



# HwpCtrl API

## 1. Action Object

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.

각 **Prototype**에서 명시한 파라미터의 타입의 종류는 다음과 같다.

Type	설명	VARIANT type	member of VARIANT
BSTR	문자열	VT_BSTR	bstrVal
boolean / BOOL	BOOL	VT_BOOL	boolVal
byte	1 byte unsigned char	VT_UI1	bVal
unsigned short	2 byte unsigned integer	VT_UI2	uiVal
unsigned long	4 byte unsigned integer	VT_UI4	ulVal
char	1 byte signed char	VT_I1	cVal
short	2 byte signed integer	VT_I2	iVal
long	4 byte signed integer	VT_I4	lVal
int	signed integer	VT_INT	intVal
unsigned int	unsigned integer	VT_UINT	uintVal

한글 액션을 이용하면서 주의해야할 점은 하나의 액션을 **Create**한 후 **Execute**하기 전까지 또 다른 액션을 **Create**하면 안된다.

예를 들어,

```
act1.CreateAction();
act2.CreateAction();
...
act1.Execute();
act2.Execute();
```

위와 같이 사용할 경우 잘못된 동작을 수행하게 되는 경우가 발생할 수 있다.

- Example

HTML

```
var act;  
var set;  
act = HwpControl.HwpCtrl.CreateAction("PageSetup"); // 액션 생성  
set = act.CreateSet(); // parameter set 생성  
act.GetDefault(set); // parameter set 초기화  
if (act.PopupDialog(set) == 1) // 대화상자 띄우기  
    act.Execute(set); // 액션 실행
```

## 가. 속성 (Property)

### 1) ActID[읽기전용]

액션 ID를 나타낸다.

- 구문(Syntax)

*C++*

```
CString Getactid()
```

*Javascript*

```
(string) ActID
```

- 설명(Remarks)

액션 ID의 종류 및 설명은 별도 문서 참조.

- 예제(Example)

- 참고 항목(See Also)

## 2) SetID[읽기전용]

액션이 사용하는 `ParameterSet` ID를 나타낸다.

- 구문(Syntax)

*C++*

```
CString Getsetid()
```

*javascript*

```
(string) SetID
```

- 설명(Remarks)

`parameter set`을 사용하지 않는 액션은 빈 문자열을 리턴한다.

- 예제(Example)

- 참고 항목(See Also)

## 나. 메소드 (Method)

### 1) CreateSet

액션과 대응하는 `parameter set`을 생성한다.

- 구문(Syntax)

*C++*

```
LPDISPATCH CreateSet()
```

*Javascript*

```
ParameterSet CreateSet()
```

- 매개변수(Parameters)
- 반환값(Return)  
해당 Action과 연관되는 ParmeterSet Object.
- 설명(Remarks)  
`parameter set`을 사용하지 않는 액션의 경우 NULL이 리턴된다.  
이 method는 다음과 같이 수행한 것과 동일하다.

```
Set param = HwpCtrl.CreateSet(action.SetID)
```

- 예제(Example)
- 참고 항목(See Also)

## 2) GetDefault

현재 상태에 따라 액션 실행에 필요한 인수를 구한다.

- 구문(Syntax)

*C++*

```
void GetDefault(LPDISPATCH set)
```

*Javascript*

```
GetDefault(ParameterSet set)
```

- 매개변수(Parameters)

*set*

인수를 저장할 ParameterSet

- 반환값(Return)

해당 Action과 연관되는 ParmeterSet Object.

- 설명(Remarks)

예를 들어 글자모양의 액션의 경우, 현재 셀렉션 상태에 따라 param의 아이템들이 채워진다.  
서브셋을 만들 경우에는 서브셋을 만든 후에 GetDefault를 사용한다.

- 예제(Example)

- 참고 항목(See Also)

### 3) PopupDialog

액션의 대화상자를 띄운다.

- 구문(Syntax)

*C++*

```
long PopupDialog(LPDISPATCH set)
```

*javascript*

```
number PopupDialog(ParameterSet set)
```

- 매개변수(Parameters)

*set*

여기에 지정한 아이템의 값에 따라 대화상자의 각 컨트롤의 초기 값이 결정되고, 대화상자가 닫힌 후에는 사용자가 지정한 값들이 담겨 돌아온다.

- 반환값(Return)

액션이 정의하기에 따라 다르지만, 일반적으로 다음과 같은 **modal dialog result**를 리턴한다.

ID	값	설명
hwpOK	IDOK	다이얼로그 박스의 확인버튼을 눌렀을 경우 리턴 되는 값
hwpCancel	IDCANCEL	다이얼로그 박스의 취소버튼을 눌렀을 경우 리턴 되는 값
hwpError	-1	실행시 에러가 발생 하였을 경우 리턴 되는 값

- 설명(Remarks)

- 예제(Example)

- 참고 항목(See Also)

#### 4) Execute

지정한 인수로 액션을 실행한다.

- 구문(Syntax)

*C++*

```
long Execute(LPDISPATCH set)
```

*javascript*

```
number Execute(ParameterSet set)
```

- 매개변수(Parameters)

*set*

액션의 실행을 제어할 인수. **parameter set**의 종류와 아이템의 의미는 액션이 정의한 바에 따라 다르다.

- 반환값(Return)

액션이 성공하면 1, 실패하면 0을 반환한다.

- 설명(Remarks)

- 예제(Example)

- 참고 항목(See Also)

[Run](#), `HwpCtrl.Run(HwpCtrl Object 메소드 중 Run부분)`



## 5) Run

액션을 실행한다.

- 구문(Syntax)

*C++*

```
void Run()
```

*javascript*

```
Run()
```

- 매개변수(Parameters)

- 반환값(Return)

- 설명(Remarks)

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

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

- 예제(Example)

- 참고 항목(See Also)

## 2. ctrlCode object

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

## 가. 속성 (Property)

### 1) ctrlCh[읽기전용]

컨트롤 문자.

- 구문(Syntax)

C++

```
short GetCtrlCh()
```

javascript

```
(number) ctrlCh
```

- 설명(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 1부터 31사이의 값을 사용한다.

Ch	설명	Ch	설명
1	예약	17	각주 / 미주
2	구역/단 정의	18	자동 번호
3	필드 시작	19	예약
4	필드 끝	20	예약
5	예약	21	쪽바꿈
6	예약	22	책갈피 / 찾아보기 표시
7	예약	23	덧말 / 글자 겹침
8	예약	24	하이픈
9	탭	25	예약
10	강제 줄 나눔	26	예약
11	그리기 개체 / 표	27	예약
12	예약	28	예약
13	문단 나누기	29	예약
14	예약	30	목록 빈칸
15	주석	31	고정 폭 빈칸
16	머리말 / 꼬리말		

- 예 제 (Example)
- 참고 항목 (See Also)

## 2) CtrlID[읽기전용]

컨트롤 ID.

- 구문(Syntax)

C++

```
CString GetCtrlId()
```

javascript

```
(string) CtrlID
```

- 설명(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	번호 넣기
nwno	AutoNum	AutoNum	새 번호로
pgct	PageNumCtrl	PageNumCtrl	페이지 번호 제어 (97의 홀수 쪽에서 시작)
pghd	PageHiding	PageHiding	감추기
pgnp	PageNumPos	PageNumPos	쪽 번호 위치
head	HeaderFooter	HeaderFooter	머리말
foot	HeaderFooter	HeaderFooter	꼬리말
%dte	FieldCtrl	FieldCtrl	현재의 날짜/시간 필드
%ddt	FieldCtrl	FieldCtrl	파일 작성 날짜/시간 필드

ID	Property Set	Initialization Set	설명
%pat	FieldCtrl	FieldCtrl	문서 경로 필드
%bm	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
%h1k	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

- 예제 (Example)

#### Visual Basic

```
'문서 중의 각주를 카운트
Dim ctrlcode As CtrlCode
cnt = 0
Set ctrlcode = HwpCtrl.HeadCtrl
while Not ctrl Is Nothing
    If ctrlcode.CtrlID = "fn" Then
        cnt = cnt + 1
    End If
    Set ctrlcode = ctrlcode.Next
wend
MsgBox cnt
```

#### C++

```
// 그리기 개체를 찾아서 그 위치로 이동하는 예제
void FindCtrl()
```

```

{
    DHwpCtrlCode code = m_CHwpCtrl.GetHeadCtrl();
    DHwpParameterSet paramSet;
    VARIANT list, para, pos;
    while (code && code != m_CHwpCtrl.GetLastCtrl()) {
        CString strID = code.GetCtrlid();
        if (strcmp(strID, _T("gso")) == 0) {
            paramSet = code.GetAnchorPos(0);
            list = paramSet.Item("List");
            para = paramSet.Item("Para");
            pos = paramSet.Item("Pos");

            m_CHwpCtrl.SetPos(list.lVal, para.lVal, pos.lVal);
            break;
        }
        code = code.GetNext();
    }
}

```

- 참고 항목(See Also)

[ctrlCh](#)

### 3) Next[읽기전용]

다음 컨트롤.

- 구문(Syntax)

*C++*

```
LPDISPATCH GetNext()
```

*javascript*

```
(CtrlCode) Next
```

- 설명(Remarks)

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

- 예제(Example)

- 참고 항목(See Also)

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



#### 4) Prev[읽기전용]

앞 컨트롤.

- 구문(Syntax)

*C++*

```
LPDISPATCH GetPrev()
```

*javascript*

```
(CtrlCode) Prev
```

- 설명(Remarks)

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

- 예제(Example)

- 참고 항목(See Also)

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

## 5) Properties

컨트롤의 속성을 나타낸다.

- 구문(Syntax)

*C++*

```
LPDISPATCH GetProperties()  
void SetProperties(LPDISPATCH value)
```

*Javascript*

```
(ParameterSet) Properties
```

- 설명(Remarks)

모든 컨트롤은 대응하는 **parameter set**으로 속성을 읽고 쓸 수 있다.

- 예제(Example)

*Visual Basic*

```
'현재 캐럿이 위치한 표의 셀 간격을 1pt로 설정한다.  
Dim ctrlcode As CtrlCode  
Dim tbset As ParameterSet  
Set ctrlcode = HwpCtrl.ParentCtrl  
If ctrlcode = "tbl" Then  
    Set tbset = HwpCtrl.CreateSet("Table")  
    tbset.SetItem "Cellspacing", 100  
    ctrlcode.Properties = tbset  
End If
```

- 참고 항목(See Also)

## 6) UserDesc[읽기전용]

컨트롤의 종류를 사용자에게 보여줄 수 있는 `localize`된 문자열로 나타낸다.

- 구문(Syntax)

*C++*

```
CString GetUserDesc()
```

*javascript*

```
(string) UserDesc
```

- 설명(Remarks)
- 예제(Example)
- 참고 항목(See Also)

## 나. 메소드 (Method)

### 1) GetAnchorPos

컨트롤의 **anchor**의 위치를 반환한다.

- 구문(Syntax)

*C++*

```
LPDISPATCH GetAnchorPos(long type)
```

[ver:0x05050115](#)

*Javascript*

```
ParameterSet GetAnchorPos(number type)
```

- 매개변수(Parameters)

*type*

기준위치

값	설명	비고
0	바로 상위 리스트에서의 <b>anchor position</b>	default
1	탭 레벨 리스트에서의 <b>anchor position</b>	
2	루트 리스트에서의 <b>anchor position</b>	

- 반환값(Return)

성공했을 경우 **ListParaPos** **ParameterSet**이 반환된다.

실패했을 경우 **null**이 반환된다.

- 설명(Remarks)

- 예제(Example)

*C++*

```
BOOL CTestHwpCtrlDlg::OnFindMailMerge()  
{  
    DHwpCtrlCode code = m_cHwpCtrl.GetHeadCtrl();  
    DHwpParameterSet paramSet;
```

```

    VARIANT list, para, pos;

    while (code && code != m_CHwpCtrl.GetLastCtrl()) {

        CString strID = code.GetCtrlid();
        if (strcmp(strID, _T("%mmg")) == 0) {
            paramSet = code.GetAnchorPos(0);
            list = paramSet.Item("List");
            para = paramSet.Item("Para");
            pos = paramSet.Item("Pos");

            m_CHwpCtrl.SetPos(list.lval, para.lval, pos.lval);
            m_CHwpCtrl.SetFocus();
            return TRUE;
        }
        code = code.GetNext();
    }
    return FALSE;
}

```

- 참고 항목(See Also)

### 3. HwpCtrl Object

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.

## 가. 속성 (Property)

### 1) AutoShowHideToolBar

(2002.7.4)

자동 툴바 보이기/숨기기 작동여부

- 구문(Syntax)

*C++*

```
BOOL GetAutoShowHideToolBar()  
void SetAutoShowHideToolBar(BOOL value)
```

[ver:0x0505011B](#)

*Javascript*

```
(boolean) AutoShowHideToolBar
```

- 설명(Remarks)

ShowToolBar 등을 사용하여 툴바를 만들었을 때만 효과가 있다.

바탕쪽, 머리말/꼬리말, 각주, 숨은 설명에 대한 도구상자를 필요에 따라 자동으로 보이거나 감추도록 한다.

- 예제(Example)

- 참고 항목(See Also)

## 2) CellShape

현재 선택되어 있는 표와 셀의 모양 정보를 나타낸다.

- 구문(Syntax)

*C++*

```
LPDISPATCH GetCellShape()  
void SetCellShape(LPDISPATCH value)
```

*Javascript*

```
(ParameterSet) CellShape
```

- 설명(Remarks)

ParameterSet/Table로 표의 속성에 대한 기본 정보를 나타내며, 이 가운데 "Cell" 아이템이 ParameterSet/Cell로 셀의 속성을 나타낸다.

셀 블록이 잡혀있지 않은 상태이면 현재 커서가 위치한 셀 하나만을 대상으로 한다. 현재 표 내부에 커서가 위치하지 않으면 에러가 발생한다.

- 예제(Example)

*Visual Basic*

```
Dim tp As ParameterSet  
Dim cp As ParameterSet  
Set tp = HwpCtrl.CreateSet("Table")  
Set cp = tp.CreateItemSet("Cell", "Cell")  
tp.SetItem "CellSpacing", 200  
cp.SetItem "Header", 1  
HwpCtrl.CellShape = tp
```

- 참고 항목(See Also)



### 3) CharShape

현재 **selection**의 글자 모양을 나타낸다.

- 구문(Syntax)

*C++*

```
LPDISPATCH GetCharShape()  
void SetCharShape(LPDISPATCH newValue)
```

*javascript*

```
(ParameterSet) CharShape
```

- 설명(Remarks)

**property get**을 수행하면 현재 **selection** 내의 글자 모양을 구할 수 있다.

**selection**이 존재하지 않으면 현재 커서가 위치한 곳의 글자 모양을 돌려준다.

글자 모양 중 특정 항목이 **selection** 내에서 서로 다른 속성을 가지고 있으면 아예 아이템 자체가 존재하지 않는다.

**property set**을 수행하면 아이템이 존재하는 항목에 대해서만 속성을 설정한다.

**ParameterSet**의 형식은 **ParameterSet/CharShape** 참조.

- 예제(Example)

*visual basic*

```
' selection 내의 글자 크기가 모두 동일할 때만 20pt로 설정한다. (Visual Basic version)  
Dim cs As ParameterSet  
Set cs = HwpCtrl.CharShape  
If cs.ItemExist("Height") Then  
    cs.SetItem "Height", 2000  
    HwpCtrl.CharShape = cs  
End If
```

- 참고 항목(See Also)

[ParaShape](#)

#### 4) curFieldState[읽기전용]

캐럿이 위치한 필드의 상태정보를 구한다. 읽기전용

- 구문(Syntax)

*C++*

```
long GetCurFieldState()
```

*javascript*

```
(long) curFieldState
```

- 설명(Remarks)

얻어지는 내용은 일종의 **flag**값으로 비트가 **set**되어 있으면 다음과 같은 의미를 가진다.

BIT 0	현재 필드는 셀
BIT 1	현재 필드는 누름틀
BIT 4	필드명이 존재
나머지	예약

- 예제(Example)

- 참고 항목(See Also)

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

## 5) curSelectedCtrl[읽기전용]

. 현재 선택된 컨트롤 객체. 읽기전용

- 구문(Syntax)

*C++*

```
LPDISPATCH GetCurSelectedCtrl()
```

*javascript*

```
(Ctrl) curSelectedCtrl
```

- 설명(Remarks)

현재 마우스 커서에 의해 선택된 컨트롤 객체 또는 컨트롤 객체 앞에 캐럿이 있는 경우, 해당 컨트롤 객체에 대한 정보를 가져올 수 있다. **CtrlCode** 객체를 리턴 받아, **CtrlCode** 객체의 속성과 함수를 사용할 수 있다.

- 예제(Example)

*C++*

```
CDHwpCtrlCode code = m_HwpCtrl1.GetCurSelectedCtrl();  
MessageBox(code.Getctrlid());
```

*javascript*

```
var code = vHwpCtrl.CurSelectedCtrl;  
alert(code.CtrlID());
```

- 참고 항목(See Also)

[CtrlCode](#)

## 6) EditMode

(2001.12.4)

현재 편집모드

- 구문(Syntax)

C++

```
long GetEditMode()  
void SetEditMode(long value)
```

[ver:0x05050101](#)

javascript

```
(long) EditMode
```

- 설명(Remarks)

현재 편집모드를 나타낸다.

0	읽기전용
1	일반 편집모드
2	양식모드(양식 사용자 모드): cell과 누름틀 중 양식 모드에서 편집 가능한 속성을 가진 것만 편집가능하다
16	예약

- 예제(Example)

visual C++

```
// 읽기전용모드로 전환한다  
m_CHwpCtrl.SetEditMode(0);
```

- 참고 항목(See Also)

[SetFieldViewOption](#)

## 7) EngineProperties

환경설정 정보

- 구문(Syntax)

C++

```
LPDISPATCH GetEngineProperties()  
void SetEngineProperties(LPDISPATCH value)
```

javascript

```
(ParameterSet) EngineProperties
```

- 설명(Remarks)

환경설정에서 지정할 수 있는 옵션 값을 설정할 수 있다.

ParameterSet의 형식은 ParameterSet/EngineProperties 참조

Item ID	Type	Description
DoAnyCursorEdit	PIT_UI1	마우스로 두 번 누르기 한곳에 입력가능
DoOutLineStyle	PIT_UI1	개요 번호 삽입 문단에 개요 스타일 자동 적용
InsertLock	PIT_UI1	삽입 잠금
TabMoveCell	PIT_UI1	표 안에서 <Tab>으로 셀 이동
FaxDriver	PIT_BSTR	팩스 드라이버
PDFDriver	PIT_BSTR	PDF 드라이버
EnableAutoSpell	PIT_UI1	맞춤법 도우미 작동
CtrlMaskAs2002	PIT_UI1	한글 2002 방식으로 조판 부호 표시하기 (한글 버전 : 7.5.11.603)
ShowGuildLines	PIT_UI1	투명선 보이기 (한글 버전 : 7.5.11.603)

- 예제(Example)

javascript

```
var vSet = vHwpCtrl.CreateSet("EngineProperties");  
vSet.SetItem("CtrlMaskAs2002", 1); // 조판부호 1: 2002 모드, 0 : 2007 모드  
vSet.SetItem("ShowGuildLines", 0); // 표 투명선 1: 보이기, 0 : 안보이기  
vHwpCtrl.EngineProperties = vSet;
```

- 참고 항목(See Also)

## 8) GetIsEmpty[읽기전용]

아무 내용도 들어있지 않은 빈 문서인지 여부를 나타낸다. 읽기전용

- 구문(Syntax)

*C++*

```
BOOL GetIsEmpty()
```

*javascript*

```
(boolean) IsEmpty
```

- 설명(Remarks)
- 예제(Example)
- 참고 항목(See Also)

## 9) HeadCtrl[읽기전용]

문서 중 첫 번째 컨트롤. 읽기전용

- 구문(Syntax)

*C++*

```
LPDISPATCH GetHeadCtrl()
```

*javascript*

```
(Ctrl) HeadCtrl
```

- 설명(Remarks)

문서 중의 모든 컨트롤(표, 그림 등의 특수 문자들)은 **Linked List**로 서로 연결되어 있는데, 그 **List**의 시작 컨트롤을 나타낸다. 이 컨트롤로부터 시작, **Ctrl.Next**를 이용해 순방향 탐색을 수행할 수 있다.

- 예제(Example)

*visual basic*

```
' 문서중의 모든 표를 삭제
Dim ctrl As ctrl
Dim nxtctrl As ctrl
Set ctrl = HwpCtrl.HeadCtrl
while Not ctrl Is Nothing
    Set nxtctrl = ctrl.Next
    If ctrl.CtrlID = "tbl" Then
        HwpCtrl.DeleteCtrl ctrl
    End If
    Set ctrl = nxtctrl
wend
```

- 참고 항목(See Also)

[LastCtrl](#)



## 10) HyperlinkMode

(2004.3.15)

하이퍼링크 모드 지정

- 구문(Syntax)

C++

```
long GetHyperlinkMode()
void SetHyperlinkMode(long value)
```

[ver:0x05070145](#)

javascript

(ParameterSet) HyperlinkMode

- 설명(Remarks)

하이퍼링크 액션을 한글에서 실행할지 외부프로그램에서 실행할지 결정한다.

0	기본값. 기존에 수행되던 방식으로, 하이퍼링크 클릭시 한글에서 액션을 수행한다. 컨트롤에는 하이퍼링크 정보가 돌아간다.
1	하이퍼링크 클릭 시 한글에서 액션이 수행되지 않고 URL정보만 알려준다. 액션은 외부프로그램에서 만들어 수행해야 한다.

- 예제(Example)

C++

```
// 내부 실행 모듈 (하이퍼링크 모드 설정)
void OnHyperlingMode()
{
    m_chwpCtrl.SetHyperlinkMode(1);
}

// 외부프로그램 모듈 (하이퍼링크를 실행시킨다)
void CTestHwpCtrlDlg::OnNotifyMessageHwpctrlCtrl(LPCTSTR Msg, long wParam, long lParam)
{
    CString szMsg;

    if (!strcmp(Msg, _T("HyperLinkClicked"))) {
```

```

if (WParam != 0){          // wParam 값이 0이 아니면 하이퍼링크
    VARIANT var;
    DHwpParameterSet set = m_cHwpCtrl.GetMessageSet();
    var = set.Item("String");
    szMsg = var.bstrVal;
    if (m_cHwpCtrl.GetHyperlinkMode())
        ShellExecute(NULL, _T("open"), szMsg, NULL, NULL, SW_SHOW);
}
else{          //wParam값이 0 이면 문서내 이동
    TRACE(_T("문서내 이동"));
}
}

```

- 참고 항목(See Also)

## 11) IsModified[읽기전용]

문서가 변경되었는지 나타낸다. 읽기전용

- 구문(Syntax)

*C++*

```
short GetIsModified()
```

*Javascript*

```
(number) IsModified
```

- 설명(Remarks)

0	변경되지 않은 깨끗한 상태
1	변경된 상태
2	변경되었으나 자동 저장된 상태

- 예제(Example)

- 참고 항목(See Also)

## 12) IsPreviewMode[읽기전용]

(2005.3.3) 한글2004

인쇄 미리보기 상태인지를 나타낸다. 읽기전용

- 구문(Syntax)

*C++*

```
BOOL GetIsPreviewMode()
```

[ver:0x0600010B, 0x0605010B](#)

*javascript*

```
(boolean) IsPreviewMode
```

- 설명(Remarks)

현재 뷰의 상태가 미리보기 상태인지 알아온다.

- 예제(Example)

*javascript*

```
function PreviewClose() {  
    if (pHwpCtrl.IsPreviewMode)  
        pHwpCtrl.PreviewCommand(10);    // 미리보기 닫기  
}
```

- 참고 항목(See Also)

[PreviewCommand](#)

### 13) IsPrivateInfoProtected[읽기전용]

. 개인정보가 보호되어 있는지 여부를 알 수 있는 속성. 읽기전용

- 구문(Syntax)

*C++*

```
BOOL GetIsPrivateInfoProtected()
```

*javascript*

```
(boolean) IsPrivateInfoProtected
```

- 설명(Remarks)

문서 내에 개인정보가 보호된 문서인지 아닌지를 알 수 있다.

- 예제(Example)

*C++*

```
if (m_HwpCtrl.GetIsPrivateInfoProtected() == FALSE)
    MessageBox(_T("개인정보보호 문서 아님!"));
else if (m_HwpCtrl.GetIsPrivateInfoProtected() == TRUE)
    MessageBox(_T("개인정보보호 문서!"));
```

*javascript*

```
if (vHwpCtrl.IsPrivateInfoProtected == false)
    alert("개인정보보호 문서 아님!");
else if (vHwpCtrl.IsPrivateInfoProtected == true)
    alert("개인정보보호 문서!");
```

- 참고 항목(See Also)

[SetPrivateInfoPassword](#), [RegisterPrivateInfoPattern](#), [ProtectPrivateInfo](#), [FindPrivateInfo](#)

#### 14) LastCtrl[읽기전용]

문서 중 마지막 컨트롤. 읽기전용

- 구문(Syntax)

*C++*

```
LPDISPATCH GetLastCtrl()
```

*javascript*

```
(Ctrl) LastCtrl
```

- 설명(Remarks)

문서 중의 모든 컨트롤(표, 그림 등의 특수 문자들)은 **linked list**로 서로 연결되어 있는데, 그 **list**의 마지막 컨트롤을 나타낸다. 이 컨트롤로부터 시작, **Ctrl.Prev**를 이용해 역방향 탐색을 수행할 수 있다.

- 예제(Example)

- 참고 항목(See Also)

[HeadCtrl](#)

## 15) PageCount[읽기전용]

문서의 페이지 수. 읽기전용

- 구문(Syntax)

*C++*

```
long GetPageCount()
```

*javascript*

```
(number) PageCount
```

- 설명(Remarks)
- 예제(Example)
- 참고 항목(See Also)

[CreatePageImage](#)

## 16) ParaShape

현재 `selection`의 문단 모양을 나타낸다.

- 구문(Syntax)

*C++*

```
LPDISPATCH GetParaShape()  
void SetParaShape(LPDISPATCH value)
```

*javascript*

```
(ParameterSet) ParaShape
```

- 설명(Remarks)

개념과 사용법은 `CharShape`과 동일하다.

`ParameterSet`의 형식은 `ParameterSet/ParaShape` 참조.

- 예제(Example)

- 참고 항목(See Also)

[CharShape](#)



## 17) ParentCtrl[읽기전용]

현재 캐럿 위치의 상위컨트롤. 읽기전용

- 구문(Syntax)

*C++*

```
LPDISPATCH GetParentCtrl()
```

*javascript*

```
(Ctrl) ParentCtrl
```

- 설명(Remarks)

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

현재 캐럿이 본문 레벨에 위치해 상위 컨트롤이 없을 때는 **NULL**이 반환된다.

- 예제(Example)

*visual basic*

```
If HwpCtrl.ParentCtrl Is Nothing Then  
    ... 본문 내 위치  
End If
```

- 참고 항목(See Also)

## 18) Path

문서 파일의 경로

- 구문(Syntax)

*C++*

```
CString GetPath()  
void SetPath(CString value)
```

*javascript*

```
(string) Path
```

- 설명(Remarks)

현재 편집 중인 문서 파일의 경로를 나타낸다. 문서의 경로는 마지막으로 수행한 **open**, **SaveAs**에 따라 결정되지만, 이 **property**에 값을 대입함으로써 직접 경로를 설정할 수도 있다.

- 예제(Example)

- 참고 항목(See Also)

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

## 19) ReadOnlyMode[읽기전용]

문서가 읽기 전용인지 아닌지 여부를 알려준다

- 구문(Syntax)

*C++*

```
BOOL GetReadOnlyMode();
```

*javascript*

```
(boolean) ReadOnlyMode
```

- 설명(Remarks)

이미 다른 프로그램에서 해당 문서를 사용 중이거나, 문서의 속성이 읽기 전용일 때, 해당 문서가 읽기 전용인지를 알 수 있다.

- 예제(Example)

*C++*

```
if (m_HwpCtrl.GetReadOnlyMode() == FALSE)
    MessageBox(_T("읽기전용 모드 아님!"));
else if (m_HwpCtrl.GetReadOnlyMode() == TRUE)
    MessageBox(_T("읽기전용 모드!"));
```

*javascript*

```
if (vHwpCtrl.ReadOnlyMode == false)
    alert("읽기전용 모드 아님!");
else if (vHwpCtrl.ReadOnlyMode == true)
    alert("읽기전용 모드!");
```

- 참고 항목(See Also)

## 20) ScrollPosInfo

(2003.2.27)

스크롤바 위치정보

- 구문(Syntax)

C++

```
LPDISPATCH GetScrollPosInfo()  
void SetScrollPosInfo(LPDISPATCH value)
```

[ver:0x05070125](#)

javascript

(ParameterSet) ScrollPosInfo

- 설명(Remarks)

현재 스크롤바(수직, 수평)의 위치정보를 "ScrollPosInfo" ParameterSet으로 얻어오거나 설정한다.

