혼글 OLE Automation

1.1. 혼글 Objet Model

```
IHwpObject
├─ IXHwpDocuments
│   └─ IXHwpDocument
│       ├─ IXHwpFormPushButtons
│       │   └─ IXHwpFormPushButton
│       ├─ IXHwpFormCheckButtons
│       │   └─ IXHwpFormCheckButton
│       ├─ IXHwpFormRadioButtons
│       │   └─ IXHwpFormRadioButton
│       ├─ IXHwpFormComboBoxs
│       │   └─ IXHwpFormComboBox
│       └─ IXHwpFormEdits
│           └─ IXHwpFormEdit
├─ IXHwpWindows
│   └─ IXHwpWindow
│       ├─ IXHwpTabs
│       │   └─ IXHwpTab
│       └─ IXHwpToobarLayout
│           ├─ IXHwpToolbar
│           ├─ IXHwpToolbarButton
│           └─ IXHwpToolBarMenuButton
├─ IXHwpMessageBox
├─ IXHwpODBC
├─ HAction
├─ HParameterSet
│   └─ H[SetObjects]
│       ├─ HSet
│       └─ HArray
├─ IDHwpAction
├─ IDHwpParameterSet
├─ IDHwpParameterArray
└─ IDHwpCtrlCode
```

Interface : Single Object

Interface : Collection Object

1.2. 혼글 Objets

OLE Automation Object

Automation Object는 Automation Server가 Automation Client에게 제공하는 프로퍼티, 메소드, 이벤트로 구성된 개체이다.

IHwpObject는 혼글 Automation Server의 최상위 오브젝트이며 단일 오브젝트와 Collection 오브젝트로 구성된 하위의 오브젝트는 모두 IHwpObject로부터 생성된다.

```
(JavaScript에서 IHwpObject 최상위 개체 생성)
var App = new ActiveXObject("HWPFrame.HwpObject.1");        //IXHwpObject 생성
var Docs = App.XHwpDocuments;

(Visual C++에서 IHwpObject 최상위 개체 생성)
IHwpObject hwpobject;
if (hwpobject.CreateDispatch("HWPFrame.HwpObject.1")) {
        MessageBox("한글에 연결을 성공하였습니다.", "성공", MB_OK);
} else {
        MessageBox("한글에 연결을 실패하였습니다.", "오류", MB_OK);
        PostQuitMessage(0);
}
```

OLE Automation Collection Object

OLE Automation 단일 Object를 그룹화 하는 Object

혼글에서 새글 또는 새탭으로 열려있는 문서는 IXHwpDocument Object에서 제어 가능하며 열려있는 문서를 모두 관리 할 수 있는 것은 IXHwpDocuments Collection Object이다.

IXHwpDocument Object는 IXHwpDocuments의 Add 메소드에 의하여 생성 될 수 있으며 생성된 Object는 Item 프로퍼티에 의하여 구할 수 있다.

Property

OLE Automation Object의 상태에 대한 정보를 제어할 수 있는 기능을 제공

IXHwpDocument의 EditMode 프로퍼티는 지정된 문서가 읽기 모드인지 일반 편집 모드인지 부분 편집 모드인지를 얻을 수 있거나 설정할 수 있다.

Method

OLE Automation Object가 특정한 동작을 수행하도록 하는 기능을 제공

IXHwpMessageBox의 DoModal 메소드는 메시지 박스를 보여준다.

Event

OLE Automation Object에서 Client로 특정 사건이 발생했음을 알려주는 기능을 제공 하며 IHwpObject에서 이벤트를 얻을 수 있다.

```
(Visual Basic에서 IHwpObject 최상위 개체로부터 종료 이벤트 얻기)
Private WithEvents objHwpObject As HwpObject
Private Sub Form_Load()
        Set objHwpObject = New HwpObject
```

```
        MsgBox "한글에 접속"
End Sub

Private Sub objHwpObject_Quit()
        MsgBox "한글로부터 종료 이벤트를 받았습니다."
End Sub
```

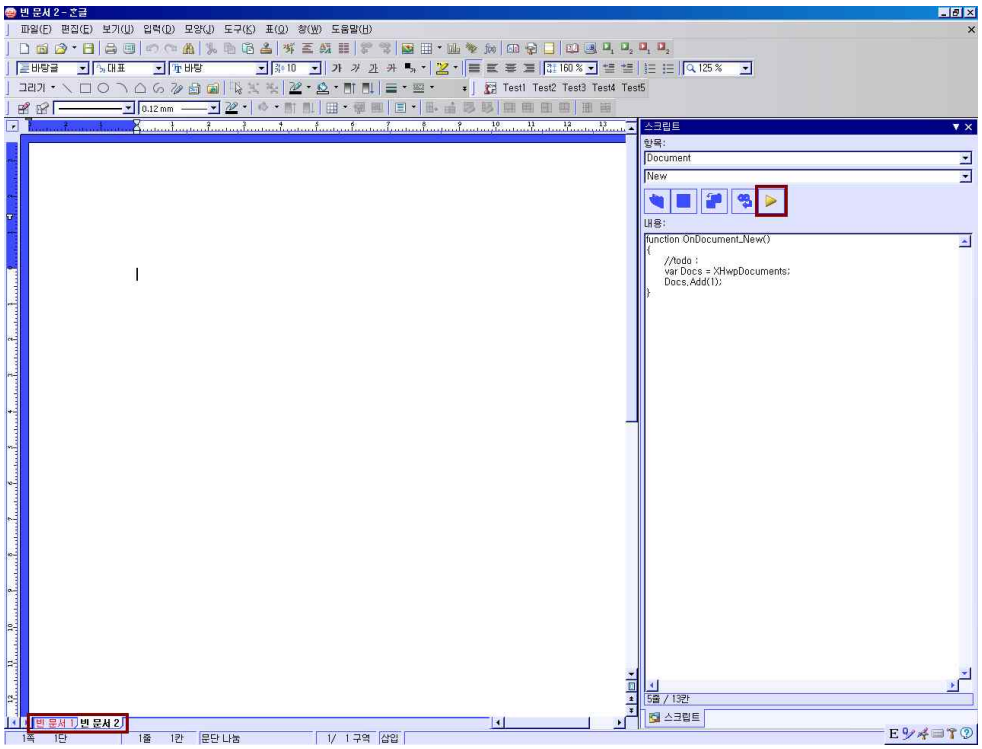
1.3. 한글 내부 개체

내부 개체

한글은 Automation Client가 제어 할 수 있듯이 한글 내부에서도 스크립트로 제어할 수 있다. 스크립트는 JavaScript를 사용한다.

내부에서 사용되므로 IHwpObject는 이미 생성 되어있고 스크립트에서 최상위 IHwpObject 생성을 위한 코드는 필요가 없다.

(내부 스크립트에서 한글 제어)



JavaScript를 이용한 예제

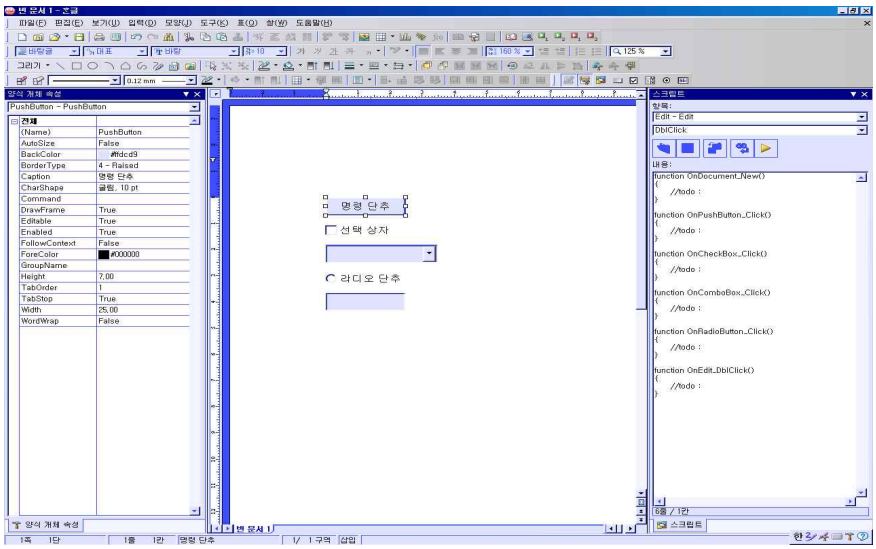
```
//IHwpObject는 이미 생성되어있으므로 IHwpObject의 하위 항목은
//XHwpDocuments와 XHwpMessageBox같이 바로 사용 가능하다.
function OnDocument_Open()
{
        var Docs = XHwpDocuments;
        Docs.Add(1);        //새 창으로 문서 생성 (0 = 새 창, 1 = 새 탭)
        var count = Docs.Count;     //생성된 Document의 개수 얻기
```

```
        var msgbox = XHwpMessageBox;
        msgbox.String = count;
        msgbox.DoModal();
}
```

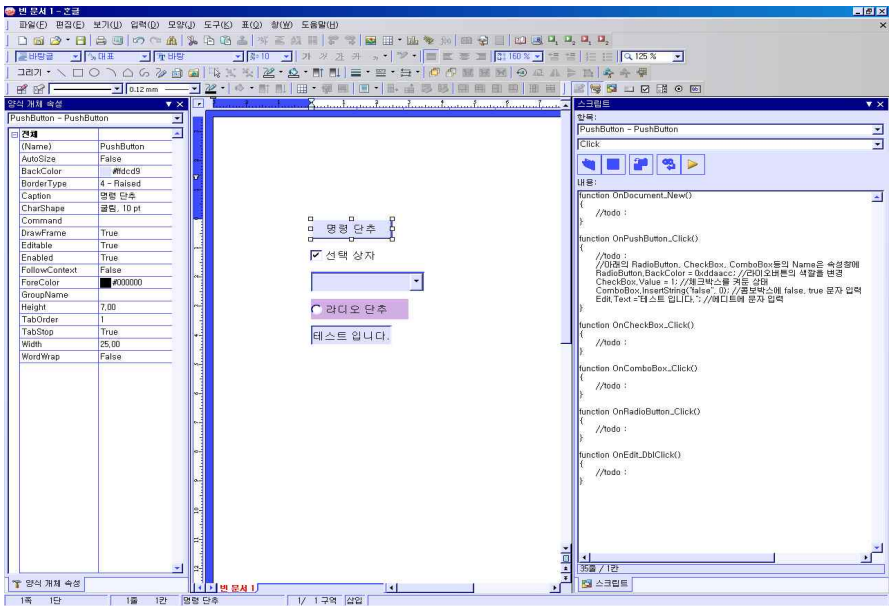
양식 개체와 내부 개체

흔글은 양식 개체와 내부 개체를 연동하여 사용할 수 있다.
양식 개체는 PushButton, CheckBox, ComboBox, RadioButton, Edit가 있다.
각 양식 개체는 Automation Object로 되어있으므로 스크립트로 제어가 가능하며
양식개체의 특정 이벤트에따라 스크립트가 수행되도록 할 수 있다.

