData)
```

### Description

현재 캐럿 위치에 문서 파일을 삽입한다.

### Parameters

5

**path** : URL, 서버 path, blob

서버 path : 서버에 올라가 있는 파일의 경로를 받을 수 있다.

blob : input을 통해서 blob파일을 인자로 받을 수 있다.

**format** : 문서 형식. [Open API 참고](#). 별도 설명 참조. 빈 문자열을 지정하면 자동으로 디텍트한다. 생략하면 빈 문자열이 지정된다.

**arg** : 세부 옵션. [Open API 참고](#). 의미는 format에 지정한 파일 형식에 따라 다르

다. 생략하면 빈 문자열이 지정된다.

**Callback** : Insert 함수가 동작을 마친 후 수행될 함수

- Parameter 1

result: 성공 실패 여부, FileName, Size, CallbackUserData를 포함한 Object

**CallbackUserData** : Callback함수에 Argument로 넘겨 받을 result에 포함될 UserData, 생략 가능.

## Return Values

성공하면 true, 실패하면 false

Argument가 제대로 들어왔는지 정도의 판단을 할 수 있으며 실제로 문서 Insert의 성공 여부는 콜백함수의 인자로 들어가는 result에서 확인할 수 있다.

## Remarks

브라우저에서는 로컬파일 접근 권한이 없어 로컬 파일은 사용할 수 없다.  
서버 path의 경우 이후 제외될 수 있다.

## Example

```
HwpCtrl.Insert(BasePath + "test.hwp", "HWP", "", function (res) {  
    console.log(res);  
}, {"UserData" : "testData"});
```

## See Also

[Open](#)

## 8.3.23. InsertBackgroundPicture

### Syntax

```
HwpCtrl.InsertBackgroundPicture(bordertype, path, embedded, filloption,  
watermark, effect, brightnss, contrast)
```

### Description

배경이미지를 넣는다.

## Parameters 8

**bordertype** : 배경 유형을 지정

bordertype	설명	비고
"SelectedCell"	현재 선택된 표의 셀의 배경을 변경한다.	
"SelectedCellDelete"	현재 선택된 표의 셀의 배경을 지운다.	반드시 셀이 선택되어 있어야 함. 커서가 위치하는 것만으로는 동작하지 않음.

**path** : 이미지 URL

**embedded** : 이미지 파일을 문서내에 포함할지 여부 (True/False). 생략하면 True

**filloption** : 삽입할 그림의 크기를 지정하는 옵션

filloption	설명	비고
0	바둑판식으로 - 모두	
1	바둑판식으로 - 가로/위	
2	바둑판식으로 - 가로/아로	
3	바둑판식으로 - 세로/왼쪽	
4	바둑판식으로 - 세로/오른쪽	
5	크기에 맞추어	설정하지 않았을 때 기본값
6	가운데로	
7	가운데 위로	
8	가운데 아래로	
9	왼쪽 가운데로	
10	왼쪽 위로	
11	왼쪽 아래로	
12	오른쪽 가운데로	
13	오른쪽 위로	
14	오른쪽 아래로	

**effect** : 이미지효과

**watermark** : watermark효과 유무 (True/False)

이 옵션이 true이면 brightness 와 contrast 옵션이 무시된다.

### effect

effect	설명	비고
0	원래 그림	설정하지 않았을 때 기본값
1	그레이 스케일	
2	흑백으로	

**brightness** : 밝기 지정(-100 ~ 100), 기본값 : 0



**contrast** : 선명도 지정(-100 ~ 100), 기본값 : 0

## Return Values

성공했을 경우 true, 실패했을 경우 false

## Remarks

CellBorderFill의 SetItem 중 FillAttr 의 SetItem FileName 에 이미지의 binary data 를 지정해 줄 수가 없어서 만든 함수다.

기타 배경에 대한 다른 조정은 Action과 ParameterSet의 조합으로 가능하다.

브라우저에서는 로컬파일 접근 권한이 없어 로컬 파일은 사용할 수 없다.

## Example

현재 캐럿이 셀안에 위치할 때 현재 셀의 셀 테두리 배경이미지가 존재 한다면 삭제하고, 없다면 새롭게 넣는다.

```
function InsertBgImg() {
    HwpCtrl.InsertBackgroundPicture("SelectedCell", BaseURL+
    "logo1.bmp", 1, 5, 0, 0, 0, 0);
}

var DeleteBgImg= function() {
    HwpCtrl.InsertBackgroundPicture("SelectedCellDelete", 0, 0, 0, 0, 0,
    0, 0);
};

var TestBgImg = function() {
    var act;
    var set;
    var subset;
    if (HwpCtrl.ParentCtrl != null && HwpCtrl.ParentCtrl.CtrlID == "tbl"){
        HwpCtrl.Run("TableCellBlock"); // 반드시 셀을 선택해야 한다.
        act = HwpCtrl.CreateAction("CellBorderFill");
        set = act.CreateSet();
        act.GetDefault(set);

        if (set.ItemExist("FillAttr")) {
            subset = set.Item("FillAttr");
            if(subset.ItemExist("Type")) {
                if((subset.Item("Type") & 2) == 2) {
                    alert("셀 테두리 배경이미지 존재");
                }
            }
        }
    }
}
```

```

        return {IsCell: true, hasImage : true };
    } else {
        alert("셀 테두리 배경이미지가 존재하지
        않음");
        return {IsCell: true, hasImage : false };
    }
}
} else {
    alert("셀 테두리 배경에 대한 정보가 존재하지 않음");
    return {IsCell: true, hasImage : false };
}
HwpCtrl.Run("Cancel");
} else {
    alert("현재 위치가 표의 셀이 아님.");
    return {IsCell: false, hasImage : false };
}
};

var InsertBackGroundPictureTest = function () {
    var result = TestBgImg();
    if(result.IsCell == true){
        if(result.hasImage == false){
            InsertBgImg();
        } else {
            DeleteBgImg();
        }
    }
};

```

## See Also

[InsertPicture](#)

## 8.3.24. InsertCtrl

### Syntax

```
HwpCtrl.InsertCtrl(ctrlid, initparam)
```

### Description

현재 캐럿 위치에 컨트롤을 삽입한다.

## Parameters 2

**ctrlid** : 삽입할 컨트롤의 ID

**initparam** : 컨트롤의 초기 속성. 생략하면 디폴트 속성으로 생성한다.

## Return Values

생성된 CtrlCode 객체를 리턴한다.

## Remarks

ctrlid에 지정할 수 있는 컨트롤 ID는 HwpCtrl.CtrlID가 리턴하는 ID와 동일하다. 자세한 것은 Ctrl 오브젝트 Properties인 CtrlID를 참조.

initparam에는 컨트롤의 초기 속성을 지정한다. 대부분의 컨트롤은 Ctrl.Properties와 동일한 포맷의 parameter set을 사용하지만, 컨트롤 생성시에는 다른 포맷을 사용하는 경우도 있다. 예를 들어 표의 경우 Ctrl.Properties에는 "Table" 셋을 사용하지만, 생성시 initparam에 지정하는 값은 "TableCreation" 셋이다.

## Example

5x5 크기의 표를 삽입한다.