Item ID	설명
HorzPos	수평 스크롤바 위치
HorzLimitPos	수평 스크롤바 한계 위치 값 (읽기전용)
VertPos	수직 스크롤바 위치
VertLimitPos	수직 스크롤바 한계 위치 값 (읽기전용)

- 예제(Example)

C++

```
DHwpParameterSet set = m_CHwpCtrl.GetScrollPosInfo();  
VARIANT var;  
var = set.Item("HorzPos");  
int hp = var.intVal;  
var = set.Item("VertPos");  
int vp = var.intVal;  
var = set.Item("HorzLimitPos");  
int hlp = var.intVal;  
var = set.Item("VertLimitPos");  
int vlp = var.intVal;
```

- 참고 항목(See Also)

## 21) SelectionMode[읽기전용]

문서의 내용이 어떤 **selection** 상태인가를 알려준다. 읽기전용

- 구문(Syntax)

*C++*

```
short GetSelectionMode()  
void SetSelectionMode(short value)
```

*javascript*

```
(number) SelectionMode
```

- 설명(Remarks)

일반 블록이 아닌 F3키나 F4키에 의해 블록이 지정된 경우,

HWPSEL\_STRICT\_MODE ( = 0x10, 십진수 16)으로 OR 마스크되어 오기 때문에,

항상 0x0F(십진수15)로 AND 마스크한 결과로 판단하도록 한다.

아래의 예제 참조.

- 예제(Example)

*visual basic*

```
Dim mode As Integer  
  
Const HWPSEL_NONE = 0  
Const HWPSEL_NORMAL = 1  
Const HWPSEL_COLUMN = 2  
Const HWPSEL_TABLE = 3  
Const HWPSEL_CTRL = 4  
Const HWPSEL_MODE_MASK = 15  
Const HWPSEL_STRICT_MODE = 16  
  
mode = HwpCtrl1.SelectionMode() And HWPSEL_MODE_MASK  
  
If mode = 0 Then  
    ' 블록지정이 되지 않은 경우  
    MsgBox "No Selection ..."
```

```
ElseIf mode = HWPSEL_NORMAL Then
    ' 일반 블록이거나 F3키에 의한 블록인 경우 (TEXT)
    MsgBox "Normal Selection ..."
ElseIf mode = HWPSEL_COLUMN Then
    ' F4키에 의한 블록인 경우 (TEXT)
    MsgBox "Column Selection ..."
ElseIf mode = HWPSEL_TABLE Then
    ' 테이블 셀 블록 상태
    MsgBox "Table Selection ..."
ElseIf mode = HWPSEL_CTRL Then
    ' 그리기 개체나 표, 그림 등 ShapeObject에 해당하는 컨트롤이 선택된 상태
    MsgBox "Ctrl Selection ..."
End If
```

- 참고 항목(See Also)

## 22) viewProperties

view의 상태정보

- 구문(Syntax)

*C++*

```
LPDISPATCH GetViewProperties()  
void SetViewProperties(LPDISPATCH value)
```

*javascript*

```
(ParameterSet) ViewProperties
```

- 설명(Remarks)

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

- 예제(Example)

*visual basic*

```
'화면 확대 비율을 100%로 맞춘다.  
Dim vp As ParameterSet  
Set vp = HwpCtrl.CreateSet("ViewProperties")  
vp.SetItem "ZoomType", hwpZoomCustom  
vp.SetItem "ZoomRatio", 100  
HwpCtrl.ViewProperties = vp
```

- 참고 항목(See Also)

### 23) version[읽기전용]

한글과 한글 컨트롤의 버전정보를 구한다. 읽기전용

- 구문(Syntax)

C++

```
long GetVersion()
```

javascript

```
(long) version
```

- 설명(Remarks)

한글 2004컨트롤 버전이 이전에는 0x050701xx이였으나 한글 6.0.1.689버전 이후부터 0x06000101 로 시작하도록 수정한다.  
(2004.3.26) 현재 2002컨트롤 버전은 0x05070135 이고, 2004컨트롤 버전은 0x06000101 이다.

BYTE 3	한글 컨트롤의 major version
BYTE 2	한글 컨트롤의 minor version(1)
BYTE 1	한글 컨트롤의 minor version(2)
BYTE 0	한글 컨트롤의 minor version(3)

- 예제(Example)

C++

```
long ver = m_CHwpCtrl.GetVersion();  
if (ver & 0x05000000)  
    TRACE("Hwp2002 Ctrl");  
else if (ver & 0x06000000)  
    TRACE("Hwp2004 Ctrl");
```

- 참고 항목(See Also)



## 24) XHwpDocuments[읽기전용]

(2009.1.19) 한글2007

문서(document)의 집합을 관리하는 XHwpDocuments 속성. 읽기전용

- 구문(Syntax)

C++

```
LPDISPATCH GetXHwpDocuments()
```

[ver:0x07050206](#)

javascript

```
(XHwpDocuments) XHwpDocuments
```

- 설명(Remarks)

문서(document)의 집합을 관리하는 객체이다.

한글 컨트롤은 하나의 document를 사용하기 때문에 XHwpDocuments로 얻는 문서객체는 유일하나 한글 Automation과의 구조적 일치성을 위해 추가되었다. 실제로 한글 Automation의 XHwpDocuments 객체와 동일하다. XHwpDocuments 객체의 주요 속성과 메소드는 hwpAutomation.hwp문서를 참고한다.

- 예제(Example)

C++

```
// 한글 컨트롤의 문서객체를 얻어온다.  
CXHwpDocuments docs = m_CHwpCtrl.GetXHwpDocuments();  
CXHwpDocument doc = docs.get_Active_XHwpDocument();
```

- 참고 항목(See Also)

## 나. 메소드 (Method)

### 1) AboutBox

도움말을 본다.

- 구문(Syntax)

*C++*

```
void AboutBox()
```

*javascript*

```
AboutBox()
```

- 매개변수(Parameters)

- 반환값(Return)

- 설명(Remarks)

도움말 대화상자가 나타난다.

- 예제(Example)

*javascript*

```
m_HwpCtrl1.AboutBox();
```

- 참고 항목(See Also)

## 2) Clear

현재 편집중인 문서의 내용을 닫고 빈문서 편집 상태로 돌아간다.

- 구문(Syntax)

*C++*

```
void Clear(VARIANT& option)
```

*Javascript*

```
Clear([number option])
```

- 매개변수(Parameters)

*option*

편집중인 문서의 내용에 대한 처리 방법, 생략하면 **hwpAskSave**가 선택된다.

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

- 반환값(Return)

- 설명(Remarks)

**hwpSaveIfDirty**, **hwpSave**가 지정된 경우 현재 문서경로가 지정되어 있지 않으면 “새 이름으로 저장” 대화상자를 띄워 사용자에게 경로를 묻는다.

- 예제(Example)

*C++*

```
enum {  
    // 문서를 clear할 경우 다음의 옵션을 사용 가능하다.  
    AskSave= 0,    // 문서의 내용이 변경되었을 때 사용자에게 저장할지 묻는 대화상자를 띄운다.  
    Discard= 1,    // 문서 내용을 버린다.  
    SaveIfDirty= 2, // 문서가 변경된 경우 저장한다.  
    Save= 3        // 무조건 저장한다.  
};  
pHwpCtrl->Clear(COLEvariant((short)AskSave));
```

- 참고 항목(See Also)

[Open](#)

### 3) createAction

Action 객체를 생성한다.

- 구문(Syntax)

*C++*

```
LPDISPATCH createAction(LPCTSTR actionID)
```

*Javascript*

```
Action createAction(string actionID)
```

- 매개변수(Parameters)

*actionID*

액션 ID (ActionIDTable.hwp 참조)

- 반환값(Return)

Action object

- 설명(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.

- 예제(Example)

*visual basic*

```
'사용자에게 글자모양 대화상자를 띄우고 그 결과에서 글자크기를 제외한 항목만 적용한다.  
Dim ac As Action  
Dim cs As ParameterSet  
Set ac = HwpCtrl.CreateAction("CharShape")  
Set cs = ac.CreateSet  
ac.GetDefault cs  
if ac.PopupDialog(cs) = hwpOK Then  
    cs.RemoveItem "Height"  
    ac.Execute cs  
End if
```

'GetDefault, PopupDialog, Execute는 Action의 Method부분을 참고.  
'RemoveItem은 ParameterSet Object의 Method부분을 참고.

- 참고 항목(See Also)

[Run](#)

#### 4) CreateField

캐럿의 현재 위치에 누름틀을 생성한다.

- 구문(Syntax)

*C++*

```
BOOL CreateField(LPCTSTR direction, VARIANT& memo, VARIANT& name)
```

*javascript*

```
boolean CreateField(string direction, string memo, string name)
```

- 매개변수(Parameters)

*direction*

누름틀에 입력이 안 된 상태에서 보이는 안내문/지시문.

*memo*

누름틀에 대한 설명/도움말

*name*

누름틀 필드에 대한 필드 이름

- 반환값(Return)

성공이면 **true**, 실패면 **false**

- 설명(Remarks)

- 예제(Example)

- 참고 항목(See Also)

## 5) CreatePageImage

지정한 페이지를 이미지파일로 생성한다.

- 구문(Syntax)

*C++*

```
BOOL CreatePageImage(LPCTSTR Path, VARIANT& pgno, VARIANT& resolution, VARIANT& depth, VARIANT& format)
```

*javascript*

```
boolean CreatePageImage(string Path, [number pgno], [number resolution], [number depth], [string format])
```

- 매개변수(Parameters)

*path*

생성할 이미지 파일의 경로

*pgno*

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

*resolution*

이미지 해상도. DPI단위(96, 300, 1200 등)로 지정한다. 생략하면 96이 사용된다.

*depth*

이미지파일의 Color Depth(1, 4, 8, 24)를 지정한다.

*format*

이미지파일의 포맷. "bmp", "gif"중의 하나. 생략하면 "bmp"가 사용된다.

- 반환값(Return)

성공하면 true, 실패하면 false

- 설명(Remarks)

지정된 페이지를 이미지파일로 저장한다. 저장되는 이미지파일의 포맷은 비트맵 또는 GIF 이미지이다.

만약 이 외의 포맷이 입력되었다면 비트맵으로 저장한다.

- 예제(Example)



- 참고 항목(See Also)  
[PageCount](#)

## 6) CreateSet

ParameterSet을 생성한다.

- 구문(Syntax)

*C++*

```
LPDISPATCH CreateSet(LPCTSTR setID)
```

*Javascript*

```
ParameterSet CreateSet(string setID)
```

- 매개변수(Parameters)

*setID*

생성할 ParameterSet의 ID (SetIDTable.hwp)

- 반환값(Return)

생성된 ParameterSet Object

- 설명(Remarks)

ParameterSet은 일종의 정보를 지니는 객체이다.

어떤 Action들은 그 Action이 수행되기 위해서 정보가 필요한데 이때 사용되는 정보를 ParameterSet으로 넘겨준다. 또한 한글 컨트롤은 특정정보(ViewProperties, cellShape, CharShape 등)를 ParameterSet으로 변환하여 넘겨주기도 한다.

사용 가능한 ParameterSet의 ID는 SetIDTable.hwp문서를 참조한다.

- 예제(Example)

- 참고 항목(See Also)

## 7) DeleteCtrl

문서 내 컨트롤을 삭제한다.

- 구문(Syntax)

*C++*

```
BOOL DeleteCtrl(LPDISPATCH ctrl)
```

*javascript*

```
boolean DeleteCtrl(Ctrl ctrl)
```

- 매개변수(Parameters)

*ctrl*

삭제할 문서 내 컨트롤

- 반환값(Return)

성공하면 **true**, 실패하면 **false**

- 설명(Remarks)

- 예제(Example)

- 참고 항목(See Also)

[InsertCtrl](#)

## 8) ExportStyle

현재 문서의 Style을 Export한다.

- 구문(Syntax)

*C++*

```
BOOL ExportStyle(LPDISPATCH styleset)
```

*javascript*

```
boolean ExportStyle(ParameterSet styleset)
```

- 매개변수(Parameters)

*styleset*

- 반환값(Return)

- 설명(Remarks)

- 예제(Example)

*javascript*

- 참고 항목(See Also)

[ImportStyle](#)

## 9) FieldExist

문서에 지정된 데이터 필드가 존재하는지 검사한다.

- 구문(Syntax)

*C++*

```
BOOL FieldExist(LPCTSTR field)
```

*javascript*

```
boolean FieldExist(string field)
```

- 매개변수(Parameters)

*field*

필드이름

- 반환값(Return)

필드가 존재하면 **true**, 존재하지 않으면 **false**

- 설명(Remarks)

- 예제(Example)

- 참고 항목(See Also)

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

## 10) FindPrivateInfo

개인정보를 찾는다.

- 구문(Syntax)

*C++*

```
long FindPrivateInfo(long PrivateType, VARIANT PrivateString);
```

*javascript*

```
number FindPrivateInfo(int PrivateTypes, [optional] string PrivateString);
```

- 매개변수(Parameters)

*PrivateTypes*

보호할 개인정보 유형. 다음의 값을 하나이상 조합한다.

0x0001 : 전화번호  
0x0002 : 주민등록번호  
0x0004 : 외국인등록번호  
0x0008 : 전자우편  
0x0010 : 계좌번호  
0x0020 : 신용카드번호  
0x0040 : IP 주소  
0x0080 : 생년월일  
0x0100 : 주소  
0x0200 : 사용자 정의  
0x0400 : 기타

*PrivateString*

기타 문자열. 0x0400 유형이 존재할 경우에만 유효하므로, 생략가능하다

- 반환값(Return)

찾은 개인정보의 유형 값. 다음과 같다.

0x0001 : 전화번호  
0x0002 : 주민등록번호  
0x0004 : 외국인등록번호  
0x0008 : 전자우편  
0x0010 : 계좌번호  
0x0020 : 신용카드번호  
0x0040 : IP 주소  
0x0080 : 생년월일  
0x0100 : 주소  
0x0200 : 사용자 정의  
0x0400 : 기타

개인정보가 없는 경우에는 0을 반환한다.

또한, 검색 중 문서의 끝(end of document)을 만나면 -1을 반환한다. 이는 함수가 무한히 반복하는 것을 막아준다.

- 설명(Remarks)

개인 정보 유형으로 등록된 정보를 찾아 이동한다.

동작방식은 한/글의 찾기와 비슷하다.

함수가 호출되면 현재 캐럿 위치에서부터 개인정보를 검색하기 시작한다. 개인정보가 검색되면 검색된 개인정보를 블록 선택한다. 만약 문서의 끝까지 개인정보가 검색되지 않을 경우에는 함수는 -1을 반환하여 무한으로 검색하는 것을 막는다.

**PrivateTypes**는 검색할 개인정보 유형을 나타낸다. 단 하나의 유형을 검색할 수도 있지만 여러 유형을 **bit flag**로 조합해서 검색할 수 있다. 예를 들면, 주민등록번호와 외국인등록번호를 한꺼번에 검색할 수 있다.

**PrivateTypes**에 0(Null)을 줄 수 있는데 이것은 기타유형을 제외한 모든 유형을 검색하게 한다. (0x03FF 값을 입력한 것과 동일함)

**PrivateTypes**의 값 중 0x400 값은 단어를 직접 검색할 때 사용한다. 0x400 값은 위의 **PrivateString**과 조합되어 사용될 수 있으며, 다음과 같이 사용된다.

예) 신용카드번호 또는 단어 “신한카드”를 직접 검색  
`FindPrivateInfo(0x20+0x400, “신한카드”);`

`PrivateString`은 `PrivateTypes`에 `0x400`(기타) 값이 존재할 경우에만 유효하다. `PrivateTypes`에 `0x400`값이 없을 경우에는 함수에 아무런 영향을 끼치지 않으므로 생략 가능하다.

`FindPrivateInfo()` 함수는 리턴값으로 현재 선택된 개인정보의 유형을 반환한다. 사용자는 리턴값을 확인함으로 선택된 개인정보의 유형을 확인할 수 있으며, 선택적으로 정보보호가 가능하다.

함수는 문서 내 개인정보가 없을 경우에는 `0`을 반환한다. 그리고, 문서의 끝에 도달한 경우에는 `-1`을 반환한다. 이 두 값은 이전 리턴값보다 의미상 중요한데, 이유는 루프문의 탈출코드로 사용되기 때문이다.

다음의 예제코드를 보자.

예) 문제가 있는 코드 - 기타유형을 제외한 모든 유형의 개인정보를 보호

```
while (FindPrivateInfo(0)) {  
    ProtectPrivateInfo(“*”, 0);  
}
```

위 코드는 개인정보가 없을 경우에 루프를 탈출한다. 하지만 개인정보가 한 개라도 존재할 경우에는 무한루프에 빠지게 된다. 그러므로 위 코드는 다음과 같이 수정해야 한다.

예) 정상 코드 - 기타유형을 제외한 모든 유형의 개인정보를 보호

```
while (true) {  
    var pType = FindPrivateInfo(0);  
    if (pType == 0 || pType == -1)  
        break;  
    ProtectPrivateInfo(“*”, 0);  
}
```

- 예제 (Example)
- 참고 항목 (See Also)

[IsPrivateInfoProtected](#), [SetPrivateInfoPassword](#), [RegisterPrivateInfoPattern](#), [ProtectPrivateInfo](#)



## 11) GetActionCmdUIStatus

(2001.10.9)

Action의 UI상태를 얻어온다.

- 구문(Syntax)

C++

```
BOOL GetActionCmdUIStatus(LPCTSTR actid, long bwithKey, long * bEnabled, long * bChecked, long * bRadio,  
                           BSTR * szText) ver:0x05050101
```

```
BOOL GetActionCmdUIStatus(LPCTSTR actid, BOOL bwithKey, BOOL * bEnabled, short * bChecked, BOOL * bRadio,  
                           BSTR * szText) ver:0x05050100
```

- 매개변수(Parameters)

*actid*

액션 ID (ActionIDTable.Hwp 참조)

*bwithKey*

true: 메뉴 스트링 끝에 단축키를 같이 붙여준다.

false: 메뉴 이름만 붙여준다.

*bEnabled*

Action이 실행 가능한 상태인가를 나타낸다.

*bCheck*

Action이 실행중인가를 나타낸다. (실행중인 상태를 갖는 것들 예를 들면 “매크로 기록”, “상황선” 등과 같은 메뉴들은 실행중이면 메뉴에 check 표시가 된다.)

*bRadio*

UI 중 radio button 상태를 나타낸다.

*szText*

메뉴 문자열이 저장된다.

bwithKey 가 true인 경우 단축키가 뒤에 붙는다.

- 반환값(Return)

성공하면 true, 실패하면 false

- 설명(Remarks)

컨트롤 바깥쪽에서 메뉴나, 툴바 버튼의 상태를 표시할 때 유용하다.

※ 이 함수는 빠른 속도가 요구되므로 **parameter**로 포인터를 받는다. 따라서 포인터를 사용할 수 없는 언어에서는 사용이 불가능하다.

- 예제(Example)

C++

```
void CMainFrame::OnUpdateMenuItemTest1(CCmdUI* pCmdUI)
{
    LPCTSTR szActionName;
    long bIsEnabled;
    long bRadio;
    long nChecked;
    CComBSTR text;

    szActionName = GetActionNameByID(pCmdUI->m_nID);
    if (IsWindow(m_CHwpCtrl.m_hwnd))
    {
        // HwpCtrl 에게 메뉴 Action의 상태를 물어본다.
        if (szActionName&&m_CHwpCtrl.GetActionCmdUIStatus(szActionName, TRUE, &bIsEnabled,
                                                         &nChecked, &bRadio, &text))
        {
            pCmdUI->Enable(bIsEnabled);
            pCmdUI->SetCheck(nChecked);
            pCmdUI->SetText((CString)text);
            return;
        }
        _ASSERT(!_T("GetActionCmdUIStatus로 처리할 수 없는 잘못된 ID"));
    }
}
```

- 참고 항목(See Also)

## 12) GetBinDataPath

Binary Data의 경로를 가져온다.

- 구문(Syntax)

*C++*

```
CString GetBinDataPath(long binid)
```

*javascript*

```
string GetBinDataPath(number binid)
```

- 매개변수(Parameters)

바이너리 데이터의 ID 값 (1부터 시작)

- 반환값(Return)

바이너리 데이터의 경로

- 설명(Remarks)

- 예제(Example)

*C++*

```
CString path = m_HwpCtrl1.GetBinDataPath(1);  
MessageBox(path);  
  
path = m_HwpCtrl1.GetBinDataPath(2);  
MessageBox(path);
```

- 참고 항목(See Also)

### 13) GetContextMenu

ContextIDString에 맞는 HwpMenu 객체를 얻어온다.

- 구문(Syntax)

C++

```
LPDISPATCH GetContextMenu(LPCTSTR contextID);
```

javascript

```
HwpMenu GetContextMenu(string contextID);
```

- 매개변수(Parameters)

*ContextIDString*

아래 표 참조

ContextIDString	설명
HWPCTX_TEXT_SEL	// 텍스트 블록
HWPCTX_CTRL_SEL	// 그리기 개체 선택 (1개)
HWPCTX_CTRL_SEL2	// 그리기 개체 선택 (2개 이상)
HWPCTX_TABLE_SEL	// 셀 편집
HWPCTX_NORMAL	// 일반 텍스트 편집
HWPCTX_IN_TABLE	// 표 내부
HWPCTX_IN_TEXTBOX	// 글상자 내부
HWPCTX_IN_MASTER_PAGE	// 바탕쪽 내부
HWPCTX_IN_FOOTNOTE	// 각주/미주 내부
HWPCTX_IN_HEADER	// 머리말/꼬리말 내부
HWPCTX_IN_CAPTION	// 캡션 내부
HWPCTX_IN_FIELD	// 필드의 내부
HWPCTX_IN_MEMO	// 메모의 내부

- 반환값(Return)

ContextIDString에 맞는 HwpMenu 객체

- 설명(Remarks)  
ContextIDString에 맞는 HwpMenu 객체를 리턴한다.

- 예제(Example)

*javascript*

```
CDHwpMenu hwpMenu = m_HwpCtrl1.GetContextMenu("HWPCTX_TEXT_SEL");  
hwpMenu.AppendMenu("CharShapeBold", "");
```

- 참고 항목(See Also)

[AppendMenu](#), [RemoveMenu](#)

## 14) GetCtrlHorizontalOffset

개체의 가로기준 위치에 대한 현재 위치 값을 얻어온다.

- 구문(Syntax)

*C++*

```
long GetCtrlHorizontalOffset(LPDISPATCH ctrl, long relTo)
```

*javascript*

```
number GetCtrlHorizontalOffset(HwpCtrlCode ctrl, number relTo)
```

- 매개변수(Parameters)

*ctrl*

개체 컨트롤

*relTo*

가로 기준 (종이 : 0, 쪽 : 1, 단 : 2, 문단 : 3)

- 반환값(Return)

가로 위치 값 (HWPUNIT)

- 설명(Remarks)

개체의 가로기준 위치에 대한 현재 위치값을 반환하는 함수

- 예제(Example)

*javascript*

```
var selctrl = pHwpCtrl.CurSelectedCtrl;           // 현재 선택된 개체를 구한다.  
val newoffset = pHwpCtrl.GetCtrlHorizontalOffset(selctrl, 1); // 해당 개체의 가로기준을 "쪽"으로  
설정했을 때의 위치값을 구한다.
```

- 참고 항목(See Also)

[GetCtrlVerticalOffset](#)

## 15) GetCtrlVerticalOffset

개체의 세로기준 위치에 대한 현재 위치 값을 얻어온다.

- 구문(Syntax)

*C++*

```
long GetCtrlVerticalOffset(LPDISPATCH ctrl, long relTo)
```

*javascript*

```
number GetCtrlVerticalOffset(HwpCtrlCode ctrl, number relTo)
```

- 매개변수(Parameters)

*ctrl*

개체 컨트롤

*relTo*

세로 기준 (종이 : 0, 쪽 : 1, 문단 : 2)

- 반환값(Return)

세로 위치 값 (HWPUNIT)

- 설명(Remarks)

개체의 세로기준 위치에 대한 현재 위치 값을 반환하는 함수

- 예제(Example)

*javascript*

```
var selctrl = pHwpCtrl.CurSelectedCtrl;           // 현재 선택된 개체를 구한다.  
val newoffset = pHwpCtrl.GetCtrlVerticalOffset(selctrl, 1); // 해당 개체의 가로기준을 "쪽"으로 설정했을 때의 위치값을 구한다.
```

- 참고 항목(See Also)

[GetCtrlHorizontalOffset](#)

## 16) GetCurFieldName

현재 캐럿이 위치하는 곳의 필드이름을 구한다.

- 구문(Syntax)

C++

```
CString GetCurFieldName(VARIANT& option)
```

javascript

```
string GetCurFieldName([number option])
```

- 매개변수(Parameters)

*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.

- 반환값(Return)

필드이름이 돌아온다. 필드이름이 없는 경우 빈 문자열이 돌아온다.

- 설명(Remarks)

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

- 예제(Example)

- 참고 항목(See Also)

[GetFieldList](#), [SetCurFieldName](#), [CurFieldState](#), [FieldExist](#)



## 17) GetFieldList

문서에 존재하는 필드의 목록을 구한다.

- 구문(Syntax)

*C++*

```
CString GetFieldList(VARIANT& number, VARIANT& option)
```

*javascript*

```
string GetFieldList([number number], [number option])
```

- 매개변수(Parameters)

*number*

문서 내에서 동일한 이름의 필드가 여러 개 존재할 경우 이를 구별하기 위한 식별방법을 지정한다.  
생략하면 `hwpFieldPlain`이 지정된다.

ID	값	설명
<code>hwpFieldPlain</code>	0	아무 기호 없이 순서대로 필드의 이름을 나열한다.
<code>hwpFieldNumber</code>	1	필드이름 뒤에 일련번호가 <code>{{#}}</code> 과 같은 형식으로 붙는다.
<code>hwpFieldCount</code>	2	필드이름 뒤에 그 필드의 개수가 <code>{{#}}</code> 과 같은 형식으로 붙는다.

*option*

다음과 같은 옵션을 조합할 수 있다. 0을 지정하면 모두 `off`이다.

생략하면 0이 지정된다.

ID	값	설명
<code>hwpFieldCell</code>	0x01	셀에 부여된 필드 리스트만을 구한다. <code>hwpFieldClickHere</code> 과는 함께 지정할 수 없다.
<code>hwpFieldClickHere</code>	0x02	누름틀에 부여된 필드 리스트만을 구한다. <code>hwpFieldCell</code> 과는 함께 지정할 수 없다.
<code>HwpFieldSelection</code>	0x04	선택된 내용 안에 존재하는 필드 리스트를 구한다.

- 반환값(Return)

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

"필드이름#1\x2필드이름#2\x2...\필드이름#n"

- 설명(Remarks)

문서 중에 동일한 이름의 필드가 여러 개 존재할 때는 **number**에 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.

<b>hwpFieldPlain</b>	"title\x2body\x2title\x2body\x2footer"
<b>hwpFieldNumber</b>	"title{{0}}\x2body{{0}}\x2title{{1}}\x2body{{1}}\x2footer{{0}}"
<b>hwpFieldCount</b>	"title{{2}}\x2body{{2}}\x2footer{{1}}"

- 예제(Example)

- 참고 항목(See Also)

[FieldExist](#), [SetCurFieldName](#), [CurFeildState](#)

## 18) GetFieldText

지정한 필드에서 문자열을 구한다.