(스크립트 적용 전)



(스크립트 적용 후)



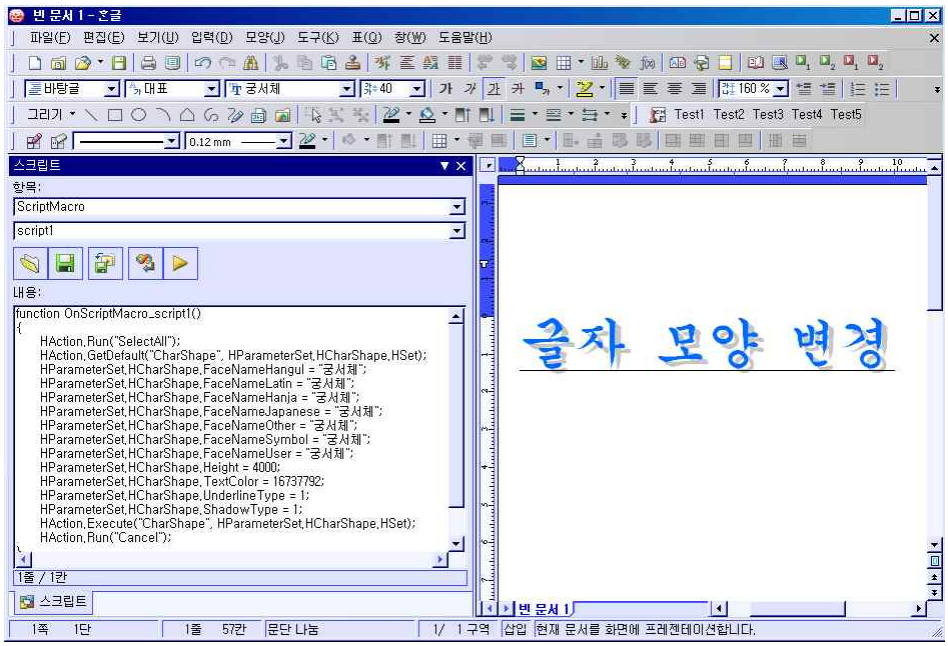
양식개체 스크립트 적용 예

```
function OnPushButton_Click()        //PushButton을 클릭 했을때 수행되는 함수
{
        //아래의 RadioButton, CheckBox, ComboBox등의 Name은
        //속성 창에 Name을 보면 알 수 있다.
        RadioButton.BackColor = 0xddaacc; //라이오버튼의 색깔을 변경
        CheckBox.Value = 1; //체크박스를 켜둔 상태
        ComboBox.InsertString("false", 0); //콤보박스에 false, true 문자 입력
                ComboBox.InsertString("true", 1);
        Edit.Text ="테스트 입니다."; //에디트에 문자 입력
}
```

1.4. 외부 클라이언트와 한글

한글은 외부 클라이언트(응용 프로그램)와 연결되어 외부 클라이언트는 한글을 제어할 수 있다. 외부 클라이언트가 한글에 접속을 요청할 때 한글은 자동으로 실행 된다.

(내부 스크립트로 글자 모양을 변경한 예)

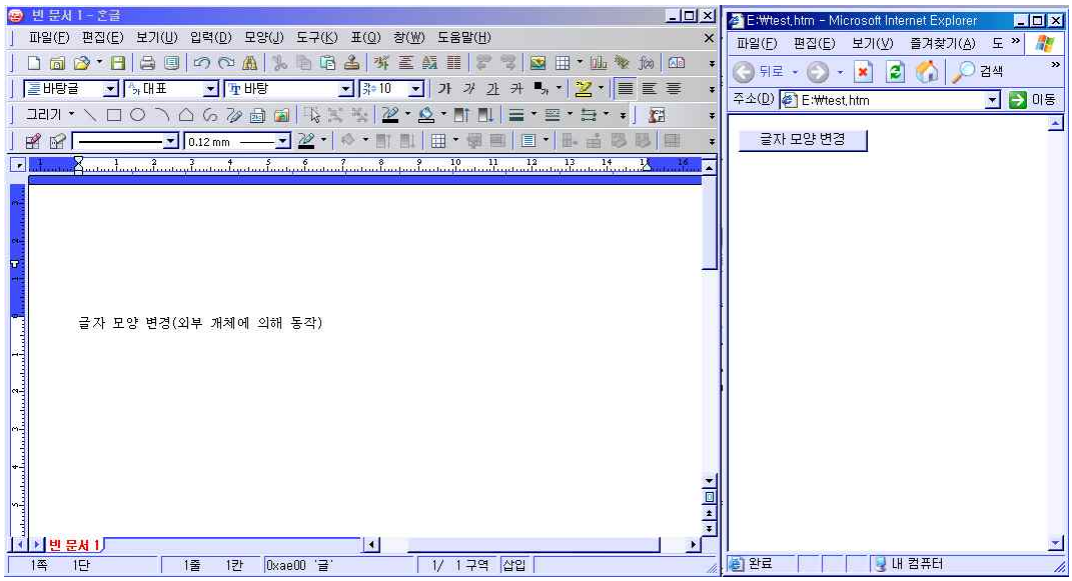


(사용된 스크립트 – 스크립트 매크로로 생성)

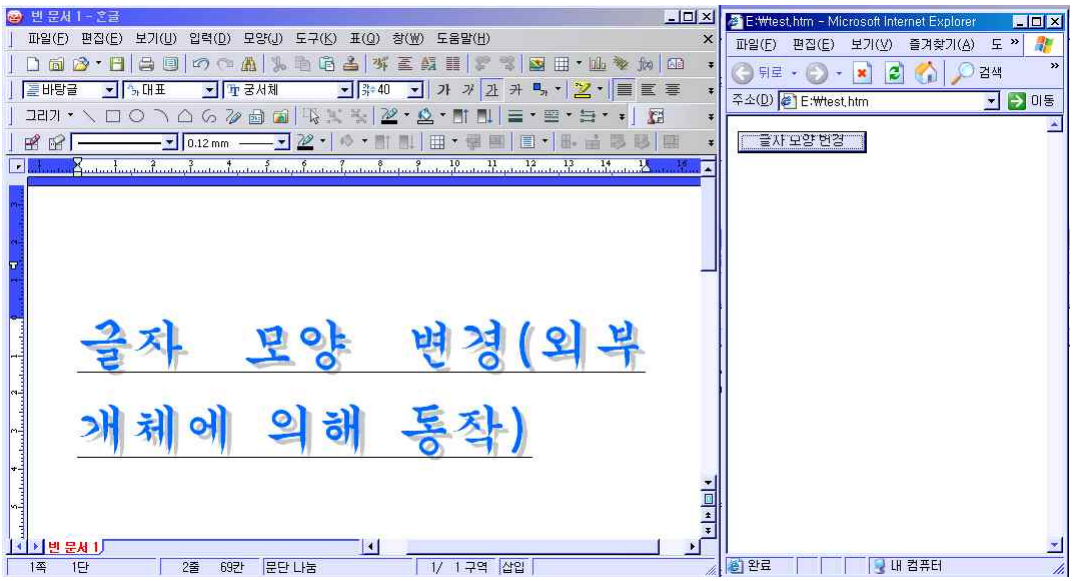
```
HAction.Run("SelectAll");
HAction.GetDefault("CharShape", HParameterSet.HCharShape.HSet);
HParameterSet.HCharShape.FaceNameHangul = "궁서체";
HParameterSet.HCharShape.FaceNameLatin = "궁서체";
HParameterSet.HCharShape.FaceNameHanja = "궁서체";
HParameterSet.HCharShape.FaceNameJapanese = "궁서체";
HParameterSet.HCharShape.FaceNameOther = "궁서체";
HParameterSet.HCharShape.FaceNameSymbol = "궁서체";
HParameterSet.HCharShape.FaceNameUser = "궁서체";
HParameterSet.HCharShape.Height = 4000;
HParameterSet.HCharShape.TextColor = 16737792;
```

```
HParameterSet.HCharShape.UnderlineType = 1;
HParameterSet.HCharShape.ShadowType = 1;
HAction.Execute("CharShape", HParameterSet.HCharShape.HSet);
HAction.Run("Cancel");
```

(외부 클라이언트(인터넷 익스플로러)가 한글을 제어 – 적용 전)



(외부 클라이언트(인터넷 익스플로러)가 한글을 제어 – 적용 후)

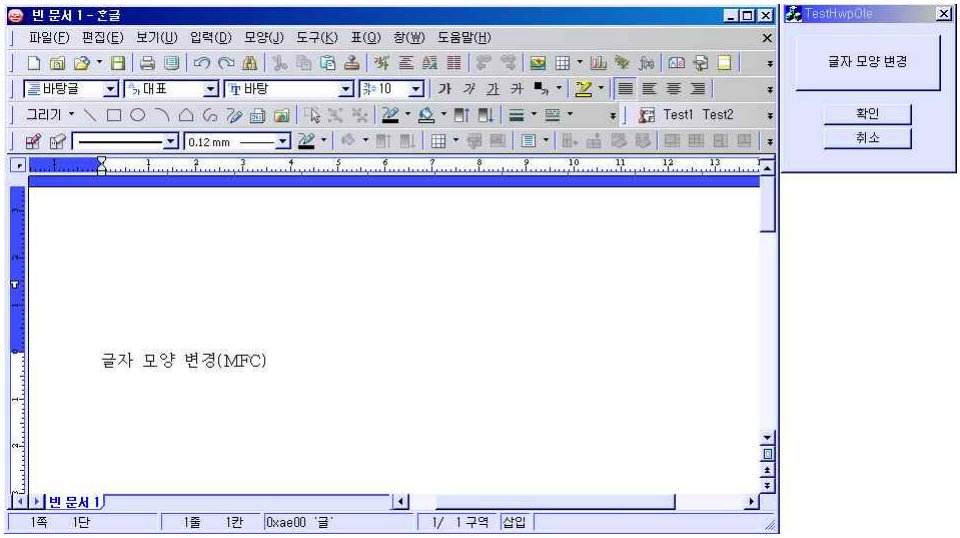


(사용된 스크립트)