```
var tbact = HwpCtrl.CreateAction("TableCreate");
var tbset = tbact.CreateSet();
tbset.SetItem("Rows", 5);
tbset.SetItem("Cols", 5);
var table = HwpCtrl.InsertCtrl("tbl", tbset);
```

## See Also

[CreateAction](#), [Action.CreateSet](#)

## 8.3.25. InsertPicture

### Syntax

```
HwpCtrl.InsertPicture(path, embedded, sizeoption, reverse, watermark, effect,
width, height, callback)
```

### Description

현재 캐럿의 위치에 그림을 삽입한다.

## Parameters 8

**path** : 이미지 URL

**embedded** : 이미지 파일을 문서내에 포함할지 여부 (True/False). 생략하면 True

**sizeoption** : 삽입할 그림의 크기를 지정하는 옵션

sizeoption	설명
0	이미지 원래의 크기로삽입한다. width와 height를 지정할 필요없다.
1	width와 height에 지정한 크기로 그림을 삽입한다.
2	width는 셀의 width만큼, height는 셀의 height만큼 확대/축소된다.
3	현재 캐럿이 표의 셀안에 있을 경우, 셀의 크기에 맞추어 원본 이미지의 가로 세로의 비율이 동일하게 확대/축소하여 삽입한다.

**reverse** : 이미지의 반전 유무 (True/False)

**watermark** : watermark효과 유무 (True/False)

**effect** : 그림 효과

sizeoption	설명
0	실제 이미지 그대로
1	그레이 스케일
2	흑백효과

**width** : 그림의 가로 크기 지정. 단위는 mm

**height** : 그림의 높이 크기 지정. 단위는 mm

**callback** : InsertPicture 함수가 동작을 마친 후 수행될 함수

- Parameter 1

ctrl: 성공시 생성된 컨트롤 오브젝트, 실패시 null

### Remark

브라우저에서는 로컬파일 접근 권한이 없어 로컬 파일은 사용할 수 없다.

### Example

```
HwpCtrl.InsertPicture(BaseURL + 'logo1.png', true, 1, false, false, 0, 17, 17,
function(ctrl){
    if(ctrl){
        console.log('성공');
    } else {
        console.log('실패');
    }
});
```

## 8.3.26. IsCommandLock

### Syntax

```
HwpCtrl.IsCommandLock(actionID)
```

## Description

특정 액션이 잠금 상태인지 여부를 조사한다.

## Parameters 3

**actionID** : 액션 ID

## Return Values

해당 액션이 잠금상태이면 True를 리턴하고, 잠금상태가 아니면 False를 리턴한다.

## Example

Bold Action을 잠금한다.

```
HwpCtrl.LockCommand('CharShapeBold', true);  
var isCommandLock = HwpCtrl.IsCommandLock('CharShapeBold');  
if(isCommandLock){  
    console.log('Bold Action을 실행할 수 없습니다.')  
}else{  
    console.log('Bold Action을 실행할 수 있습니다.')  
}
```

## See Also

[LockCommand](#)

## 8.3.27. KeyIndicator

### Syntax

```
HwpCtrl.KeyIndicator();
```

## Description

상태바에 나타날 정보를 알아낸다.

## Return Values

상태바에 나타날 정보를 Object형태로 리턴한다.

**result** : 성공 실패 여부

**seccnt** : 총 구역

**secno** : 현재 구역

prnpageno : 쪽  
colno : 단  
line : 줄  
pos : 칸  
over :  
true:수정  
false:삽입

## Remarks

컨트롤 바깥쪽에서 상태바를 만들어서 상태바에 표시할 정보들의 내용을 알아낼 때 유용하다.

## Example

상태바에 나타낼 정보들을 얻어낸다.

```
var result = HwpCtrl.KeyIndicator();  
alert("result : "+ result.result + "Wn" +  
      "총 구역 : " + result.seccnt + "Wn" +  
      "현재 구역 : " + result.secno + "Wn" +  
      "쪽 : " + result.prnpageno + "Wn" +  
      "단 : " + result.colno + "Wn" +  
      "줄 : " + result.line + "Wn" +  
      "칸 : " + result.pos + "Wn" +  
      "over : " + result.over + "Wn");
```

## 8.3.28. LockCommand

### Syntax

```
HwpCtrl.LockCommand(actionID, lock)
```

### Description

특정 액션이 실행되지 않도록 잠근다.

### Parameters 3

**actionID** : 액션 ID

**lock** : True이면 액션의 실행을 lock시키고, False이면 unlock시킨다.

### Example

[IsCommandLock](#) 참고

## See Also

[IsCommandLock](#)

## 8.3.29. ModifyFieldProperties

### Syntax

```
HwpCtrl.ModifyFieldProperties(field, remove, add)
```

### Description

지정한 필드의 속성을 바꾼다.

### Parameters 3

**field** : 속성을 바꿀 필드 이름의 리스트. 형식은 PutFieldText과 동일하다.

**remove** : 제거될 속성

**add** : 추가될 속성

### Return Values

음수가 리턴되면 에러임을 나타낸다.

그 외의 리턴값은 아래 remark 참조

### Remarks

속성의 값은 아래와 같다.

long value	설명
0x00000001	양식모드에서 편집가능 속성 (0: 편집 불가, 1: 편집 가능)

return 값의 bit field는 다음과 같다.

long value bit mask	설명
0x00000001	양식모드에서 편집가능 속성 (0: 편집 불가, 1: 편집 가능)
0x80000000	에러
0x40000000	필드를 찾을 수 없음

remove와 add에 둘다 0이 입력되면 현재 속성을 돌려준다.

리턴값이 음수인지 확인하여 쉽게 에러임을 판별할 수 있으며 자세한 에러내용은 bit mask로 and 연산하여 알아 낼 수 있다.

리턴값은 여러 가지 추가 정보가 같이 올 수 있으므로 반드시 bit mask를 사용하여 비교해야 한다.

## Example

양식모드에서 편집 가능 옵션 변경

```
var ModifyFieldPropertiesTest = function() {  
  
    var ret;  
    ret = HwpCtrl.ModifyFieldProperties("test", 0, 0);  
    if (ret < 0 ) {  
        var strErrMsg = "test 필드의 정보를 알아오는 데 실패";  
  
        if(ret & 0x40000000)  
            strErrMsg += "Wn지정한 필드를 찾을 수 없습니다.";  
        alert(strErrMsg);  
        return;  
    }  
  
    if (ret & 0x00000001){  
        alert("현재 양식모드에서 편집 가능");  
        ret = HwpCtrl.ModifyFieldProperties("test", 1, 0);  
    } else{  
        alert("현재 양식모드에서 편집 불가능");  
        ret = HwpCtrl.ModifyFieldProperties("test", 0, 1);  
    }  
    if (ret < 0 ) {  
        alert("필드 속성 변경 실패.");  
    }  
  
    if (ret & 0x00000001)  
        alert("양식모드에서 편집 가능하도록 변경되었습니다.");  
    else  
        alert("양식모드에서 편집 불가능하도록 변경되었습니다.");  
}
```

## 8.3.30. MovePos

### Syntax