- 구문(Syntax)

C++

```
CString GetFieldText(LPCTSTR fieldlist)
```

javascript

```
string GetFieldText(string fieldlist)
```

- 매개변수(Parameters)

*fieldlist*

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

```
"필드이름#1\x2필드이름#2\x2...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}}\x2본문\x2이름{{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.

- 반환값(Return)

텍스트 데이터가 돌아온다. 텍스트에서 탭은 '\t'(0x09), 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.

- 설명(Remarks)

- 예제(Example)

- 참고 항목(See Also)

[PutFieldText](#), [RenameField](#), [MoveToField](#), [MoveToFieldEx](#)

## 19) GetFileInfo

파일 정보를 알아낸다.

- 구문(Syntax)

C++

```
LPDISPATCH GetFileInfo(LPCTSTR FileName)
```

[ver:0x05050116](#)

javascript

```
ParameterSet GetFileInfo(string FileName)
```

- 매개변수(Parameters)

*FileName*

정보를 구하고자 하는 파일명

- 반환값(Return)

"FileInfo" ParameterSet이 반환된다.

※ FileInfo

Item ID	Type	설명
Format	string	파일의 형식. HWP : 한글 파일 UNKNOWN : 알 수 없음.
VersionStr	string	파일의 버전 문자열 ex) 5.0.0.3
VersionNum	unsigned int	파일의 버전 ex) 0x05000003
Encrypted	int	암호 여부 (현재는 파일 버전 3.0.0.0 이후 문서-한글97, 한글 워디안, 한글 2002-에 대 해서만 판단한다.) -1 : 판단 할 수 없음 0 : 암호가 걸려 있지 않음 양수 : 암호가 걸려 있음.

- 설명(Remarks)

한글 문서를 열기 전에 암호가 걸린 문서인지 확인할 목적으로 만들어졌다.

- 예제(Example)

C++

```
DHwpParameterSet dFileInfoSet = pHwpCtrl->GetFileInfo(strFileName);
TRACE(_T("---File Information: %s\n"), strFileName);
TRACE(_T("Format          : %s\n"), (CString)(dFileInfoSet.Item(_T("Format")).bstrVal));
TRACE(_T("Version String   : %s\n"), (CString)(dFileInfoSet.Item(_T("VersionStr")).bstrVal));
TRACE(_T("Version Number  : %08x\n"), dFileInfoSet.Item(_T("VersionNum")).ulVal);
TRACE(_T("Encription Type : %d\n"), dFileInfoSet.Item(_T("Encrypted")).iVal);
TRACE(_T("---File Information End\n"));
```

- 참고 항목(See Also)

## 20) GetFilterList

흔글에서 사용가능한 필터(포맷)의 리스트를 얻어온다.

- 구문(Syntax)

C++

```
BOOL GetFilterList(BSTR * szfilterlist, VARIANT& flags)
```

- 매개변수(Parameters)

szfilterlist

- 반환값(Return)

성공하면 **true**를 실패하면 **false**를 반환한다.

- 설명(Remarks)

BSTR값으로 필터 리스트가 넘어오나 내부적으로 **cstring**값을 사용하므로 **Unicode**로 얻을 수 없고 **ANSI Character**로 얻게 된다.

매개변수로 포인터를 사용하므로, 스크립트 언어(**javascript**, **visual basic**)에서 사용할 수 없다.

- 예제(Example)

- 참고 항목(See Also)

## 21) GetFormObjectAttr

(2004.11.12) 한글2005

양식개체의 특성 값을 얻어온다.

- 구문(Syntax)

C++

```
VARIANT GetFormObjectAttr(LPCTSTR formname, LPCTSTR attrname)
```

[ver:0x0605010A](#)

javascript

```
variant GetFormObjectAttr(string formname, string attrname)
```

- 매개변수(Parameters)

*formname*

양식개체 이름

*attrname*

특성 이름

- 반환값(Return)

양식개체의 특성 값을 반환한다.

해당 양식개체가 없거나 속성 값이 없다면 빈값을 반환한다.

- 설명(Remarks)

양식 개체별 속성의 이름

공통 속성

attrname value	type	설명
Name	String	양식 개체의 이름
AutoSize	BOOL	문자열에 맞추어서 양식개체 크기를 조절
BackColor	INT	배경색
ForeColor	INT	전경색
BorderType	Enum	테두리 유형 ("None", "Flat", "Bump", "Etched", "Raised", "Sunken")

DrawFrame	BOOL	양식 개체의 표현할지를 선택(에디트, 콤보만 적용) 양식개체의 내용은 출력된다.
Editable	BOOL	편집상태
FollowContext	BOOL	양식개체 주위의 글자 속성과 일치
GroupNmae	String	양식 개체의 group name
TabOrder	INT	탭의 순서
Height	INT	높이
width	INT	폭

#### 버튼 속성

attrname value	type	설명
Caption	String	버튼 문자열
wordwrap	BOOL	단어단위 처리

#### 체크박스 속성

attrname value	type	설명
BackStyle	Enum	배경 style (Transparent, Opaque)
Caption	String	체크박스 문자열
TriState	BOOL	체크의 상태를 3가지 상태 설정( 선택, 비선택, 선택 비활성화)
value	BOOL	체크 박스 값
wordwrap	BOOL	단어단위 처리

#### 라디오상자 속성

attrname value	type	설명
BackStyle	Enum	배경 style (Transparent, Opaque)
Caption	String	라디오 상자 문자열
RadioGroupName	String	라디오 그룹 문자열
TriState	BOOL	3가지 상태 설정( 선택, 비선택, 선택 비활성화)
value	BOOL	라디오 상자의 값
wordwrap	BOOL	단어단위 처리



#### 콤보박스 속성

attrname value	type	설명
EditEnable	BOOL	콤보박스의 에디트를 편집가능하게 한다.
ListBoxRows	INT	콤보박스를 펼치면 나오는 리스트의 수
ListBoxWidth	INT	콤보박스를 펼치면 나오는 리스트의 폭
Text	String	콤보박스의 값
wordwrap	BOOL	단어단위 처리

#### 에디트상자 속성

attrname value	type	설명
MaxLength	INT	에디트 상자의 text의 길이
PasswordChar	CHAR	password를 표현할 문자
ReadOnly	BOOL	편집 상태
TabKeyBehavior	ENUM	에디트 상자에서 tab 키를 처리 방법(NextObject, InsertTab)
Text	String	에디트 상자의 값

- 예제(Example)
- 참고 항목(See Also)  
[SetFormObjectAttr](#)

## 22) GetHeadingString

글머리표/문단번호/개요번호를 추출한다.

- 구문(Syntax)

*C++*

```
CString GetHeadingString();
```

*Javascript*

```
string GetHeadingString();
```

- 매개변수(Parameters)

없음

- 반환값(Return)

문자열이 반환된다.

글머리표/문단번호/개요번호가 있다면 해당 문자열이 반환된다.

- 설명(Remarks)

글머리표/문단번호/개요번호가 있는 경우, 해당 문자열을 얻어올 수 있다.

문단에 글머리표/문단번호/개요번호가 없는 경우, 빈 문자열이 추출된다.

- 예제(Example)

*C++*

```
CString head = m_HwpCtrl1.GetHeadingString();  
MessageBox(head);
```

- 참고 항목(See Also)

### 23) GetMessageBoxMode

현재 메시지 박스의 Mode를 int로 얻어온다.

- 구문(Syntax)

*C++*

```
long GetMessageBoxMode()
```

*javascript*

```
number GetMessageBoxMode()
```

- 매개변수(Parameters)

- 반환값(Return)

SetMessageBoxMode 참고

- 설명(Remarks)

- 예제(Example)

*javascript*

- 참고 항목(See Also)

[SetMessageBoxMode](#)

## 24) GetMessageSet

(2003.2.27)

이벤트에서 전달된 문자열을 `ParameterSet`으로 얻어온다.

- 구문(Syntax)

*C++*

```
LPDISPATCH GetMessageSet()
```

[ver:0x05070125](#)

*javascript*

```
ParameterSet GetMessageSet()
```

- 매개변수(Parameters)

- 반환값(Return)

※차후 업데이트

- 설명(Remarks)

OCX Event가 외부에 알려질 때 전달 값이 BSTR인 경우 GetMessageSet을 통하여 Event에 의한 BSTR 전달 값을 얻을 수 있다.

- 예제(Example)

- 참고 항목(See Also)

[HyperlinkMode](#), [NotifyMessage](#)

## 25) GetMousePos

마우스의 현재 위치를 얻어온다.

- 구문(Syntax)

C++

```
LPDISPATCH GetMousePos(long Xrelto, long Yrelto)
```

[ver:0x05050111](#)

javascript

```
ParameterSet GetMousePos(number Xrelto, number Yrelto)
```

- 매개변수(Parameters)

*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.

- 반환값(Return)

"MousePos" ParameterSet이 반환된다.

※ MousePos

Item ID	Type	설명
XRelTo	unsigned long	가로 상대적 기준 0 : 종이 1 : 쪽
YRelTo	unsigned long	세로 상대적 기준 0 : 종이

		1 : 쪽
Page	unsigned long	페이지 번호 ( 0 based)
X	long	가로 클릭한 위치 (HWPUNIT)
Y	long	세로 클릭한 위치 (HWPUNIT)

- 설명(Remarks)

단위가 HWPUNIT임을 주의해야 한다.

※ 1 inch = 7200 HWPUNIT, 1mm = 283.465 HWPUNIT

- 예제(Example)

*javascript*

```
function HwpCtrl_OnMouseDown(x,y)
{
    var MousePosSet = pHwpCtrl.GetMousePos(0, 0); // 쪽기준
    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");
    HwpControl.page.value = page + 1; // 페이지
    HwpControl.page_mousex_mm.value = Math.floor(pagex / 283.465); // 1mm == 283.465 HWPUNIT
    HwpControl.page_mousey_mm.value = Math.floor(pagey / 283.465);
    MousePosSet = pHwpCtrl.GetMousePos(1, 1); // 종이기준
    var paperx = MousePosSet.Item("X");
    var papery = MousePosSet.Item("Y");
    HwpControl.paper_mousex_mm.value = Math.floor(paperx / 283.465);
    HwpControl.paper_mousey_mm.value = Math.floor(papery / 283.465);
}
```

- 참고 항목(See Also)

## 26) GetPageText

- 페이지 단위의 텍스트 추출

- 구문(Syntax)

*C++*

```
CString GetPageText(long pgno, VARIANT& option)
```

*javascript*

```
string GetPageText(number pgno, variant option)
```

- 매개변수(Parameters)

*pgno*

텍스트를 추출 할 페이지의 번호(0부터 시작)

*option*

추출 대상을 다음과 같은 옵션을 조합하여 지정할 수 있다. 생략(또는 0xffffffff)하면 모든 텍스트를 추출한다.

ID	값	설명
maskNormal	0x00	본문 텍스트를 추출한다. (항상 기본)
maskTable	0x01	표에대한 텍스트를 추출한다.
maskTextbox	0x02	글상자 텍스트를 추출한다.
maskCaption	0x04	캡션 텍스트를 추출한다. (표, ShapeObject)

- 반환값(Return)

해당 페이지의 텍스트가 추출된다.

- 설명(Remarks)

일반 텍스트(글자처럼 취급 도형 포함)를 우선적으로 추출하고, 도형(표, 글상자) 내의 텍스트를 추출한다.

- 예제(Example)

*C++*

```
CString str = m_HwpCtrl.GetPageText(1, CComVariant(0));  
MessageBox(str);
```

*javascript*

```
var str = vHwpCtrl.GetPageText(0, 1);  
alert(str);
```

- 참고 항목(See Also)



## 27) GetPos

캐럿의 위치를 얻어온다.

- 구문(Syntax)

C++

```
void GetPos(long * list, long * para, long * pos)
```

- 매개변수(Parameters)

*list*

캐럿이 위치한 문서 내 `list` ID

*para*

캐럿이 위치한 문단 ID

*pos*

캐럿이 위치한 문단 내 글자 위치

- 반환값(Return)

- 설명(Remarks)

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

리스트 아이디는 문서 내 위치 정보 중 하나로서 `SelectText`에 넘겨줄 때 사용한다.

매개변수로 포인터를 사용하므로, 포인터를 사용할 수 없는 언어에서는 사용이 불가능 하다.

포인터를 사용하지 않는 언어를 지원하기 위해서 `ParameterSet`을 사용하는 `GetPosBySet()`가 존재한다.

- 예제(Example)

- 참고 항목(See Also)

[SetPos](#), [SelectText](#), [GetPosBySet](#), [SetPosBySet](#), [GetSelectedPos](#), [GetSelectedPosBySet](#)

## 28) GetPosBySet

현재 캐럿의 위치 정보를 `ParameterSet`으로 얻어온다.

- 구문(Syntax)

*C++*

```
LPDISPATCH GetPosBySet()
```

[ver:0x05050104](#)

*javascript*

```
ParameterSet GetPosBySet()
```

- 매개변수(Parameters)

- 반환값(Return)

캐럿 위치에 대한 `ParameterSet`

- 설명(Remarks)

캐럿의 위치를 `ParameterSet`으로 얻어온다. 포인터를 사용할 수 없는 언어에서도 사용가능하다.

- 예제(Example)

*javascript*

```
var lpp;
function GetPosTest()
{
    lpp = HwpControl.MyHwpCtrl.GetPosBySet();

    alert(lpp.Item("List") + " " + lpp.Item("Para") + " " + lpp.Item("Pos"));
}
function SetPosTest()
{
    if ( lpp != null)
    {
        if (!HwpControl.MyHwpCtrl.SetPosBySet(lpp))
        {
```

```
        alert ("GetPos err");
    }
}
else
{
    alert("GetPos를 먼저 실행하세요.");
}
}
```

- 참고 항목(See Also)

[GetPos](#), [SetPos](#), [SelectText](#), [SetPosBySet](#), [GetSelectedPos](#), [GetSelectedPosBySet](#)

## 29) GetScriptSource

헌글에서 사용하는 매크로의 소스코드를 가져온다.

- 구문(Syntax)

*C++*

```
CString GetScriptSource(LPCTSTR filepath)
```

*javascript*

```
string GetScriptSource(string filepath)
```

- 매개변수(Parameters)

*filepath*

매크로 소스를 가져올 헌글문서

- 반환값(Return)

script source code.

- 설명(Remarks)

- 예제(Example)

*javascript*

```
<HTML>
<BODY>
<FORM name = "HwpControl">
<INPUT type="file" name="FileName" />
<INPUT type="button" value="스크립트 얻어오기" onClick="javascript:OnGetScript()" /> <BR>
<INPUT type="textarea" name="Scrp" />
<OBJECT id=HwpCtrl" style="LEFT:0px; TOP:0px;" width="640" height="480"
        classid="CLSID:BD9C32DE-3155-4691-8972-097D53B10052" />
</FORM>

<SCRIPT language="javascript">
var vHwpCtrl = HwpControl.HwpCtrl;
```

```
function OnGetScript() {  
    var scrpt = vHwpCtrl.GetScriptSource(HwpControl.FileName.value);  
    HwpControl.Scrpt.value = scrpt;  
}  
</SCRIPT>  
</BODY>  
</HTML>
```

- 참고 항목(See Also)  
[RunScriptMacro](#)

### 30) GetSelectedPos

(2005.5.20) 한글2005

현재 설정된 블록의 위치정보를 얻어온다.

- 구문(Syntax)

C++

```
BOOL GetSelectedPos(long * slist, long * spara, long * spos, long * elist, long * epara, long * epos)  
ver:0x06050110
```

- 매개변수(Parameters)

*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.

- 반환값(Return)

성공하면 **true**, 실패하면 **false**

- 설명(Remarks)

위의 리스트란, 문단과 컨트롤들이 연결된 한글문서 내 구조를 뜻한다.

리스트 아이디는 문서 내 위치정보중 하나로서 **SelectText**에 넘겨줄 때 사용한다.

매개변수로 포인터를 사용하므로, 포인터를 사용할 수 없는 언어에서는 사용이 불가능 하다.

포인터를 사용하지 않는 언어를 지원하기 위해서 **ParameterSet**을 사용하는 **GetSelectedPosBySet()**이 존재한다.

- 예제(Example)

C++

```
long slist, spara, spos;  
long elist, epara, epos;  
m_CHwpCtrl.GetSelectedPos(&slist, &spara, &spos, &elist, &epara, &epos);
```

- 참고 항목(See Also)

[GetPos](#), [SetPos](#), [SelectText](#), [GetPosBySet](#), [SetPosBySet](#), [GetSelectedPosBySet](#)

### 31) GetSelectedPosBySet

(2005.5.20) 한글2005

현재 설정된 블록의 위치정보를 얻어온다. (GetSelectedPos의 ParameterSet버전)

- 구문(Syntax)

*C++*

```
BOOL GetSelectedPosBySet(LPDISPATCH sset, LPDISPATCH eset)
```

[ver:0x06050110](#)

*javascript*

```
boolean GetSelectedPosBySet(ParameterSet sset, ParameterSet eset)
```

- 매개변수(Parameters)

*sset*

설정된 블록의 시작 파라미터셋 (ListParaPos)

*eset*

설정된 블록의 끝 파라미터셋 (ListParaPos)

- 반환값(Return)

성공하면 true, 실패하면 false

- 설명(Remarks)

- 예제(Example)

*javascript*

```
function {  
    var sset, eset;  
    sset = pHwpCtrl.CreateSet("ListParaPos");  
    eset = pHwpCtrl.CreateSet("ListParaPos");  
    var ret = pHwpCtrl.GetSelectedPosBySet(sset, eset);  
  
    var slist, spara, spos;  
    var elist, epara, epos;  
  
    slist = sset.Item("List");
```



```
spara = sset.Item("Para");  
spos = sset.Item("Pos");  
  
elist = eset.Item("List");  
epara = eset.Item("Para");  
epos = eset.Item("Pos");  
}
```

- 참고 항목(See Also)

[GetPos](#), [SetPos](#), [SelectText](#), [GetPosBySet](#), [SetPosBySet](#), [GetSelectedPos](#)

## 32) GetTableCellAddr

(2005.8.4) 한글2005

현재 커서가 위치한 셀의 위치정보를 얻어온다

- 구문(Syntax)

C++

```
long GetTableCellAddr(long type)
```

[ver:0x06070113](#)

javascript

```
number GetTableCellAddr(number type)
```

- 매개변수(Parameters)

*type*

행 또는 열 (0:행, 1:열)

- 반환값(Return)

0~n까지의 행, 열의 위치정보를 얻어온다.

커서가 셀에 있지 않거나, 실패하면 -1을 반환한다.

- 설명(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)

- 예제(Example)

javascript

```
function {  
    var col = pHwpCtrl.GetTableCellAddr(0);  
    var row = pHwpCtrl.GetTableCellAddr(1);  
}
```

- 참고 항목(See Also)

### 33) GetText

문서 내에서 텍스트를 얻어온다.

- 구문(Syntax)

C++

```
long GetText(BSTR * text)
```

- 매개변수(Parameters)

*text*

텍스트 데이터가 돌아온다. 텍스트에서 탭은 '\t'(0x9), 문단 바꿈은 CR/LF(0x0D/0x0A)로 표현되며, 이외의 특수 코드는 포함되지 않는다.

- 반환값(Return)

다음과 같은 결과 값을 반환한다.

0	텍스트 정보 없음
1	리스트의 끝
2	일반 텍스트
3	다음 문단
4	제어문자 내부로 들어감
5	제어문자를 빠져나옴
101	초기화 안 됨 (InitScan()) 실패 또는 InitScan()을 실행하지 않은 경우)
102	텍스트 변환 실패

- 설명(Remarks)

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.

매개변수로 포인터를 사용하므로, 포인터를 사용할 수 없는 언어에서는 사용이 불가능 하다.

포인터를 사용하지 않는 언어를 지원하기 위해서 ParameterSet을 사용하는 GetTextBySet()가 존재한다.

- 예제(Example)

- 참고 항목(See Also)

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

### 34) GetTextBySet

문서 내에서 텍스트를 얻어온다. (GetText의 ParameterSet 버전)

- 구문(Syntax)

C++

```
long GetTextBySet(LPDISPATCH text)
```

[ver:0x05050104](#)

javascript

```
number GetTextBySet(ParameterSet text)
```

- 매개변수(Parameters)

*text*

Item의 "Text"에 텍스트 데이터가 돌아온다. 텍스트에서 탭은 '\t'(0x9), 문단 바꿈은 CR/LF(0x0D/0x0A)로 표현되며, 이외의 특수 코드는 포함되지 않는다.

- 반환값(Return)

다음과 같은 결과 값을 반환한다.

0	텍스트 정보 없음
1	리스트의 끝
2	일반 텍스트
3	다음 문단
4	제어문자 내부로 들어감
5	제어문자를 빠져나옴
101	초기화 안 됨 (InitScan() 실패 또는 InitScan()을 실행하지 않은 경우)
102	텍스트 변환 실패

- 설명(Remarks)

GetText의 ParameterSet 버전이다. 포인터를 사용할 수 없는 언어를 지원하기 위해 추가되었다.

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.

- 예제(Example)

*javascript*

```
function GetTextTest()
{
    var text;
    var ret;
    HwpControl.MyHwpCtrl.InitScan();
    text = HwpControl.MyHwpCtrl.CreateSet("GetText");
    ret = HwpControl.MyHwpCtrl.GetTextBySet(text);
    alert(ret + ":" + text.Item("Text"));
    HwpControl.MyHwpCtrl.ReleaseScan();
}
```

- 참고 항목(See Also)

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

### 35) GetTextFile

현재 열린 문서를 문자열로 넘겨준다.

- 구문(Syntax)

C++

```
VARIANT GetTextFile(LPCTSTR format, LPCTSTR option)
```

[ver:0x05050109](#)

javascript

```
string GetTextFile(string format, string option)
```

- 매개변수(Parameters)

*format*

파일의 형식

format	설명	비고
HWP	HWP native format	BASE64로 인코딩되어 있다. 저장된 내용을 다른 곳에서 보여줄 필요가 없다면 이 포맷을 사용하기를 권장합니다. <a href="#">ver:0x0505010B</a>
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	일반 텍스트	유니코드에만 있는 정보(한자, 고어, 특수문자 등)는 모두 손실된다.

*option*

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

- 반환값(Return)

지정된 포맷에 맞춰 파일을 문자열로 변환한 값을 반환한다.

- 설명(Remarks)

이 함수는 JScript나 VBScript와 같이 직접적으로 local disk를 접근하기 힘든 언어를 위해 만들어졌으므로 disk를 접근할 수 있는



언어에서는 사용하지 않기를 권장합니다.

disk를 접근할 수 있다면, **Save**나 **SaveBlockAction**을 사용하십시오.

이 함수 역시 내부적으로는 **save**나 **SaveBlockAction**을 호출하도록 되어있고 텍스트로 저장된 파일이 메모리에서 3~4번 복사되기 때문에 느리고, 메모리를 낭비합니다.

- 예제(Example)

*javascript*

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

- 참고 항목(See Also)

[SetTextFile](#)

### 36) GetViewStatus

(2003.10.30) 한글2004

현재 보이고 있는 view의 절대 위치 값을 알려준다.

- 구문(Syntax)

C++

```
LPDISPATCH GetViewStatus(long nType)
```

[ver:0x05070129](#)

javascript

```
ParameterSet GetViewStatus(number nType)
```

- 매개변수(Parameters)

nType

view의 정보를 얻기 위한 Flag (0일 경우 view의 절대 위치 값을 알려준다)

- 반환값(Return)

view의 상태정보를 지는 ParameterSet

※ ViewStatus

Item ID	Type	설명
Type	unsigned int	0 (현재 View의 절대 값 Pos를 알려준다.)
ViewPosX	int	현재 뷰의 X값
ViewPosY	int	현재 뷰의 Y값

- 설명(Remarks)

- 예제(Example)

C++

```
LONG lHwpViewLeft = 0; //한글 전체 페이지에 대하여 현재 보이는 뷰의Left 좌표  
LONG lHwpViewTop = 0; //한글 전체 페이지에 대하여 현재 보이는 뷰의Top 좌표  
DHwpParameterSet set = m_ctrlHwp.GetViewStatus(0);  
VARIANT var = {0,};  
var = set.Item("ViewPosX");
```

```
lHwpViewLeft = var.intVal; // 어떤 변수에 값이 들어오는지 체크하기  
var = set.Item("ViewPosY");  
lHwpViewTop = var.intVal; // 어떤 변수에 값이 들어오는지 체크하기  
  
TRACE(_T("[GetHwpViewStatus()] lHwpViewLeft = %d, lHwpViewTop = %d\n"),  
      lHwpViewLeft, lHwpViewTop);
```

- 참고 항목(See Also)

### 37) GetVersionHistoryCount

(2005.12.19) 한글2005

현재문서의 버전정보 개수를 반환한다.

- 구문(Syntax)

*C++*

```
long GetVersionHistoryCount() ver:0x06070115
```

*javascript*

```
number GetVersionHistoryCount()
```

- 매개변수(Parameters)

- 반환값(Return)

현재문서의 버전정보의 개수를 반환한다.

- 설명(Remarks)

- 예제(Example)

*javascript*

```
var count = pHwpCtrl.GetVersionHistoryCount();
```

- 참고 항목(See Also)

[GetVersionInfo](#), [MakeVersionDiffAll](#), [VersionSave](#), [VersionDelete](#)

### 38) GetVersionInfo

(2005.12.19) 한글2005

현재문서의 버전정보 아이템을 얻어온다.

- 구문(Syntax)

C++

```
LPDISPATCH GetVersionInfo(long index)
```

[ver:0x06070115](#)

javascript

```
ParameterSet GetVersionInfo(number index)
```

- 매개변수(Parameters)

*index*

가지고 올 버전정보 아이템의 index

- 반환값(Return)

VersionInfo의 ParameterSet을 반환한다.

※ VersionInfo: 버전정보

Item ID	Type	설명
SourcePath	string	버전 비교용 소스 패스
TargetPath	string	버전 비교용 타겟 패스
ItemStartIndex	byte	버전 비교를 보여줄 시작 히스토리 인덱스
ItemEndIndex	byte	버전 비교를 보여줄 마지막 히스토리 인덱스
ItemOverWrite	byte	히스토리 정보를 저장할 때 마지막 버전으로 덮어쓰는 플랙 (on : 1/off : 0)
ItemSaveDescription	byte	히스토리 정보를 저장할 때 설명을 입력하는 대화상자를 띄우는 플랙 (on : 1/off : 0) 지은이 정보가 직전 버전의 정보와 다를 경우 저장을 하지 않기 때문에 대화상자를 생략할 경우 지은이정보를 꼭 써야만 한다.
TempFilePath	ParameterArray	버전 비교용 임시파일 경로
ItemInfoIndex	unsigned long	버전 정보 얻어오기 및 삭제시 사용될 인덱스
SaveFilePath	string	버전저장 파일 경로 (OCX 컨트롤 용)

ItemInfowriter	string	지은이 정보
ItemInfoDescription	string	해당 버전에 대한 설명
ItemInfoTimeHi	unsigned long	날짜 정보, FILETIME의 HIWORD
ItemInfoTimeLo	unsigned long	날짜 정보, FILETIME의 LOWORD
ItemInfoLock	byte	버전정보 잠그기 (on : 1/off :0)

- 설명 (Remarks)

- 예제 (Example)

*javascript*

```
var set = pHwpCtrl.GetVersionInfo(0);
var str = set.Item("ItemInfowriter");
var desc = set.Item("ItemSaveDescription");
```

- 참고 항목 (See Also)

[MakeVersionDiffAll](#), [GetVersionHistoryCount](#), [VersionSave](#), [VersionDelete](#)

### 39) ImportStyle

현재 문서의 style을 Import한다.

- 구문(Syntax)

*C++*

```
BOOL ImportStyle(LPDISPATCH styleset)
```

*javascript*

```
boolean ImportStyle(ParameterSet styleset)
```

- 매개변수(Parameters)

*styleset*

Import할 스타일 Set

- 반환값(Return)

성공하면 true, 실패하면 false

- 설명(Remarks)

ImportStyle을 이용하여 해당 문서의 스타일을 변경할 수 있다.

- 예제(Example)

*C++*

```

CDHwpAction dact;
CDHwpParameterSet dset;
CDHwpParameterArray dNameLocals, dNameEngs;

dact = m_hwpobj.CreateAction(_T("StyleTemplate"));
dset = dact.CreateSet();

dset.SetItem(_T("FileName"), (COleVariant)(LPCTSTR)_T("C:\\Normal.hwp"));
dNameLocals = dset.CreateItemArray(_T("NameLocals"), 1);
dNameEngs = dset.CreateItemArray(_T("NameEngs"), 1);

dNameLocals.SetItem(0, (COleVariant)(LPCTSTR)_T("바탕글"));
dNameEngs.SetItem(0, (COleVariant)(LPCTSTR)_T("Normal"));

BOOL bRet = m_hwpobj.ImportStyle(dset);

```

- 참고 항목(See Also)

[ExportStyle](#)



#### 40) InitScan

문서의 내용을 검색하기 위해 초기설정을 한다.

- 구문(Syntax)

*C++*

```
BOOL InitScan(VARIANT& option, VARIANT& range, VARIANT& spara, VARIANT& spos, VARIANT& epara, VARIANT& epos)
```

*javascript*

```
boolean InitScan([number option], [number range], [number spara], [number spos], [number epara], [number epos])
```

- 매개변수(Parameters)

*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, 문단 끝, 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, 프레젠테이션, 다단, 누름틀 필드 시작, Shape object, 머리말, 꼬리말, 각주, 미주, 번호관련 컨트롤, 새 번호 관련 컨트롤, 감추기, 찾아보기, 글자 겹침, 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)

<code>scanSposControl</code>	<code>0x0060</code>	컨트롤의 시작부터. (시작 위치)
<code>scanSposDocument</code>	<code>0x0070</code>	문서의 시작부터. (시작 위치)
<code>scanEposCurrent</code>	<code>0x0000</code>	캐럿 위치까지. (끝 위치)
<code>scanEposSpecified</code>	<code>0x0001</code>	특정 위치까지. (끝 위치)
<code>scanEposLine</code>	<code>0x0002</code>	줄의 끝까지. (끝 위치)
<code>scanEposParagraph</code>	<code>0x0003</code>	문단의 끝까지. (끝 위치)
<code>scanEposSection</code>	<code>0x0004</code>	구역의 끝까지. (끝 위치)
<code>scanEposList</code>	<code>0x0005</code>	리스트의 끝까지. (끝 위치)
<code>scanEposControl</code>	<code>0x0006</code>	컨트롤의 끝까지. (끝 위치)
<code>scanEposDocument</code>	<code>0x0007</code>	문서의 끝까지. (끝 위치)
<code>scanWithinSelection</code>	<code>0x00ff</code>	검색의 범위를 블록으로 제한.
<code>scanForward</code>	<code>0x0000</code>	정방향. (검색 방향)
<code>scanBackward</code>	<code>0x0100</code>	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.

- 반환값(Return)

성공하면 `true`, 실패하면 `false`

- 설명(Remarks)

문서의 검색 과정은 `InitScan()`으로 검색위한 준비 작업을 하고 `GetText()`를 호출하여 본문의 텍스트를 얻어온다.

`GetText()`를 반복호출하면 연속하여 본문의 텍스트를 얻을 수 있다. 검색이 끝나면 `ReleaseScan()`을 호출하여 관련 정보를 Release해야 한다.

- 예제(Example)

- 참고 항목(See Also)  
[GetText](#), [ReleaseScan](#), [MovePos](#), [GetTextBySet](#)

## 41) Insert

현재 커서 위치에 문서파일을 삽입한다.

- 구문(Syntax)

*C++*

```
BOOL Insert(LPCTSTR path, VARIANT& format, VARIANT& arg)
```

*Javascript*

```
boolean Insert(string path, [string format], [string arg])
```

- 매개변수(Parameters)

*path*

문서파일의 경로, URL 사용가능([ver:0x05050111](#))

*format*

문서형식. 별도 설명 참조. 빈 문자열을 지정하면 자동으로 선택한다. 생략하면 빈 문자열이 지정된다.

*arg*

세부옵션. 의미는 **format**에 지정한 파일형식에 따라 다르다. 생략하면 빈 문자열이 지정된다.

- 반환값(Return)

성공하면 **true**, 실패하면 **false**

- 설명(Remarks)

**format**, **arg**에 대해서는 **open** 참조

- 예제(Example)

- 참고 항목(See Also)

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

## 42) InsertCtrl

현재 캐럿 위치에 컨트롤을 삽입한다.

- 구문(Syntax)

*C++*