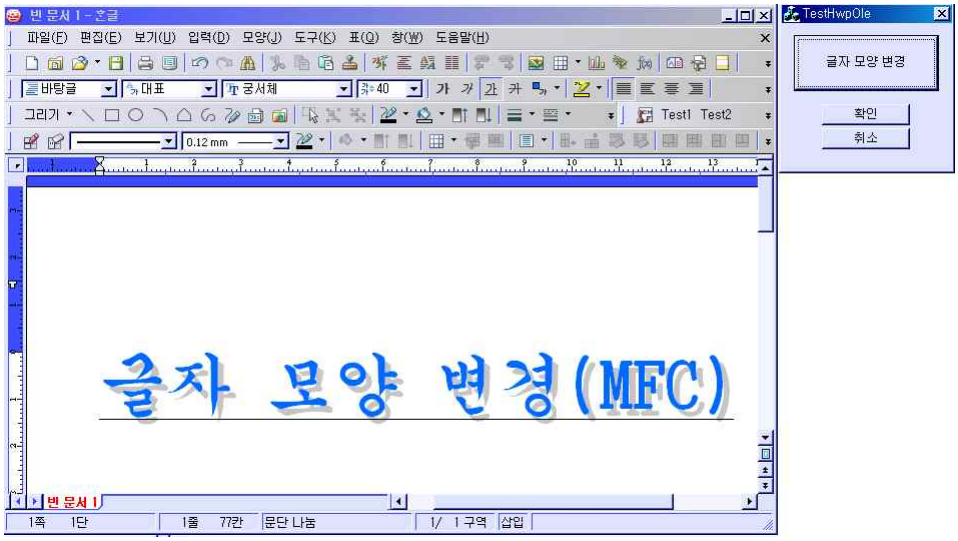
```
<html>
        <script language="jscript">
                var App = new ActiveXObject("HWPFrame.HwpObject.1");
                function OnChangeCharShape() {
                        App.HAction.Run("SelectAll");
                        App.HAction.GetDefault("CharShape"
```

```
                                        , App.HParameterSet.HCharShape.HSet);
                        App.HParameterSet.HCharShape.FaceNameHangul = "궁서체";
                        App.HParameterSet.HCharShape.FaceNameLatin = "궁서체";
                        App.HParameterSet.HCharShape.FaceNameHanja = "궁서체";
                        App.HParameterSet.HCharShape.FaceNameJapanese = "궁서체";
                        App.HParameterSet.HCharShape.FaceNameOther = "궁서체";
                        App.HParameterSet.HCharShape.FaceNameSymbol = "궁서체";
                        App.HParameterSet.HCharShape.FaceNameUser = "궁서체";
                        App.HParameterSet.HCharShape.Height = 4000;
                        App.HParameterSet.HCharShape.TextColor = 16737792;
                        App.HParameterSet.HCharShape.UnderlineType = 1;
                        App.HParameterSet.HCharShape.ShadowType = 1;
                        App.HAction.Execute("CharShape"
                                        , App.HParameterSet.HCharShape.HSet);
                        App.HAction.Run("Cancel");
                }
        </script language="jscript">
        <body>
                <button onclick="OnChangeCharShape()">  글자 모양 변경 </button>
        </body>
</html>
```

(외부 클라이언트(exe 파일)가 한글을 제어 – 적용 전)



(외부 클라이언트(exe 파일)가 한글을 제어 – 적용 전)



(적용 코드)

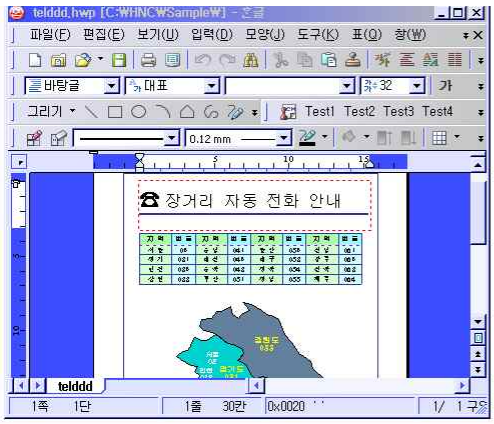
```
void CTestHwpOleDlg::OnButton1()
{
        // TODO: Add your control notification handler code here

        HAction hAction;
        HParameterSet hParameterSet;
        HCharShape hCharShape;
        HSet hSet;

        hAction.AttachDispatch(m_pHwpObject.GetHAction());
        hParameterSet.AttachDispatch(m_pHwpObject.GetHParameterSet());
        hCharShape.AttachDispatch(hParameterSet.GetHCharShape());
        hSet.AttachDispatch(hCharShape.GetHSet());
```

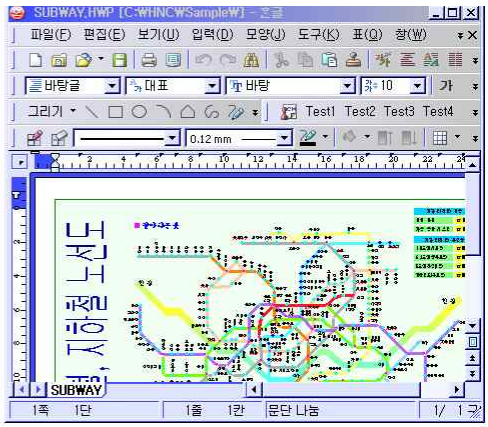
```
        hAction.Run("SelectAll");
        hAction.GetDefault("CharShape", hSet);
        hCharShape.SetFaceNameHangul("궁서체");
        hCharShape.SetFaceNameLatin("궁서체");
        hCharShape.SetFaceNameHanja("궁서체");
        hCharShape.SetFaceNameJapanese("궁서체");
        hCharShape.SetFaceNameOther("궁서체");
        hCharShape.SetFaceNameSymbol("궁서체");
        hCharShape.SetFaceNameUser("궁서체");
        hCharShape.SetHeight(4000);
        hCharShape.SetTextColor(16737792);
        hCharShape.SetUnderlineType(1);
        hCharShape.SetShadowType(1);
        hAction.Execute("CharShape", hSet);
        hAction.Run("Cancel");
}
```

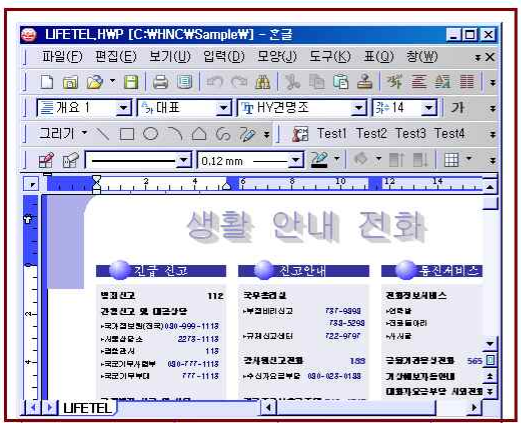
1.5. 외부 클라이언트와 실행 중인 혼글

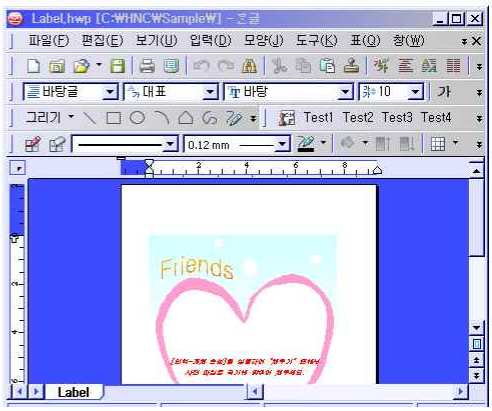
이미 실행 중인 혼글에 외부 클라이언트가 접속하여 혼글을 제어할 수 있다. (Running Object Table을 액세스 할 수 있는 프로그래밍 언어로 작성 된 프로그램에서 가능함)

(이미 실행 중인 혼글에 외부 클라이언트 제어 – 적용 전)

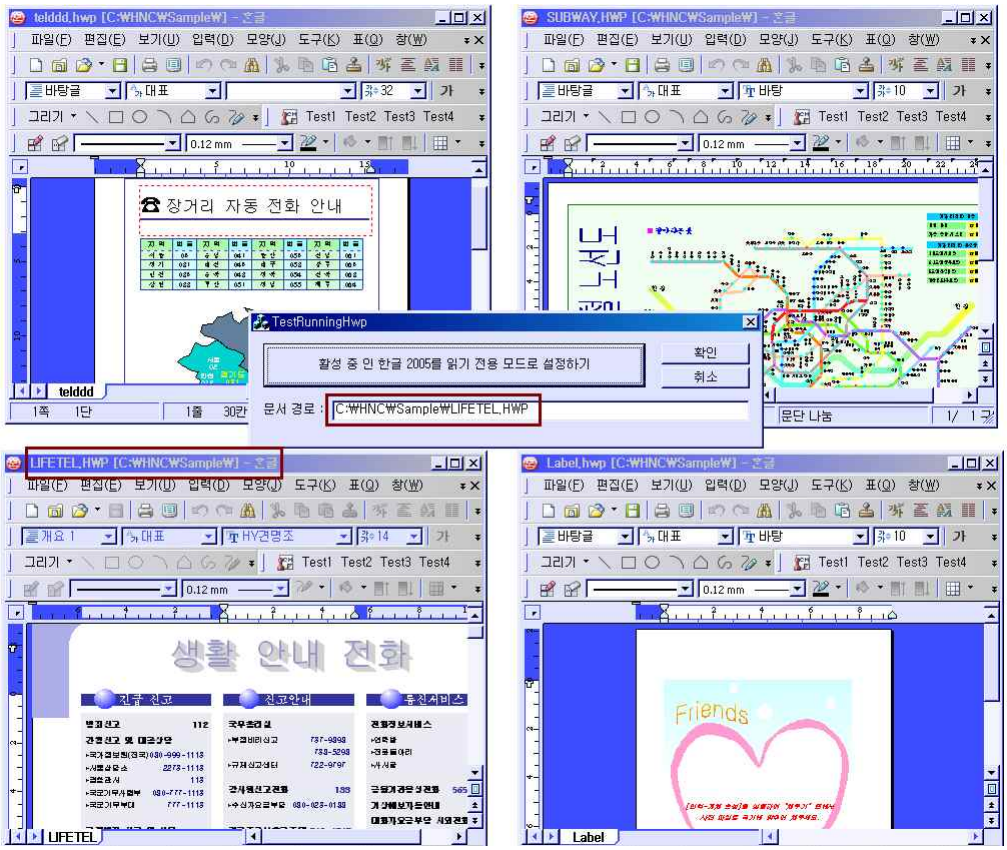








(이미 실행 중인 훈글에 외부 클라이언트 제어 – 적용 후)