```
HwpCtrl.MovePos(moveID, para, pos)
```

### Description

캐럿의 위치를 옮긴다.

### Parameters 3

**moveID** : 다음과 같은 값을 지정할 수 있다. 생략하면 moveCurList가 지정된다.

ID	값	설명
moveMain	0	루트 리스트의 특정 위치.(para pos로 위치 지정)
moveCurList	1	현재 리스트의 특정 위치.(para pos로 위치 지정)
moveTopOfFile	2	문서의 시작으로 이동.
moveBottomOfFile	3	문서의 끝으로 이동.



ID	값	설명
moveTopOfList	4	현재 리스트의 시작으로 이동
moveBottomOfList	5	현재 리스트의 끝으로 이동
moveStartOfPara	6	현재 위치한 문단의 시작으로 이동
moveEndOfPara	7	현재 위치한 문단의 끝으로 이동
moveStartOfWord	8	현재 위치한 단어의 시작으로 이동. (현재 리스트만을 대상으로 동작한다.)
moveEndOfWord	9	현재 위치한 단어의 끝으로 이동. (현재 리스트만을 대상으로 동작한다.)
moveNextPara	10	다음 문단의 시작으로 이동. (현재 리스트만을 대상으로 동작한다.)
movePrevPara	11	앞 문단의 끝으로 이동. (현재 리스트만을 대상으로 동작한다.)
moveNextPos	12	한 글자 뒤로 이동. (서브 리스트를 옮겨다닐 수 있다.)
movePrevPos	13	한 글자 앞으로 이동. (서브 리스트를 옮겨다닐 수 있다.)
moveNextPosEx	14	한 글자 뒤로 이동. (서브 리스트를 옮겨다닐 수 있다.) (머리말/꼬리말, 각주/미주, 글상자 포함.)
movePrevPosEx	15	한 글자 앞으로 이동. (서브 리스트를 옮겨다닐 수 있다.) (머리말/꼬리말, 각주/미주, 글상자 포함.)
moveNextChar	16	한 글자 뒤로 이동. (현재 리스트만을 대상으로 동작한다.)
movePrevChar	17	한 글자 앞으로 이동. (현재 리스트만을 대상으로 동작한다.)
moveNextWord	18	한 단어 뒤로 이동. (현재 리스트만을 대상으로 동작한다.)
movePrevWord	19	한 단어 앞으로 이동. (현재 리스트만을 대상으로 동작한다.)
moveNextLine	20	한 줄 아래로 이동.
movePrevLine	21	한 줄 위로 이동.
moveStartOfLine	22	현재 위치한 줄의 시작으로 이동.
moveEndOfLine	23	현재 위치한 줄의 끝으로 이동.
moveParentList	24	한 레벨 상위로 이동한다.
moveTopLevelList	25	탑레벨 리스트로 이동한다.
moveRootList	26	루트 리스트로 이동한다. 추가 설명 : 현재 루트 리스트에 위치해 있어 더 이상 상위 리스트가 없을 때는 위치 이동 없이 리턴한다. 이동한 후의 위치는 상위 리스트에서 서브리스트가 속한 컨트롤 코드가 위치한 곳이다. 위치 이동시 셀렉션은 무조건 풀린다.
moveCurrentCaret	27	현재 캐럿이 위치한 곳으로 이동한다. (캐럿 위치가 뷰의 맨 위쪽으로 올라간다. )
moveLeftOfCell	100	현재 캐럿이 위치한 셀의 왼쪽
moveRightOfCell	101	현재 캐럿이 위치한 셀의 오른쪽
moveUpOfCell	102	현재 캐럿이 위치한 셀의 위쪽
moveDownOfCell	103	현재 캐럿이 위치한 셀의 아래쪽
moveStartOfCell	104	현재 캐럿이 위치한 셀에서 행(row)의 시작

ID	값	설명
moveEndOfCell	105	현재 캐럿이 위치한 셀에서 행(row)의 끝
moveTopOfCell	106	현재 캐럿이 위치한 셀에서 열(column)의 시작
moveBottomOfCell	107	현재 캐럿이 위치한 셀에서 열(column)의 끝
moveScrPos	200	한/글 문서장에서의 screen 좌표로서 위치를 설정 한다.
moveScanPos	201	GetText() 실행 후 위치로 이동한다.

**para** : 이동할 문단의 번호.

moveMain 또는 moveCurList가 지정되었을 때만 사용된다. moveScrPos가 지정되었을 때는 문단번호가 아닌 스크린 좌표로 해석된다.

(스크린 좌표 : LOWORD = x 좌표, HIWORD = y 좌표)

**pos** : 이동할 문단 중에서 문자의 위치.

moveMain 또는 moveCurList가 지정되었을 때만 사용된다.

## Return Values

성공하면 true, 실패하면 false

## Remarks

moveScrPos가 지정되었을 때는 스크린 좌표는 마우스 커서의 (x,y) 좌표를 그대로 넘겨 주면 된다.

moveScanPos는 문서를 검색하는 중에 캐럿을 이동 시키려 할 경우에만 사용이 가능하다.

## Example

특정 위치로 위치를 이동한다.

```
....  
HwpCtrl.MovePos(0 , selectedPos.spara, selectedPos.spos);
```

## 8.3.31. MoveToField

### Syntax

```
HwpCtrl.MoveToField(field, text, start, select)
```

## Description

지정한 필드로 캐럿을 이동한다.

## Parameters 4

**field** : 필드 이름. GetFieldText/PutFieldText과 같은 형식으로 이름 뒤에 '{{#}}'로 번호를 지정할 수 있다.

**text** : 필드가 누름틀일 경우 누름틀 내부의 텍스트로 이동할지(True) 누름틀 코드로 이동할지(False)를 지정한다. 누름틀이 아닌 필드일 경우 무시된다. 생략하면 True가 지정된다.

**start** : 필드의 처음(True)으로 이동할지 끝(False)으로 이동할지 지정한다. select를 True로 지정하면 무시된다. 생략하면 True가 지정된다.

**select** : 필드 내용을 블록으로 선택할지(True), 캐럿만 이동할지(False) 지정한다. 생략하면 False가 지정된다.

## Return Values

성공하면 true, 실패하면 false

## Remarks

누름틀은 한글 메뉴 중 [입력 - 개체 - 필드 입력] 메뉴를 선택하면 누름틀을 만드실 수 있습니다.

## Example

본문 필드로 이동한다.

```
HwpCtrl.MoveToField('본문', true, true, false);
```

## See Also

## 8.3.32. MoveToFieldEx

### Syntax

```
HwpCtrl.MoveToFieldEx(field, text, start, boolean, select)
```

### Description

지정한 필드로 캐럿을 이동한 후 캐럿 위치로 화면을 이동한다.

## Parameters 4

**field** : 필드 이름. GetFieldText/PutFieldText과 같은 형식으로 이름 뒤에 '{{#}}'로 번호를 지정할 수 있다.

**text** : 필드가 누름틀일 경우 누름틀 내부의 텍스트로 이동할지(True) 누름틀 코드

로 이동할지(False)를 지정한다. 누름틀이 아닌 필드일 경우 무시된다. 생략하면 True가 지정된다.

**start** : 필드의 처음(True)으로 이동할지 끝(False)으로 이동할지 지정한다. select를 True로 지정하면 무시된다. 생략하면 True가 지정된다.

**select** : 필드 내용을 블록으로 선택할지(True), 캐럿만 이동할지(False) 지정한다. 생략하면 False가 지정된다.

## Return Values

성공하면 true, 실패하면 false

## Remarks

누름틀은 한글 메뉴 중 [입력 - 개체 - 필드 입력] 메뉴를 선택하면 누름틀을 만드실 수 있습니다.

## Example

본문 필드로 이동한다.

```
HwpCtrl.MoveToFieldEx('본문', true, true, false);
```

## See Also

## 8.3.33. Open

### Syntax