```
LPDISPATCH InsertCtrl(LPCTSTR ctrlid, VARIANT& initparam)
```

*javascript*

```
Ctrl InsertCtrl(string ctrlid, [ParameterSet initparam])
```

- 매개변수(Parameters)

*ctrlid*

삽입할 컨트롤 ID

*initparam*

컨트롤 초기속성. 생략하면 **default** 속성으로 생성한다.

- 반환값(Return)

생성된 컨트롤 **object**

- 설명(Remarks)

*ctrlid*에 지정할 수 있는 컨트롤 ID는 **HwpCtrl.CtrlID**가 반환하는 ID와 동일하다. 자세한 것은 **Ctrl** 오브젝트 **Properties**인 **CtrlID**를 참조.

*initparam*에는 컨트롤의 초기 속성을 지정한다. 대부분의 컨트롤은 **Ctrl.Properties**와 동일한 포맷의 **parameter set**을 사용하지만, 컨트롤 생성 시에는 다른 포맷을 사용하는 경우도 있다. 예를 들어 표의 경우 **Ctrl.Properties**에는 **"Table"** 셋을 사용하지만, 생성 시 *initparam*에 지정하는 값은 **"TableCreation"** 셋이다.

- 예제(Example)

*visual basic*

```
' 5x5 크기의 표를 삽입한다.  
Dim tbset As ParameterSet  
Dim table As Ctrl
```

```
Set tbset = HwpCtrl.CreateSet("TableCreation")
tbset.SetItem "Rows", 5
tbset.SetItem "Cols", 5
Set table = HwpCtrl.InsertCtrl("tbl", tbset)
```

- 참고 항목(See Also)

[DeleteCtrl](#)

### 43) InsertBackgroundPicture

배경 이미지를 넣는다.

- 구문(Syntax)

*C++*

```
VARIANT InsertBackgroundPicture(LPCTSTR bordertype, LPCTSTR Path, VARIANT& embedded, VARIANT& filloption, VARIANT& watermark, VARIANT& effect, VARIANT& brightness, VARIANT& contrast)  
ver:0x05050106
```

*javascript*

```
variant InsertBackgroundPicture(string bordertype, string path, [boolean embedded], [number filloption],  
[boolean watermark], [number effect], [number brightness], [number contrast]);
```

- 매개변수(Parameters)

*bordertype*

배경 유형을 지정

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

*path*

삽입할 이미지 파일, URL 사용 가능([ver:0x05050111](#))

*embedded*

이미지 파일을 문서 내에 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 기본 값
6	가운데로	<a href="#">ver:0x05050111</a>
7	가운데 위로	<a href="#">ver:0x05050111</a>
8	가운데 아래로	<a href="#">ver:0x05050111</a>
9	왼쪽 가운데로	<a href="#">ver:0x05050111</a>
10	왼쪽 위로	<a href="#">ver:0x05050111</a>
11	왼쪽 아래로	<a href="#">ver:0x05050111</a>
12	오른쪽 가운데로	<a href="#">ver:0x05050111</a>
13	오른쪽 위로	<a href="#">ver:0x05050111</a>
14	오른쪽 아래로	<a href="#">ver:0x05050111</a>

#### *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 기본 값
1	그레이 스케일	
2	흑백으로	

#### *brightness*

밝기 지정(-100 ~ 100), 기본 값 : 0

#### *contrast*

선명도 지정(-100 ~ 100), 기본 값 : 0

- 반환값(Return)

성공했을 경우 true, 실패했을 경우 false

- 설명(Remarks)

CellBorderFill의 SetItem 중 FillAttr 의 SetItem FileName 에 이미지의 binary data를 지정해 줄 수가 없어서 만든 함수



다.

기타 배경에 대한 다른 조정은 **Action**과 **ParameterSet**의 조합으로 가능하다.

- 예제(Example)

*C++*

```
#define OLELONG(X) (ColeVariant)(long)(X)
void CHwpXCtrlFrameView::OnMenuItemSetCellBgImg()
{
    m_cHwpCtrl.InsertBackgroundPicture("SelectedCell", "test.bmp", OLELONG(1), OLELONG(5),
                                       OLELONG(1), OLELONG(1), OLELONG(0), OLELONG(0));
}
```

*javascript*

```
function InsertBgImg()
{
    HwpControl.HwpCtrl.InsertBackgroundPicture("SelectedCell", BasePath+"test.bmp",1,5,0,0,0,0);
}
function DeleteBgImg()
{
    HwpControl.HwpCtrl.InsertBackgroundPicture("SelectedCellDelete", 0, 0, 0, 0, 0, 0, 0);
}
function TestBgImg()
{
    var act;
    var set;
    var subset;
    if (HwpControl.HwpCtrl.ParentCtrl != null && HwpControl.HwpCtrl.ParentCtrl.Ctrlid == "tbl")
    {
        HwpControl.HwpCtrl.Run("TableCellBlock"); // 반드시 셀을 선택해야 한다.
        act = HwpControl.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");
        }
        else
        {
            alert("셀 테두리 배경이미지가 존재하지 않음");
        }
    }
}
else
{
    alert("셀 테두리 배경에 대한 정보가 존재하지 않음");
}
HwpControl.HwpCtrl.Run("Cancel");
}
else
{
    alert("현재 위치가 표의 셀이 아님.");
}
}

```

- 참고 항목(See Also)

[InsertPicture](#)

#### 44) InsertMenu

(2009.1.19) 한글2007

메뉴를 추가한다.

- 구문(Syntax)

C++

```
BOOL InsertMenu(LPCTSTR menuidx, LPCTSTR menustr, LPCTSTR actionstr, long menutype) ver:0x07050206
```

javascript

```
boolean InsertMenu(string menuidx, string menustr, string actionstr, long menutype)
```

- 매개변수(Parameters)

*menuidx*

메뉴의 위치를 나타내는 문자열. Remark 참고

*menustr*

메뉴에 들어가는 문자열.

*actionstr*

메뉴가 수행해야 할 Action ID.

*menutype*

메뉴의 유형. 다음의 값 중 하나이다.

값	설명	비고
0	추가되는 메뉴는 action을 수행한다.	
1	추가되는 메뉴는 서브메뉴를 가진다.	actionID 무시
2	추가되는 메뉴는 구분자(separator)이다.	text, actionID 무시

- 반환값(Return)

메뉴 추가에 성공하면 true를 실패하면 false를 반환한다.

- 설명(Remarks)

메뉴 바에 메뉴를 추가한다.

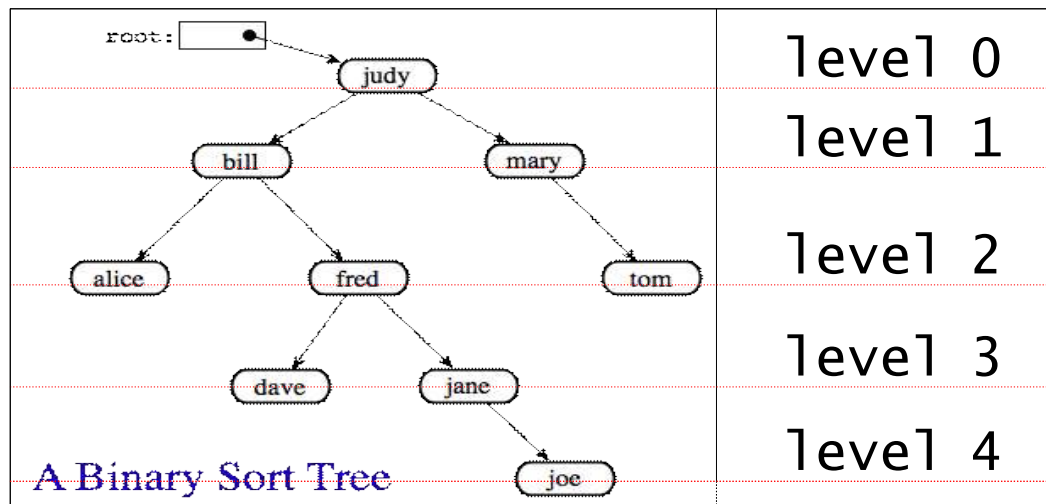
만약 한글 컨트롤에 메뉴 바가 존재하지 않으면 내부적으로 다음의 코드를 호출하여 메뉴 바를 생성한다.

```
"SetToolBar(-1, "TOOLBAR_MENU");
```

한글 컨트롤의 `InsertMenu()`와 MFC의 `InsertMenu()`는 기능 및 형태가 매우 유사하다. (실제로 내부에서는 MFC의 `InsertMenu()`를 호출한다) 유일한 차이점이라면 메뉴가 삽입될 위치를 찾아가는 방법이 다르다.

메뉴는 전형적인 **Tree**구조를 지니고 있어서 특정 위치를 탐색하기 위해서는 트리의 **Root**부터 찾아가야 한다. (MFC의 `CMenu` class가 이런 방식으로 메뉴를 탐색한다) 이때의 문제는 특정 위치까지 도달하기 위해서 중간 단계를 거쳐야 한다는 점이다. 이러한 불편함을 해소하고자 한글 컨트롤은 **Tree**의 각 **level**마다 위치를 표현하여 탐색하는 방법을 사용하였다.

다음의 그림을 보자.



위에서 **dave**는 **level 3**에 첫 번째에 위치하고 있다.

**Root**부터 **dave**까지의 경로를 탐색하면 **judy(Root) -> bill -> fred -> dave**의 경로를 얻을 수 있다. 위치를 **0**을 **base**로 해서 매기고, 같은 부모를 가진 형제노드일 경우에만 위치를 산정한다고 했을 경우에 각 경로의 위치를 표현하면 다음과 같다.

Judy	Bill	Fred	Dave
0	0	1	0

얻어진 위치를 ':'로 구분하여 표현하면 최종적으로 **dave**의 위치는 다음의 문자열로 표현된다.

dave의 위치: "0:0:1:0" (zero based number)

위의 예와 마찬가지로 메뉴도 **tree**구조를 가지므로, 다음 메뉴의 특정 항목을 위의 문자열로 표현할 수 있다.



그림 6. 한글의 메뉴창

“입력 - 한자 입력 - 한자 새김 입력”의 위치 : “3:3:3”

메뉴의 위치를 표현할 수 있으므로 이번에는 메뉴가 삽입되는 방법을 생각해보자. 이는 MFC의 **InsertMenu()**와 동일하다. **InsertMenu()**는 해당 위치에 새로운 메뉴를 추가한다. 만약 추가하려는 곳에 이미 메뉴가 존재하면 기존 메뉴들은 한 칸씩 뒤로 밀린다. 위의 메뉴에서 “한자 입력”의 서브메뉴 첫 번째 위치에 “한글로 바꾸기” 메뉴를 넣으려 한다면 다음의 위치 값을 가지고 **InsertMenu()**를 호출한다.

“3:3:0”

“한자로 바꾸기”, “한자 단어 등록”, “한자 부수/총획수”, “한자 새김 입력” 메뉴들은 한자리씩 뒤로 밀리게 된다.

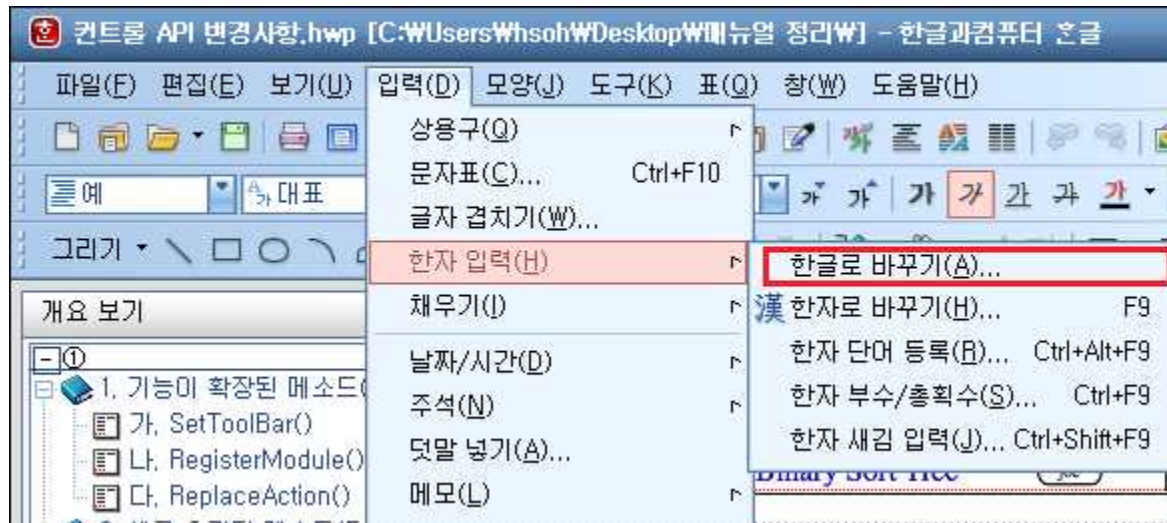


그림 7. “한글로 바꾸기” 메뉴가 추가된 모습

text로 들어가는 문자열에 '&'를 넣어주면 Alt키로 조합되는 단축키를 사용할 수 있다. 이를테면 다음과 같다.

“문서 불러오기(&O)”

한글 컨트롤은 사용자 Accelerator를 지원하지 않는다. '\t' 옵션은 단순히 내용만 우측으로 옮겨준다.

#### • 예제(Example)

C++

```
// 한글이 실행될 때
BOOL OnLoad(CDHwpCtrl& rHwpCtrl)
{
    CString str;
    str.Format("#0;1:myToolbar, %s, %s, %s, Separator, %s, %s",
        , FileOpen
        , FileSave
        , Print
        , g_arUserActions[1].szActionName
        , g_arUserActions[2].szActionName
    );
}
```

```

rHwpCtrl.SetToolBar(-2, CComVariant(str.AllocSysString()));

rHwpCtrl.SetToolBar(-1, CComVariant("#0;1:TOOLBAR_MENU"));
rHwpCtrl.InsertMenu((BSTR)"0:3", (BSTR)"내 메뉴", (BSTR)"", 1);
rHwpCtrl.InsertMenu((BSTR)"0:3:1", (BSTR)"파일열기(&O)...", (BSTR)"FileOpen", 0);
rHwpCtrl.InsertMenu((BSTR)"0:3:2", (BSTR)"파일저장(&V)...", (BSTR)"FileSave", 0);
rHwpCtrl.InsertMenu((BSTR)"0:3:3", (BSTR)"인쇄(&D)...", (BSTR)"Print",
                    MF_STRING|MF_MENUBREAK);
rHwpCtrl.InsertMenu((BSTR)"0:3:3", (BSTR)"", (BSTR)"", 2);
rHwpCtrl.InsertMenu((BSTR)"0:3:4", (BSTR)"UserAction1(&L)...",
                    (BSTR)USERACTION1, 0);
rHwpCtrl.InsertMenu((BSTR)"0:3:5", (BSTR)"UserAction2(&S)...",
                    (BSTR)USERACTION2, 0);

return TRUE;
}

```

- 참고 항목(See Also)

[RemoveMenu](#)

## 45) InsertPicture

현재 캐럿의 위치에 그림을 삽입한다.

- 구문(Syntax)

*C++*

```
LPDISPATCH InsertPicture(LPCTSTR path, VARIANT& embedded, VARIANT& sizeoption, VARIANT& reverse,
    VARIANT& watermark, VARIANT& effect, VARIANT& width, VARIANT& height)
```

*javascript*

```
Ctrl InsertPicture(string path, [boolean embedded], [number sizeoption], [boolean reverse], [boolean
    watermark], [number effect], [number width], [number height]) ver:0x05050102
```

```
boolean InsertPicture(BSTR path, [boolean embeded], [short sizeoption], [boolean reverse], [boolean
    watermark], [short effect], [long width], [long height]) ver:0x05050100
```

```
boolean InsertPicture(BSTR path, [byte embeded], [byte sizeoption], [byte reverse], [byte watermark],
    [byte effect], [unsigned int width], [unsigned int height]) ver:0x05000100
```

- 매개변수(Parameters)

*path*

삽입할 이미지 파일, URL 사용 가능 ([ver:0x05050111](#))

*embeded*

이미지 파일을 문서 내에 포함할지 여부 (True/False). 생략하면 True

*sizeoption*

삽입할 그림의 크기를 지정하는 옵션

ID	값	설명
realSize	0	이미지 원래의 크기로 삽입한다. <b>width</b> 와 <b>height</b> 를 지정할 필요 없다.
specificSize	1	<b>width</b> 와 <b>height</b> 에 지정한 크기로 그림을 삽입한다.
cellSize	2	현재 캐럿이 표의 셀 안에 있을 경우, 셀의 크기에 맞게 자동 조절하여 삽입한다. <b>width</b> 는 셀의 <b>width</b> 만큼, <b>height</b> 는 셀의 <b>height</b> 만큼 확대/축소된다. 캐럿이 셀 안에 있지 않으면 이미지의 원래 크기대로 삽입된다.
cellSizeWithSameRatio	3	현재 캐럿이 표의 셀 안에 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

ID	값	설명
RealPicture	0	실제 이미지 그대로
GrayScale	1	그레이 스케일
Blackwhite	2	흑백효과

### *width*

그림의 가로 크기 지정. 단위는 mm

### *height*

그림의 높이 크기 지정. 단위는 mm

- 반환값(Return)

생성된 컨트롤 object([ver:0x05050102](#))

성공하면 TRUE, 실패하면 FALSE([ver:0x05050100](#))

- 설명(Remarks)

- 예제(Example)

C++

```
CComPtr<IDispatch> pctrl;  
COleVariant oleDefault(0);  
  
pctrl = m_CHwpCtrl.InsertPicture(_T("c:\\windows\\부채.bmp"), COleVariant(TRUE), COleVariant(2),  
                                oleDefault, oleDefault, oleDefault, oleDefault, oleDefault);  
  
if (!pctrl)  
{  
    AfxMessageBox("그림을 삽입할 수 없습니다.");  
}
```

```

else
{
    //그림 삽입 후에 컨트롤의 속성을 추가적으로 더 변경할 수 있다.
    DHwpParameterSet dset;
    CComPtr<IDispatch> pset;
    DHwpCtrlCode dctrlcode;
    dctrlcode.AttachDispatch(pctrl);
    pset = m_CHwpCtrl.CreateSet(_T("ShapeObject"));
    dset.AttachDispatch(pset);
    dset.SetItem(_T("TreatAsChar"), (COleVariant)(short)FALSE); //글자처럼 취급
    dset.SetItem(_T("Textwrap"), (COleVariant)(short)2); //본문과의 배치 : 0:어울림, 1:자리차지,
                                                    //2:글 뒤로, 3:글 앞으로
    dctrlcode.SetProperties(pset); //주의 : Ctrl의 속성을 직접 수정했으므로 Undo history에 기록이 되지 않는다.
    dset.DetachDispatch();
    dctrlcode.DetachDispatch();
}

```

- 참고 항목(See Also)

[InsertBackgroundPicture](#)

## 46) IsCommandLock

해당 액션이 잠겨있는지 확인한다.

- 구문(Syntax)

*C++*

```
BOOL IsCommandLock(LPCTSTR actionID)
```

*javascript*

```
boolean IsCommandLock(string actionID)
```

- 매개변수(Parameters)

*actionID*

액션 ID. (ActionIDTable.Hwp 참조)

- 반환값(Return)

잠겨있으면 **true**, 잠겨있지 않으면 **false**를 반환한다.

- 설명(Remarks)

- 예제(Example)

- 참고 항목(See Also)

[LockCommand](#)

## 47) IsSpellCheckCompleted

맞춤법검사가 완료되었는지 상태 여부를 판단한다.

- 구문(Syntax)

*C++*

```
BOOL IsSpellCheckCompleted();
```

*Javascript*

```
boolean IsSpellCheckCompleted();
```

- 매개변수(Parameters)

없음

- 반환값(Return)

**Boolean** 타입의 값.

맞춤법검사가 완료된 상태이면 **True**를 반환하며, 아직 맞춤법검사를 완료하지 않은 상태이면 **False**를 반환한다.

맞춤법검사를 실행 중 취소한 경우에도 **False**를 반환한다.

- 설명(Remarks)

현재 한글컨트롤에 열려진 문서가 맞춤법검사를 완료한 상태인지 여부를 확인해주는 메소드이다.

문서를 처음 연 상태에서는 항상 **False**를 반환합니다. 또한, 맞춤법검사를 완료한 문서라 하여도 수정을 하거나, 저장을 한 후 호출하면 **False**를 반환한다. (Open, Save시 관련 상태플래그가 초기화되기 때문임)

- 예제(Example)

*javascript*

```
function OnSubmit()
{
    // 맞춤법 검사 완료 여부를 체크한다.
    if (!HwpControl.HwpCtrl.IsSpellCheckCompleted()) {

        // 맞춤법 액션을 실행하기 위해서 포커스를 컨트롤로 이동한다.
        HwpControl.HwpCtrl.focus();

        // 맞춤법 검사 액션을 실행한다.
        var act = HwpControl.HwpCtrl.CreateAction("SpellingCheck");
        var set = act.CreateSet();
        act.GetDefault(set);

        // 결과를 통보받을지 여부를 설정 (결과는 HwpCtrl_NotifyMessage 로 통보 됨)
        set.SetItem("NotifyResult", 1);
        // 맞춤법 대화상자가 표시되면서 맞춤법 검사가 시작되는 옵션
        set.SetItem("CheckStart", 1);

        act.Popupdialog(set);

        return;
    }

    _OnSubmit();
}
```

- 참고 항목(See Also)  
“SpellingCheck” Action

## 48) KeyIndicator

상태 바의 정보를 얻어온다.

- 구문(Syntax)

C++

```
BOOL KeyIndicator(long * secCnt, long * secno, long * prnpageno, long * colno, long * line, long * pos,  
short * over, BSTR * ctrlname)
```

- 매개변수(Parameters)

*secCnt*

총 구역

*secno*

현재 구역

*prnpageno*

쪽

*colno*

단

*line*

줄

*pos*

칸

*over*

삽입모드 (true: 수정, false: 삽입)

*ctrlname*

캐럿이 위치한 곳의 컨트롤 이름

- 반환값(Return)

성공하면 true, 실패하면 false

- 설명(Remarks)

컨트롤 바깥쪽에서 상태 바를 만들어서 상태 바에 표시할 정보들의 내용을 알아낼 때 유용하다.  
매개변수로 포인터를 사용하므로, 포인터를 사용할 수 없는 언어에서는 사용이 불가능 하다.

- 예제(Example)

C++

```
// idle time에 UI를 업데이트한다.
// 이함수는 App이 호출한다.
void CMainFrame::OnIdleUpdateUI()
{
    static int nTick = 0;
    if(!(nTick++ & 0x1F)) //너무 자주 업데이트 할 필요는 없다. 업데이트 주기:1/32
    {
        CHwpXCtrlFrameView *pView = (CHwpXCtrlFrameView*)GetActiveView();
        if (pView)
        {
            long seccnt; // 총 구역
            long secno; // 현재 구역
            long prnpageno; // 쪽
            long colno; //단
            long line; //줄
            long pos; //칸
            short over; //삽입/ 겹침
            CComBSTR ctrlname; // 컨트롤 종류
            CString strCtrlName;
            CString strPaneText;

            pView->m_CHwpCtrl.KeyIndicator(&seccnt, &secno, &prnpageno, &colno, &line, &pos,
            &over, &ctrlname);

            strCtrlName = ctrlname;
            strPaneText.Format(_T("   %5d쪽 %5d단"), prnpageno, colno);
            m_wndStatusBar.SetPaneText(0, strPaneText);
            strPaneText.Format(_T(" %5d줄 %5d칸"), line, pos);
            m_wndStatusBar.SetPaneText(1, strPaneText);
            strPaneText.Format(_T("%s"), strCtrlName);
            m_wndStatusBar.SetPaneText(2, strPaneText);
            strPaneText.Format(_T(" %d/%d구역"), secno, seccnt);
            m_wndStatusBar.SetPaneText(3, strPaneText);
        }
    }
}
```

```
        strPaneText.Format(_T("%s"), (over)?_T("수정"):_T("삽입"));
        m_wndStatusBar.SetPaneText(4, strPaneText);
    }
}
```

- 참고 항목(See Also)



## 49) LockCommand

특정 액션이 실행되지 않도록 잠근다.

- 구문(Syntax)

*C++*

```
void LockCommand(LPCTSTR actionID, BOOL lock)
```

*javascript*

```
LockCommand(string actionID, boolean lock)
```

- 매개변수(Parameters)

*actionID*

액션 ID. (ActionIDTable.Hwp 참조)

*lock*

true이면 액션의 실행을 잠그고, false이면 액션이 실행되도록 한다.

- 반환값(Return)

- 설명(Remarks)

Action을 잠근다. 잠긴 Action은 툴바 및 메뉴에서 비활성화 되며, 실행되지 않는다.

- 예제(Example)

*C++*

```
HwpCtrl.LockCommand("Print", TRUE);  
HwpCtrl.LockCommand("Undo", FALSE);
```

- 참고 항목(See Also)

[IsCommandLock](#)

## 50) LoadState

툴바와 보기 옵션을 불러온다.

- 구문(Syntax)

*C++*

```
BOOL LoadState(LPCTSTR FileName)
```

[ver:0x05050118](#)

*javascript*

```
boolean LoadState(string FileName)
```

- 매개변수(Parameters)

*FileName*

불러올 환경 파일

- 반환값(Return)

성공여부, 환경 파일이 잘못된 경우에는 **false**

- 설명(Remarks)

불러올 파일의 내용은 윈도우의 **profile(\*.ini)** 형식을 사용한다.

**filename**이 ;로 시작하면, **Profile**의 내용 자체가 들어있는 것이다.

**Profile**에 없는 항목에 대해서는 현재 상태를 유지하는 것을 원칙으로 한다.

- 예제(Example)

*javascript*

```
function SaveStateFile()
{
    pHwpCtrl.SaveState("PROFILE2.INI");
}
function ViewState()
{
    alert(pHwpCtrl.SaveState(""));
}
```

- 참고 항목(See Also)  
[SaveState](#)

## 51) MakeDocumentDiff

문서를 비교한다.

- 구문(Syntax)

*C++*

```
BOOL MakeDocumentDiff (LPCTSTR originDocument, LPCTSTR comparedDocument, LPCTSTR originOutput, LPCTSTR comparedOutput, BOOL viewingMemo);
```

*javascript*

```
boolean MakeDocumentDiff (string originDocument, string comparedDocument, string originOutput, string comparedOutput, boolean viewingMemo);
```

- 매개변수(Parameters)

*originDocument*

비교원본파일의 경로. 한/글문서(.hwp)여야 한다.

*compareDocument*

비교대상파일의 경로. 한/글문서(.hwp)여야 한다.

*originOutput*

생성할 비교결과물 파일의 경로.

원본파일을 기준으로 대상파일과 차이점을 메모 또는 교정부호로 표시한다.

HWPML파일(.hml)이어야 한다.

*compareOutput*

생성할 비교결과물 파일의 경로.

대상파일을 기준으로 원본파일과 차이점을 메모 또는 교정부호로 표시한다.

HWPML파일(.hml)이어야 한다.

### *viewingMemo*

비교자료를 메모로 출력할지, 교정부호로 출력할지 결정한다.

viewingMemo	설명
True	비교내용을 메모로 표시한다.
False	비교내용을 교정부호로 표시한다.

- 반환값(Return)

BOOL 타입의 값.

문서비교를 완료하여 비교결과물을 생성하면 **True**를, 실패한 경우에는 **False**를 반환한다.

- 설명(Remarks)

비교 원본문서와 대상문서를 파라미터로 받으며 비교결과를 **hwpm1** 문서로 반환한다.

문서비교는 단순히 문자열만 비교하는 것이 아니라 문서의 서식정보를 포함, 한/글에서 표현되는 모든 정보를 비교한다.

비교결과는 파라미터로 전달받은 파일로 각각 전달되며, 각 비교문서 대비 다른 정보들은 한/글의 메모기능을 이용해서 표현된다.

만약, **viewingMemo** 파라미터를 **False**로 전달하면, 메모가 아닌 교정부호로 표시하게 된다.

차이점을 상세히 보기 위해서는 메모로 결과물을 출력하는 것이 좋으며, 교정부호를 잘 아는 경우에는 교정부호가 사용된 깔끔한 결과물을 얻을 수 있다.

- 예제(Example)

*javascript*

```

function OnMakeDocumentDiff()
{
    var document1, document2, output1, output2;
    document1 = document.getElementById("File1").value;
    document2 = document.getElementById("File2").value;
    output1 = BasePath + "output1.html";
    output2 = BasePath + "output2.html";

    if (document1 == "" || document2 == "") {
        alert("파일을 선택하세요!");
        return;
    }

    if (!vHwpCtrl1.MakeDocumentDiff(document1, document2, output1, output2, 0)) {
        alert("문서비교 실패!");
        return;
    }

    vHwpCtrl1.Open(output1);
    vHwpCtrl2.Open(output2);

    vHwpCtrl1.style.width = "50%";
    vHwpCtrl2.style.display = "inline";
    document.getElementById("HwpCtrlRegion").style.display = "block";
}

```

- 참고 항목(See Also)

[MakeDocumentMergeDiff](#)

## 52) MakeDocumentMergeDiff

문서를 비교한다. (결과가 하나의 파일로 통합되어 나온다.)

- 구문(Syntax)

*C++*

```
BOOL MakeDocumentMergeDiff (LPCTSTR originDocument, LPCTSTR comparedDocument , LPCTSTR mergedOutput,
BOOL viewingMemo);
```

*Javascript*