(4개의 실행 중인 훈글에서 활성화 되어 있는 훈글의 편집 모드를 외부 클라이언트가 읽기 모드로 변경함)

적용된 코드

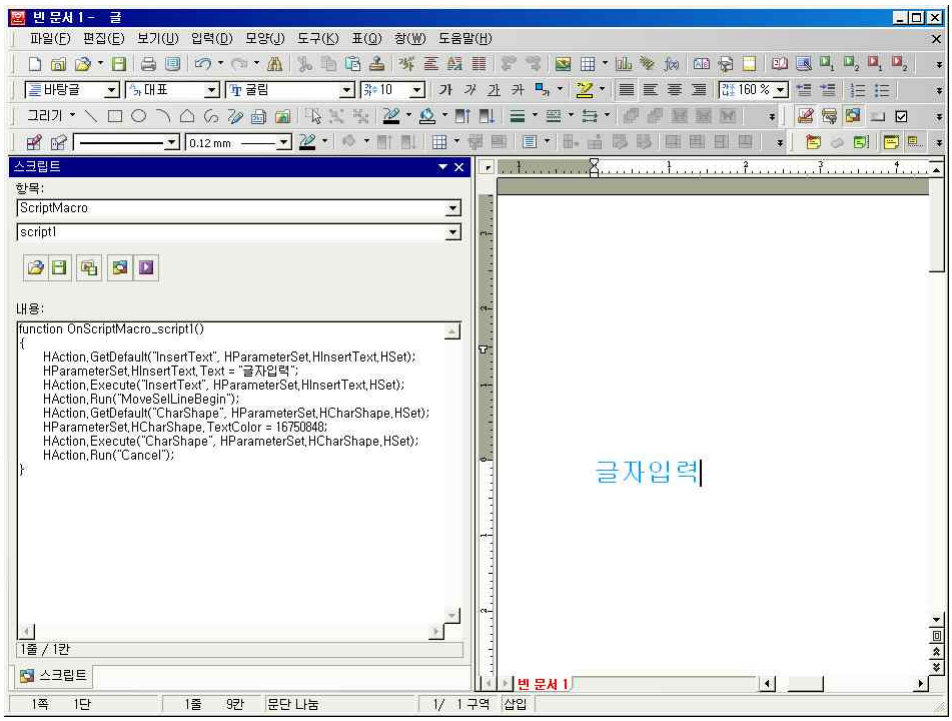
```
void CTestRunningHwpDlg::OnConnectButton()
{
        TCHAR szobjectname[] = "!HwpObject65.";
        WCHAR* szname = NULL;

        IEnumMoniker* pemo = NULL;
        IRunningObjectTable* pobjtable = NULL;
        IMoniker* iMoniker = NULL;
        IUnknown* iUnknown = NULL;
        IBindCtx* iBindCtx = NULL;
        ULONG uelement = 0;

        IHwpObject cActiveHwpObj;
        IXHwpDocuments cDocuments;
        IXHwpDocument cActiveDocument;

        GetRunningObjectTable(0, &pobjtable);
        pobjtable->EnumRunning(&pemo);
        while (SUCCEEDED(pemo->Next(1, &iMoniker, &uelement))) {
                if (uelement == 0)
```

```
                break;

            pobjtable->GetObject(iMoniker, &iUnknown);
            CreateBindCtx(0, &iBindCtx);
            iMoniker->GetDisplayName(iBindCtx, NULL, (OLECHAR**)&szname);
            CString strobject;
            strobject = szname;
            if (strobject.Find(szobjectname, 0) != -1) {
                LPDISPATCH pdisp;
                iUnknown->QueryInterface(IID_IDispatch, (void**) &pdisp);
                if (pdisp) {
                    cActiveHwpObj.AttachDispatch(pdisp);
                    LPDISPATCH pdocsdisp;
                    pdocsdisp = cActiveHwpObj.GetXHwpDocuments();
                    cDocuments.AttachDispatch(pdocsdisp);
                    LPDISPATCH pdocdisp;
                    pdocdisp = cDocuments.GetActive_XHwpDocument();
                    if (pdocdisp) {
                        cActiveDocument.AttachDispatch(pdocdisp);
                        cActiveDocument.SetEditMode(0);
                        m_strGet = cActiveDocument.GetFullName();
                        UpdateData(FALSE);
                    } else {
                        continue;
                    }
                }
            }
        }
}
```

1.6. 스크립트 매크로를 이용한 코드 적용 예제

스크립트 매크로 기록으로 코드 생성

1. "글자입력" 문자열 입력
2. 전체 블록 설정
3. 글자모양에서 색깔을 파란색으로 설정
4. 블록 설정 해제



스크립트 매크로로 생성된 스크립트 코드

```
function OnScriptMacro_script1()
{
        HAction.GetDefault("InsertText", HParameterSet.HInsertText.HSet);
        HParameterSet.HInsertText.Text = "글자입력";
        HAction.Execute("InsertText", HParameterSet.HInsertText.HSet);
        HAction.Run("MoveSelLineBegin");
        HAction.GetDefault("CharShape", HParameterSet.HCharShape.HSet);
        HParameterSet.HCharShape.TextColor = 16750848;
        HAction.Execute("CharShape", HParameterSet.HCharShape.HSet);
        HAction.Run("Cancel");
}
```

생성된 스크립트 코드를 인터넷 익스플로러에 적용한 예제

```
<HTML>
<SCRIPT LANGUAGE="JSCRIPT">
var App = new ActiveXObject("HWPFrame.HwpObject.1");
function OnScriptMacro_script1()
{
App.HAction.GetDefault("InsertText", App.HParameterSet.HInsertText.HSet);
App.HParameterSet.HInsertText.Text = "글자입력";
App.HAction.Execute("InsertText", App.HParameterSet.HInsertText.HSet);
App.HAction.Run("MoveSelLineBegin");
App.HAction.GetDefault("CharShape", App.HParameterSet.HCharShape.HSet);
App.HParameterSet.HCharShape.TextColor = 16750848;
App.HAction.Execute("CharShape", App.HParameterSet.HCharShape.HSet);
App.HAction.Run("Cancel");
}
</SCRIPT>
<BODY>
<BUTTON OnClick="OnScriptMacro_script1()"> 글자 테스트</BUTTON>
</BODY>
</HTML>
```

생성된 스크립트 코드를 MFC에 적용한 예제

```
IHwpObject pHwpObject;
void CTestText::OnButtonClick()
{
BOOL bres = pHwpObject.CreateDispatch("HWPFrame.HwpObject.1");
if (bres == FALSE)
        return ;
HAction haction;
haction.AttachDispatch(pHwpObject.GetHAction());
HParameterSet hparameterset;
hparameterset.AttachDispatch(pHwpObject.GetHParameterSet());
HInsertText hinserttext;
hinserttext.AttachDispatch(hparameterset.GetHInsertText);
HSet hset1;
hset1.AttachDispatch(hinserttext.GetHSet);
haction.GetDefault("InsertText", hset1);
hinserttext.Text = "글자입력";
haction.Execute("InsertText", hset1);
haction.Run("MoveSelLineBegin");
HCharShape hcharshape;
hcharshape.AttachDispatch(hparameterset.GetHCharShape);
HSet hset2;
hset2.AttachDispatch(hcharshape.GetHSet);
haction.GetDefault("CharShape", hset2);
hcharshape.TextColor = 16750848;
haction.Execute("CharShape", hset2);
haction.Run("Cancel");
}
```

1.7. OLE Automation 오브젝트

IHwpObject : 최상위 개체 (모든 Automation Object의 최상위 Object이다.

Item Name	Description
IsModified(Property)	Description 문서가 변경되어있는지 나타낸다. (읽기 전용) Remark 0 = 변경되지 않은 깨끗한 상태 1 = 변경된 상태 2 = 변경되었으나 자동 저장된 상태
IsEmpty(Property)	Description 아무 내용도 들어있지 않은 빈 문서인지 여부를 나타낸다.(읽기 전용)
EditMode(Property)	Description 현재 편집 모드 Remark 0 : 읽기 전용 1 : 일반 편집모드 2 : 양식 모드(양식 사용자 모드) : Cell과 누름틀 중 양식 모드에서 편집 가능 속성을 가진 것만 편집 가능하다. 16 : 배포용 문서 (SetEditMode로 지정 불가능)
SelectionMode(Property)	Description 문서의 내용이 어떤 Selection 상태인가를 알려준다.(읽기 전용) Remark 일반 블록이 아닌 F3키나 F4키에 의해 블록이 지정된 경우, HWPSEL_STRICT_MODE (= 0x10, 십진수 16)으로 OR 마스크되어 오기때문에, 항상 0x0F(십진수15)로 AND 마스크한 결과로 판단하도록 한다.
CurFieldState(Property)	Description 캐럿이 위치한 필드의 상태 정보를 구한다.(읽기 전용) Remark bit 5 - 31 = 예약 bit 4 = 필드명의 존재 여부 (1 = 있음, 0 = 없음) bit 0 - 3 = 필드의 종류 (0 = 없음, 1 = 셀, 2 = 누름틀)
PageCount(Property)	Description 문서 페이지 수 (읽기 전용) Remark 문서의 전체 페이지 수를 나타낸다. 문서 전체에 대한 pagination이 수행되지 않은 상태에서 이 property를 참조하면 먼저 문서 전체의 pagination을 먼저 수행하므로 긴 문서에 대해 문서 내용 변경과 참조를 반복하면 속도가 심각하게 느려질 수 있다.
CellShape(Property)	Description 현재 선택되어있는 표와 셀의 모양 정보를 나타낸다. Remark ParameterSet/Table로 표의 속성에 대한 기본 정보를 나타내며, 이 가운데 "Cell" 아이템이 ParameterSet/Cell로 셀의 속성을 나타낸다. 셀 블록이 잡혀있지 않은 상태이면 현재 캐럿이 위치한 셀 하나만을 대상으로 한다. 현재 표 내부에 캐럿이 위치하지 않으면 에러가 발생한다.
CharShape(Property)	Description 현재 Selection의 글자 모양을 나타낸다. Remark property get을 수행하면 현재 selection 내의 글자 모양을 구할 수 있다. selection이 존재하지 않으면 현재 캐럿이 위치한 곳의 글자 모양을 돌려준다. 글자 모양 중 특정 항목이 selection 내에서 서로 다른