```
HwpCtrl.Open(path, format, arg, callback, callbackUserData)
```

### Description

문서 파일을 연다

### Parameters 5

**path** : URL, 서버 path, blob

서버 path : 서버에 올라가 있는 파일의 경로를 받을 수 있다.

blob : input을 통해서 blob파일을 인자로 받을 수 있다.

**format** : 문서 형식. 별도 설명 참조. 빈 문자열을 지정하면 자동으로 디텍트한다. 생략하면 빈 문자열이 지정된다.

**arg** : 세부 옵션. 의미는 format에 지정한 파일 형식에 따라 다르다. 생략하면 빈 문자열이 지정된다.

**Callback** : Open 함수가 동작을 마친 후 수행될 함수

- Parameter 1

result: 성공 실패 여부, FileName, Size, CallbackUserData를 포함한 Object

**CallbackUserData** : Callback함수에 Argument로 넘겨 받을 result에 포함될 UserData, 생략 가능.

## Return Values

성공하면 true, 실패하면 false

Argument가 제대로 들어왔는지 정도의 판단을 할 수 있으며 실제로 문서 Open의 성공 여부는 콜백함수의 인자로 들어가는 result에서 확인할 수 있다.

## Remarks

브라우저에서는 로컬파일 접근 권한이 없어 로컬 파일은 사용할 수 없다.  
서버 path의 경우 이후 제외될 수 있다.

일부 옵션이 변경될 수 있다.

format에 지정할 수 있는 문서 형식은 다음과 같은 종류가 있다.

format은 아래에 써있는대로 대문자로만 써야 한다.

"HWP"	한글 native format
"HWP30"	한글 3.X/96/97
"HTML"	인터넷 문서
"TEXT"	아스키 텍스트 문서
"UNICODE"	유니코드 텍스트 문서
"HWP20"	한글 2.0
"HWP21"	한글 2.1/2.5
"HWP15"	한글 1.X
"HWPML1X"	HWPML 1.X 문서 (Open만 가능)
"HWPML2X"	HWPML 2.X 문서 (Open / SaveAs 가능)
"RTF"	서식있는 텍스트 문서
"DBF"	DBASE II/III 문서
"HUNMIN"	훈민정음 3.0/2000
"MSWORD"	마이크로소프트 워드 문서
"HANA"	하나워드 문서
"ARIRANG"	아리랑 문서
"ICHITARO"	一太郎 문서 (일본 워드프로세서)
"WPS"	WPS 문서

"DOCIMG"	인터넷 프레젠테이션 문서(SaveAs만 가능)
"SWF"	Macromedia Flash 문서(SaveAs만 가능)
"PUBDOCBODY"	공문서(xml)

arg에 지정할 수 있는 옵션의 의미는 필터가 정의하기에 따라 다르지만, 신택스는 다음과 같이 공통된 형식을 사용한다.

key:value;key:value;...

\* key는 A-Z, a-z, 0-9, \_ 로 구성된다.

\* value는 타입에 따라 다음과 같은 3 종류가 있다.

boolean: ex) fullsave:true (== fullsave)

integer: ex) type:20

string: ex) prefix:\_This\_

\* value는 생략 가능하며, 이때는 콜론도 생략한다.

\* arg에 지정할 수 있는 옵션

"모든 파일"

setcurdir(boolean, FALSE) : 로드한 후 해당 파일이 존재하는 폴더로 현재 위치를 변경한다. hyperlink 정보가 상대적인 위치로 되어 있을 때 유용하다.

"HWP"

lock (boolean, TRUE) : 로드한 후 해당 파일을 계속 오픈한 상태로 lock을 걸지 여부

notext (boolean, FALSE) : 텍스트 내용을 읽지 않고 헤더 정보만 읽을지 여부. (스타일 로드 등에 사용)

template (boolean, FALSE): 새로운 문서를 생성하기 위해 템플릿 파일을 오픈한다. 이 옵션이 주어지면 lock은 무조건 FALSE로 처리된다.

suspendpassword (boolean, FALSE): TRUE로 지정하면, 암호가 있는 파일일 경우 암호를 묻지 않고 무조건 읽기에 실패한 것으로 처리한다.

forceopen (boolean, FALSE): TRUE로 지정하면, 읽기 전용으로 읽어야 하는 경우 대화상자를 띄우지 않는다.

versionwarning (boolean, FALSE): TRUE로 지정하면, 문서가 상위버전일 경우 메시지 박스를 띄우게 된다.

"HTML"

code(string, codepage) : 문서 변환시 사용되는 코드 페이지를 지정할 수 있으며

code키가 존재할 경우 필터 사용시 사용자 다이얼로그를 띄우지 않는다.

textunit(boolean, pixel) : Export될 Text의 크기의 단위 결정 pixel, point, mili 지정 가능.

formatunit(boolean, pixel) : Export될 문서 포맷 관련 (마진, Object 크기 등) 단위 결정. pixel, point, mili 지정 가능

"DOCIMG"

asimg(boolean, FALSE) : 저장할 때 페이지를 image로 저장

ashtml(boolean, FALSE) : 저장할 때 페이지를 html로 저장

[codepage 종류]

ks : 한글 KS 완성형 kssm: 한글 조합형 |sjis : 일본 |utf8 : UTF8 | unicode: 유니코드 | gb : 중국 간체 | big5 : 중국 번체 | acp : Active Codepage 현재 시스템의 코드 페이지

"TEXT"

code(string, codepage): 문서 변환시 사용되는 코드 페이지를 지정할 수 있으며 code키가 존재할 경우 필터 사용시 사용자 다이얼로그를 띄우지 않는다.

[codepage 종류]

ks : 한글 KS 완성형 | kssm : 한글 조합형 |sjis : 일본 | gb : 중국 간체 | big5 : 중국 번체

kps: 북한(한글 2004) | acp : Active Codepage 현재 시스템의 코드 페이지

## Example

```
HwpCtrl.Open(BasePath + "test.hwp", "HWP", "", function (res) {
    console.log(res);
}, {"UserData" : "testData"});
```

## See Also

[Insert](#)

## 8.3.34. PutFieldText

### Syntax

```
HwpCtrl.PutFieldText(fieldlist, textlist)
```

### Description

지정한 필드의 내용을 채운다.

## Parameters 2

**fieldlist** : 내용을 채울 필드 이름의 리스트. 한 번에 여러 개의 필드를 지정할 수 있으며, 형식은 GetFieldText와 동일하다. 다만 필드 이름 뒤에 '#'로 번호를 지정하지 않으면 해당 이름을 가진 모든 필드에 동일한 텍스트를 채워 넣는다. 즉, PutFieldText에서는 '필드이름'과 '필드이름0'의 의미가 다르다.

**textlist** : 필드에 채워 넣을 문자열의 리스트. 형식은 필드 리스트와 동일하게 필드의 개수만큼 텍스트를 0x02로 구분하여 지정한다.

## Return Values

### Remarks

현재 필드에 입력되어 있는 내용은 지워진다. 채워진 내용의 글자모양은 필드에 지정해 놓은 글자모양을 따라간다.

fieldlist의 필드 개수와, textlist의 텍스트 개수는 동일해야 한다.

존재하지 않는 필드에 대해서는 무시한다.

## Example

```
HwpCtrl.PutFieldText('이름1', '홍길동');
```

## See Also

## 8.3.35. ReleaseScan

### Syntax

```
HwpCtrl.ReleaseScan()
```

### Description

InitScan()으로 설정된 정보를 초기화 한다.

## Example

[GetText](#) 참고

## See Also

[InitScan](#), [GetText](#)

## 8.3.36. RenameField

### Syntax



```
HwpCtrl.RenameField(oldname, newname)
```

## Description

지정한 필드의 이름을 바꾼다.

## Parameters 2

**oldname** : 이름을 바꿀 필드 이름의 리스트. 형식은 PutFieldText과 동일하다.

**newname** : 새로운 필드 이름의 리스트. oldname과 동일한 개수의 필드 이름을 0x02로 구분하여 지정한다.

## Return Values

### Remarks

예를 들어 oldname에 "title{{0}}\wx02title{{1}}", newname에 "tt1\wx02tt2"로 지정하면 첫 번째 title은 tt1로, 두 번째 title은 tt2로 변경된다.

oldname의 필드 개수와, newname의 필드 개수는 동일해야 한다.  
존재하지 않는 필드에 대해서는 무시한다.

## Example

```
HwpCtrl.PutFieldText('이름1', '이름2');
```

## See Also

## 8.3.37. ReplaceAction

### Syntax

```
HwpCtrl.ReplaceAction(oldActionID, newActionID)
```

## Description

메뉴나 단축키로 호출되는 Action을 교체한다.

## Parameters 2

**oldActionID** : 교체할 원본 Action의 이름

**newActionID** : 교체될 새로운 Action의 이름

## Return Values

성공여부, ActionID 가 잘못된 경우에는 실패.

## Remarks

메뉴나 단축키로 호출되는 Action을 교체할 뿐, CreateAction이나, Run 등의 API를 이용할 때는 아무런 영향을 주지 못한다.

교체된 OldActionID를 원래대로 되돌려 놓기 위해서는 OldActionID와 NewActionID의 이름을 동일하게 하여 호출한다.

OldActionID와 NewActionID가 구조적으로 연결된 경우에는 해당 Action을 실행하면 무한 루프를 돌게 된다.

예:

```
ReplaceAction(FileSave, FileSaveAs);
```

```
ReplaceAction(FileSaveAs, FileSave);
```

를 둘다 호출한 경우 FileSave나, FileSaveAs를 호출하면 Stack overflow가 발생한다.

## Example

CharShapeBold Action의 이름을 BoldAction으로 변경한다.