```
boolean MakeDocumentDiff (string originDocument, string comparedDocument, string mergedOutput, boolean
viewingMemo);
```

- 매개변수(Parameters)

*originDocument*

- 비교원본파일의 경로. 한/글문서(.hwp)여야 한다.

*compareDocument*

- 비교대상파일의 경로. 한/글문서(.hwp)여야 한다.

*mergedOutput*

- 생성할 비교결과물 파일의 경로.
- 원본파일을 기준으로 대상파일과 차이점을 메모 또는 교정부호로 표시한다.
- HWPML파일(.hml)이어야 한다.

*viewingMemo*

- 비교자료를 메모로 출력할지, 교정부호로 출력할지 결정한다.

viewingMemo	설명
True	비교내용을 메모로 표시한다.
False	비교내용을 교정부호로 표시한다.

- 반환값(Return)

BOOL 타입의 값.

문서비교를 완료하여 비교결과물을 생성하면 **True**를, 실패한 경우에는 **False**를 반환합니다.

- 설명(Remarks)

비교 원본문서와 대상문서를 파라메터로 받으며 비교결과를 **hwpm1** 문서로 반환한다.

문서비교는 단순히 문자열만 비교하는 것이 아니라 문서의 서식정보를 포함, 한/글에서 표현되는 모든 정보를 비교한다.

비교결과는 파라메터로 전달받은 파일로 각각 전달되며, 각 비교문서 대비 다른 정보들은 한/글의 메모기능을 이용해서 표현된다.

만약, **viewingMemo** 파라메터를 **False**로 전달하면, 메모가 아닌 교정부호로 표시하게 된다.

차이점을 상세히 보기 위해서는 메모로 결과물을 출력하는 것이 좋으며, 교정부호를 잘 아는 경우에는 교정부호가 사용된 깔끔한 결과물을 얻을 수 있다.

**MakeDocumentDiff()**와 기능이 동일하며, 유일한 차이점은 결과가 하나의 파일로 통합되어 나온다는 점이다.



- 예제(Example)

*javascript*

```
function OnMakeDocumentMergeDiff() {
    var document1, document2, output;
    document1 = document.getElementById("File1").value;
    document2 = document.getElementById("File2").value;
    output = BasePath + "output.html";

    if (document1 == "" || document2 == "") {
        alert("파일을 선택하세요!");
        return;
    }

    if (!vHwpCtrl1.MakeDocumentMergeDiff(document1, document2, output, 0)) {
        alert("문서비교 실패!");
        return;
    }

    vHwpCtrl1.Open(output);

    vHwpCtrl1.style.width = "100%";
    document.getElementById("HwpCtrlRegion").style.display = "block";
}
```

- 참고 항목(See Also)

[MakeDocumentDiff](#)

### 53) MakeVersionDiffAll

(2005.12.19) 한글2005

현재문서의 버전비교문서를 만든다.

- 구문(Syntax)

C++

```
LPDISPATCH MakeVersionDiffAll(VARIANT& filepath)
```

[ver:0x06070115](#)

javascript

```
ParameterSet MakeVersionDiffAll(variant filepath)
```

- 매개변수(Parameters)

*filepath*

버전비교결과를 저장할 임시파일의 경로, 생략할 경우 시스템의 임시폴더에 임의로 생성한다.

- 반환값(Return)

VersionInfo ParameterSet을 반환한다.

※ VersionInfo: 버전정보

Item ID	Type	설명
SourcePath	string	버전 비교용 소스 패스
TargetPath	string	버전 비교용 타겟 패스
ItemStartIndex	byte	버전 비교를 보여줄 시작 히스토리 인덱스
ItemEndIndex	byte	버전 비교를 보여줄 마지막 히스토리 인덱스
ItemOverWrite	byte	히스토리 정보를 저장할 때 마지막 버전으로 덮어쓰는 플랙 (on : 1/off : 0)
ItemSaveDescription	byte	히스토리 정보를 저장할 때 설명을 입력하는 대화상자를 띄우는 플랙 (on : 1/off : 0) 지은이 정보가 직전 버전의 정보와 다를 경우 저장을 하지 않기 때문에 대화상자를 생략할 경우 지은이정보를 꼭 써야만 한다.
TempFilePath	ParameterArray	버전 비교용 임시파일 경로
ItemInfoIndex	unsigned long	버전 정보 얻어오기 및 삭제시 사용될 인덱스
SaveFilePath	string	버전저장 파일 경로 (OCX 컨트롤 용)

ItemInfoWriter	string	지은이 정보
ItemInfoDescription	string	해당 버전에 대한 설명
ItemInfoTimeHi	unsigned long	날짜 정보, FILETIME의 HIWORD
ItemInfoTimeLo	unsigned long	날짜 정보, FILETIME의 LOWORD
ItemInfoLock	byte	버전정보 잠그기 (on : 1/off :0)

- 설명 (Remarks)

- 예제 (Example)

*javascript*

```
var set = phwpCtrl.MakeVersionDiffAll("");
var array = set.Item("TempFilePath");
```

- 참고 항목 (See Also)

[GetVersionInfo](#), [GetVersionHistoryCount](#), [VersionSave](#), [VersionDelete](#)

## 54) ModifyFieldProperties

(2001.12.4)

지정한 필드의 속성을 바꾼다.

- 구문(Syntax)

*C++*

```
long ModifyFieldProperties(LPCTSTR field, long remove, long add) ver:0x05050101
```

*javascript*

```
number ModifyFieldProperties(string field, number remove, number add)
```

- 매개변수(Parameters)

*field*

속성을 바꿀 필드 이름의 리스트. 형식은 PutFieldText와 동일하다.

*remove*

제거될 속성

*add*

추가될 속성

- 반환값(Return)

음수가 반환되면 에러이다.

그 외 리턴 값은 Remark 참조.

- 설명(Remarks)

속성의 값은 아래와 같다.

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

return 값의 bit field는 다음과 같다.

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

`remove`와 `add`에 둘 다 0이 입력되면 현재 속성을 돌려준다.

반환값이 음수인지 확인하여 쉽게 에러임을 판별할 수 있으며 자세한 에러내용은 **bit mask**로 **and** 연산하여 알아 낼 수 있다.

반환값은 여러 가지 추가 정보가 같이 올 수 있으므로 반드시 **bit mask**를 사용하여 비교해야 한다.

- 예제(Example)

*C++*

```

long ret;
ret = m_CHwpCtrl.ModifyFieldProperties(_T("test"), 0, 0);
if (ret < 0 )
{
    CString strErrMsg = _T("test 필드의 정보를 알아오는 데 실패");
    if(ret & 0x40000000)
        strErrMsg += _T("\n지정한 필드를 찾을 수 없습니다.");
    AfxMessageBox(strErrMsg);
    return;
}
if (ret & 0x00000001)
{
    AfxMessageBox(_T("현재 양식모드에서 편집 가능"));
    ret = m_CHwpCtrl.ModifyFieldProperties(_T("test"), 1, 0);
}
else
{
    AfxMessageBox(_T("현재 양식모드에서 편집 불가능"));
    ret = m_CHwpCtrl.ModifyFieldProperties(_T("test"), 0, 1);
}
if (ret < 0 )
{
    AfxMessageBox(_T("필드 속성 변경 실패."));
}
if (ret & 0x00000001)
    AfxMessageBox(_T("양식모드에서 편집 가능하도록 변경되었습니다."));
else
    AfxMessageBox(_T("양식모드에서 편집 불가능하도록 변경되었습니다."));

```

- 참고 항목(See Also)

## 55) MovePos

캐럿의 위치를 옮긴다.

- 구문(Syntax)

*C++*

```
BOOL MovePos(VARIANT& moveID, VARIANT& para, VARIANT& pos)
```

*javascript*

```
boolean MovePos([number moveID], [number para], [number pos])
```

- 매개변수(Parameters)

*moveID*

다음과 같은 값을 지정할 수 있다. 생략하면 `moveCurList`가 지정된다.

ID	값	설명
<code>moveMain</code>	0	루트 리스트의 특정 위치.( <code>para pos</code> 로 위치 지정)
<code>moveCurList</code>	1	현재 리스트의 특정 위치.( <code>para pos</code> 로 위치 지정)
<code>moveTopOfFile</code>	2	문서의 시작으로 이동.
<code>moveBottomOfFile</code>	3	문서의 끝으로 이동.
<code>moveTopOfList</code>	4	현재 리스트의 시작으로 이동
<code>moveBottomOfList</code>	5	현재 리스트의 끝으로 이동
<code>moveStartOfPara</code>	6	현재 위치한 문단의 시작으로 이동
<code>moveEndOfPara</code>	7	현재 위치한 문단의 끝으로 이동
<code>moveStartOfword</code>	8	현재 위치한 단어의 시작으로 이동.(현재 리스트만을 대상으로 동작한다.)
<code>moveEndOfword</code>	9	현재 위치한 단어의 끝으로 이동.(현재 리스트만을 대상으로 동작한다.)
<code>moveNextPara</code>	10	다음 문단의 시작으로 이동.(현재 리스트만을 대상으로 동작한다.)
<code>movePrevPara</code>	11	앞 문단의 끝으로 이동.(현재 리스트만을 대상으로 동작한다.)
<code>moveNextPos</code>	12	한 글자 뒤로 이동.(서브 리스트를 옮겨 다닐 수 있다.)
<code>movePrevPos</code>	13	한 글자 앞으로 이동.(서브 리스트를 옮겨 다닐 수 있다.)
<code>moveNextPosEx</code>	14	한 글자 뒤로 이동.(서브 리스트를 옮겨 다닐 수 있다. 머리말/꼬리말, 각주/미주, 글상자 포함.)

ID	값	설명
movePrevPosEx	15	한 글자 앞으로 이동.(서브 리스트를 옮겨 다닐 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. 현재 루트 리스트에 위치해 있어 더 이상 상위 리스트가 없을 때는 위치 이동 없이 반환한다. 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. ) <a href="#">ver:0x0507013B, 0x06000105</a>
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
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 `moveCurList`가 지정되었을 때만 사용된다. `moveScrPos`가 지정되었을 때는 문단번호가 아닌 스크린 좌표로 해석된다.

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

*pos*

이동할 문단 중에서 문자의 위치.

`moveMain` 또는 `moveCurList`가 지정되었을 때만 사용된다.

- 반환값(Return)  
성공하면 `true`, 실패하면 `false`
- 설명(Remarks)  
`moveScrPos`가 지정한 경우에는 스크린 좌표로 마우스 커서의 (x,y)좌표를 그대로 넘겨주면 된다.  
`moveScanPos`는 문서를 검색하는 중 캐럿을 이동시키려 할 경우에만 사용이 가능하다.
- 예제(Example)
- 참고 항목(See Also)  
[InitScan](#), [GetText](#), [ReleaseScan](#), [GetTextBySet](#)

## 56) MoveToField

지정한 필드로 캐럿을 이동한다.

- 구문(Syntax)

*C++*

```
BOOL MoveToField(LPCTSTR field, VARIANT& text, VARIANT& start, VARIANT& select)
```

*Javascript*

```
boolean MoveToField(string field, [boolean text], [boolean start], [boolean select])
```

- 매개변수(Parameters)

*field*

필드이름. `GetFieldText()`/`PutFieldText()`와 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 **True**가 지정된다.

*start*

필드의 처음(**True**)으로 이동할지 끝(**False**)으로 이동할지 지정한다. **select**를 **True**로 지정하면 무시된다. 생략하면 **True**가 지정된다.

*select*

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

- 반환값(Return)

성공하면 **true**, 실패하면 **false**

- 설명(Remarks)

누름틀은 한글97의 메뉴 중, 입력 메뉴에 문서마당정보를 선택하면 누름틀을 만들 수 있다.

- 예제(Example)

- 참고 항목(See Also)

[GetFieldText](#), [PutFieldText](#), [RenameField](#), [MoveToFieldEx](#)

## 57) MoveToFieldEx

지정된 필드로 캐럿을 이동한 후 캐럿 위치로 화면을 이동한다.

- 구문(Syntax)

*C++*

```
BOOL MoveToFieldEx(LPCTSTR field, VARIANT& text, VARIANT& start, VARIANT& select)
```

*Javascript*

```
boolean MoveToFieldEx(string field, [boolean text], [boolean start], [boolean select])
```

- 매개변수(Parameters)

*field*

필드 이름. GetFieldText/PutFieldText와 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.

- 반환값(Return)

성공하면 true, 실패하면 false

- 설명(Remarks)

누름틀은 한글97메뉴 중 입력 메뉴에 문서마당 정보를 선택하면 누름틀을 만드실 수 있습니다.

- 예제(Example)

- 참고 항목(See Also)

[GetFieldText](#), [PutFieldText](#), [RenameField](#), [MoveToField](#)

## 58) open

문서를 연다.

- 구문(Syntax)

*C++*

```
BOOL Open(LPCTSTR path, VARIANT& format, VARIANT& arg)
```

*Javascript*

```
boolean Open(string path, [string format], [string arg])
```

- 매개변수(Parameters)

*path*

문서 파일의 경로, URL 사용 가능([ver:0x05050111](#))

*format*

문서 형식. 별도 설명 참조. 빈 문자열을 지정하면 자동으로 인식한다. 생략하면 빈 문자열이 지정된다.

*arg*

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

- 반환값(Return)

성공하면 **true**, 실패하면 **false**

- 설명(Remarks)

**format**에 지정할 수 있는 문서 형식은 현재 시스템에 설치된 문서 필터(\*.dft)의 종류에 따라 달라진다. 일반적으로 설치되는 형식들에는 다음과 같은 종류가 있다.

※ **format**은 아래에 쓰여 있는 대로 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	서식 있는 텍스트 문서
DBF	DBASE II/III 문서
HUNMIN	훈민정음 3.0/2000
MSWORD	마이크로소프트 워드 문서
DOCRTF	MS 워드 문서 (doc)
OOXML	MS 워드 문서 (docx)
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)

arg에 지정할 수 있는 옵션의 의미는 필터가 정의하기에 따라 다르지만, **syntax**는 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 파일포맷"

```
setcurdir(boolean, FALSE)
```

"HWP"

로드한 후 해당 파일이 존재하는 폴더로 현재 위치를 변경한다.  
**hyperlink** 정보가 상대적인 위치로 되어 있을 때 유용하다.

<code>lock (boolean, TRUE)</code>	로드한 후 해당 파일을 계속 오픈한 상태로 <code>lock</code> 을 걸지 여부
<code>notext (boolean, FALSE)</code>	텍스트 내용을 읽지 않고 헤더 정보만 읽을지 여부. (스타일 로드 등에 사용)
<code>template (boolean, FALSE)</code>	새로운 문서를 생성하기 위해 템플릿 파일을 오픈한다. 이 옵션이 주어진다면 <code>lock</code> 은 무조건 <code>FALSE</code> 로 처리된다.
<code>suspendpassword (boolean, FALSE)</code>	<code>TRUE</code> 로 지정하면 암호가 있는 파일일 경우 암호를 묻지 않고 무조건 읽기에 실패한 것으로 처리한다.
<code>forceopen (boolean, FALSE)</code>	<code>TRUE</code> 로 지정하면 읽기 전용으로 읽어야 하는 경우 대화상자를 띄우지 않는다.
<code>versionwarning (boolean, FALSE)</code>	<code>TRUE</code> 로 지정하면 문서가 상위버전일 경우 메시지 박스를 띄우게 된다.

#### "HTML"

<code>code(string, codepage)</code>	문서변환 시 사용되는 코드 페이지를 지정할 수 있으며 <code>code</code> 키가 존재할 경우 필터 사용 시 사용자 다이얼로그를 띄우지 않는다.
<code>textunit(boolean, pixel)</code>	<code>Export</code> 될 <code>Text</code> 의 크기의 단위 결정. <code>pixel</code> , <code>point</code> , <code>mili</code> 지정 가능.
<code>formatunit(boolean, pixel)</code>	<code>Export</code> 될 문서 포맷 관련 (마진, <code>Object</code> 크기 등) 단위 결정. <code>pixel</code> , <code>point</code> , <code>mili</code> 지정 가능

#### "DOCIMG"

<code>asing(boolean, FALSE)</code>	저장할 때 페이지를 <code>image</code> 로 저장
<code>ashtml(boolean, FALSE)</code>	저장할 때 페이지를 <code>html</code> 로 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>code(string, codepage)</code>	문서 변환 시 사용되는 코드 페이지를 지정할 수 있으며 <code>code</code> 키가 존재할 경우 필터 사용 시 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

#### • 예제 (Example)

*javascript*

```
HwpControl.HwpCtrl.Open("C:/GetFieldText.hwp", "HWP");
HwpControl.HwpCtrl.Open("C:/GetFieldText.hwp", "HWP", "versionwarning:true");
```

C++

```
m_CHwpCtrl.Open(path, COleVariant(""), COleVariant("suspendpassword:true;forceopen:true"));
```

- 참고 항목(See Also)  
[Save](#), [SaveAs](#), [Path](#)

## 59) PreviewCommand

(2005.3.3) 한글2004

미리보기 화면에서 view를 제어한다.

- 구문(Syntax)

C++

```
BOOL PreviewCommand(long previewmode)
```

[ver:0x0600010B, 0x0605010B](#)

javascript

```
boolean PreviewCommand(number previewmode)
```

- 매개변수(Parameters)

*previewmode*

미리보기 액션

ID	값	설명
PV_PRINTER_PAPER_LIST	1	공급 용지 바꾸기
PV_QUICK_PRINT	2	빠른 인쇄.
PV_MARGIN_BORDER	3	여백 보기
PV_PAGE_BORDER	4	편집 용지 보기
PV_ZOOM_LIST	5	확대 축소
PV_ONE_PAGE	6	한쪽 보기
PV_MULTI_PAGES	7	여러쪽 보기
PV_MIRROR_VIEW	8	맞쪽 보기
PV_EDIT_CURRENT_PAGE	9	현재 쪽 편집
PV_CLOSE	10	미리 보기 - 닫기
PV_ZOOM_PAGE_4	11	네 쪽 보기
PV_ZOOM_PAGE_3	12	세 쪽 보기
PV_ZOOM_PAGE_2	13	두 쪽 보기
PV_ZOOM_PAGE_1	14	쪽 맞춤
PV_ZOOM_100	15	미리 보기 - 100 %
PV_ZOOM_150	16	미리 보기 - 150 %
PV_ZOOM_200	17	미리 보기 - 200 %



PV_ZOOM_250	18	미리 보기 - 250 %
PV_ZOOM_300	19	미리 보기 - 300 %
PV_ZOOM_500	20	미리 보기 - 500 %
PV_LEFT	21	미리 보기 - 왼쪽 방향 키
PV_RIGHT	22	미리 보기 - 오른쪽 방향 키
PV_UP	23	미리 보기 - 위 방향 키
PV_DOWN	24	미리 보기 - 아래 방향 키
PV_PRIOR	25	미리 보기 - 이전 키 (Page UP)
PV_NEXT	26	미리 보기 - 다음 키 (Page Down)
PV_PRIOR_PAGE	27	이전 쪽
PV_NEXT_PAGE	28	다음 쪽
PV_FIRST_PAGE	29	처음 쪽
PV_LAST_PAGE	30	마지막 쪽

- 반환값(Return)

성공하면 **true**, 실패하면 **false**

- 설명(Remarks)

미리보기상태에서 **view**를 제어한다. 단, 미리보기상태가 아닐 경우에 사용하면 동작하지 않기 때문에 미리보기상태인지를 미리 확인한 후 사용한다.

- 예제(Example)

*javascript*

```
function PreviewClose{
    if (pHwpCtrl.IsPreviewMode)
        pHwpCtrl.PreviewCommand(10);    // 미리보기 닫기
}
```

- 참고 항목(See Also)

[IsPreviewMode](#)

## 60) PrintDocument

문서를 인쇄한다.

- 구문(Syntax)

*C++*

```
BOOL PrintDocument()
```

*javascript*

```
boolean PrintDocument()
```

- 매개변수(Parameters)

- 반환값(Return)

항상 true

- 설명(Remarks)

현재 문서를 인쇄하기 위해서 인쇄 대화상자를 띄운다.

- 예제(Example)

*javascript*

```
CHwpCtrl.PrintDocument();
```

- 참고 항목(See Also)

## 61) ProtectPrivateInfo

개인정보를 보호한다.

- 구문(Syntax)

C++

```
long ProtectPrivateInfo(LPCTSTR ProtectingChar, VARIANT PrivatePatternType);
```

javascript

```
boolean ProtectPrivateInfo(string ProtectingChar, [optional] int ProtectingType);
```

- 매개변수(Parameters)

*ProtectingChar*

보호문자. 개인정보는 해당문자로 가려진다.

*ProtectingType*

보호유형. 개인정보 유형마다 설정할 수 있는 값이 다르다.

0값은 기본 보호유형으로 모든 개인정보를 보호문자로 보호한다.

개인정보 유형에 따른 보호유형은 다음과 같다

개인정보 유형	보호 유형	보호유형 형태
전화번호 (0x0001)	0	*****
	1	NNN-***-NNNN
	2	NNN-NNN-****
주민등록번호 (0x0002)	0	*****
	1	NNNNNN-*****
	2	NNNNNN-N*****
	3	NNNNNN-N*****N
외국인등록번호 (0x0004)	0	*****
	1	NNNNNN-*****
	2	NNNNNN-N*****
	3	NNNNNN-N*****N
전자우편	0	*****

(0x0008)	1	***@TT.TT.TT
	2	T**@TT.TT.TT
	3	TT*@TT.TT.TT
	4	TTT@***.***.***
계좌번호 (0x0010)	0	*****
	1	NNN-**-****-NNN
	2	***-NN-NNNN-***
	3	NNN-**-NNNN-***
신용카드번호 (0x0020)	0	*****
	1	NNNN-****-****-NNNN
	2	NNNN-****-NNNN-****
	3	****-NNNN-NNNN-****
IP 주소 (0x0040)	0	*****
	1	NNN. *. *. N
	2	NNN. *. N. *
	3	***. N. N. *
생년월일 (0x0080)	0	*****
	1	NNNN-**-NN
	2	NNNN-**-**
	3	NNNN-NN-**

이 값은 생략 가능하며, 생략할 경우 0(기본 보호유형)값을 설정한 것과 동일하게 동작한다.  
사용자가 선택한 글자를 보호하는 경우(선택 글자 보호 기능) 이 값은 항상 0이다.

- 반환값(Return)

true / false

개인정보를 보호문자로 치환한 경우에 true를 반환한다.

개인정보를 보호하지 못할 경우 false를 반환한다.

문자열이 선택되지 않은 상태이거나, 개체가 선택된 상태에서는 실패한다.

또한, 보호유형이 잘못된 설정된 경우에도 실패한다.

마지막으로 보호암호가 설정되지 않은 경우에도 실패하게 된다.

- 설명(Remarks)

선택된 문자열을 보호한다.

한/글의 경우 “찾아서 보호”와 “선택 글자 보호”를 다른 기능으로 구현되었지만, API에서는 하나의 함수로 구현한다.

**ProtectingChar**는 개인정보를 보호할 때 개인정보 대신 입력될 문자이다. 어떤 문자라도 상관없다.

예) 개인정보를 숫자 ‘1’로 보호  
`ProtectPrivateInfo("1", 0);`

**ProtectingType**은 보호유형으로 개인정보의 전체를 보호할지 부분적으로 보호할지를 정해주는 값이다. 보호유형은 개인정보 유형에 따라 각각 다르다. (단, 기본 보호유형은 모든 보호유형에서 동일한 값(0)을 가진다)

예) 주민등록번호의 뒷부분만 보호  
`ProtectPrivateInfo("*", 1);`

**ProtectingType**은 생략이 가능한데 생략할 경우에 기본 보호유형이 설정된다.

**ProtectPrivateInfo()** 함수는 **FindPrivateInfo()** 함수로 선택된 문자열뿐만 아니라 사용자가 임의로 선택한 문자열도 보호할 수 있다. 예를 들어 “RepeatFind” 액션으로 찾아 선택한 문자열의 경우에도 보호가 가능하다.

예) “RepeatFind”로 문자열을 찾은 후 보호  
`var dact = hwpctrl.CreateAction("RepeatFind");  
var dset = dact.CreateSet();  
dact.GetDefault(dset);  
dset.SetItem("FindString", "한글과컴퓨터");  
dset.SetItem("IgnoreMessage", 1);  
dact.Execute(dset);  
if (hwpctrl.SelectionMode & 0x0F == 1) // 텍스트 선택상태일 경우  
    ProtectPrivateInfo("*");`

개인정보 보호가 성공할 경우에는 **true**를 반환한다.

문자열이 선택된 경우에는 대부분 성공하지만 이전에 보호암호를 설정하지 않은 경우에는 실패하게 된다. 또한, 문자열이 선택되지 않은 경우와 개인정보 유형과 맞지 않는 보호유형을 설정할 경우에도 실패하게 된다.

- 예 제 (Example)

- 참고 항목 (See Also)

[SetPrivateInfoPassword](#), [RegisterPrivateInfoPattern](#), [FindPrivateInfo](#), [IsPrivateInfoProtected](#)

## 62) PutFieldText

지정한 필드의 내용을 채운다.

- 구문(Syntax)

*C++*

```
void PutFieldText(LPCTSTR fieldlist, LPCTSTR textlist)
```

*javascript*

```
PutFieldText(string fieldlist, string textlist)
```

- 매개변수(Parameters)

*fieldlist*

내용을 채울 필드 이름의 리스트. 한 번에 여러 개의 필드를 지정할 수 있으며, 형식은 **GetFieldText**와 동일하다. 다만 필드 이름 뒤에 ‘# ’로 번호를 지정하지 않으면 해당 이름을 가진 모든 필드에 동일한 텍스트를 채워 넣는다. 즉, **PutFieldText**에서는 ‘필드이름’과 ‘필드이름0 ’의 의미가 다르다.

*textlist*

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

- 반환값(Return)

- 설명(Remarks)

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

**fieldlist**의 필드 개수와, **textlist**의 텍스트 개수는 동일해야 한다.

존재하지 않는 필드에 대해서는 무시한다.

- 예제(Example)

- 참고 항목(See Also)

[GetFieldText](#), [RenameField](#), [MoveToField](#), [MoveToFieldEx](#)

## 63) RegisterModule

(2003.12.23)

- 구문(Syntax)

C++

```
BOOL RegisterModule(LPCTSTR ModuleType, VARIANT& ModuleData)
```

[ver:0x05070130](#)

javascript

```
boolean RegisterModule(string ModuleType, variant ModuleData)
```

- 매개변수(Parameters)

*ModuleType*

모듈의 유형(Type). Remark 참고

ModuleType	설명	비고
FilePathCheckProc	파일경로 승인모듈을 callback Procedure 형태로 추가한다.	파일경로 승인모듈 Remark참고
FilePathCheckDLL	파일경로 승인모듈을 DLL 형태로 추가한다.	
FilePathCheckHandle	파일경로 승인모듈에 제공할 Instance Handle	
UserAction	UserAction DLL 모듈을 추가한다. ( <a href="#">ver:0x07050206</a> )	

*ModuleData*

모듈의 유형에 따른 Data 정보

ModuleType	ModuleData	설명	비고
FilePathCheckDLL	Registry에 등록된 DLL 모듈 ID	Registry에 등록된 DLL 모듈 ID	
UserAction			
FileCheckProc	Function Pointer	사용자가 정의한 callback 함수의 함수 포인터	
FilePathCheckHandle	Control ID	Control을 구분할 목적으로 사용되는 Control ID	구분할 필요가 없다면 생략 가능

- 반환값(Return)

추가모듈등록에 성공하면 true를 실패하면 false를 반환한다.



- 설명(Remarks)

흔글 컨트롤에 추가적인 모듈을 등록한다. 현재 등록할 수 있는 모듈의 종류는 크게 2가지이며, 향후 필요에 의해 더 늘어날 수 있다.

### 1. FilePathCheck 모듈

흔글 컨트롤은 사용자가 모르는 사이에 파일이 수정되거나 서버로 전송되는 것을 막기 위해 파일을 불러오거나 저장할 때 사용자로부터 승인을 받도록 되어있다. 그러나 이미 검증받은 웹페이지이거나, 이미 사용자의 파일 시스템에 대해 강력한 접근 권한을 갖는 응용프로그램의 경우에는 이러한 승인절차가 아무런 의미가 없으며 오히려 불편하기만 하다. 그러므로 흔글 컨트롤이 가지는 기본적인 확인절차 이외에 사용자가 직접 작성한 보안모듈을 사용할 수 있도록 지원하기 위해 **RegisterModule()**은 다음의 3가지 모드를 만들었다.

<b>FilePathCheckProc</b>	C++ 개발자를 위한 <b>Callback Function</b> 형태의 보안모듈
<b>FilePathCheckDLL</b>	C++ 이외의 개발환경을 위한 <b>DLL</b> 형태의 보안모듈
<b>FilePathCheckHandle</b>	보안모듈에서 컨트롤을 구분하기 위한 컨트롤 <b>ID (Instance ID)</b>

#### FilePathCheckProc

Windows 응용프로그램 상에서 흔글 컨트롤을 **Embedding**하는 경우에는 **DLL**을 호출하는 것보다 직접적인 **callback**함수를 호출하는 것이 빠르다. **FilePathCheckProc**는 그런 목적을 위해 추가된 **ModuleType**이다. **callback**함수의 형태는 다음과 같다.