<table>
<tr><td></td><td>속성을 가지고 있으면 아예 아이템 자체가 존재하지 않는다.
property set을 수행하면 아이템이 존재하는 항목에 대해서만 속성을 설정한다. (ParameterSet의 형식은 ParameterSet/CharShape 참조.)</td></tr>
<tr><td>HeadCtrl(Property)</td><td>Description
문서 중 첫 번째 컨트롤(읽기 전용)
Remark
문서 중의 모든 컨트롤(표, 그림 등의 특수 문자들)은 linked list로 서로 연결되어 있는데, 그 list의 시작 컨트롤을 나타낸다. 이 컨트롤로부터 시작, Ctrl.Next를 이용해 forward iteration을 수행할 수 있다.</td></tr>
<tr><td>LastCtrl(Property)</td><td>Description
문서 중 마지막 컨트롤 (읽기 전용)
Remark
문서 중의 모든 컨트롤(표, 그림 등의 특수 문자들)은 linked list로 서로 연결되어 있는데, 그 list의 마지막 컨트롤을 나타낸다. 이 컨트롤로부터 시작, Ctrl.Prev를 이용해 backward iteration을 수행할 수 있다.</td></tr>
<tr><td>CurSelectedCtrl(Property)</td><td>Description
현재 선택되어 있는 컨트롤(읽기 전용)</td></tr>
<tr><td>ParaShape(Property)</td><td>Description
현재 Selection의 문단 모양을 나타낸다.
Remark
개념과 사용법은 CharShape과 동일하다.
(ParameterSet의 형식은 ParameterSet/ParaShape 참조.)</td></tr>
<tr><td>ParentCtrl(Property)</td><td>Description
현재 캐럿의 상위 컨트롤(읽기 전용)
Remark
상위 컨트롤은 현재 캐럿이 위치한 리스트를 보유한 컨트롤이다. 예를 들어 셀 내부에 위치하면 표, 각주 내용에 위치하면 각주, 바탕쪽이면 구역 컨트롤이 상위 컨트롤이다. 현재 캐럿이 본문 레벨에 위치해 상위 컨트롤이 없을 때는 NULL이 리턴된다.</td></tr>
<tr><td>ViewProperties(Property)</td><td>Description
뷰의 상태 정보
Remark
조판 부호, 화면 확대 비율과 같은 view에 관련된 정보를 나타낸다.
(ParameterSet의 형식은 ParameterSet/ViewProperties 참조.)</td></tr>
<tr><td>Path(Property)</td><td>Description
문서 파일의 경로</td></tr>
<tr><td>Open(Method)</td><td>Description
문서 파일을 연다.
Declaration
Open(BSTR path, BSTR format, BSTR arg)
Parameters
path : 문서 파일의 경로(URL 사용 가능)
format : 문서 형식. 별도 설명 참조. 빈 문자열을 지정하면 자동으로 인식한다. 생략하면 빈 문자열이 지정된다.
arg : 세부 옵션. 의미는 format에 지정한 파일 형식에 따라 다르다. 생략하면 빈 문자열이 지정된다.
Remark
format에 지정할 수 있는 문서 형식은 현재 시스템에 설치된 문서 필터(*.dft)의 종류에 따라 달라진다. 일반적으로 설치되는 형식들에는 다음과 같은 종류가 있다.
<table>
<tr><td>HWP</td><td>워디안 native format</td></tr>
<tr><td>HWP30</td><td>한글 3.X/96/97</td></tr>
</table></td></tr>
</table>

HTML	인터넷 문서
TEXT	아스키 텍스트 문서
UNICODE	유니코드 텍스트 문서
HWP20	한글 2.0
HWP21	한글 2.1/2.5
HWP15	한글 1.X
HWPML1X	HWPML 1.X 문서 (Open만 가능)
HWPML2X	HWPML 2.X 문서 (Open / SaveAs 가능)
RTF	서식있는 텍스트 문서
DBF	DBASE II/III 문서
HUNMIN	훈민정음 3.0/2000
MSWORD	마이크로소프트 워드 문서
HANA	하나워드 문서
ARIRANG	아리랑 문서
ICHITARO	一太郎 문서 (일본 워드프로세서)
WPS	WPS 문서
DOCIMG	인터넷 프레젠테이션 문서(SaveAs만 가능)
SWF	Macromedia Flash 문서(SaveAs만 가능)

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

key:value;key:value;...

* key는 A-Z, a-z, 0-9, _ 로 구성된다.
* value는 타입에 따라 다음과 같은 3 종류가 있다.
  boolean: ex) fullsave:true (== fullsave)
  integer: ex) type:20
  string: ex) prefix:_This_
* value는 생략 가능하며, 이때는 콜론도 생략한다.
* arg에 지정할 수 있는 옵션

"모든 파일"	setcurdir(boolean, FALSE) : 로드한 후 해당 파일이 존재하는 폴더로 현재 위치를 변경한다. hyperlink 정보가 상대적인 위치로 되어 있을 때 유용하다.
"HWP"	- lock (boolean, TRUE) : 로드한 후 해당 파일을 계속 오픈한 상태로 lock을 걸지 여부 - notext (boolean, FALSE) : 텍스트 내용을 읽지 않고 헤더 정보만 읽을지 여부. (스타일 로드 등에 사용) - template (boolean, FALSE): 새로운 문서를 생성하기 위해 템플릿 파일을 오픈한다. 이 옵션이 주어지면 lock은 무조건 FALSE로 처리된다. - suspendpassword (boolean, FALSE): TRUE로 지정하면, 암호가 있는 파일일 경우 암호를 묻지 않고 무조건 읽기에 실패한 것으로 처리한다. - forceopen (boolean, FALSE): TRUE로 지정하면, 읽기 전용으로 읽어야 하는 경우 대화상자를 띄우지 않는다.
"HTML"	- code(string, codepage) : 문서 변환시 사용되는 코드 페이지를 지정할 수 있이며 code키가 존재할 경우 필터 사용시 사용자 다이얼로그를 띄우지 않는다. - textunit(boolean, pixel) : Export될 Text의 크기의 단위 결정(pixel, point, mili 지정 가능.) - formatunit(boolean, pixel) : Export될 문서 포맷 관련 (마진, Object 크기 등) 단위 결정. pixel, point, mili 지정 가능
"DOCIMG"	- asimg(boolean, FALSE) : 저장할 때 페이지를 image로 저장 - ashtml(boolean, FALSE) : 저장할 때 페이지를 html

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>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</td></tr>
<tr><td>"TEXT"</td><td>- code(string, codepage): 문서 변환시 사용되는 코드 페이지를 지정할 수 있이며 code키가 존재할 경우 필터 사용시 사용자 다이얼로그를 띄우지 않는다.

[codepage 종류]
- ks : 한글 KS 완성형 | kssm : 한글 조합형 | sjis : 일본 | gb : 중국 간체 | big5 : 중국 번체 | acp : Active Codepage 현재 시스템의 코드 페이지</td></tr>
</table>
</td></tr>
<tr><td>Save(Method)</td><td>Description
현재 편집중인 문서를 저장한다.
Declaration
BOOL Save(VARIANT save_if_dirty)
Parameters
save_if_dirty : True를 지정하면 문서가 변경된 경우에만 저장한다. False를 지정하면 변경 여부에 관계없이 무조건 저장한다. 생략하면 True가 지정된다.
Remark
문서의 경로가 지정되어 있지 않으면 "새이름으로 저장" 대화상자가 떠서 사용자에게 경로를 묻는다.</td></tr>
<tr><td>SaveAs(Method)</td><td>Description
현재 편집중인 문서를 지정한 이름으로 저장한다.
Declaration
BOOL Save(BSTR path, VARIANT format, VARIANT arg)
Parameters
path : 문서 파일의 경로
format : 문서 형식. 별도 설명 참조. 생략하면 "HWP"가 지정된다.
arg : 세부 옵션. 의미는 format에 지정한 파일 형식에 따라 다르다. 생략하면 빈 문자열이 지정된다.
Remark
format, arg의 일반적인 개념에 대해서는 Open 참조.
"HWP" 형식으로 파일 저장시 arg에 지정할 수 있는 옵션은 다음과 같다.
<table>
<tr><th>함수</th><th>인자 타입</th><th>기본 값 (Default)</th><th>설명</th></tr>
<tr><td>lock</td><td>boolean</td><td>TRUE</td><td>저장한 후 해당 파일을 계속 오픈한 상태로 lock을 걸지 여부</td></tr>
<tr><td>backup</td><td>boolean</td><td>FALSE</td><td>백업 파일 생성 여부</td></tr>
<tr><td>compress</td><td>boolean</td><td>TRUE</td><td>압축 여부</td></tr>
<tr><td>fullsave</td><td>boolean</td><td>FALSE</td><td>스토리지 파일을 완전히 새로 생성하여 저장</td></tr>
<tr><td>prvimage</td><td>int</td><td>2</td><td>미리보기 이미지
(0=off, 1=BMP, 2=GIF)</td></tr>
<tr><td>prvtext</td><td>int</td><td>1</td><td>미리보기 텍스트(0=off, 1=0)</td></tr>
</table>
</td></tr>
<tr><td>Insert(Method)</td><td>Description
현재 캐럿 위치에 문서 파일을 삽입한다.
Declaration</td></tr>
</table>