```
HwpCtrl.ReplaceAction('CharShapeBold', 'BoldAction');
```

## See Also

### 8.3.38. Run

#### Syntax

```
HwpCtrl.Run(actionID)
```

#### Description

액션을 실행한다.

#### Parameters 1

**actionID** : 액션 ID (별도 문서 참조)

#### Example

[FieldExist](#) 참고

## See Also

[CreateAction](#)

### 8.3.39. SaveAs

#### Syntax

`HwpCtrl.SaveAs(fileName, format, arg, callback, callbackUserData)`

#### Description

현재 편집중인 문서를 지정한 이름으로 저장한다.

#### Parameters 5

**fileName** : 로컬에 다운로드할 때에 파일의 이름을 지정한다.

**arg** 인자에 `download:true` 포함되어 있을 때 로컬에 저장된다. 그때 사용되는 파일 이름.

**format** : 문서 형식. 별도 설명 참조. 생략하면 "HWP"가 지정된다.

**arg** : 세부 옵션. 의미는 `format`에 지정한 파일 형식에 따라 다르다. 생략하면 빈 문자열이 지정된다.

`download:true`를 넣으면 문서가 로컬에 다운로드 된다.

**Callback** : SaveAs 함수가 동작을 마친 후 수행될 함수

- Parameter 1

**result**: 성공 실패 여부, **path**: 저장경로, **fileName**, **Size**, **CallbackUserData**를 포함한 Object

**CallbackUserData** : Callback함수에 Argument로 넘겨 받을 **result**에 포함될 UserData, 생략 가능.

#### Return Values

성공하면 `true`, 실패하면 `false`

#### Remarks

`format`, `arg`의 일반적인 개념에 대해서는 Open 참조.

"HWP" 형식으로 파일 저장시 `arg`에 지정할 수 있는 옵션은 다음과 같다.

함수	인자 타입	기본 값(Default)	설명
lock	Boolean	true	저장한 후 해당 파일을 계속 오픈한 상태로 lock을 걸지 여부
backup	Boolean	false	백업 파일 생성 여부
compress	Boolean	true	압축 여부
fullsave	Boolean	false	스토리지 파일을 완전히 새로 생성하여 저장
prvimage	Number	2	미리보기 이미지(0=off, 1=BMP, 2=GIF)
prvtext	Number	1	미리보기 텍스트(0=off, 1=on)
autosave	Boolean	false	자동저장 파일로 저장할 지 여부 (true: 자동저장, false: 지정 파일로 저장)
export			다른이름으로 저장은 하고 열린문서는 바꾸지 않는다. (lock:false와 함께 설정 되어 있을 시 동작)
download	Boolean	false	true: DownLoad폴더에 다운로드, 다운로드 하지 않음.

Example

lock:TRUE

backup:FALSE

prvimage:1

여러 개를 한꺼번에 할 경우에는 세미콜론으로 구분하여 연속적으로 사용할 수 있다.

lock:TRUE;backup:FALSE;prvtext:1

### Example

현재 열려있는 문서를 다운로드 폴더에 다운로드한다.

```
HwpCtrl.SaveAs("test.hwp", "HWP", "download:true");
```

### See Also

[Open](#), [Save](#)

## 8.3.40. SelectText

### Syntax

```
HwpCtrl.SelectText(spara, spos, epara, epos)
```

### Description

블록을 설정한다.

#### Parameters 4

**spara** : 블록 시작 위치의 문단 번호.  
**spos** : 블록 시작 위치의 문단 중에서 문자의 위치.  
**epara** : 블록 끝 위치의 문단 번호.  
**epos** : 블록 끝 위치의 문단 중에서 문자의 위치.

#### Return Values

성공하면 true, 실패하면 false

#### Remarks

epos가 가리키는 문자는 포함되지 않는다.

#### Example

```
..  
HwpCtrl.SelectText(selectedPos.spara, selectedPos.spos, selectedPos.epara,  
selectedPos.epos);
```

#### See Also

[GetPos](#), [GetPosBySet](#), [GetSelectedPos](#), [GetSelectedPosBySet](#)

### 8.3.41. SetCurFieldName

#### Syntax

```
HwpCtrl.SetCurFieldName(fieldname, option, direction, memo)
```

#### Description

현재 캐럿 위치의 데이터 필드 이름을 설정한다.

#### Parameters 4

**fieldname** : 데이터 필드 이름.  
**option** : 다음과 같은 옵션을 지정할 수 있다. 0을 지정하면 모두 off이다.  
생략하면 0이 지정된다.

ID	값	설명
hwpFieldCell	1	셀에 부여된 필드 리스트만을 구한다. hwpFieldClickHere와는 함께 지정할 수 없다.
hwpFieldClickHere	2	누름틀에 부여된 필드 리스트만을 구한다. hwpFieldCell과는 함께 지정할 수 없다.

**direction** : 누름틀 필드의 안내문. 누름틀 필드일 때만 유효하다.

**memo** : 누름틀 필드의 메모. 누름틀 필드일 때만 유효하다.

## Return Values

성공하면 true, 실패하면 false

## Remarks

GetFieldList()의 option 중에 hwpFieldSelection (= 4) 옵션은 사용하지 않는다.

## Example

[GetCurFieldName](#) 참고

## See Also

[GetFieldList](#), [GetCurFieldName](#), [CurFieldState](#)

## 8.3.42. SetFieldViewOption

### Syntax

```
HwpCtrl.SetFieldViewOption(option)
```

### Description

양식 모드와 읽기 전용모드일때 현재 열린 문서의 필드의 겉보기 속성(『』표시)을 바꾼다.

### Parameters 1

**option** : 겉보기 속성 bit

option	누름틀	개인정보/문서요약/날짜 시간/파일경로	비고
1	『』을 표시하지 않음	『』을 표시하지 않음	
2	『』을 빨간색으로 표시	『』을 흰색으로 표시	설정하지 않았을 때 기본값
3	『』을 흰색으로 표시	『』을 흰색으로 표시	

### Return Values

설정된 속성이 리턴된다.

에러일 경우 0 이 리턴된다.

### Remarks

EditMode와 비슷하게 현재 열려있는 문서에 대한 속성이다. 따라서 저장되지 않는다.

### Example

누름틀 표시색을 흰색으로 변경한다.