```
BOOL CALLBACK modulename(HWND hwnd, Long nID, LPCTSTR szFilePath, LPCTSTR szSiteInfo);
```

Parameter	설명
hwnd	Window의 핸들. 주로 컨트롤을 <b>Embedding</b> 한 Windows의 핸들이 넘어온다.
nID	<b>RegisterModule(FilePathCheckHandle, nID)</b> 로 넘겨준 <b>nID</b> 값.
szFilePath	파일의 경로
szSiteInfo	컨트롤을 <b>Embedding</b> 한 Host의 정보.

**hwnd**의 값은 컨트롤을 사용하는 부모의 윈도우 핸들 값이 넘어온다. 만약 인터넷 익스플로러(IE)에서 컨트롤을 사용한다면 IE의 윈도우 핸들이 넘어오며, 응용프로그램인 경우에는 응용프로그램의 윈도우 핸들이 넘어온다.

**nID**는 **RegisterModule()**을 **FilePathCheckHandle**값과 같이 호출했을 때 **ModuleData**로 넘겨준 값이 넘어오게 된다. (일반적으로 컨트롤 ID) 자세한 내용은 **FilePathCheckHandle** 참고

**szFilePath**는 보안상 사용자의 승인이 필요한 파일의 경로가 넘어온다.

**szSiteInfo**는 컨트롤의 부모에 대한 정보로서, IE의 경우에는 URL, **PROCESS\_PATH**, **PROCESS**의 정보가 담긴 문자열이 넘어오며, 응용프로그램인 경우에는 **PROCESS\_PATH**와 **PROCESS**정보가 넘어오게 된다. 각 정보는 '\r\n'의 값으로 구분되어 넘어온다.

```
URL:http://www.haansoft.com/SWLab/Test_Control.html
PROCESS_PATH:C:\Program Files\Internet Explorer\iexplore.exe
PROCESS:iexplore.exe
```

그림 8. szSiteInfo로 넘어오는 문자열의 예

### FilePathCheckDLL

windows 응용프로그램이 아닌 경우에는 **callback**함수를 직접 정의할 수 없으므로, 사용자가 직접 작성한 보안모듈을 등록하여 승인절차를 변경해준다. 보안모듈 DLL의 작성법은 작성된 예제를 참고한다. (SWLAB의 보안승인모듈.zip)

작성된 보안모듈을 한글 컨트롤에 등록하기 위해서는 우선 보안모듈 DLL을 **Registry**에 등록해야 한다. **Registry**의 경로는 다음과 같다.

```
HKEY_CURRENT_USER/Software/HNC/HwpCtrl/Module
```

다음과 같은 이름과 값으로 보안모듈을 등록한다.

이름	FilePathCheck (무엇을 적어도 상관없으나 보안모듈임을 알 수 있도록 적는다)
값	보안모듈이 설치된 local 경로

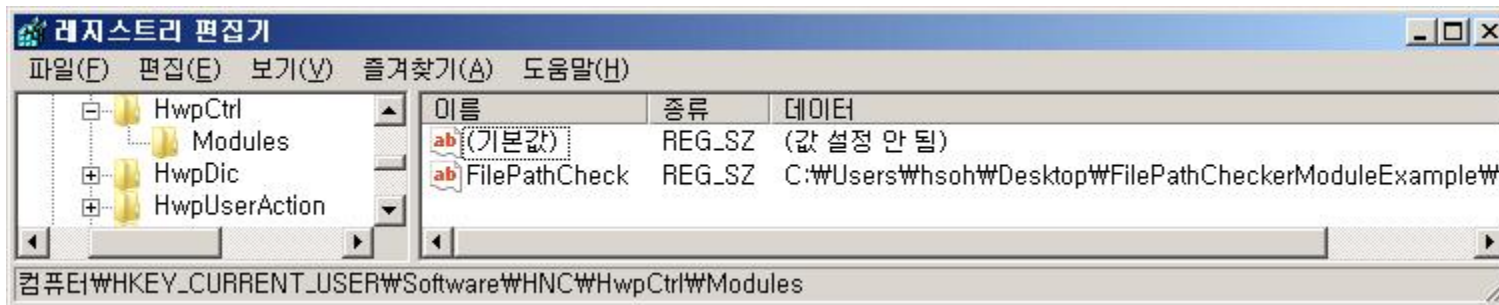


그림 9. Registry에 보안모듈이 등록된 모습

Registry에 등록된 이름을 가지고 **RegisterModule**을 호출하면 된다.

```
RegisterModule(FilePathCheckDLL, FilePathCheck);
```

### FilePathCheckHandle

보안모듈이 컨트롤을 구분할 수 있도록 컨트롤에 **ID**를 부여한다.

종종 컨트롤에 ID를 부여하여 구분해야 할 경우가 있다. 만약 다음과 같은 상황이 발생했다고 하자.

인터넷 익스플로러(IE)의 각 페이지(normal.html, critical.html)마다 한글 컨트롤을 사용하고 있다. normal.html은 중요한 문서를 다루지 않으므로 일반 보안승인절차가 필요하지만, critical.html은 매우 중요한 문서만 다루기 때문에 특별히 고안된 보안모듈을 사용할 예정이다.

위 경우에서 보안모듈은 각 페이지마다 다른 보안수준을 적용해야 한다. 물론 Page에 ID를 부여하여 구분할 수도 있겠지만, 만약 한 Page에 컨트롤이 2개가 사용된 경우라면 보안모듈은 각 컨트롤들을 구분해줘야 한다. 이런 문제를 해결하기 위해서 FilePathCheckHandle이 사용되었다. 컨트롤의 개수와 상관없이 동일한 보안수준을 적용해야 한다면 FilePathCheckHandle을 생략하고 사용해도 된다.

## 2. UserAction 모듈

[ver:0x07050206](#)

한글컨트롤 7.5.2.6([ver:0x07050206](#)) 버전 이 후 사용자 정의 Action(User defined action. 줄여서 UserAction)을 지원한다. UserAction을 동작시키는 방식은 2가지가 있는데 하나는 자동로딩방식이고 다른 하나는 수동로딩방식이다. 자동로딩방식은 한글컨트롤이 불러지는 순간에 자동으로 로딩 되는 방식을 말하며, 수동로딩방식은 RegisterModule()을 이용해서 특정시점에 DLL을 로딩하는 방식이다. (UserAction을 작성하는 방법은 컨트롤용 UserActionDLL 만들기.hwp를 참조한다.)

RegisterModule()을 이용해서 UserAction모듈(UserAction DLL)을 로딩하기 위해서는 우선 UserAction DLL을 Registry에 등록해야 한다. 등록해야 하는 경로는 FilePathCheckDLL과 동일하다. 다음과 같은 이름으로 보안모듈을 등록한다.

이름	ModuleName (무엇을 적어도 상관없으나 UserAction DLL임을 알 수 있도록 적는다)
값	UserAction DLL이 설치된 local 경로

FilePathCheckDLL과 마찬가지로 Registry에 등록된 이름을 가지고 RegisterModule을 호출한다.

```
RegisterModule(UserAction, ModuleName);
```

### • 예제(Example)

C++

보안승인함수(콜백함수) 작성

```
static BOOL __stdcall IsAccessiblePath(HWND hwnd, LONG nID, LPCTSTR szFilePath, LPCTSTR szSiteInfo)
{
    // 무조건 허용한다.
    return TRUE;
}
```

```

}

BOOL CTestHwpCtrlFilePathCheckDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    //... Class wizard가 생성해준 코드는 생략했습니다. .. /
    // TODO: Add extra initialization here

    if (m_cHwpCtrl.GetVersion() >= 0x05070130) {
        VARIANT vProc;
        AfxVariantInit(&vProc);

        vProc.vt = VT_BYREF;
        vProc.byref = IsAccessiblePath;
        m_cHwpCtrl.RegisterModule(_T("FilePathCheckProc"), vProc);
    }
    m_cHwpCtrl.Open(_T("C:\\spy1.txt"), (COleVariant)(LPCTSTR)_T(""), (COleVariant)(LPCTSTR)_T(""));

    return TRUE; // return TRUE unless you set the focus to a control
}

```

#### javascript

보안승인함수가 작성된 DLL을 RegisterModule로 등록

```

<HTML>
<HEAD>
<SCRIPT language="JScript">

var pHwpCtrl;

function OnStart()
{
    HwpControl.HwpCtrl.focus();
    pHwpCtrl = HwpControl.HwpCtrl;
    pHwpCtrl.RegisterModule("FilePathCheckDLL", "FilePathCheckerExample");
    pHwpCtrl.RegisterModule("FilePathCheckHandle", 132); // 132 = 임의 ID 사용하지 않는 경우 필요 없음.
}

```

- 참고 항목(See Also)

## 64) RegisterPrivateInfoPattern

개인정보의 패턴을 등록한다.

- 구문(Syntax)

*C++*

```
long RegisterPrivateInfoPattern(long PrivateType, LPCTSTR PrivatePattern);
```

*javascript*

```
boolean RegisterPrivateInfoPattern(int PrivateType, string PrivatePatterns);
```

- 매개변수(Parameters)

*PrivateType*

등록할 개인정보 유형. 다음의 값 중 하나다.

0x0001 : 전화번호

0x0002 : 주민등록번호

0x0004 : 외국인등록번호

0x0008 : 전자우편

0x0010 : 계좌번호

0x0020 : 신용카드번호

0x0040 : IP 주소

0x0080 : 생년월일

0x0100 : 주소

0x0200 : 사용자 정의

//0x0400 : 기타 - 기타는 등록할 수 없다

*PrivatePatterns*

등록할 개인정보 패턴. 예를 들면 이런 형태로 입력한다.

(예) 주민등록번호 - "NNNNNN-NNNNNNN"

한/글이 이미 정의한 패턴은 정의하면 안 된다.

함수를 여러 번 호출하는 것을 피하기 위해 패턴을 “;”기호로 구분  
반속해서 입력할 수 있도록 한다.

- 반환값(Return)

`true / false`

등록이 성공하였으면 `true`, 실패하였으면 `false`

- 설명(Remarks)

찾아낼 개인정보 패턴을 정의한다. (찾아서 보호 기능에서 사용됨)

한/글이 이미 정의하고 있는 패턴은 새로 등록할 수 없다. `PrivateType`으로 등록할 개인정보 유형을 선택한 뒤 `PrivatePattern`으로 패턴을 정의하는 식이다. 대략 다음의 예와 같이 사용할 수 있다.

예) 전화번호 패턴을 정의

```
RegisterPrivateInfoPattern(0x01, "NNNN-NNNN"); // 1580-xxxx 형태의 전화번호 유형을 등록함
```

한/글이 기본적으로 등록한 패턴은 새로 등록할 수 없다. 또한 이미 등록한 패턴 역시 등록이 실패하며 `false` 반환한다.

또한, 한글은 패턴을 등록할 수 있는 단어를 제한해두고 있다. 사용할 수 있는 문자는 'N', '-', '.', 빈 칸 등이며, IP 주소 유형의 경우에만 '@'를 허용한다. 잘못된 문자를 사용하여 패턴을 등록하면 위와 마찬가지로 `false`를 반환한다.

마지막으로 동일 유형에는 여러 패턴을 반복적으로 입력할 수 있다. 이때 함수를 한번만 호출할 수 있도록 ';' 구분자를 두어 반복해서 등록한다.

예) 여러 패턴을 한 번에 등록

```
// 상호용 전화번호와 국제 전화번호 패턴을 입력
```

```
int ret = RegisterPrivateInfoPattern(0x01, "NNNN-NNNN;NN-NN-NNNN-NNNN");
```

- 예제(Example)

- 참고 항목(See Also)

[SetPrivateInfoPassword](#), [ProtectPrivateInfo](#), [FindPrivateInfo](#), [IsPrivateInfoProtected](#)

## 65) ReleaseScan

InitScan()으로 설정된 초기화 정보를 해제한다.

- 구문(Syntax)

*C++*

```
void ReleaseScan()
```

*javascript*

```
ReleaseScan()
```

- 매개변수(Parameters)
- 반환값(Return)
- 설명(Remarks)

텍스트 검색작업이 끝나면 반드시 호출하여 설정된 정보를 해제해야 한다.
- 예제(Example)
- 참고 항목(See Also)

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



## 66) RemoveMenu

(2009.1.19) 한글2007

메뉴를 제거한다.

- 구문(Syntax)

C++

```
long RemoveMenu(LPCTSTR menuidx, long menutype)
```

[ver:0x07050206](#)

javascript

```
number RemoveMenu(string menuidx, number menutype)
```

- 매개변수(Parameters)

*menuidx*

메뉴의 위치를 나타내는 문자열. 또는 메뉴 action ID. InsertMenu의 Remark를 참조

*menutype*

menuidx의 type을 나타내는 값. 다음과 같다.

값	설명
0	menuidx는 action ID를 뜻한다.
1	menuidx는 메뉴의 위치를 나타내는 문자열이다.

- 반환값(Return)

성공하면 true(1), 실패하면 false(0)를 반환한다.

- 설명(Remarks)

사용자의 메뉴를 삭제한다.

상위 메뉴를 삭제하면 자동으로 하위 메뉴도 삭제된다. 모든 메뉴를 삭제하면 빈 툴바만 남게 된다.

menutype 매개변수는 어떤 방식으로 메뉴를 접근할 것인지 판단하는데 사용된다.

값이 0일 경우에는 action ID를 기준으로 메뉴를 제거하므로 위치를 모르지만 action ID를 알 경우에 사용하면 편하다. 동일한 action ID를 사용하는 메뉴가 둘 이상 존재하면 처음으로 발견되는 메뉴를 삭제한다. action ID를 기준으로 메뉴를 삭제하기 때문에 Popup메뉴나 Separator는 삭제할 수 없다.

`menutype`의 값이 1일 경우에는 메뉴의 위치를 기준으로 제거한다. 이때 `menuidx`는 메뉴의 위치를 나타내는 문자열이다. (위치를 표기하는 법은 `InsertMenu()`를 참고) 잘못된 위치를 참조한 경우에는 실패한다. 모든 메뉴(실행메뉴, `popup`메뉴, `separator`)에 접근 가능하므로 모든 메뉴를 삭제할 수 있다.

- 예제(Example)

*javascript*

```
pHwpCtrl.RemoveMenu("Print", 0);           // 인쇄메뉴를 삭제한다.  
pHwpCtrl.RemoveMenu("0:3", 0);           // 첫번째 popup메뉴의 third메뉴를 삭제한다.
```

- 참고 항목(See Also)

[InsertMenu](#)

## 67) RenameField

지정한 필드의 이름을 바꾼다.

- 구문(Syntax)

*C++*

```
void RenameField(LPCTSTR oldname, LPCTSTR newname)
```

*javascript*

```
RenameField(string oldname, string newname)
```

- 매개변수(Parameters)

*oldname*

이름을 바꿀 필드 이름의 리스트. 형식은 `PutFieldText`와 동일하다.

*newname*

새로운 필드 이름의 리스트. *oldname*과 동일한 개수의 필드 이름을 `0x02`로 구분하여 지정한다.

- 반환값(Return)

- 설명(Remarks)

예를 들어 *oldname*에 "title{{0}}\x2title{{1}}",

*newname*에 "tt1\x2tt2로 지정하면 첫 번째 `title`은 `tt1`로, 두 번째 `title`은 `tt2`로 변경된다.

*oldname*의 필드 개수와, *newname*의 필드 개수는 동일해야 한다. 존재하지 않는 필드에 대해서는 무시한다.

- 예제(Example)

- 참고 항목(See Also)

[GetFieldText](#), [PutFieldText](#), [MoveToField](#), [MoveToFieldEx](#)

## 68) ReplaceAction

특정 Action을 다른 Action으로 대체한다.

- 구문(Syntax)

C++

```
BOOL ReplaceAction(LPCTSTR OldActionID, LPCTSTR NewActionID) ver:0x05050118
```

javascript

```
boolean ReplaceAction(string OldActionID, string NewActionID)
```

- 매개변수(Parameters)

*OldActionID*

변경될 원본 Action ID. 한글 컨트롤에서 사용할 수 있는 Action ID는 ActionIDTable.hwp(별도문서)를 참고한다.

*NewActionID*

변경할 대체 Action ID. 기존의 Action ID와 UserAction ID([ver:0x07050206](#)) 모두 사용가능하다.

- 반환값(Return)

Action을 바꾸면 true를 바꾸지 못했다면 false를 반환한다.

- 설명(Remarks)

특정 Action을 다른 Action으로 대체한다.

이는 메뉴나 단축키로 호출되는 Action을 대체할 뿐, CreateAction()이나, Run() 등의 함수를 이용할 때에는 아무런 영향을 주지 않는다.

즉, ReplaceAction("Cut", "Copy")을 호출하여 "오려내기" Action을 "복사하기" Action으로 교체하면 Ctrl+X 단축키나 오려내기 메뉴/툴바 기능을 수행하더라도 복사하기 기능이 수행되지만, 코드 상에서 Run("Cut")을 실행하면 오려내기 Action이 실행된다.

```
HwpCtrl.ReplaceAction("Cut", "Copy");  
HwpCtrl.Run("Cut");
```

ReplaceAction()을 사용할 때에는 대체되는 Action들이 서로 Loop를 형성하지 않도록 조심해야 한다. 다음의 예를 보자.

```
HwpCtrl.ReplaceAction("MoveLeft", "MoveDown");
HwpCtrl.ReplaceAction("MoveDown", "MoveRight");
HwpCtrl.ReplaceAction("MoveRight", "MoveUp");
HwpCtrl.ReplaceAction("MoveUp", "MoveLeft");
```

위 예제는 키보드의 [←↓→↑]키를 한 칸씩 옆으로 옮겨 [↓→↑←]키로 재배치시켰다. 사용자는 [←]키를 누르면 [↓]키가 동작하는 것을 기대하였다.

하지만 실제로 위 예제는 동작하지 않으며 **stack overflow**를 만든다. **ReplaceAction()**은 단순히 <OldActionID,NewActionID>로 구성되는 테이블을 내부적으로 생성해준다. 단축키나 메뉴/툴바버튼이 눌러졌을 때 한글 컨트롤은 이 테이블을 참고해서 실행되어야 하는 **Action**을 선택해 수행한다. 문제는 컨트롤이 **Action**을 선택할 때 마지막으로 대체되는 **Action**을 찾을 때까지 반복해서 **Action**을 찾는다는 점이다.

MoveLeft ⇒ MoveDown ⇒ MoveRight ⇒ MoveUp ⇒ MoveLeft ⇒ & so on ...

▷ **Action**이 마지막으로 대체되는 **Action**을 무한정 반복해가며 찾는다.

대체된 **Action**을 원래의 **Action**으로 되돌리기 위해서는 **NewActionID**의 값을 원래의 **Action**으로 설정한 뒤 호출한다. 이를테면 이런 식이다.

```
ReplaceAction("Cut", "Cut")
```

한글컨트롤 7.5.2.6([ver:0x07050206](#)) 버전 이 후 사용자 정의 **Action**(User defined action. 줄여서 **UserAction**)을 지원한다.

**ReplaceAction()**의 이전버전에서는 기존 **Action**에 대해서만 **Action**을 대체하였다. 하지만 한글 컨트롤이 **UserAction**을 지원하게 됨으로서 **UserAction**에 대해서도 **Action**을 대체할 필요성이 대두되었다. 실제로 **ReplaceAction()** 함수는 기존의 **Action**을 대체하는 것보다 기존 **Action**을 **UserAction**으로 대체하는 것이 더더욱 유용하다. 기존의 **Action**을 사용자의 요구에 맞춰 확장시킬 수 있기 때문이다.

사용법은 동일하다. **NewActionID**에 **UserAction ID**를 넣어 사용하면 된다.

```
var myFileOpen = "{D90DD92B-415F-482b-BC26-5738E35D4769}";
HwpCtrl.ReplaceAction("FileOpen", myFileOpen);
```

- 예제(Example)  
*javascript*

```

var bFrameActionEnabled = false;
function FrameActionEnabled()
{
    if (bFrameActionEnabled)
    {
        pHwpCtrl.ReplaceAction("FileNew", "FileNew");
        pHwpCtrl.ReplaceAction("FileOpen", "FileOpen");
        pHwpCtrl.ReplaceAction("FileSave", "FileSave");
        pHwpCtrl.ReplaceAction("FileSaveAs", "FileSaveAs");
        pHwpCtrl.ReplaceAction("FindDlg", "FindDlg");
        pHwpCtrl.ReplaceAction("ReplaceDlg", "ReplaceDlg");
        bFrameActionEnabled = false;
        alert("Frame Action Disabled");
    }
    else
    {
        pHwpCtrl.ReplaceAction("FileNew", "HwpCtrlFileNew");
        pHwpCtrl.ReplaceAction("FileOpen", "HwpCtrlFileOpen");
        pHwpCtrl.ReplaceAction("FileSave", "HwpCtrlFileSave");
        pHwpCtrl.ReplaceAction("FileSaveAs", "HwpCtrlFileSaveAs");
        pHwpCtrl.ReplaceAction("FindDlg", "HwpCtrlFindDlg");
        pHwpCtrl.ReplaceAction("ReplaceDlg", "HwpCtrlReplaceDlg");
        bFrameActionEnabled = true;
        alert("Frame Action Enabled");
    }
}

```

- 참고 항목(See Also)

## 69) Run

액션을 실행한다.

- 구문(Syntax)

*C++*

```
void Run(LPCTSTR actionID)
```

*javascript*

```
Run(string actionID)
```

- 매개변수(Parameters)

*actionID*

액션 ID (ActionIDTable.hwp 참조)

- 반환값(Return)
- 설명(Remarks)
- 예제(Example)
- 참고 항목(See Also)  
[CreateAction](#)

## 70) RunScriptMacro

훈글문서 내에 존재하는 매크로를 실행한다.

- 구문(Syntax)

*C++*

```
BOOL RunScriptMacro(LPSTR FunctionName, long uMacroType, long uScriptType)
```

*javascript*

```
boolean RunScriptMacro(string FunctionName, number uMacroType, number uScriptType)
```

- 매개변수(Parameters)

*FunctionName*

실행할 매크로 함수이름

*uMacroType*

매크로의 유형. 밑의 값 중 하나이다.

ID	값	설명
HWP_GLOBAL_MACRO_TYPE	0	전역 매크로
HWP_DOCUMENT_MACRO_TYPE	1	문서에만 적용되는 매크로

*uScriptType*

스크립트의 유형. 현재는 javascript만을 유일하게 지원한다. reserved.

- 반환값(Return)

매크로의 실행여부와 상관없이 true를 반환한다.

- 설명(Remarks)

훈글은 매크로 기능을 지원하고 있다. 이렇게 작성된 매크로를 훈글 컨트롤에서도 실행시킬 수 있도록 지원하는 함수이다.

훈글 매크로 기능은 현재 javascript 언어만 지원하고 있으며 차후 visual basic의 언어도 추가적으로 지원할 예정이다.

- 예제(Example)

*javascript*

```
<SCRIPT language="javascript">
```



```

// Document Macro 실행 예제
//
// 밑의 코드를 실행하려면 한/글문서에 스크립트가 저장되어 있어야 한다.
// 스크립트를 저장하는 방법은 한/글문서의 메뉴 [보기-작업창-스크립트]로 실행되는 작업창에서
// 항목을 Document로 설정한 다음에 편집 창에 원하는 함수를 정의하면 된다. 여기서는 다음과 같은 함수를 정의하였다.
// function HelloWorld() {
//     alert("한/글 매크로의 세계에 오신 것을 환영합니다!");
// }

function OnRunScript(FunctionName) {
    vHwpCtrl.RunScriptMacro(FunctionName,1,0);
}

</SCRIPT>

```

- 참고 항목(See Also)

[GetScriptSource](#)

## 71) Save

현재 편집중인 문서를 저장한다.

- 구문(Syntax)

*C++*

```
BOOL Save(VARIANT& save_if_dirty)
```

*javascript*

```
boolean Save([boolean save_if_dirty])
```

- 매개변수(Parameters)

*save\_if\_dirty*

**true**를 지정하면 문서가 변경된 경우에만 저장한다. **false**를 지정하면 변경여부와 상관없이 무조건 저장한다. 생략하면 **true**가 지정된다.

- 반환값(Return)

성공하면 **true**, 실패하면 **false**

- 설명(Remarks)

문서의 경로가 지정되어있지 않으면 “새 이름으로 저장” 대화상자가 뜬다.

- 예제(Example)

- 참고 항목(See Also)

[Open](#), [SaveAs](#), [Path](#)

## 72) SaveAs

현재 편집중인 문서를 지정한 이름으로 저장한다.

- 구문(Syntax)

*C++*

```
BOOL SaveAs(LPCTSTR Path, VARIANT& format, VARIANT& arg)
```

*javascript*

```
boolean SaveAs(string Path, [string format], [string arg])
```

- 매개변수(Parameters)

*path*

문서 파일의 경로

*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.

- 반환값(Return)

성공하면 **true**, 실패하면 **false**

- 설명(Remarks)

**format**, **arg**의 일반적인 개념에 대해서는 **open()** 참조.

"Hwp" 포맷으로 파일 저장 시 **arg**에 지정할 수 있는 옵션은 다음과 같다.

이름	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	int	2	미리보기 이미지 (0=off, 1=BMP, 2=GIF)
prvtext	int	1	미리보기 텍스트 (0=off, 1=on)

autosave	boolean	false	자동저장 파일로 저장할 지 여부 (TRUE: 자동저장, FALSE: 지정 파일로 저장) <a href="#">ver:0x0507013c, 0x06000107</a>
export	void		다른 이름으로 저장하지만 열린 문서는 바꾸지 않는다. (lock:false와 함께 설정되어 있을 시 동작)

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

```
lock:TRUE;backup:FALSE;prvtext:1
```

- 예제 (Example)
- 참고 항목 (See Also)  
[Open](#), [Save](#), [Path](#)

### 73) SaveState

툴바와 보기 옵션을 저장한다.

- 구문(Syntax)

*C++*

```
CString SaveState(LPCTSTR FileName)
```

[ver:0x05050118](#)

*javascript*

```
string SaveState(string FileName)
```

- 매개변수(Parameters)

*FileName*

환경을 저장할 파일명

- 반환값(Return)

저장된 내용. 파일열기에 실패한 경우에는 빈 문자열이 반환된다.

- 설명(Remarks)

*FileName*이 세자 이상일 때만, 파일로 저장한다. 세자 미만일 경우에는 저장하지 않는 것으로 간주하고, 저장될 내용만 반환한다. 모든 항목을 저장하는 것을 원칙으로 한다.

- 예제(Example)

*javascript*

```
function LoadStateByFilePath()
{
    pHwpCtrl.LoadState(BasePath + "PROFILE2.INI");
}
function LoadStateByString()
{
    var state = ";;; [HWPCTRL PROFILE] ;;;\r\n" +
        "\r\n" +
        "[FRAME]\r\n"+
        "SHOW_STATUSBAR=1\r\n"+
```

```

        "\r\n"+
        "[VIEW]\r\n"+
        "OptionFlag=0\r\n"+
        "ZoomType=2\r\n"+
        "ZoomRatio=96\r\n"+
        "ZoomCntX=1\r\n"+
        "ZoomCntY=1\r\n"+
        "\r\n"+
        "[TOOLBAR]\r\n"+
        "REMOVE_NOT_DEFINED=1\r\n"+
        "SHOW_TOOLBAR=1\r\n"+
        "TOOLBAR_MENU=#0;2:TOOLBAR_MENU\r\n"+
        "TOOLBAR_STANDARD=#1;1:TOOLBAR_STANDARD\r\n"+
        "TOOLBAR_FORMAT=#2;1:TOOLBAR_FORMAT\r\n"+
        "TOOLBAR_DRAW=#3;1:TOOLBAR_DRAW\r\n"+
        "TOOLBAR_TABLE=#4;2:TOOLBAR_TABLE\r\n"+
        "TOOLBAR_IMAGE=#3;2:TOOLBAR_IMAGE\r\n"

        pHwpCtrl.LoadState(state);
    }

```

- 참고 항목(See Also)

[LoadState](#)

## 74) SelectText

블록을 설정한다.

- 구문(Syntax)

*C++*

```
BOOL SelectText(long spara, long spos, long epara, long epos)
```

*javascript*

```
boolean SelectText(number spara, number spos, number epara, number epos)
```

- 매개변수(Parameters)

*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.

- 반환값(Return)

성공하면 **true**, 실패하면 **false**

- 설명(Remarks)

**epos**가 가리키는 문자는 포함되지 않는다.

- 예제(Example)

- 참고 항목(See Also)

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

## 75) SetAutoSave

(2004.6.10) 한글2002

자동저장을 설정한다.

- 구문(Syntax)

C++

```
BOOL SetAutoSave(LPCTSTR FileName, long saveinterval) verL0x0507013C, 0x06000107
```

javascript