	Insert(BSTR path, VARIANT format, VARIANT arg) Parameters path : 문서 파일의 경로, URL 사용 가능 format : 문서 형식. 별도 설명 참조. 빈 문자열을 지정하면 자동으로 디텍트한다. 생략하면 빈 문자열이 지정된다. arg : 세부 옵션. 의미는 format에 지정한 파일 형식에 따라 다르다. 생략하면 빈 문자열이 지정된다. Remark format, arg에 대해서는 Open 참조
SelectText(Method)	Description 블록을 설정 한다. Declaration BOOL SelectText(long spara, long spos, long epara, long epo) 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. Remark epos가 가리키는 문자는 포함되지 않는다.
CreateField(Method)	Description 캐럿의 현재 위치에 누름틀을 생성한다. Declaration BOOL CreateField(BSTR direction, VARIANT memo, VARIANT name) Parameters direction : 누름틀에 입력이 안된 상태에서 보여지는 안내문/지시문. memo : 누름틀에 대한 설명/도움말 name : 누름틀 필드에 대한 필드 이름
MoveToField(Method)	Description 지정한 필드로 캐럿을 이동한다. Declaration BOOL MoveToField(BSTR field, VARIANT text, VARIANT start, VARIANT select) Parameters field : 필드 이름. GetFieldText/PutFieldText과 같은 형식으로 이름 뒤에 '{{#}}'로 번호를 지정할 수 있다. text : 필드가 누름틀일 경우 누름틀 내부의 텍스트로 이동할지(True) 누름틀 코드로 이동할지(False)를 지정한다. 누름틀이 아닌 필드일 경우 무시된다. 생략하면 True가 지정된다. start : 필드의 처음(True)으로 이동할지 끝(False)으로 이동할지 지정한다. select를 True로 지정하면 무시된다. 생략하면 True가 지정된다. select : 필드 내용을 블록으로 선택할지(True), 캐럿만 이동할지(False) 지정한다. 생략하면 False가 지정된다. Remark 누름틀은 한글97 메뉴 중 입력 메뉴에 문서마당 정보를 선택하면 누름틀을 만드실 수 있습니다.
FieldExist(Method)	Description 문서 중에 지정한 데이터 필드가 존재하는지 검사한다. Declaration BOOL FieldExist(BSTR field) Parameters field : 필드 이름
GetFieldText(Method)	Description 지정한 필드에서 문자열을 구한다. Declaration

	BSTR GetFieldText(BSTR fieldlist) Parameters fieldlist : 텍스트를 구할 필드 이름의 리스트. 다음과 같이 필드 사이를 문자 코드 0x02로 구분하여 한 번에 여러 개의 필드를 지정할 수 있다. "필드이름#1Wx2필드이름#2Wx2...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}}Wx2본문Wx2이름{{0}}" 과 같이 지정하면 '제목' 이라는 이름의 필드 중 두 번째, '본문' 이라는 이름의 필드 중 첫 번째, '이름' 이라는 이름의 필드 중 첫 번째를 각각 지정한다. 즉, '필드이름' 과 '필드이름0 ' 은 동일한 의미로 해석된다. return : 텍스트 데이터가 돌아온다. 텍스트에서 탭은 'Wt'(0x9), 문단 바뀜은 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.
PutFieldText(Method)	Description 지정한 필드의 내용을 채운다. Declaration void PutFieldText(BSTR fieldlist, BSTR textlist) Parameters fieldlist : 내용을 채울 필드 이름의 리스트. 한 번에 여러 개의 필드를 지정할 수 있으며, 형식은 GetFieldText와 동일하다. 다만 필드 이름 뒤에 '# ' 로 번호를 지정하지 않으면 해당 이름을 가진 모든 필드에 동일한 텍스트를 채워 넣는다. 즉, PutFieldText에서는 '필드이름' 과 '필드이름0 ' 의 의미가 다르다. textlist : 필드에 채워 넣을 문자열의 리스트. 형식은 필드 리스트와 동일하게 필드의 개수만큼 텍스트를 0x02로 구분하여 지정한다. Remark 현재 필드에 입력되어 있는 내용은 지워진다. 채워진 내용의 글자모양은 필드에 지정해 놓은 글자모양을 따라간다. fieldlist의 필드 개수와, textlist의 텍스트 개수는 동일해야 한다. 존재하지 않는 필드에 대해서는 무시한다.
RenameField(Method)	Description 지정한 필드의 이름을 바꾼다. Declaration void RenameField(BSTR oldname, BSTR newname) Parameters oldname : 이름을 바꿀 필드 이름의 리스트. 형식은 PutFieldText과 동일하다. newname : 새로운 필드 이름의 리스트. oldname과 동일한 개수의 필드 이름을 0x02로 구분하여 지정한다. Remark 예를 들어 oldname에 "title{{0}}Wx2title{{1}}", newname에 "tt1Wx2tt2로 지정하면 첫 번째 title은 tt1로, 두 번째 title은 tt2로 변경된다. oldname의 필드 개수와, newname의 필드 개수는 동일해야 한다. 존재하지 않는 필드에 대해서는 무시한다.
GetCurFieldName(Method)	Description 현재 캐럿 위치의 데이터 필드 이름을 구한다. Declaration BSTR GetCurFieldName([HwpFieldOption option]) Parameters option : 다음과 같은 옵션을 지정할 수 있다. 0을 지정하면 모두 off 이다. 생략하면 0이 지정된다. (HwpFieldOption : short type)

<table>
<tr><td></td><td>
<table>
<tr><th>ID</th><th>값</th><th>설명</th></tr>
<tr><td>hwpFieldCell</td><td>1</td><td>셀에 부여된 필드 리스트만을 구한다.
hwpFieldClickHere와는 함께 지정할 수 없다.</td></tr>
<tr><td>hwpFieldClickHere</td><td>2</td><td>누름틀에 부여된 필드 리스트만을 구한다.
hwpFieldCell과는 함께 지정할 수 없다.</td></tr>
</table>
return : 필드 이름이 돌아온다. 필드 이름이 없는 경우 빈 문자열이 돌아온다.
Remark
GetFieldList()의 option 중에 hwpFieldSelection (= 4) 옵션은 사용하지 않는다.
</td></tr>
<tr><td>SetCurFieldName(Method)</td><td>
Description
현재 캐럿 위치의 데이터 필드 이름을 설정한다.
Declaration
BOOL SetCurFieldName(BSTR fieldname, [HwpFieldOption option], [BSTR direction], [BSTR memo])
Parameters
fieldname : 데이터 필드 이름.
option : 다음과 같은 옵션을 지정할 수 있다. 0을 지정하면 모두 off 이다. 생략하면 0이 지정된다. (HwpFieldOption : short type)
direction : 누름틀 필드의 안내문. 누름클 필드일 때만 유효하다.
memo : 누름틀 필드의 메모. 누름클 필드일 때만 유효하다.
Remark
GetFieldList()의 option 중에 hwpFieldSelection (= 4) 옵션은 사용하지 않는다.
</td></tr>
<tr><td>ModifyFieldProperties(Method)</td><td>
Description
지정한 필드의 속성을 바꾼다.
Declaration
long ModifyFieldProperties(LPCTSTR field, long remove, long add)
Parameters
field : 속성을 바꿀 필드 이름의 리스트. 형식은 PutFieldText과 동일.
remove : 제거될 속성
add : 추가될 속성
return : 음수가 리턴되면 에러임을 나타낸다.
Remark
속성의 값은 아래와 같다.
<table>
<tr><th>long value</th><th>설명</th></tr>
<tr><td>0x00000001</td><td>양식모드에서 편집가능 속성
(0: 편집 불가, 1: 편집 가능)</td></tr>
</table>
return 값의 bit field는 다음과 같다.
<table>
<tr><th>long value bit mask</th><th>설명</th></tr>
<tr><td>0x00000001</td><td>양식모드에서 편집가능 속성
(0: 편집 불가, 1: 편집 가능)</td></tr>
<tr><td>0x80000000</td><td>에러</td></tr>
<tr><td>0x40000000</td><td>필드를 찾을 수 없음</td></tr>
</table>
remove와 add에 둘다 0이 입력되면 현재 속성을 돌려준다.
리턴값이 음수인지 확인하여 쉽게 에러임을 판별할 수 있으며 자세한 에러내용은 bit mask로 and 연산하여 알아 낼 수 있다.
리턴값은 여러 가지 추가 정보가 같이 올 수 있으므로 반드시 bit mask를 사용하여 비교해야 한다.
</td></tr>
</table>

<table>
<tr>
<td>SetFieldViewOption(Method)</td>
<td>
Description
양식 모드와 읽기 전용모드일때 현재 열린 문서의 필드의 겉보기 속성(『』표시)을 바꾼다.
Declaration
long SetFieldViewOption(long option)
Parameters
option : 겉보기 속성 bit
<table>
<tr><th>option</th><th>누름틀</th><th>개인정보/문서요약/날짜시간/파일경로</th><th>비고</th></tr>
<tr><td>1</td><td>『』을 표시하지 않음</td><td>『』을 표시하지 않음</td><td></td></tr>
<tr><td>2</td><td>『』을 빨간색으로 표시</td><td>『』을 흰색으로 표시</td><td>설정하지 않았을 때 기본 값</td></tr>
<tr><td>3</td><td>『』을 흰색으로 표시</td><td>『』을 흰색으로 표시</td><td></td></tr>
</table>
return : 설정된 속성이 리턴 된다. 에러일 경우 0 이 리턴 된다.
Remark
EditMode와 비슷하게 현재 열려있는 문서에 대한 속성이다. 따라서 저장되지 않는다.
</td>
</tr>
<tr>
<td>GetFieldList(Method)</td>
<td>
Description
문서 중의 필드 리스트를 구한다.
Declaration
BSTR GetFieldList([HwpFieldNumber number], [HwpFieldOption option])
Parameters
number : 문서 중에서 동일한 이름의 필드가 여러 개 존재할 때 이를 구별하기 위한 식별 방법을 지정한다. 생략하면 hwpFieldPlain이 지정된다. (HwpFieldNumber : short type)
<table>
<tr><th>ID</th><th>값</th><th>설명</th></tr>
<tr><td>hwpFieldPlain</td><td>0</td><td>아무 기호 없이 순서대로 필드 이름이 나열된다.</td></tr>
<tr><td>hwpFieldNumber</td><td>1</td><td>필드 이름 뒤에 일련번호가 {{#}}와 같은 형식으로 붙는다.</td></tr>
<tr><td>hwpFieldCount</td><td>2</td><td>필드 이름뒤에 그 이름의 필드가 몇 개 있는지 {{#}}와 같은 형식으로 붙는다.</td></tr>
</table>
option : 다음과 같은 옵션을 조합할 수 있다. 0을 지정하면 모두 off이다. 생략하면 0이 지정된다. (HwpFieldOption : unsigned short type)
<table>
<tr><th>ID</th><th>값</th><th>설명</th></tr>
<tr><td>hwpFieldCell</td><td>1</td><td>셀에 부여된 필드 리스트만을 구한다. hwpFieldClickHere와는 함께 지정할 수 없다.</td></tr>
<tr><td>hwpFieldClickHere</td><td>2</td><td>누름틀에 부여된 필드 리스트만을 구한다. hwpFieldCell과는 함께 지정할 수 없다.</td></tr>
<tr><td>hwpFieldSelection</td><td>4</td><td>셀렉션 내에 존재하는 필드 리스트를 구한다.</td></tr>
</table>
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"
Remark
문서 중에 동일한 이름의 필드가 여러 개 존재할 때는 number에 지정한 타입에 따라 3 가지의 서로 다른 방식을 중에서 선택할 수 있다. 예를 들어 문서 중 title, body, title, body, footer 순으로 5 개의 필드가 존재할 때, hwpFieldPlain, hwpFieldNumber, HwpFieldCount 세 가지 형식에 따라 다음과 같은 내용이 돌아온다.
hwpFieldPlain “titleWx2bodyWx2titleWx2bodyWx2footer"
hwpFieldNumber "title{{0}}Wx2body{{0}}Wx2title{{1}}Wx2body{{1}}Wx2footer{{0}} “
hwpFieldCount "title{{2}}Wx2body{{2}}Wx2footer{{1}}"
</td>
</tr>
</table>