```
HwpCtrl.SetFieldViewOption(2);
```

### See Also

[EditMode](#)

## 8.3.43. SetPos

### Syntax

```
HwpCtrl.SetPos(list, para, pos)
```

### Description

캐럿을 문서 내 특정 위치로 위치시킨다.

### Parameters 3

**list** : 캐럿이 위치할 문서 내 리스트 아이디.  
**para** : 캐럿이 위치할 문단 아이디.  
**pos** : 캐럿이 위치할 문단 내 글자 단위 위치.

### Return Values

성공시 true, 실패시 false

### Example

[GetPos](#) 참고

### See Also

[GetPos](#), [SelectText](#)

## 8.3.44. SetPosBySet

### Syntax

HwpCtrl.SetPosBySet(pos)

### Description

캐럿을 문서 내 특정 위치로 위치시킨다. (SetPos의 ParameterSet 버전)

### Parameters 1

**pos** : 캐럿의 위치에 대한 ParameterSet

### Return Values

성공하면 true, 실패하면 false

### Example

[GetPosBySet](#) 참고

### See Also

[SetPos](#)

## 8.3.45. SetTextFile

### Syntax

HwpCtrl.SetTextFile(data, format, option, callback, callbackUserData)

### Description

문서를 문자열로 지정한다.

### Parameters 3

**data** : 문자열로 변경된 text 파일

**format** : 파일의 형식



format	설명	비고
HWP	HWP native format	BASE64 로 인코딩되어 있어야 한다. 저장된 내용을 다른 곳에서 보여줄 필요가 없다면 이 포맷을 사용하기를 권장합니다.
HWPML2X	HWP 형식과 호환	문서의 모든 정보를 유지
HTML	인터넷 문서 HTML 형식	한글 고유의 서식은 손실된다.
UNICODE	유니코드 텍스트	서식정보가 없는 텍스트만 저장 Text 옵션과 동일하게 동작한다. ActiveX 한/글 Ctrl과 호환을 위해 존재하는 옵션으로 사용을 권장하지 않는다.
TEXT	유니코드 텍스트	서식정보가 없는 텍스트만 저장

#### option

option	설명	비고
insertfile	현재커서 이후에 지정된 파일 삽입	

**callback** : SetTextFile 함수가 동작을 마친 후 수행될 함수

- Parameter 1

result: 성공 실패 여부, CallbackUserData를 포함한 Object를 인자로 받는다.

**callbackUserData** : Callback함수에 Argument로 넘겨 받을 result에 포함될 UserData, 생략 가능.

#### Return Values

성공하면 true, 실패하면 false

#### Remark

return받은 boolean 값은 Argument가 제대로 들어왔고, 함수가 제대로 동작을 하였는지를 판별할 수 있고, 서버를 통해서 문서가 제대로 SetTextFile 됐는지 여부는 콜백함수의 인자로 들어오는 result에서 확인할 수 있다.

#### Example

[GetTextFile](#) 참고

### 8.3.46. SetToolBar

#### Syntax

```
HwpCtrl.SetToolBar(option, ToolBarID)
```

## Description

툴바를 등록/보이기/감추기/삭제할 수 있다.

## Parameters 2

**option :**

option	설명
0	추가
1	삭제
2	보이기
3	감추기

현재는 0, 1 옵션만 제공한다.

**ToolBarID :** 미리 정의된 툴바의 이름이 올 수 있다.

Defined Toolbar name	설명
TOOLBAR_MENU	메뉴
TOOLBAR_STANDARD	기본 도구 상자

차후 추가 예정.

## Return Values

성공하면 true, 실패하면 false

## Remarks

기존 ActiveX 한글컨트롤이 제공하던 커스텀 툴바는 향후 지원 예정이다.

## Example

메뉴만 남기고 리본, 툴박스를 제거한다.

```
HwpCtrl.SetToolBar(1, "TOOLBAR_MENU");  
HwpCtrl.SetToolBar(0, "TOOLBAR_STANDARD");  
HwpCtrl.ShowToolBar(true);  
HwpCtrl.ShowRibbon(false);
```

## See Also

[ShowToolBar](#)

## 8.3.47. ShowStatusBar

## Syntax

```
HwpCtrl.ShowStatusBar(show)
```

## Description

상태바를 표시하거나 감춘다.

## Parameters

1

**Show** : 상태바의 표시 여부 (true or false)

## Return Values

동작 여부(true or false)

## Remarks

아무 동작도 하지 않았으면 false를 리턴한다. 즉, 현재 상태바가 보여진 상태에서 ShowStatusBar(true)을 호출하거나, 현재 상태바가 감춰진 상태에서 ShowStatusBar(false)를 호출하면 false를 리턴한다.

## Example

상태바를 숨긴다.

```
HwpCtrl.ShowStatusBar(false);
```

## See Also

## 8.3.48. ShowToolBar

### Syntax

```
HwpCtrl.ShowToolBar(show)
```

### Description

툴바를 화면에 Show/Hide한다.

### Parameters

1

**show** : 툴바 화면 표시 여부

true : show

false : hide

### Return Values

성공하면 true, 실패하면 false

## Remarks

SetToolBar로 등록한 툴바 정보에 따라 툴바를 화면 표시>Show)하거나, 숨긴다(Hide).

주의: 툴바를 숨길 경우 SetToolBar로 등록한 메뉴역시 툴바의 일종이므로 같이 사라진다.

## Example

툴바를 숨긴다.

```
HwpCtrl.ShowToolBar(false);
```

## See Also

[SetToolBar](#)

## 8.3.49. ShowRibbon

### Syntax

```
HwpCtrl.ShowRibbon(show)
```

### Description

리본 메뉴를 화면에 Show/Hide한다.

### Parameters 1

**show** : 리본 메뉴 표시 여부

true : show

false : hide

### Return Values

성공하면 true, 실패하면 false

## Example

리본을 숨긴다.

```
HwpCtrl.ShowRibbon(false);
```

## See Also

[ShowToolBar](#)

## 8.3.50. OpenDocument

### Syntax

```
HwpCtrl.OpenDocument(path, format, callback)
```

### Parameters 3

**path** : URL, 서버 path, blob

서버 path : 서버에 올라가 있는 파일의 경로를 받을 수 있다.

blob : input을 통해서 blob파일을 인자로 받을 수 있다.

**format** : 문서 형식. 별도 설명 참조. 빈 문자열을 지정하면 자동으로 디텍트한다.  
생략하면 빈 문자열이 지정된다.

**Callback** : OpenDocument 함수가 동작을 마친 후 수행될 함수

- Parameter 1

result: 성공 실패 여부, FileName, Size

### Return Values

성공하면 true, 실패하면 false

Argument가 제대로 들어왔는지 정도의 판단을 할 수 있으며 실제로 문서 Open의 성공 여부는 콜백함수의 인자로 들어가는 result에서 확인할 수 있다.

### Remarks

브라우저에서는 로컬파일 접근 권한이 없어 로컬 파일은 사용할 수 없다.  
서버 path의 경우 이후 제외될 수 있다.

### Example

```
HwpCtrl.OpenDocument(BasePath + "test.hwp", "HWP", function (res)
{
    console.log(res);
});
```

### See Also

[Open](#)

## 8.3.51. SaveDocument

### Syntax

```
HwpCtrl.SaveDocument(fileName, format, callback)
```

## Parameters 2

**fileName** : 로컬에 다운로드할 파일의 이름을 지정한다.

**format** : 문서 형식. 별도 설명 참조. 생략하면 "HWP"가 지정된다.

**Callback** : SaveDocument 함수가 동작을 마친 후 수행될 함수

- Parameter 1

result: 성공 실패 여부, path: 저장경로, fileName, Size, CallbackUserData를 포함한 Object

## Return Values

성공하면 true, 실패하면 false

## Remarks

성공하면 Download 폴더에 문서가 저장된다.

## Example

현재 열려있는 문서를 다운로드 폴더에 다운로드한다.