```
boolean SetAutoSave(string FileName, number saveinterval)
```

- 매개변수(Parameters)

*filename*

자동 저장될 파일의 확장자는 **sav**로 한다, 파일이름을 지정할 때는 “파일이름.sav” 로 하거나 "파일이름"만을 사용한다. 다른 확장자를 붙이거나 잘못된 파일이름을 사용할 경우에는 자동저장이 작동하지 않는다. 파일이름이 NULL일 경우에는 자동저장이 작동하지 않는다.

*saveinterval*

자동저장을 할 시간 간격을 설정한다. 단위는 ms이다. 0을 넣으면 자동 저장이 해제된다.

- 반환값(Return)

자동 저장된 파일을 열 경우에는 **true**를 그 외에는 **false**를 반환한다.

- 설명(Remarks)

자동저장은 자동저장 파일이름으로 설정한 파일로 저장되며 컨트롤의 타이머에 의해 주기적으로 저장을 한다. 자동저장 파일이 저장되는 경로는 컨트롤 내부에서 정한 경로에 저장하고 이를 수정할 수는 없다.

**SetAutoSave**는 파일 이름으로 지정된 자동저장 파일을 찾아서 있을 경우에는, 사용자에게 메시지 박스를 띄워 자동저장 파일을 열거나 열지 않기를 결정한다.

파일 이름은 확장자를 생략하고 파일이름만을 쓴다. 확장자를 붙여서 파일이름을 만들 경우에는 확장자를 모두 제거하고 파일 이름과 컨트롤 내부에서 정의한 확장자를 이용한다. 파일 이름으로 사용할 수 없는 파일이름이 들어올 경우에는 자동저장이 정상적으로 작동되지 않는다.

자동저장은 사용자의 승인 하에 컨트롤 내부에 지정된 폴더에 저장하고 열기 때문에 보안승인 모듈을 거치지 않는다.

특정 이름을 지정해준 파일만 복구하는 이유는 컨트롤은 기본적으로 창을 하나만 띄우는 구조이기 때문에 모든 자동 저장 파일을 복구 할



경우 G/W환경이 이상하게 동작하는 문제가 발생한다. 그러므로 자동 저장 파일 이름을 미리 알고 있어야만 복구가 가능하다.

### 주의사항

훅글 컨트롤에서 자동 저장 파일을 복구해주는 기능은 말 그대로 문서를 복구해주는 기능이지 작업 환경을 복구해주는 기능이 아니다. 예를 들어 기안문을 작성하다가 비정상적인 종료로 되었을 때, 다시 기안문을 열 경우에는 자동 저장된 기안문서가 오픈될 수 있지만, 다음 번 실행 때는 결재 문을 열 경우에 결재 환경에서 기안문서가 뜰 경우가 발생 할 수 있다. 이러한 작업 환경은 컨트롤에서 해결 할 수 없는 문제로 컨테이너에서 해결해야 하는 상황이다. 보통의 경우 각 작업 환경마다 자동 저장 파일 이름을 다르게 주어 어느 정도 해결 할 수 있지만, 이 역시도 컨테이너에서 관리를 해줘야 한다.

### 자동저장 및 복구 시나리오

저장	초기화	자동저장 단계	문서 작성 종료 (저장)	컨트롤 종료
Container (G/W)	자동저장 파일 이름 설정		문서작성이 종료되면 문서를 저장하고 서버로 전송	자동 저장 타이머 종료
HwpCtrl	자동저장 파일이름이 있을 경우에만 자동저장 상태로 인식	컨트롤 내부에서 타이머 동작하여 문서가 변경되었을 때만 지정된 자동저장 파일로 저장		자동 저장 타이머가 종료될 때만 자동저장 파일 삭제 (함수 자체가 on/off 스위치 역할)
복구	초기화	복구		
Container (G/W)	자동저장 파일 이름 설정	SetAutoSave가 TRUE를 리턴하면 자동저장 파일을 오픈한 경우이고 그 외에는 전부 FALSE이다. 이 값을 체크하여 사용자는 템플릿을 오픈할지 말지를 결정한다.		
HwpCtrl		SetAutoSave에 의해 자동저장 파일 복구. 메시지 박스를 띄워 사용자에게 물어본 후 자동저장 파일은 항상 삭제		

- 예제 (Example)

javascript

```
function OnStart()
{
    // 컨트롤이 초기화 될 때 자동저장을 설정할 수 있다.
    pHwpCtrl.SetAutoSave("기안", 5000);
}
function TestAutoSave()
{
    // 결재.asv로 자동 저장된 문서를 찾아서 있으면 사용자에게 물어본 후 열거나
    // 열지 않을 경우에는 지정된 파일을 오픈한다. 파일을 열지 않을 경우에는 자동저장 파
    // 일을 삭제한다. 자동저장 파일을 연 후 항상 다른 이름으로 저장 하도록 한다.
    if (!pHwpCtrl.SetAutoSave("결재", 5000)){
```

```

        pHwpCtrl.Open("c:\\works\\결재.hwp", "HWP", "");
    }
}
function OnQuit()
{
    // 문서 작성이 끝나면 서버로 전송 하고 자동 저장을 종료한다.
    // 자동저장이 종료되면 자동저장 파일은 삭제된다.
    pHwpCtrl.SetAutoSave("", 0);
}

```

- 참고 항목(See Also)

## 76) SetBarcodeImage

바코드 이미지를 삽입한다.

- 구문(Syntax)

*C++*

```
BOOL SetBarcodeImage(LPCTSTR lpImagePath, long pgno, long index, long x, long y, VARIANT width, VARIANT height)
```

*javascript*

```
boolean SetBarcodeImage(string lpImagePath, number pgno, number index, number x, number y, [number width], [number height])
```

- 매개변수(Padrameters)

*ImagePath*

바코드 이미지 경로

*pgno*

바코드 이미지를 삽입 할 페이지 번호(0부터 시작)

*index*

페이지에 삽입될 바코드 이미지의 번호(0부터 시작)

같은 번호를 지정하면 이전에 삽입된 바코드의 이미지가 수정된다.

바코드 이미지를 공백으로 지정하면 해당 번호의 바코드 이미지가 삭제된다.

*x*

바코드 이미지가 삽입 될 위치의 x좌표 (mm단위)

바코드 이미지의 좌측 상단 기준으로 페이지(종이 기준) 좌측 상단에서의 거리이다.

*y*

바코드 이미지가 삽입 될 위치의 y좌표 (mm단위)

바코드 이미지의 좌측 상단 기준으로 페이지(종이 기준) 좌측 상단에서의 거리이다.

*width*

바코드 이미지의 너비

생략하면 원본 이미지의 너비가 적용된다.

*height*

바코드 이미지의 높이

생략하면 원본 이미지의 높이가 적용된다.

- 반환값(Return)

성공하면 `true`, 실패하면 `false`

- 설명(Remarks)

- 예제(Example)

*C++*

```
pHwpCtrl.SetBarcodeImage(HwpControl.image.value, HwpControl.page.value, HwpControl.index.value,  
HwpControl.x.value, HwpControl.y.value, HwpControl.width.value, HwpControl.height.value);
```

- 참고 항목(See Also)

## 77) SetClientName

Client 프로그램의 특성에 따라 오동작 하는 것을 막기 위한 함수

- 구문(Syntax)

C++

```
void SetClientName(LPCTSTR szClient)
```

javascript

```
SetClientName(string szClient)
```

- 매개변수(Parameters)

*szClient*

Client를 지정하며 “;”을 구분자로 하여 동시에 여러 개를 지정할 수 있다.

PB7.0	POWERBUILDER 7.0인 경우 BSTR처리가 잘못되는 것을 방지
DEBUG	DEBUG Message를 출력하도록 한다. (ver:0x05050105)
RELEASE	DEBUG Message를 출력하지 않도록 한다. (ver:0x05050105)

- 반환값(Return)

- 설명(Remarks)

본래 Power Builder 7.0의 BSTR 처리에 문제가 있어서 만든 것이지만, 그 외 Client에 따라 다르게 동작해야할 필요가 있을 때 사용하게 되었다.

PB7.0 옵션의 경우 프로그램 실행도중에 Client가 바뀌는 일은 없다고 보므로 해제하는 방법은 없다.

- 예제(Example)

C++

```
void CHwpCtrlEx::InitControl()
{
#ifdef _DEBUG
    SetClientName(_T("DEBUG"));
#endif
}
```

```
_CreateStdToolBar();  
_LockCommands();  
}
```

- 참고 항목(See Also)

## 78) SetCurFieldName

현재 캐럿이 위치하는 곳의 필드이름을 설정한다.

- 구문(Syntax)

*C++*

```
BOOL SetCurFieldName(LPCTSTR fieldname, VARIANT& option, VARIANT& direction, VARIANT& memo)
```

*javascript*

```
boolean SetCurFieldName(string fieldname, [number option], [string direction], [string memo])
```

- 매개변수(Parameters)

*fieldname*

데이터 필드 이름.

*option*

다음과 같은 옵션을 지정할 수 있다. 0을 지정하면 모두 **off**이다. 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.

- 반환값(Return)

성공하면 **true**, 실패하면 **false**

- 설명(Remarks)

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

- 예제(Example)

- 참고 항목(See Also)  
[GetCurFieldName](#), [GetFieldList](#), [CurFieldState](#), [FieldExist](#)



## 79) SetFieldViewOption

양식모드와 읽기전용모드일 때 현재 열린 문서의 필드의 겉보기 속성(『』 표시)을 바꾼다.

- 구문(Syntax)

C++

```
long SetFieldViewOption(long option)
```

[ver:0x05050108](#)

javascript

```
number SetFieldViewOption(number option)
```

- 매개변수(Parameters)

*option*

겉보기 속성 bit

option	누름틀	개인정보/문서요약/날짜시간/파일경로	비고
1	『』을 표시하지 않음	『』을 표시하지 않음	
2	『』을 빨간색으로 표시	『』을 흰색으로 표시	설정하지 않았을 때 기본 값
3	『』을 흰색으로 표시	『』을 흰색으로 표시	

- 반환값(Return)

설정된 속성이 반환된다.

에러일 경우 0이 반환된다.

- 설명(Remarks)

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

- 예제(Example)

javascript

```
function OnStart()  
{  
    HwpControl.HwpCtrl.SetClientName("DEBUG"); //For debug message  
    HwpControl.HwpCtrl.Open("누름틀문서.hwp")  
    HwpControl.HwpCtrl.SetFieldViewOption(3);  
    HwpControl.HwpCtrl.EditMode = 2;  
}
```

- 참고 항목(See Also)

[EditMode](#)

## 80) SetFormObjectAttr

(2004.11.12) 한글2005

양식개체의 특성 값을 설정한다.

- 구문(Syntax)

*C++*

```
BOOL SetFormObjectAttr(LPCTSTR formname, LPCTSTR attrname, VARIANT& value)
```

[ver:0x0605010A](#)

*javascript*

```
boolean SetFormObjectAttr(string formname, string attrname, variant value)
```

- 매개변수(Parameters)

*formname*

양식개체 이름

*attrname*

특성 이름

*value*

특성 값

- 반환값(Return)

양식개체의 값을 설정하면 **true**, 해당 양식개체가 없거나 속성 값을 설정하지 못하면 **false**를 반환한다.

- 설명(Remarks)

GetFormObjectAttr함수의 Remark를 참조

- 예제(Example)

*javascript*

```
var length;  
  
length = 20;  
if (HwpControl.MyHwpCtrl.SetFormObjectAttr("EditBoxform", "MaxLength", length)  
{  
    alert ("에디트 상자의 최대 글자길이를 20으로 설정");  
}
```

```
}  
else  
{  
    alert ("SetFormObjectAttr err");  
}
```

- 참고 항목(See Also)  
[GetFormObjectAttr](#)

## 81) SetMessageBoxMode

한/글에서 쓰는 다양한 메시지박스가 뜨지 않고, 자동으로 특정 버튼을 클릭한 효과를 주기 위해 사용한다.

- 구문(Syntax)

*C++*

```
void SetMessageBoxMode(long mode)
```

*javascript*

```
SetMessageBoxMode(long mode)
```

- 매개변수(Parameters)

```
// 메시지 박스의 종류
#define MB_MASK 0x00FFFFFF
// 1. 확인(MB_OK) : IDOK(1)
#define MB_OK_IDOK 0x00000001
#define MB_OK_MASK 0x0000000F
// 2. 확인/취소(MB_OKCANCEL) : IDOK(1), IDCANCEL(2)
#define MB_OKCANCEL_IDOK 0x00000010
#define MB_OKCANCEL_IDCANCEL 0x00000020
#define MB_OKCANCEL_MASK 0x000000F0
// 3. 종료/재시도/무시(MB_ABORTRETRYIGNORE) : IDABORT(3), IDRETRY(4), IDIGNORE(5)
#define MB_ABORTRETRYIGNORE_IDABORT 0x00000100
#define MB_ABORTRETRYIGNORE_IDRETRY 0x00000200
#define MB_ABORTRETRYIGNORE_IDIGNORE 0x00000400
#define MB_ABORTRETRYIGNORE_MASK 0x00000F00
// 4. 예/아니오/취소(MB_YESNOCANCEL) : IDYES(6), IDNO(7), IDCANCEL(2)
#define MB_YESNOCANCEL_IDYES 0x00001000
#define MB_YESNOCANCEL_IDNO 0x00002000
#define MB_YESNOCANCEL_IDCANCEL 0x00004000
#define MB_YESNOCANCEL_MASK 0x0000F000
// 5. 예/아니오(MB_YESNO) : IDYES(6), IDNO(7)
```

```

#define MB_YESNO_IDYES                0x00010000
#define MB_YESNO_IDNO                0x00020000
#define MB_YESNO_MASK                0x000F0000
// 6. 재시도/취소(MB_RETRYCANCEL) : IDRETRY(4), IDCANCEL(2)
#define MB_RETRYCANCEL_IDRETRY        0x00100000
#define MB_RETRYCANCEL_IDCANCEL      0x00200000
#define MB_RETRYCANCEL_MASK          0x00F00000

```

- 반환값(Return)

`SetMessageBoxMode` 참고

- 설명(Remarks)

한/글에서 한/글이 로드된 후 `SetMessageBoxMode()`를 호출해서 사용한다.

`SetMessageBoxMode`는 하나의 파라미터를 받으며, 해당 파라미터는 자동으로 스킵할 버튼의 값으로 설정된다.

예를들어, `MB_OK_IDOK` (`0x00000001`)값을 주면, `MB_OK`형태의 메시지박스에서 `OK`버튼이 눌린 효과를 낸다.

- 예제(Example)

*C++*

```

// Empty example code block

```

- 참고 항목(See Also)

[`GetMessageBoxMode`](#)

## 82) SetPos

캐럿을 문서 내 특정 위치로 옮긴다.

- 구문(Syntax)

*C++*

```
BOOL SetPos(long list, long para, long pos)
```

*Javascript*

```
boolean SetPos(number list, number para, number pos)
```

- 매개변수(Parameters)

*list*

캐럿이 위치한 문서 내 list ID

*para*

캐럿이 위치한 문단 ID

*pos*

캐럿이 위치한 문단 내 글자 위치

- 반환값(Return)

성공하면 true, 실패하면 false

- 설명(Remarks)

지정된 위치로 캐럿을 옮겨준다.

- 예제(Example)

- 참고 항목(See Also)

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

### 83) SetPosBySet

캐럿을 `ParameterSet`으로 얻어지는 위치로 옮긴다.

- 구문(Syntax)

*C++*

```
BOOL SetPosBySet(LPDISPATCH pos)
```

[ver:0x05050104](#)

*javascript*

```
boolean SetPosBySet(ParameterSet pos)
```

- 매개변수(Parameters)

*Pos*

캐럿을 옮길 위치에 대한 `ParameterSet` 정보

- 반환값(Return)

성공하면 `true`, 실패하면 `false`

- 설명(Remarks)

- 예제(Example)

*javascript*

```
var lpp;
function GetPosTest()
{
    lpp = HwpControl.MyHwpCtrl.GetPosBySet();

    alert(lpp.Item("List") + " " + lpp.Item("Para") + " " + lpp.Item("Pos"));
}
function SetPosTest()
{
    if ( lpp != null)
    {
        if (!HwpControl.MyHwpCtrl.SetPosBySet(lpp))
```



```
        {  
            alert ("GetPos err");  
        }  
    }  
    else  
    {  
        alert("GetPos를 먼저 실행하세요.");  
    }  
}
```

- 참고 항목(See Also)

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

## 84) SetPrivateInfoPassword

개인정보보호를 위한 암호를 등록한다.

- 구문(Syntax)

*C++*

```
long SetPrivateInfoPassword(LPCTSTR password);
```

*javascript*

```
boolean SetPrivateInfoPassword(string password);
```

- 매개변수(Parameters)

*password*

새 암호

- 반환값(Return)

true / false

정상적으로 암호가 설정되면 **true**를 반환한다.

암호설정에 실패하면 **false**를 반환한다. **false**를 반환하는 경우는 다음과 같다

1. 암호의 길이가 너무 짧거나 너무 길 때 (영문: 5~44자, 한글: 3~22자)
2. 암호가 이미 설정되었음. 또는 암호가 이미 설정된 문서임

- 설명(Remarks)

개인정보 보호를 설정하기 위해서는 우선 개인정보 보호 암호를 먼저 설정해야 한다. 그러므로 개인정보 보호 함수를 실행하기 이전에 반드시 이 함수를 호출해야 한다.

- 예제(Example)

*javascript*

```
vHwpCtrl.SetPrivateInfoPassword("12345");
```

- 참고 항목(See Also)

[RegisterPrivateInfoPattern](#), [ProtectPrivateInfo](#), [FindPrivateInfo](#), [IsPrivateInfoProtected](#)

## 85) SetTextFile

문서를 문자열로 지정한다.

- 구문(Syntax)

*C++*

```
long SetTextFile(VARIANT& data, LPCTSTR format, LPCTSTR option) ver:0x05050109
```

*javascript*

```
number SetTextFile(string data, string format, string option)
```

- 매개변수(Parameters)

*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. <a href="#">ver:0x0505010B</a>
HWPML2X	HWP 형식과 호환	문서의 모든 정보를 유지
HTML	인터넷 문서 HTML 형식	훈글 고유의 서식은 손실된다.
UNICODE	유니코드 텍스트	서식정보가 없는 텍스트만 저장
TEXT	일반 텍스트	유니코드에만 있는 정보(한자, 고어, 특수문자 등)는 모두 손실된다.

*option*

option	설명	비고
insertfile	현재커서 이후에 지정된 파일 삽입	

- 반환값(Return)

성공이면 1을, 실패하면 0을 반환한다.

- 설명(Remarks)

이 함수는 JScript나 VBScript와 같이 직접적으로 local disk를 접근하기 힘든 언어를 위해 만들어졌으므로 disk를 접근할 수 있는 언어에서는 사용하지 않기를 권장합니다.

disk를 접근할 수 있다면, Open이나 Insert를 사용하십시오.

이 함수 역시 내부적으로는 Open이나 Insert를 호출하도록 되어있고 텍스트로 저장된 파일이 메모리에서 3~4번 복사되기 때문에 느리고, 메모리를 낭비합니다.

- 예제(Example)

*javascript*

```
function SetTextFile()
{
    HwpControl.HwpCtrl.SetTextFile("HWP", HwpControl.format.value, "");
}
```

- 참고 항목(See Also)

[GetTextFile](#)

## 86) SetToolBar

툴바를 등록/보임/숨김/삭제할 수 있다.

- 구문(Syntax)

C++

```
BOOL SetToolBar(long lToolBarID, VARIANT& varID)
```

Javascript

```
boolean SetToolBar(number lToolBarID, variant varID)
```

- 매개변수(Parameters)

**lToolBarID**

툴바의 ID. 0부터 1씩 증가한다. varID가 0일 경우에는 투바의 위치 Index. 마찬가지로 0부터 1씩 증가한다. 그 외 특별 용도로 음수를 사용한다. Remark 참고

lToolBarID	설명	비고
0 이상	툴바 ID or 투바 위치	varID가 0일 경우에만 투바 위치로 사용됨
-1	시스템에 미리 정의된 투바	
-2	사용자 정의 Action 투바	

**varID**

Action ID로 구성된 문자열. 투바 버튼을 나열한다. 또는 등록된 투바를 Show/Hide 하거나 삭제하는 flag 값을 가진다. 역시 Remark 참고

varID	설명	비고
Action List	Action List로 구성된 투바를 생성 및 등록	lToolBarID는 -2 또는 0 이상
Toolbar Name	시스템에 미리 정의된 투바를 생성 및 등록	lToolBarID는 -1
0	등록된 투바를 삭제	lToolBarID는 투바의 위치
1	등록된 투바를 보이게 함	lToolBarID는 투바의 ID
2	등록된 투바를 안보이게 함 (숨기기)	lToolBarID는 투바의 ID
3	등록된 투바를 삭제	lToolBarID는 투바의 ID

- 반환값(Return)

툴바 설정에 성공하면 true를, 실패하면 false를 반환한다.

- 설명(Remarks)

한글 컨트롤에서 툴바를 등록(생성)한다. 또는 이미 등록된 툴바를 보이게 하거나 숨기고, 등록된 툴바를 삭제한다.

### 1. 툴바 등록(생성)

한글 컨트롤에 툴바를 등록한다. 툴바의 버튼들은 각각 하나의 **Action**에 대응된다.

시스템에 미리 등록된 툴바를 사용할 수 있으며, 사용자가 툴바를 직접 구성할 수도 있다. 한글에서는 메뉴도 하나의 툴바로 인식하는데, 메뉴의 경우 사용자가 직접 구현하지 못하며 오직 시스템에 등록된 메뉴만 사용한다. (**InsertMenu()**를 사용하면 시스템에 등록된 메뉴 안에 새로운 메뉴를 추가할 수 있다. 자세한 내용은 **InsertMenu()**를 참고한다.)

한글컨트롤 7.5.2.6([ver:0x07050206](#)) 버전 이 후 사용자 정의 **Action**(User defined action. 줄여서 **UserAction**)을 지원하므로, **UserAction**을 사용하는 툴바를 등록할 수 있다.

매개변수로 주어지는 **lToolBarID**는 등록되는 툴바의 유형을 구분한다. 자세한 내용은 밑의 표를 참고한다.

값	버튼 정의 방식	사용되는 Action	varID의 내용
0 이상	사용자가 직접 정의	기존 액션	Action ID List
-1	시스템에 정의됨	기존 액션	미리 정의된 툴바 ID (*Remark 참조)
-2	사용자가 직접 정의	사용자 정의 액션(UserAction)	Action/UserAction ID List ( <a href="#">ver:0x07050206</a> )

표 1. lToolBarID값에 따른 툴바 유형

※ 기존의 사용방식(0)은 **UserAction ID**를 인식하지 못한다.

**lToolBarID**의 값이 0 이상일 경우에는 기존의 **Action**을 사용하는 툴바를 생성한다. 이 경우 **lToolBarID**는 명칭 그대로 툴바의 ID가 된다.

**lToolBarID**가 -1일 경우에는 시스템 정의 툴바를 생성한다. 시스템에서 정의하는 툴바는 다음과 같다. (**varID**로 넘겨진다)

Toolbar ID	의미
TOOLBAR_MENU	메뉴
TOOLBAR_STANDARD	기본 도구 상자
TOOLBAR_FORMAT	서식 도구 상자
TOOLBAR_DRAW	그리기 도구 상자

TOOLBAR_TABLE	표 도구 상자	
TOOLBAR_IMAGE	그림 도구 상자	
TOOLBAR_NUMBERBULLET	번호/글머리표 도구 상자	<a href="#">ver:0x0505011B</a>
TOOLBAR_HEADER_FOOTER	머리말/꼬리말 도구 상자	<a href="#">ver:0x0505011B</a>
TOOLBAR_MASTERPAGE	바탕쪽 도구 상자	<a href="#">ver:0x0505011B</a>
TOOLBAR_NOTE	각주/미주 도구 상자	<a href="#">ver:0x0505011B</a>
TOOLBAR_COMMENT	숨은 설명 도구 상자	<a href="#">ver:0x0505011B</a>

표 2. 시스템정의 툴바 표  
[ver:0x07050206](#)이후부터 `lToolBarID`가 -2일 경우에는 `UserAction`을 사용하는 툴바를 생성한다. `UserAction`은 DLL 모듈에서만 정의가 가능하므로 스크립트에서는 `UserAction`을 생성할 수 없다. 다만 정의된 `UserAction`의 ID를 알고 있으면 ID를 통해 툴바를 구성할 수 있다.

`varID`는 툴바의 버튼을 구성하는 문자열이다. `Action ID`와 `Separator` 그리고 그들을 구분해주는 구분자로 쉼표(,)로 구성된다.

"FileNew, FileOpen, Print, Separator, Cut, Copy, Paste, Separator, Undo, Redo ..."

위 문자열은 `Action list`로 구성되어 있다. 각 `Action`을 그룹으로 묶어주기 위해 사이사이에 `Separator`를 사용하고 있다.

`lToolBarID`의 값이 0 이상이면 `Action ID`는 시스템에 등록된 `Action ID`이며, -2이면 `Action ID`와 함께 `UserAction ID`를 포함한다. -2일 경우에만 `Action ID`와 `UserAction ID`를 혼용해서 사용할 수 있다.

`lToolBarID`가 -1인 경우에는 `Action List` 대신 위의 시스템정의 툴바 표의 `Toolbar ID`를 사용한다.

[ver:0x05050117](#)이후부터 툴바에 세부옵션을 주기위해 `Action List`의 첫 번째 요소를 이용한다.

줄 수 있는 세부옵션은 다음과 같다.

#	POSITION	;	SHOW	:	TOOLBAR NAME
#	위치 번호	구분자	1:Show/2:Hide	구분자	툴바 이름

그림 1. 세부옵션의 구조도

`POSITION`은 툴바 사이의 위치를 구분해주는 숫자이다. 앞이 #으로 시작하며 #0부터 시작해서 값이 1씩 증가된다. 숫자가 커질수록 툴바가 하단에 위치한다. 만약 동일한 숫자를 가지는 툴바가 존재하면 나중에 생성된 툴바가 기존툴바의 위에 위치하게 된다. 값이 -1인 경우에는 마지막에 툴바를 위치시킨다.

SHOW는 툴바가 보일지(show) 안보일지(hide) 결정하는 값이다. 1이면 툴바가 보이는 상태로 등록되며, 2이면 툴바가 안 보이는 상태로 등록된다.

TOOLBAR NAME은 툴바의 이름이다. 컨트롤의 [보기-도구상자]메뉴를 통해 등록된 툴바의 이름을 확인할 수 있다. 시스템정의 툴바를 사용하는 경우(ToolBarID가 -1)에는 이곳에 툴바 ID를 적어준다.

구분	값	설명
POSITION	0 이상	툴바의 위치 값. 숫자가 작을수록 상위에 위치
	-1	툴바의 마지막 위치에 등록 (Append)
SHOW	1	툴바가 보이는 상태로 등록
	2	툴바가 안 보이는 상태로 등록
TOOLBAR NAME	String	툴바의 이름 또는 시스템정의 툴바 ID

표 3. 세부옵션 요약

한글 컨트롤에서 사용할 수 있는 Action ID는 ActionIDTable.hwp(별도문서)를 참고한다.

UserAction ID는 DLL의 UserAction ID가 정의된 헤더파일(대부분 UserAction.h)을 참고한다.

다음은 완성된 varID 문자열 값이다.

"#0;1:TOOLBAR_MENU"	시스템 정의 툴바
"#1;2:MyTools,Print,Separator,Undo,Redo,Separator,Cut, Copy, Paste"	Action List
"{6EF744D6-157F-42a9-87BA-7EF608560CA0},{7411C643-057B-46c0-BF15-CE64C415B45C}"	UserAction List

그림 2. varID 문자열 예제

툴바를 등록한 후에는 반드시 ShowToolBar()를 호출해줘야 툴바가 컨트롤에 나타난다. 자세한 내용은 ShowToolBar()를 참고한다.

## 2. 툴바 제어(보이기/숨기기/삭제)

한글 컨트롤에 등록된 툴바에 대해 다음과 같은 동작을 수행할 수 있다.

Show, Hide, Delete

ToolBarID는 제어할 툴바의 ID를 varID는 수행되어야 할 동작을 나타낸다. varID에 따라 수행되는 동작은 다음과 같다.



varID	설명
1	툴바를 컨트롤에 나타낸다.
2	툴바를 컨트롤에서 숨긴다.
3	툴바를 컨트롤에서 삭제한다.

표 4. varID에 따른 툴바 제어

툴바 ID가 없이 등록된 툴바의 경우(lToolBarID가 -1, -2일 때)에는 Show/Hide가 불가능하다. 다만 툴바의 삭제는 가능하며 이때 lToolBarID는 툴바의 위치를 varID는 0을 넘겨주면 된다.

위 내용을 종합한 SetToolBar()의 툴바 제어표이다.

varID	lToolBarID	설명
1	Toolbar ID	툴바를 컨트롤에 나타낸다.
2	Toolbar ID	툴바를 컨트롤에서 숨긴다.
3	Toolbar ID	툴바를 컨트롤에서 삭제한다.
0	Toolbar Position	툴바를 컨트롤에서 삭제한다.

표 5. varID에 따른 툴바 제어(종합)

미리 정의된 툴바이름은 HTML의 <OBJECT><PARAM> TAG를 사용하여 툴바를 지정하는 데도 사용된다. HwpCtrl.ShowToolBar(TRUE) 대신, SHOW\_TOOLBAR속성을 사용할 수 있다.(예제 참고)

미리 정의된 툴바나 PARAM TAG를 이용한 툴바 지정기능은 [ver 0x050510E](#)이후부터 지원된다.

- 예제(Example)

*visual basic*

```
<SCRIPT language="javascript">
// 툴바 customize
function InitToolBarJS()
{
    HwpControl.HwpCtrl.SetToolBar(0, "FilePreview, Print, Separator, Undo, Redo, Separator, "
        + "Cut, Copy, Paste, Separator, ParaNumberBullet, MultiColumn, SpellingCheck, HwpDic, "
        + "Separator, PictureInsertDialog, MacroPlay1");
}
```

```

        HwpControl.HwpCtrl.SetToolBar(1, "DrawObjCreatorLine, DrawObjCreatorRectangle,
        + "DrawObjCreatorEllipse, DrawObjCreatorArc, DrawObjCreatorPolygon, DrawObjCreatorCurve, "
        + "DrawObjCreator, DrawObjTemplateLoad, Separator, ShapeObjSelect, ShapeObjGroup, "
        + "ShapeObjUngroup, Separator, ShapeObjBringToFront, ShapeObjSendToBack, ShapeObjDialog, "
        + "ShapeObjAttrDialog");

        HwpControl.HwpCtrl.SetToolBar(2, "StyleCombo, CharShapeLanguage, CharShapeTypeFace, "
        + "CharShapeHeight, CharShapeBold, CharShapeItalic, CharShapeUnderline, "
        + "ParagraphShapeAlignJustify, ParagraphShapeAlignLeft, ParagraphShapeAlignCenter, "
        + "ParagraphShapeAlignRight, Separator, ParaShapeLineSpacing, "
        + "ParagraphShapeDecreaseLeftMargin, ParagraphShapeIncreaseLeftMargin");