MovePos(Method)

Description

캐럿의 위치를 옮긴다.

Declaration

BOOL MovePos([unsigned int moveID], [unsigned int para], [unsigned int pos])

Parameters

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

ID	값	설명
moveMain	0	루트 리스트의 특정 위치.(para pos로 위치 지정)
moveCurList	1	현재 리스트의 특정 위치.(para pos로 위치 지정)
moveTopOfFile	2	문서의 시작으로 이동.
moveBottomOfFile	3	문서의 끝으로 이동.
moveTopOfList	4	현재 리스트의 시작으로 이동
moveBottomOfList	5	현재 리스트의 끝으로 이동
moveStartOfPara	6	현재 위치한 문단의 시작으로 이동
moveEndOfPara	7	현재 위치한 문단의 끝으로 이동
moveStartOfWord	8	현재 위치한 단어의 시작으로 이동. (현재 리스트만을 대상으로 동작한다.)
moveEndOfWord	9	현재 위치한 단어의 끝으로 이동. (현재 리스트만을 대상으로 동작한다.)
moveNextPara	10	다음 문단의 시작으로 이동. (현재 리스트만을 대상으로 동작한다.)
movePrevPara	11	앞 문단의 끝으로 이동. (현재 리스트만을 대상으로 동작한다.)
moveNextPos	12	한 글자 앞으로 이동. (서브 리스트를 옮겨다닐 수 있다.)
movePrevPos	13	한 글자 뒤로 이동. (서브 리스트를 옮겨다닐 수 있다.)
moveNextPosEx	14	한 글자 앞으로 이동. (서브 리스트를 옮겨다닐 수 있다.) (머리말/꼬리말, 각주/미주, 글상자 포함.)
movePrevPosEx	15	한 글자 뒤로 이동. (서브 리스트를 옮겨다닐 수 있다.) (머리말/꼬리말, 각주/미주, 글상자 포함.)
moveNextChar	16	한 글자 앞으로 이동. (현재 리스트만을 대상으로 동작한다.)
movePrevChar	17	한 글자 뒤로 이동. (현재 리스트만을 대상으로 동작한다.)
moveNextWord	18	한 단어 앞으로 이동. (현재 리스트만을 대상으로 동작한다.)
movePrevWord	19	한 단어 뒤로 이동. (현재 리스트만을 대상으로 동작한다.)

ID	값	설명
moveNextLine	20	한 줄 위로 이동.
movePrevLine	21	한 줄 아래로 이동.
moveStartOfLine	22	현재 위치한 줄의 시작으로 이동.
moveEndOfLine	23	현재 위치한 줄의 끝으로 이동.
moveParentList	24	한 레벨 상위로 이동한다.
moveTopLevelList	25	탑레벨 리스트로 이동한다.
moveRootList	26	루트 리스트로 이동한다. 추가 설명 : 현재 루트 리스트에 위치해 있어 더 이상 상위 리스트가 없을 때는 위치 이동 없이 리턴한다. 이동한 후의 위치는 상위 리스트에서 서브리스트가 속한 컨트롤 코드가 위치한 곳이다. 위치 이동시 셀렉션은 무조건 풀린다.
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 moveCurList가 지정되었을 때만 사용된다. moveScrPos가 지정되었을 때는 문단번호가 아닌 스크린 좌표로 해석된다. (스크린 좌표 : LOWORD = x 좌표, HIWORD = y 좌표)

pos : 이동할 문단 중에서 문자의 위치. moveMain 또는 moveCurList가 지정되었을 때만 사용된다.

Remark

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

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

InitScan(Method)

Description

문서 검색을 위한 준비 작업을 한다.

Declaration

BOOL InitScan([unsigned int option], [unsigned int rang], [unsigned int spara], [unsigned int spos], [unsigned int epara], [unsigned int epos])

Parameters

option : 찾을 대상을 다음과 같은 옵션을 조합하여 지정할 수있다. 생략하면 모든 컨트롤을 찾을 대상으로 한다.

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

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

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

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

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

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

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

	Remark 문서의 검색 과정은 InitScan()으로 검색위한 준비 작업을 하고 GetText()를 호출하여 본문의 텍스트를 얻어온다. GetText()를 반복호출하면 연속하여 본문의 텍스트를 얻어올 수 있다. 검색이 끝나면 ReleaseScan()을 호출하여 관련 정보를 초기화 해 주면 된다.
ReleaseScan(Method)	Description InitScan()으로 설정된 정보를 초기화 한다. Declaration void ReleaseScan()
GetText(Method)	Description 문서 중에서 텍스트를 얻는다. Declaration long GetText(BSTR FAR* text) Parameters text : 텍스트 데이터가 돌아온다. 텍스트에서 탭은 'Wt'(0x9), 문단 바뀜은 CR/LF(0x0D/0x0A)로 표현되며, 이외 특수 코드는 포함되지 않는다. return : 다음과 같은 결과값을 리턴한다. 0 = 텍스트 정보 없음. 1 = 리스트의 끝. 2 = 일반 텍스트. 3 = 다음 문단. 4 = 제어문자 내부로 들어감. 5 = 제어 문자를 빠져 나옴. 101 = 초기화 안됨. (InitScan() 실패 또는 InitScan()를 실행하지 않음) 102 = 텍스트 변환 실패. Remark GetText()의 사용이 끝나면 ReleaseScan()을 반드시 호출하여 관련 정보를 초기화 해주어야 한다. 텍스트가 있는 문단으로캐럿을 이동 시키려면 moveScanPos를 주고 MovePos()를 호출하면 된다.
GetPos(Method)	Description 현재 캐럿의 위치 정보를 얻어온다. Declaration void GetPos(long FAR* list, long FAR* para, long FAR* pos) Parameters list : 캐럿이 위치한 문서 내 리스트 아이디. para : 캐럿이 위치한 문단 아이디. pos : 캐럿이 위치한 문단 내 글자 단위 위치. Remark 위의 리스트란, 문단과 컨트롤들이 연결된 한/글 문서 내 구조를 뜻한다. 리스트 아이디는 문서내 위치 정보 중 하나로서 SelectText에 넘겨줄 때 사용한다.
SetPos(Method)	Description 캐럿을 문서 내 특정 위치로 위치시킨다. Declaration void SetPos(long list, long para, long pos) Parameters list : 캐럿이 위치할 문서 내 리스트 아이디. para : 캐럿이 위치할 문단 아이디. pos : 캐럿이 위치할 문단 내 글자 단위 위치.
KeyIndicator(Method)	Description 상태바에 나타날 정보를 알아낸다.