```
HwpCtrl.SaveDocument("test.hwp", "HWP");
```

## See Also

[SaveAs](#)

## 8.3.52. InsertDocument

### Syntax

```
HwpCtrl.InsertDocument(path, callback)
```

### Description

현재 캐럿 위치에 문서 파일을 삽입한다.

## Parameters 2

**path** : URL, 서버 path, blob

서버 path : 서버에 올라가 있는 파일의 경로를 받을 수 있다.

blob : input을 통해서 blob파일을 인자로 받을 수 있다.

**callback** : InsertDocument 함수가 동작을 마친 후 수행될 함수

- Parameter 1

result: 성공 실패 여부, FileName, Size를 포함한 Object

## Return Values

성공하면 true, 실패하면 false

Argument가 제대로 들어왔는지 정도의 판단을 할 수 있으며 실제로 문서 Insert의 성공 여부는 콜백함수의 인자로 들어가는 result에서 확인할 수 있다.

## Remarks

브라우저에서는 로컬파일 접근 권한이 없어 로컬 파일은 사용할 수 없다.  
서버 path의 경우 이후 제외될 수 있다.

## Example

현재 캐럿 위치에 test.hwp 파일을 삽입한다.

```
HwpCtrl.InsertDocument(BasePath + "test.hwp", "HWP", function (res) {  
    console.log(res);  
});
```

## See Also

[Insert](#)

## 8.3.53. IsSpellCheckCompleted

### Syntax

```
HwpCtrl.IsSpellCheckCompleted()
```

### Description

맞춤법검사가 완료되었는지를 판단한다.

**Parameters**      **0**

### Return Values

성공하면 true, 실패하면 false

### Example

```
if (HwpCtrl.IsSpellCheckCompleted()){  
    alert("맞춤법검사가 완료되었습니다.")  
}
```

## 8.3.54. GetHeadingString

### Syntax

```
HwpCtrl.GetHeadingString()
```

### Description

현재 캐럿이 위치한 문단의 문단번호/글머리표를 얻어낸다.

### Parameters 0

### Return Values

문단번호 또는 글머리표를 String 타입으로 return한다.  
문단번호나 글머리표가 없다면 빈문자열을 return한다.

### Example

```
alert("문단번호/글머리표 : " + HwpCtrl.GetHeadingString());
```

## 8.3.55. GetCtrlHorizontalOffset

### Syntax

```
HwpCtrl.GetCtrlHorizontalOffset(ctrl, relTo)
```

### Description

컨트롤의 X 좌표를 얻어낸다.



## Parameters 1

**ctrl** : 대상 컨트롤

**relTo** : X좌표계의 기준 위치

value	설명	비고
0	종이 기준	종이 기준으로 좌표를 가져온다.
1	쪽 기준	쪽 기준으로 좌표를 가져온다.
2	단 기준	단 기준으로 좌표를 가져온다.
3	문단 기준	문단 기준으로 좌표를 가져온다.

## Return Values

X좌표를 Number 타입으로 return 한다.

## Remarks

단위가 HWPUNIT임을 주의하십시오.

## Example

```
var GetCtrlHorizontalOffset = function () {  
    var ctrl = HwpCtrl.CurSelectedCtrl;  
    if(ctrl){  
        var offsetByPaper = HwpCtrl.GetCtrlHorizontalOffset(ctrl, 0);  
        var offsetByPage = HwpCtrl.GetCtrlHorizontalOffset(ctrl, 1);  
        var offsetByPara = HwpCtrl.GetCtrlHorizontalOffset(ctrl, 2);  
  
        alert('PaperOffset : ' + offsetByPaper + 'Wn'  
            + 'PageOffset : ' + offsetByPage + 'Wn'  
            + 'ParaOffset : ' + offsetByPara + 'Wn')  
    } else {  
        alert('Ctrl 객체를 선택해주세요');  
    }  
};
```

## See Also

## 8.3.56. GetCtrlVerticalOffset

### Syntax

```
HwpCtrl.GetCtrlVerticalOffset(ctrl, relTo)
```

### Description

컨트롤의 Y 좌표를 얻어낸다.

### Parameters 1

**ctrl** : 대상 컨트롤

**relTo** : Y좌표계의 기준 위치

value	설명	비고
0	종이 기준	종이 기준으로 좌표를 가져온다.
1	쪽 기준	쪽 기준으로 좌표를 가져온다.
2	문단 기준	문단 기준으로 좌표를 가져온다.

### Return Values

Y좌표를 Number 타입으로 return 한다.

### Remarks

단위가 HWPUNIT임을 주의하십시오.

### Example

```
var GetCtrlVerticalOffset = function () {  
    var ctrl = HwpCtrl.CurSelectedCtrl;  
    if(ctrl){  
        var offsetByPaper = HwpCtrl.GetCtrlVerticalOffset(ctrl, 0);  
        var offsetByPage = HwpCtrl.GetCtrlVerticalOffset(ctrl, 1);  
        var offsetByPara = HwpCtrl.GetCtrlVerticalOffset(ctrl, 2);  
  
        alert('PaperOffset : ' + offsetByPaper + 'Wn'  
            + 'PageOffset : ' + offsetByPage + 'Wn'  
            + 'ParaOffset : ' + offsetByPara + 'Wn')  
    } else {  
        alert('Ctrl 객체를 선택해주세요');  
    }  
}
```

```
}  
};
```

#### See Also

[GetCtrlHorizontalOffset](#)

### 8.3.57. SolarToLunar

#### Syntax

```
HwpCtrl.SolarToLunar(solarYear, solarMonth, solarDay)
```

#### Description

양력을 음력으로 변환한다.

#### Parameters 1

**solarYear** : 양력 연도

**solarMonth** : 양력 달

**solarDay** : 양력 날

#### Return Values

날짜 정보를 Object 타입으로 return 한다. Object의 Property는 다음과 같다.

**year** : 음력 연도

**month** : 음력 달

**day** : 음력 날

**leap** : 윤달 여부

**result** : 성공시 true, 실패시 false

#### Example

```
var lunarDate = HwpCtrl.SolarToLunar(2018, 8, 8);  
alert('양력 2018/8/8 은 음력 ' + lunarDate.year + '/' + lunarDate.month +  
      '/' + lunarDate.day + '입니다.');
```

#### See Also

[LunarToSolar](#)

### 8.3.58. SolarToLunarBySet

#### Syntax

```
HwpCtrl.SolarToLunarBySet(solarYear, solarMonth, solarDay)
```

## Description

양력을 음력으로 변환한다.

## Parameters 1

**solarYear** : 양력 연도

**solarMonth** : 양력 달

**solarDay** : 양력 날

## Return Values

날짜 정보를 ParameterSet 객체 타입으로 return 한다.

## Example

```
var lunarDateSet = HwpCtrl.SolarToLunarBySet(2018, 8, 8);  
alert('양력 2018/8/8 은 음력 ' + lunarDateSet.Item('Year') + '/' +  
lunarDateSet.Item('Month') + '/' + lunarDateSet.Item('Day') + '입니다.');
```

## See Also

[LunarToSolarBySet](#)

## 8.3.59. LunarToSolar

### Syntax

```
HwpCtrl.LunarToSolar(lunarYear, lunarMonth, lunarDay, leap)
```

## Description

음력을 양력으로 변환한다.

## Parameters 1

**lunarYear** : 음력 연도

**lunarMonth** : 음력 달

**lunarDay** : 음력 날

**leap** : 음력 날

## Return Values

날짜 정보를 Object 타입으로 return 한다. Object의 Property는 다음과 같다.

**year** : 양력 연도  
**month** : 양력 달  
**day** : 양력 날  
**result** : 성공시 true, 실패시 false

## Example

```
var solarDate = HwpCtrl.LunarToSolar(2018, 8, 8, false);  
alert('음력 2018/8/8 은 양력 ' + solarDate.year + '/' + solarDate.month +  
      '/' + solarDate.day + '입니다.');
```

## See Also

[SolarToLunar](#)

## 8.3.60. LunarToSolarBySet

### Syntax

```
HwpCtrl.LunarToSolarBySet(lunarYear, lunarMonth, lunarDay, leab)
```

### Description

음력을 양력으로 변환한다.

### Parameters 1

**lunarYear** : 음력 연도  
**lunarMonth** : 음력 달  
**lunarDay** : 음력 날  
**leap** : 음력 날

### Return Values

날짜 정보를 ParameterSet 객체 타입으로 return 한다.

## Example

```
var solarDateSet = HwpCtrl.LunarToSolarBySet(2018, 8, 8, false);  
alert('음력 2018/8/8 은 양력 ' + solarDateSet.Item('Year') + '/' +  
      solarDateSet.Item('Month') + '/' + solarDateSet.Item('Day') + '입니다.');
```

## See Also

### 8.3.61. ShowVerticalScroll

#### Syntax

```
HwpCtrl.ShowVerticalShow(show)
```

#### Description

세로 스크롤을 숨김/보이기 처리한다.

#### Parameters 1

**show** : 세로 스크롤 표시 여부

true : show

false : hide

#### Example

세로 스크롤을 숨긴다.