        HwpControl.HwpCtrl.ShowToolBar(true);
    }
    // 정의된 툴바 사용
    function InitToolBarJS()
    {
        HwpControl.HwpCtrl.SetToolBar(-1, "TOOLBAR_MENU");
        HwpControl.HwpCtrl.SetToolBar(-1, "TOOLBAR_STANDARD");
        HwpControl.HwpCtrl.SetToolBar(-1, "TOOLBAR_FORMAT");
        HwpControl.HwpCtrl.SetToolBar(-1, "TOOLBAR_DRAW");
        HwpControl.HwpCtrl.SetToolBar(-1, "TOOLBAR_TABLE");
        HwpControl.HwpCtrl.SetToolBar(-1, "TOOLBAR_IMAGE");
        HwpControl.HwpCtrl.SetToolBar(-1, "TOOLBAR_NUMBERBULLET");
        HwpControl.HwpCtrl.ShowToolBar(true);
    }
</SCRIPT>
...
<!-- HTML Code -->
<OBJECT id=HwpCtrl style="LEFT: 0px; TOP: 0px" height=600 width=800 align=center
        classid=CLSID:BD9C32DE-3155-4691-8972-097D53B10052>
</OBJECT>

```

HTML

```

<OBJECT id=HwpCtrl style="LEFT: 0px; TOP: 0px" height=600 width=800 align=center
        classid=CLSID:BD9C32DE-3155-4691-8972-097D53B10052>
    <PARAM NAME="TOOLBAR_MENU" VALUE="TRUE">
    <PARAM NAME="TOOLBAR_STANDARD" VALUE="TRUE">

```

```
<PARAM NAME="TOOLBAR_FORMAT" VALUE="TRUE">
<PARAM NAME="TOOLBAR_DRAW" VALUE="TRUE">
<PARAM NAME="TOOLBAR_TABLE" VALUE="TRUE">
<PARAM NAME="TOOLBAR_IMAGE" VALUE="TRUE">
<PARAM NAME="SHOW_TOOLBAR" VALUE="TRUE">
</OBJECT>
```

- 참고 항목 (See Also)

[ShowToolBar](#), [AutoShowHideToolBar](#), [CreateAction](#)

## 87) ShowStatusBar

상태 바를 표시하거나 감춘다.

- 구문(Syntax)

*C++*

```
BOOL ShowStatusBar(long Show)
```

[ver:0x05050115](#)

*Javascript*

```
boolean ShowStatusBar(number Show)
```

- 매개변수(Parameters)

*Show*

상태 바의 표시 여부 (0 or 1)

- 반환값(Return)

동작 여부 (true or false)

- 설명(Remarks)

아무 동작도 하지 않았으면 **false**를 반환한다. 즉, 현재 상태 바가 보이는 상태에서 **ShowStatusBar(1)**를 호출하거나, 현재 상태 바가 감춰진 상태에서 **ShowStatusBar(0)**를 호출하면 **false**를 반환한다.

- 예제(Example)

- 참고 항목(See Also)

[ShowToolBar](#)

## 88) ShowToolBar

툴바를 화면에 보임/숨긴다.

- 구문(Syntax)

*C++*

```
BOOL ShowToolBar(BOOL bShow)
```

*Javascript*

```
boolean ShowToolBar(boolean bShow)
```

- 매개변수(Parameters)

*bShow*

툴바화면 표시여부.

- 반환값(Return)

성공하면 **true**, 실패하면 **false**

- 설명(Remarks)

**ShowToolBar()**로 등록한 투바정보에 따라 화면표시(**Show**)하거나, 숨긴다(**Hide**).

※ 투바를 숨길 경우 **ShowToolBar()**로 등록한 메뉴 역시 투바의 일종이므로 같이 사라진다.

- 예제(Example)

- 참고 항목(See Also)

[SetToolBar](#)

## 89) versionDelete

(2005.12.19) 한글2005

현재문서의 버전정보 아이템을 지운다.

- 구문(Syntax)

*C++*

```
BOOL VersionDelete(long index)
```

[ver:0x06070115](#)

*javascript*

```
boolean VersionDelete(number index)
```

- 매개변수(Parameters)

*index*

삭제할 버전정보 아이템의 index(0~n)

-1을 입력할 경우 현재문서의 모든 버전정보 아이템을 삭제한다.

- 반환값(Return)

성공하면 true, 실패하면 false

- 설명(Remarks)

한번 삭제한 버전정보는 다시 되돌릴 수 없기 때문에 사용에 주의해야 한다.

- 예제(Example)

*javascript*

```
pHwpCtrl.VersionDelete(0);
```

- 참고 항목(See Also)

[GetVersionInfo](#), [MakeVersionDiffAll](#), [GetVersionHistoryCount](#), [VersionSave](#)

## 90) VersionSave

(2005.12.19) 한글2005

현재 편집중인 문서를 버전정보문서로 저장한다.

- 구문(Syntax)

*C++*

```
BOOL VersionSave(LPWSTR filepath, BOOL overwrite, BOOL infolock, VARIANT& writer, VARIANT& description)  
ver:0x06070115
```

*javascript*

```
boolean VersionSave(string filepath, boolean overwrite boolean infolock, variant writer, variant  
description)
```

- 매개변수(Parameters)

*filepath*

빈문서일 경우 저장될 문서의 경로

빈문서가 아닐 경우 현재 편집중인 문서에 그대로 저장한다.

*overwrite*

최근 버전정보 아이템을 덮어쓸지 여부를 결정한다.

최근 버전의 지은이와 저장할 지은이가 같은 경우에만 덮어쓰고, 다른 이름이면 새 버전으로 저장한다.

*infolock*

버전정보 내용 잠그기 여부를 결정한다.

*writer*

지은이 이름

*description*

설명

- 반환값(Return)

성공하면 true, 실패하면 false

- 설명(Remarks)

- 예제(Example)

*javascript*

```
pHwpCtrl.VersionSave("C:\\work\\ver.hwp", 0, 0, "abc", "new");  
pHwpCtrl.VersionSave("C:\\work\\ver.hwp", 1, 0, "xyz", "overwrite");
```

- 참고 항목(See Also)

[GetVersionInfo](#), [MakeVersionDiffAll](#), [GetVersionHistoryCount](#), [VersionDelete](#)



## 4. HwpCtrl Event

HwpCtrl은 사용자의 입력에 대한 이벤트를 감지하여 이벤트 핸들러에게 전달해준다. 다음에 소개되는 함수들은 일종의 이벤트 핸들러로 사용자에게 맞게 재정의 되어 사용된다.

## 가. 이벤트 (Event)

### 1) NotifyMessage

NotifyMessage 발생을 알린다.

- 구문(Syntax)

*C++*

```
void NotifyMessage(BSTR Msg, long WParam, long LParam)
```

*Javascript*

```
NotifyMessage(string Msg, number WParam, number LParam)
```

- 매개변수(Parameters)

*Msg*

메시지의 종류

*WParam*

정보

*LParam*

정보

- 반환값(Return)

- 설명(Remarks)

win32의 Message와 비슷한 개념으로 생각하면 된다.

Msg	WParam	LParam	참고
"FieldClickHereClicked"	LOWORD = left HIWORD = top	LOWORD = right HIWORD = bottom	필드 컨트롤이 눌렸을 때 필드의 영역을 화면 좌표계로 알려준다. 주의: <b>Event</b> 처리 함수가 리턴 된 후, 필드에 대해 나머지 작업(필드 속성 세팅등...)을 하기 때문에 필드를 지우면 안된다.
"HyperLinkClicked"	0: 문서내이동 1: 하이퍼링크		하이퍼링크가 클릭되었을 때 클릭된 URL정보를 알려준다. URL주소는 GetMessageSet()을 이용하여 얻어

---

2: mail
3: 외부 프로그램

온다.

- 예제(Example)

C++

```
OnNotifyMessageHwpctrl(LPCTSTR Msg, long WParam, long LParam)
{
    // TODO: Add your control notification handler code here
    if (!_tcscmp(Msg, _T("FieldClickHereClicked")))
    {
        CString strFieldName;
        strFieldName = m_CHwpCtrl.GetCurFieldName((COleVariant)(short)2);
        if (!strFieldName.IsEmpty()) // 이름이 있을 때만...
        {
            long left = LOWORD(WParam), top = HIWORD(WParam);
            long right = LOWORD(LParam), bottom= HIWORD(LParam);
            CString msg;
            msg.Format(_T("%s 누름틀이 눌림. left:%d, top:%d, right:%d, bottom:%d"), strFieldName,
            left, top, right, bottom);
            AfxMessageBox(msg);
        }
    }
}
```

- 참고 항목(See Also)

## 2) OnMouseDown

마우스의 왼쪽 버튼이 눌렸음을 알린다.

- 구문(Syntax)

*C++*

```
void OnMouseDown(long x, long y)
```

[ver:0x05050111](#)

*Javascript*

```
OnMouseDown(number x, number y)
```

- 매개변수(Parameters)

*x*

x 좌표

*y*

y 좌표

- 반환값(Return)

- 설명(Remarks)

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

- 예제(Example)

*HTML*

```
<HTML>
<HEAD>
<META content="text/html; charset=ks_c_5601-1987" http-equiv=Content-Type>
<SCRIPT language="JScript">
```

//\_GetBasePath()가 작동하지 않으면, OnStart()함수의 BasePath=\_GetBasePath();를 지우고, 이 예제 파일이 있는 곳을 지정해 준다.

```
var BasePath = "C:/Documents and Settings/guest/Examples/";
var MinVersion = 0x05050111;
var data;
```

```

var EventCapture = false;
var pHwpCtrl;

function OnStart()
{
    BasePath = _GetBasePath();
    pHwpCtrl = HwpControl.HwpCtrl;

    if(pHwpCtrl.getAttribute("version") == null)
    {
        alert("한글 2002 컨트롤이 설치되지 않았습니다.");
        return;
    }

    pHwpCtrl.SetClientName("DEBUG");
    _VerifyVersion();

    if(!pHwpCtrl.Open(BasePath + "MousePos.hwp"))
    {
        alert("Base Path가 잘못 지정된 것 같습니다. 소스에서 BasePath 를 수정하십시오");
    }
    ToggleCapture();
}

function _VerifyVersion()
{
    //버전 확인
    CurVersion = pHwpCtrl.Version;
    if(CurVersion < MinVersion)
    {
        alert("HwpCtrl의 버전이 낮아서 정상적으로 동작하지 않을 수 있습니다.\n"+
            "최신 버전으로 업데이트하기를 권장합니다.\n\n"+
            "현재 버전:" + CurVersion + "\n"+
            "권장 버전:" + MinVersion + " 이상"
        );
    }
}

```

```

function _GetBasePath()
{
    //BasePath를 구한다.
    var loc = unescape(document.location.href);
    var path;
    path = loc.replace(/.{2,}:\/{2,}/, ""); // file:/// 를 지워버린다.
    return path.substr(0,path.lastIndexOf("/") + 1); //BasePath 생성
}

function HwpCtrl_OnMouseDown(x,y)
{
    HwpControl.view_mousex_px.value = x;
    HwpControl.view_mousey_px.value = 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");
    HwpControl.page.value = page + 1;
    HwpControl.page_mousex_mm.value = Math.floor(pagex / 283.465); // 1mm == 283.465 HWPUNIT
    HwpControl.page_mousey_mm.value = Math.floor(pagey / 283.465);
    MousePosSet = pHwpCtrl.GetMousePos(1, 1);
    var paperx = MousePosSet.Item("X");
    var papery = MousePosSet.Item("Y");
    HwpControl.paper_mousex_mm.value = Math.floor(paperx / 283.465);
    HwpControl.paper_mousey_mm.value = Math.floor(papery / 283.465);
}

function ToggleCapture()
{
    EventCapture = !EventCapture;
    if(EventCapture)
        HwpControl.statustext.value = "마우스클릭감지중..."
    else
        HwpControl.statustext.value = "마우스클릭감지해제상태"
}

```

```

</SCRIPT>

<SCRIPT LANGUAGE=JScript FOR=HwpCtrl EVENT="OnMouseDown(x, y)" >
if(EventCapture)
    HwpCtrl_OnMouseDown(x,y);
</SCRIPT>

</HEAD>

<BODY bgColor=#77bbff onload = "OnStart()" >
<H3>MousePos</H3><hr>
<form name = "HwpControl" >
<pre>♣
<button onclick = ToggleCapture()>ToggleCapture</button>
<input type = text name=statustext READONLY size=30>
마우스 위치: Page:<input type=text name=page size=5>
V i e w X:<input type=text name=view_mousex_px size=5>
px Y:<input type=text name=view_mousey_px size=5>
px 쪽 기준 X:<input type=text name=page_mousex_mm size=5>
mm Y:<input type=text name=page_mousey_mm size=5>
mm 종이기준 X:<input type=text name=paper_mousex_mm size=5>
mm Y:<input type=text name=paper_mousey_mm size=5>mm
</pre>
<OBJECT id=HwpCtrl style="LEFT: 0px; TOP: 0px" height=600 width=800 align=center
    classid=CLSID:BD9C32DE-3155-4691-8972-097D53B10052>
    <PARAM NAME="TOOLBAR_MENU" VALUE="TRUE">
    <PARAM NAME="TOOLBAR_STANDARD" VALUE="TRUE">
    <PARAM NAME="TOOLBAR_FORMAT" VALUE="TRUE">
    <PARAM NAME="TOOLBAR_DRAW" VALUE="TRUE">
    <PARAM NAME="TOOLBAR_TABLE" VALUE="FALSE">
    <PARAM NAME="TOOLBAR_IMAGE" VALUE="FALSE">
    <PARAM NAME="TOOLBAR_HEADERFOOTER" VALUE="FALSE">
    <PARAM NAME="SHOW_TOOLBAR" VALUE="TRUE">
</OBJECT>
</form>
</BODY>
</HTML>

```

- 참고 항목(See Also)  
[GetMousePos](#)



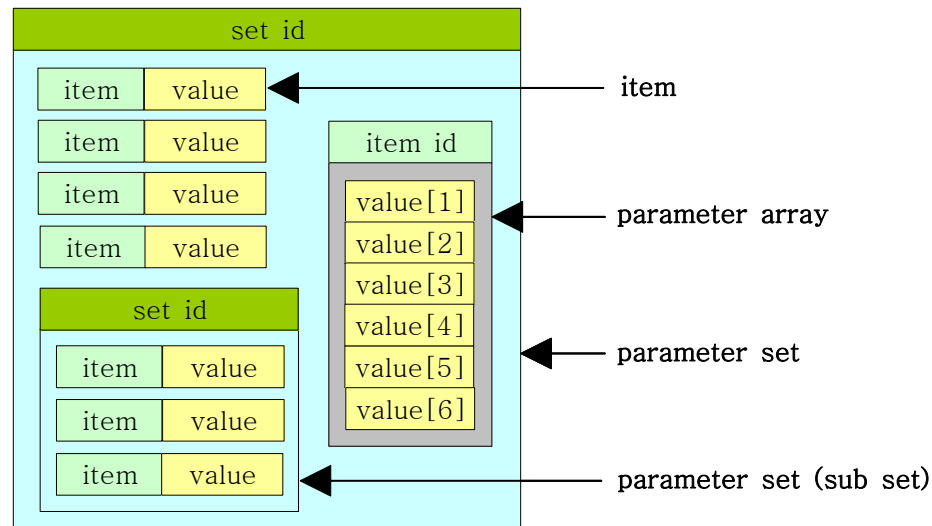
### 3) onScroll

- 구문(Syntax)
- 매개변수(Parameters)
- 반환값(Return)
- 설명(Remarks)
- 예제(Example)
- 참고 항목(See Also)

## 5. ParameterSet Object

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

어떤 액션을 수행하기 위해서는 액션이 필요로 하는 정보를 전달해주어야 하는 경우가 발생한다. 예를 들어, “표만들기”를 할 때 행과 열은 각각 얼마로 할 것인지 행의 디폴트 높이와 열의 디폴트 폭은 얼마로 할 것인지 등을 한글에게 알려주어야 한다. 이와 같이 두 객체 간에 전달해야 할 정보들의 묶음을 **parameter set**이라 하고, 묶음 내 각각의 정보에 해당하는 행의 수, 열의 수 등을 **set item**이라 한다. 각 **item**은 **item** 아이디와 **item** 값의 쌍으로 구성된다. **parameter set**의 **item**으로 **parameter set**이 올 수 있어 서브셋을 갖는 구조가 가능하며, 배열을 표현하기 위한 **parameter array**라는 구조가 **item**으로 올 수도 있다.



## 가. 속성 (Property)

### 1) count[읽기전용]

현재 존재하는 아이템의 개수를 나타낸다.

- 구문(Syntax)

*C++*

```
long GetCount()
```

*javascript*

```
(number) Count
```

- 설명(Remarks)
- 예제(Example)
- 참고 항목(See Also)

## 2) IsSet[읽기전용]

parameter set인지 여부를 나타낸다.

- 구문(Syntax)

*C++*

```
BOOL GetIsSet()
```

*javascript*

```
(boolean) IsSet
```

- 설명(Remarks)

임의의 IDispatch 포인터로부터 parameter set / parameter array를 구분하기 위해 동일한 이름의 property를 가지고 종류에 따라 True/False를 돌려준다.

ParameterSet은 True를 리턴한다.

- 예제(Example)
- 참고 항목(See Also)

### 3) SetID[읽기전용]

parameter set의 ID를 나타낸다.

- 구문(Syntax)

*C++*

```
CString Getsetid()
```

*javascript*

```
(string) SetID
```

- 설명(Remarks)

Set ID는 별도 문서를 참조한다.

- 예제(Example)

- 참고 항목(See Also)

## 나. 메소드 (Method)

### 1) clone

동일한 데이터를 가진 `parameter set`을 복사하여 반환하다.

- 구문 (Syntax)

*C++*

```
LPDISPATCH clone()
```

*Javascript*

```
ParameterSet clone()
```

- 매개변수 (Parameters)
- 반환값 (Return)  
ParameterSet Object.
- 설명 (Remarks)
- 예제 (Example)
- 참고 항목 (See Also)

## 2) CreateItemArray

아이템으로 parameter array 타입의 배열을 생성한다.

- 구문(Syntax)

*C++*

```
LPDISPATCH CreateItemArray(LPCTSTR itemid, long Count)
```

*Javascript*

```
ParameterArray CreateSet(string itemid, number Count)
```

- 매개변수(Parameters)

*itemid*

아이템 ID

*Count*

생성할 ParameterArray 배열의 초기 크기

- 반환값(Return)

생성된 ParameterArray Object

- 설명(Remarks)

동일한 ID를 가진 기존의 아이템은 삭제된다.

- 예제(Example)

*HTML*

```
function OnCreateTable()
{
    // 표만들기
    var act = pHwpCtrl.CreateAction("TableCreate");
    var set = act.CreateSet();
    var colset = set.CreateItemArray("ColWidth", 2);
    var rowset = set.CreateItemArray("RowHeight", 2);
    act.GetDefault(set);
    set.SetItem("Rows", 2);
}
```

```
set.SetItem("Cols", 2);

// 1 HWPUNIT = 1/7200 inch, 1 inch = 25.4 mm
colset.SetItem(0, 14400);
colset.SetItem(1, 7200);
rowset.SetItem(0, 3600);
rowset.SetItem(1, 7200);
act.Execute(set);
}
```

- 참고 항목(See Also)



### 3) CreateItemSet

아이템으로 `parameter set`을 생성한다.

- 구문(Syntax)

*C++*

```
LPDISPATCH CreateItemSet(LPCTSTR itemid, LPCTSTR setid)
```

*javascript*

```
ParameterSet CreateSet(string itemid, string setid)
```

- 매개변수(Parameters)

*itemid*

아이템 ID (별도 문서 참조)

*setid*

생성할 `ParameterSet` ID (별도 문서 참조)

- 반환값(Return)

생성된 `SubSet Object (ParameterSet)`

- 설명(Remarks)

`ParameterSet` 내부에 아이템으로 또 다른 `ParameterSet`을 가지는 서브셋의 개념이다.

- 예제(Example)

*Visual Basic*

```
Set act = ACTIVEXHWP.CreateAction("PageSetup")
Set set = act.CreateSet()
Set pset = set.CreateItemSet("PageDef", "PageDef")
act.GetDefault(set)
set.SetItem "ApplyTo", 3 ' 적용범위 : 문서전체

' 1mm = 283.465 HWPUNITS
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)
```

- 참고 항목(See Also)

#### 4) GetIntersection

두 **set**에 공통적으로 존재하고, 값도 동일한 아이템만으로 구성된 **intersection set**을 구한다.

- 구문(Syntax)

*C++*

```
void GetIntersection(LPDISPATCH srcset)
```

*Javascript*

```
GetIntersection(ParameterSet srcset)
```

- 매개변수(Parameters)

*srcset*

**intersection**에 사용할 원본 **set**

- 반환값(Return)

- 설명(Remarks)

현재 **set**과 **srcset**의 교집합(**intersection**)이 현재 **set**으로 저장된다.

- 예제(Example)

- 참고 항목(See Also)

## 5) IsEquivalent

두 **set**의 내용이 동일한 값을 가지고 있는지 검사한다.

- 구문(Syntax)

*C++*

```
BOOL IsEquivalent(LPDISPATCH srcset)
```

*javascript*

```
boolean IsEquivalent(ParameterSet srcset)
```

- 매개변수(Parameters)

*srcset*

현재 **set**과 비교할 원본 **set**

- 반환값(Return)

동일하면 **true**, 다르면 **false**를 반환한다.

- 설명(Remarks)

현재 **set**과 **srcset**을 비교한 결과를 반환한다.

- 예제(Example)

- 참고 항목(See Also)

## 6) Item

지정한 아이템의 값을 반환하다.

- 구문(Syntax)

*C++*

```
VARIANT Item(LPCTSTR itemid)
```

*javascript*

```
variant Item(string itemid)
```

- 매개변수(Parameters)

*itemid*

아이템 ID (별도 문서 참조)

- 반환값(Return)

아이템 값

- 설명(Remarks)

만약 지정한 아이템이 존재하지 않으면 아이템의 포맷에 따라 0 또는 빈 문자열을 반환한다.

- 예제(Example)

- 참고 항목(See Also)

## 7) ItemExist

지정한 아이템이 존재하는지 검사한다.

- 구문(Syntax)

*C++*

```
BOOL ItemExist(LPCTSTR itemid)
```

*javascript*

```
boolean ItemExist(string itemid)
```

- 매개변수(Parameters)

*itemid*

아이템 ID (별도 문서 참조)

- 반환값(Return)

존재하면 **true**, 존재하지 않으면 **false**를 반환한다.

- 설명(Remarks)

- 예제(Example)

- 참고 항목(See Also)

## 8) Merge

두 `set`의 내용을 병합한다.

- 구문(Syntax)

*C++*

```
void Merge(LPDISPATCH srcset)
```

*Javascript*

```
Merge(ParameterSet srcset)
```

- 매개변수(Parameters)

*srcset*

현재 `set`과 병합(Merge)될 원본 `set`

- 반환값(Return)

- 설명(Remarks)

결과는 현재 `set`에 존재하는 모든 `item` + `srcset`에만 존재하는 `item`이다.

- 예제(Example)

- 참고 항목(See Also)

## 9) RemoveAll

parameter set을 초기화한다.

- 구문(Syntax)

*C++*

```
void RemoveAll(LPCTSTR setid)
```

*javascript*

```
RemoveAll(string setid)
```

- 매개변수(Parameters)

*setid*

새로 적용할 set id (별도 문서 참조)

- 반환값(Return)

- 설명(Remarks)

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

- 예제(Example)

- 참고 항목(See Also)



## 10) RemoveItem

지정 한 아이템을 삭제한다.

- 구문(Syntax)

*C++*

```
void RemoveItem(LPCTSTR itemid)
```

*Javascript*

```
RemoveItem(string itemid)
```

- 매개변수(Parameters)

*itemid*

아이템 ID (별도 문서 참조)

- 반환값(Return)
- 설명(Remarks)
- 예제(Example)
- 참고 항목(See Also)

## 11) SetItem

지정한 아이템의 값을 설정한다.

- 구문(Syntax)

*C++*

```
void SetItem(LPCTSTR itemid, VARIANT& value)
```

*Javascript*

```
SetItem(string itemid, variant value)
```

- 매개변수(Parameters)

*itemid*

아이템 ID (별도 문서 참조)

*value*

설정할 값

- 반환값(Return)

- 설명(Remarks)

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

- 예제(Example)

- 참고 항목(See Also)

## 6. ParameterArray

`parameter set`의 아이템으로 배열을 표현하는 데 사용된다. 일반적인 `method`의 독립적인 인수로 사용되는 일은 없고, `parameter set`의 아이템으로만 사용된다.

## 가. 속성 (Properties)

### 1) Count

배열의 크기를 나타낸다.

- 구문(Syntax)

*C++*

```
long GetCount()  
void SetCount(long value)
```

*javascript*

```
(number) Count
```

- 설명(Remarks)

배열의 크기는 runtime에 변경이 가능하다.

- 예제(Example)

- 참고 항목(See Also)

## 2) IsSet[읽기전용]

parameter set인지 여부를 나타낸다.

- 구문(Syntax)

*C++*

```
BOOL GetIsSet()
```

*javascript*

```
(boolean) IsSet
```

- 설명(Remarks)

임의의 IDispatch 포인터로부터 parameter set / parameter array를 구분하기 위해 동일한 이름의 property를 가지고 종류에 따라 True/False를 돌려준다. ParameterArray는 False를 반환하다.

- 예제(Example)

- 참고 항목(See Also)

## 나. 메소드 (Method)

### 1) clone

동일한 크기와 데이터를 갖는 `ParameterArray` 개체를 복사하여 돌려준다.

- 구문(Syntax)

*C++*

```
LPDISPATCH clone()
```

*Javascript*

```
ParameterArray clone()
```

- 매개변수(Parameters)
- 반환값(Return)
- 설명(Remarks)
- 예제(Example)
- 참고 항목(See Also)

## 2) Copy

배열을 복사한다.

- 구문(Syntax)

*C++*

```
void Copy(LPDISPATCH srcarray)
```

*javascript*

```
Copy(ParameterArray srcarray)
```

- 매개변수(Parameters)

*srcarray*

복사할 ParameterArray Object

- 반환값(Return)

- 설명(Remarks)

*srcarray*의 내용이 그대로 현재 *ParameterArray*로 복사된다.

- 예제(Example)

- 참고 항목(See Also)

### 3) Item

지정한 원소의 값을 반환하다.

- 구문(Syntax)

*C++*

```
VARIANT Item(long index)
```

*javascript*

```
variant Item(number index)
```

- 매개변수(Parameters)

*index*

원소의 인덱스. 0부터 시작된다.

- 반환값(Return)

*index*의 위치에 존재하는 *item*이 반환된다.

- 설명(Remarks)

- 예제(Example)

- 참고 항목(See Also)



#### 4) SetItem

지정한 원소의 값을 설정한다.

- 구문(Syntax)

*C++*

```
void SetItem(long index, VARIANT& value)
```

*javascript*

```
SetItem(number index, variant value)
```

- 매개변수(Parameters)

*index*

원소의 인덱스. 0부터 시작

*value*

원소의 값.

- 반환값(Return)

해당 Action과 연관되는 ParmeterSet Object.

- 설명(Remarks)

parameter set을 사용하지 않는 액션의 경우 NULL이 리턴 된다.

이 method는 다음과 같이 수행한 것과 동일하다.

```
Set param = HwpCtrl.CreateSet(action.SetID)
```

- 예제(Example)

- 참고 항목(See Also)

## 7. HwpMenu

HwpMenu 는 HwpCtrl에서 컨텍스트 메뉴를 다루는 객체.

CLSID : 982B2F01-9EAB-4783-AE1D-EA7BAFEFCE2F

## 가. 메소드 (Method)

### 1) AppendMenu

메뉴를 덧붙임

- 구문(Syntax)

*C++*

```
BOOL AppendMenu(LPCTSTR strActionID, LPCTSTR strMenuName);
```

*javascript*

```
boolean AppendMenu(string strActionID, string strMenuName);
```

- 매개변수(Parameters)

*strMenuName*

AddOnAction일 경우 화면에 보일 String값으로 일반Action에서는 Action에 정의된 스트링으로 표시되기 때문에 무시됨.

*strActionID*

등록된 Action 명

- 반환값(Return)

True이면 정상동작, False이면 정상동작 아님.

- 설명(Remarks)

한/글컨트롤의 메뉴에 메뉴를 덧붙임

- 예제(Example)

*C++*

```
hwpMenu.AppendMenu("CharShapeBold", ""); // 진하게  
hwpMenu.AppendMenu("{D29E3545-69E9-4104-B74C-EE0E54C321F9}", "사용자 액션1") // AddOnAction일 경우 사용 예
```

- 참고 항목(See Also)

[HwpMenu](#), [RemoveMenu](#)

## 2) RemoveMenu

메뉴를 제거함

- 구문(Syntax)

*C++*

```
BOOL RemoveMenu(LPCTSTR strActionID);
```

*Javascript*

```
boolean RemoveMenu(string strActionID);
```

- 매개변수(Parameters)

*strActionID*

삭제될 action 명

- 반환값(Return)

True이면 정상동작, False이면 정상동작 아님.

- 설명(Remarks)

해당 Action ID 메뉴를 제거함

- 예제(Example)

*C++*

```
hwpMenu.RemoveMenu("CharShapeBold");
```

- 참고 항목(See Also)

[HwpMenu](#), [AppendMenu](#)