```
HwpCtrl.ShowVerticalScroll(false);
```

#### See Also

[ShowHorizontalScroll](#)

### 8.3.62. ShowHorizontalScroll

#### Syntax

```
HwpCtrl.ShowHorizontalScroll(show)
```

#### Description

가로 스크롤을 숨김/보이기 처리한다.

#### Parameters 1

**show** : 가로 스크롤 표시 여부

true : show

false : hide

#### Example

가로 스크롤을 숨긴다.

```
HwpCtrl.ShowHorizontalScroll(false);
```

#### See Also

[ShowVerticalScroll](#)

### 8.3.63. ShowCaret

#### Syntax

```
HwpCtrl.ShowCaret(show)
```

#### Description

캐럿을 숨김/보이기 처리한다.

#### Parameters 1

**show** : 캐럿 표시 여부

true : show

false : hide

#### Example

캐럿을 숨긴다.

```
HwpCtrl.ShowCaret(false);
```

### 8.3.64. CreatePageImageEx

#### Syntax

```
HwpCtrl.CreatePageImage(fileName, pgno, resolution, depth, format,  
option, callback, callbackUserData)
```

#### Description

지정한 페이지의 이미지를 파일로 생성한다.

#### Parameters 5

**fileName** : 생성할 이미지 파일의 경로

**pgno** : 페이지 번호. 0부터 PageCount - 1까지. 생략하면 0이 사용된다.

**resolution** : 이미지 해상도. DPI단위(96, 300, 1200등)로 지정한다. 생략하면 96이

사용된다.

**depth** : 이미지 파일의 color depth(1, 4, 8, 24)를 지정한다.

**format** : 이미지 파일의 포맷. "jpg", "jpeg", "png" 중의 하나. 생략하면 "png"가 사용된다.

**option** : download:true를 넣으면 문서가 로컬에 다운로드 된다.

**callback** : CreatePageImageEx 함수가 동작을 마친 후 수행될 함수

- Parameter 1

result: 성공 실패 여부, path: 저장경로, fileName, Size, CallbackUserData를 포함한 Object를 인자로 받는다.

**callbackUserData** : Callback함수에 Argument로 넘겨 받을 result에 포함될 UserData, 생략 가능.

## Example

첫 번째 페이지를 이미지로 다운로드한다.

```
HwpCtrl.CreatePageImageEx("pageimage.png", 0, 96, 24, "png",  
"download:true", function(result){  
    console.log("다운로드 완료");  
});
```

## Return Values

성공하면 true, 실패하면 false

여기서 성공/실패는 필요한 인자가 제대로 들어 있고, 비동기 로직을 제외한 부분이 제대로 동작하였는지를 판별한다.

비동기 액션이므로 return 값이 true이더라도 네트워크등의 문제로 실패 할 수 있다.

정확한 성공/실패 여부는 callback함수의 인자로 들어오는 object의 result 속성을 통해서 판별할 수 있다.

## 8.3.65. GetPageText

### Syntax

```
HwpCtrl.GetPageText(pageNo, option)
```



## Description

특정 페이지의 텍스트를 얻는다.

## Parameters 1

**pageNo** : 텍스트를 얻을 페이지 Number

**option** : 다음과 같은 옵션을 지정할 수 있다. 생략하면 모든 컨트롤에 대해 텍스트를 얻는다.

본문 텍스트는 옵션에 관계없이 얻는다.

1 : 표 내부 텍스트만 얻는다.

2 : 글상자 내부 텍스트만 얻는다

4 : 캡션 내부 텍스트만 얻는다.

option	설명	비고
1	표	
2	글상자	
4	캡션	

## Example

첫 번째 쪽의 텍스트(표 내부 텍스트를 포함하여)를 얻어낸다.

```
alert(HwpCtrl.GetPageText(0, 1));
```

## 8.3.66. PrintDocument

### Syntax

```
HwpCtrl.PrintDocument()
```

### Description

문서를 PDF로 변환하여 보여준다.

### Remarks

웹 환경에서 인쇄 드라이버에 직접 접근할 수 없다. 때문에 웹브라우저 상의 PDF 뷰어 혹은 로컬 PDF 뷰어를 통해서 인쇄 설정 및 인쇄를 할 수 있도록 문서를 PDF 포맷으로 변환한다.

## 8.3.67. AddEventListener

## Syntax

```
HwpCtrl.AddEventListener(eventType, listener)
```

## Description

이벤트 리스너를 추가한다.

## Parameters

1

**eventType** : 리스너를 추가할 이벤트의 타입

value	설명	비고
0	OnMouseDown	왼쪽 마우스 다운 Event
1	OnMouseUp	왼쪽 마우스 업 Event
2	OnScroll	스크롤 Event

**listener** : 이벤트 발생시 호출될 콜백 함수

콜백함수 원형:

OnMouseDown(x, y)

OnMouseUp(x, y)

OnScroll()

## Return Values

성공하면 true, 실패하면 false

## Example

OnMouseDown, OnMouseUp, OnScroll Event의 Listener를 등록한다.

```
HwpCtrl.AddEventListener(0, function(x, y){
    console.log('OnMouseDown:: x : ' + x + ', y : ' + y);
});

HwpCtrl.AddEventListener(1, function(x, y){
    console.log('OnMouseUp:: x : ' + x + ', y : ' + y);
});

HwpCtrl.AddEventListener(2, function(){
    console.log('OnScroll');
});
```

## See Also

[OnMouseButtonDown](#), [OnMouseButtonUp](#), [OnScroll](#)

## 8.4. Event

### 8.4.1. OnMouseButtonDown

#### Syntax

```
OnMouseButtonDown(x, y)
```

#### Description

마우스의 왼쪽 버튼이 눌렸음을 알린다.

#### Parameters 2

**x** : x 좌표

**y** : y 좌표

#### Remarks

AddEventListener API를 통해 등록한다.

좌표의 값은 한글의 View 영역의 좌측 상단을 원점으로 한 Pixel 단위의 값이다.

#### Example

[AddEventListener](#) 참고

## See Also

[OnMouseButtonUp](#), [AddEventListener](#)

### 8.4.2. OnMouseButtonUp

#### Syntax

```
OnMouseButtonUp(x, y)
```

#### Description

마우스의 왼쪽 버튼이 눌렀다가 떨어졌음을 알린다.

#### Parameters 2

**x** : x 좌표

y : y 좌표

### Remarks

AddEventListener API를 통해 등록한다.

### Example

[AddEventListener](#) 참고

### See Also

[OnMouseLButtonDown](#), [AddEventListener](#)

## 8.4.3. OnScroll

### Syntax

OnScroll
----------

**Parameters**      **0**

### Remarks

AddEventListener API를 통해 등록한다.

### Example

[AddEventListener](#) 참고