<table>
<tr><td></td><td>
Declaration
BOOL KeyIndicator(long FAR* seccnt, long FAR* secno, long FAR* prnpageno, long FAR* colno, long FAR* line, long FAR* pos, short FAR* over, BSTR FAR* ctrlname)
Parameters
seccnt : 총 구역
secno : 현재 구역
prnpageno : 쪽
colno : 단
line : 줄
pos : 칸
over : (true:수정, false:삽입)
Remark
컨트롤 바깥쪽에서 상태바를 만들어서 상태바에 표시할 정보들의 내용을 알아낼 때 유용하다.
주의: 이 함수는 빠른 속도가 요구되므로 parameter로 포인터를 받는다. 따라서 포인터를 사용할 수 없는 언어에서는 사용이 불가능하다.
</td></tr>
<tr><td>GetTextFile(Method)</td><td>
Description
현재 열린 문서를 문자열로 넘겨준다.
Declaration
VARIANT GetTextFile(BSTR format, BSTR option)
Parameters
format : 파일의 형식
<table>
<tr><th>format</th><th>설명</th><th>비고</th></tr>
<tr><td>HWP</td><td>HWP native format</td><td>BASE64로 인코딩되어 있다. 저장된 내용을 다른 곳에서 보여줄 필요가 없다면 이 포맷을 사용하기를 권장합니다.</td></tr>
<tr><td>HWPML2X</td><td>HWP 형식과 호환</td><td>문서의 모든 정보를 유지</td></tr>
<tr><td>HTML</td><td>인터넷 문서 HTML 형식</td><td>한글 고유의 서식은 손실된다.</td></tr>
<tr><td>UNICODE</td><td>유니코드 텍스트</td><td>서식정보가 없는 텍스트만 저장</td></tr>
<tr><td>TEXT</td><td>일반 텍스트</td><td>유니코드에만 있는 정보(한자, 고어, 특수문자 등)는 모두 손실된다.</td></tr>
</table>
option
<table>
<tr><th>option</th><th>설명</th><th>비고</th></tr>
<tr><td>saveblock</td><td>선택된 블록만 저장</td><td>개체 선택 상태에서는 동작하지 않는다.</td></tr>
</table>
return : 저장된 텍스트로 파일을 문자열로 바꿔서 리턴한다.
Remark
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, Save나 SaveBlockAction을 사용하십시오. 이 함수 역시 내부적으로는 save나 SaveBlockAction을 호출하도록 되어있고 텍스트로 저장된 파일이 메모리에서 3~4번 복사되기 때문에 느리고, 메모리를 낭비합니다.
</td></tr>
<tr><td>SetTextFile(Method)</td><td>
Description
문서를 문자열로 지정한다.
Declaration
long SetTextFile(VARIANT data, BSTR format, BSTR option)
Parameters
data : 문자열로 변경된 text 파일
format : 파일의 형식
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><th>format</th><th>설명</th><th>비고</th></tr>
<tr><td>HWP</td><td>HWP native format</td><td>BASE64 로 인코딩되어 있어야 한다. 저장된 내용을 다른 곳에서 보여줄 필요가 없다면 이 포맷을 사용하기를 권장합니다.</td></tr>
<tr><td>HWPML2X</td><td>HWP 형식과 호환</td><td>문서의 모든 정보를 유지</td></tr>
<tr><td>HTML</td><td>인터넷 문서 HTML 형식</td><td>한글 고유의 서식은 손실된다.</td></tr>
<tr><td>UNICODE</td><td>유니코드 텍스트</td><td>서식정보가 없는 텍스트만 저장</td></tr>
<tr><td>TEXT</td><td>일반 텍스트</td><td>유니코드에만 있는 정보(한자, 고어, 특수문자 등)는 모두 손실된다.</td></tr>
</table>
option
<table>
<tr><th>option</th><th>설명</th><th>비고</th></tr>
<tr><td>insertfile</td><td>현재커서 이후에 삽입</td><td></td></tr>
</table>
Remark
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.
</td></tr>
<tr><td>CreatePageImage(Method)</td><td>
Description
지정한 페이지의 이미지를 파일로 생성한다.
Declaration
boolean CreatePageImage(BSTR path, [long pgno], [short resolution], [short depth], [BSTR format])
Parameters
path : 생성할 이미지 파일의 경로
pgno : 페이지 번호. 0부터 PageCount - 1까지. 생략하면 0이 사용된다.
resolution : 이미지 해상도. DPI 단위(96, 300, 1200)로 지정한다. 생략하면 96이 사용된다.
depth : 이미지 파일의 color depth(1, 4, 8, 24)를 지정한다.
format : 이미지 파일의 포맷. "bmp", "gif" 중의 하나. 생략하면 "bmp"가 사용된다.
</td></tr>
<tr><td>Run(Method)</td><td>
Description
액션을 실행한다.
Declaration
void Run(BSTR action)
Parameters
action : 액션 ID (별도 문서 참조)
</td></tr>
<tr><td>LockCommand(Method)</td><td>
Description
특정 액션이 실행되지 않도록 잠근다.
Declaration
void LockCommand(BSTR actionID, boolean lock)
Parameters
actionID : 액션 ID
lock : True이면 액션의 실행을 lock시키고, False이면 unlock시킨다.
</td></tr>
<tr><td>IsCommandLock(Method)</td><td>
Description
특정 액션이 잠금 상태인지 여부를 조사한다.
Declaration
boolean IsCommandLock(BSTR actionID)
Parameters
</td></tr>
</table>

<table>
<tr><td></td><td>actionID : 액션 ID</td></tr>
<tr><td>InsertPicture(Method)</td><td>Description
현재 캐럿의 위치에 그림을 삽입한다.
Declaration
Ctrl InsertPicture(BSTR path, [boolean embeded], [short sizeoption], [boolean reverse], [boolean watermark], [short effect], [long width], [long height])
Parameters
path : 삽입할 이미지 파일, URL 사용 가능
embeded : 이미지 파일을 문서내에 포함할지 여부 (True/False). 생략하면 True
sizeoption : 삽입할 그림의 크기를 지정하는 옵션
typedef enum {
// 이미지 원래의 크기로 삽입한다.
// width와 height를 지정할 필요없다.
realSize = 0,
// width와 height에 지정한 크기로 그림을 삽입한다.
specificSize = 1,
// 현재 캐럿이 표의 셀안에 있을 경우,
// 셀의 크기에 맞게 자동 조절하여 삽입한다.
// width는 셀의 width만큼,
// height는 셀의 height만큼 확대/축소된다.
// 캐럿이 셀안에 있지 않으면
// 이미지의 원래 크기대로 삽입된다.
cellSize = 2,
// 현재 캐럿이 표의 셀안에 있을 경우,
// 셀의 크기에 맞추어 원본 이미지의 가로 세로의
// 비율이 동일하게 확대/축소하여 삽입한다.
cellSizeWithSameRatio = 3
} HwpSizeOption ;
reverse : 이미지의 반전 유무 (True/False)
watermark : watermark효과 유무 (True/False)
effect : 그림 효과
typedef enum {
RealPicture = 0, // 실제 이미지 그대로
GrayScale = 1, // 그레이스케일
BlackWhite = 2 // 흑백효과
} HwpPictureEffect;
width : 그림의 가로 크기 지정. 단위는 mm
height : 그림의 높이 크기 지정. 단위는 mm</td></tr>
<tr><td>InsertBackgroundPicture(Method)</td><td>Description
배경이미지를 넣는다.
Declaration
VARIANT InsertBackgroundPicture(BSTR bordertype, BSTR path, [bool embedded], [long filloption], [bool watermark], [long effect], [long brightness], [long contrast]);
Parameters
bordertype : 배경 유형을 지정
<table><tr><th>bordertype</th><th>설명</th><th>비고</th></tr><tr><td>"SelectedCell"</td><td>현재 선택된 표의 셀의 배경을 변경한다.</td><td></td></tr><tr><td>"SelectedCellDelete"</td><td>현재 선택된 표의 셀의 배경을 지운다.</td><td>반드시 셀이 선택되어 있어야 함. 커서가 위치하는 것만으</td></tr></table></td></tr>
</table>

		로는 동작하지 않음.

path : 삽입할 이미지 파일, URL 사용 가능

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

filloption : 삽입할 그림의 크기를 지정하는 옵션

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

effect : 이미지효과

watermark : watermark효과 유무 (True/False) 이 옵션이 true이면 brightness 와 contrast 옵션이 무시된다.

effect

effect	설명	비고
0	원래 그림	설정하지 않았을 때 기본 값
1	그래이 스케일	
2	흑백으로	

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

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

Remark

CellBorderFill의 SetItem 중 FillAttr 의 SetItem FileName 에 이미지의 binary data를 지정해 줄 수가 없어서 만든 함수다. 기타 배경에 대한 다른 조정은 Action과 ParameterSet의 조합으로 가능하다.

CreateAction(Method)

Description

오브젝트를 생성한다.

Declaration

Action CreateAction(BSTR action)

Parameters

action : 액션 ID (별도 문서 참조)

Remark

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.

InsertCtrl(Method)

Description

현재 캐럿 위치에 컨트롤을 삽입한다.

Declaration

Ctrl InsertCtrl(BSTR ctrlid, [ParameterSet initparam])

Parameters

ctrlid : 삽입할 컨트롤의 ID

initparam : 컨트롤의 초기 속성. 생략하면 디폴트 속성으로 생성한다.

<table>
<tr><td></td><td>Remark
ctrlid에 지정할 수 있는 컨트롤 ID는 HwpCtrl.CtrlID가 리턴하는 ID와 동일하다. 자세한 것은 Ctrl 오브젝트 Properties인 CtrlID를 참조.
initparam에는 컨트롤의 초기 속성을 지정한다. 대부분의 컨트롤은 Ctrl.Properties와 동일한 포맷의 parameter set을 사용하지만, 컨트롤 생성시에는 다른 포맷을 사용하는 경우도 있다. 예를 들어 표의 경우 Ctrl.Properties에는 "Table" 셋을 사용하지만, 생성시 initparam에 지정하는 값은 "TableCreation" 셋이다.</td></tr>
<tr><td>DeleteCtrl(Method)</td><td>Description
문서내 컨트롤을 삭제한다.
Declaration
boolean DeleteCtrl(HwpCtrlCode ctrl)
Parameters
ctrl : 삭제할 문서내 컨트롤</td></tr>
<tr><td>GetMousePos(Method)</td><td>Description
마우스의 현재 위치를 구한다
Declaration
ParameterSet GetMousePos(long XrelTo, long YrelTo)
Parameters
XrelTo : X좌표계의 기준 위치
<table>
<tr><th>value</th><th>설명</th><th>비고</th></tr>
<tr><td>0</td><td>종이 기준</td><td>종이 기준으로 좌표를 가져온다.</td></tr>
<tr><td>1</td><td>쪽 기준</td><td>쪽 기준으로 좌표를 가져온다.</td></tr>
</table>
YrelTo : Y좌표계의 기준 위치
<table>
<tr><th>value</th><th>설명</th><th>비고</th></tr>
<tr><td>0</td><td>종이 기준</td><td>종이 기준으로 좌표를 가져온다.</td></tr>
<tr><td>1</td><td>쪽 기준</td><td>쪽 기준으로 좌표를 가져온다.</td></tr>
</table>
Return : "MousePos" ParameterSet 이 리턴된다.
■ [Set ID] MousePos
<table>
<tr><th>Item ID</th><th>Type</th><th>Description</th></tr>
<tr><td>XRelTo</td><td>PIT_UI4</td><td>가로 상대적 기준
0 : 종이
1 : 쪽</td></tr>
<tr><td>YRelTo</td><td>PIT_UI4</td><td>세로 상대적 기준
0 : 종이
1 : 쪽</td></tr>
<tr><td>Page</td><td>PIT_UI4</td><td>페이지 번호 (0 based)</td></tr>
<tr><td>X</td><td>PIT_I4</td><td>가로 클릭한 위치 (HWPUNIT)</td></tr>
<tr><td>Y</td><td>PIT_I4</td><td>세로 클릭한 위치 (HWPUNIT)</td></tr>
</table>
Remark
단위가 HWPUNIT임을 주의하십시오.</td></tr>
<tr><td>Clear(Method)</td><td>Description
현재 편집중인 문서의 내용을 닫고 빈문서 편집 상태로 돌아간다.
Declaration
void Clear([HwpSaveOption option])
Parameters
option : 편집중인 문서의 내용에 대한 처리 방법. 생략하면 hwpAskSave가 선택된다. (HwpSaveOption : short type)
<table>
<tr><th>ID</th><th>값</th><th>설명</th></tr>
<tr><td>hwpAskSave</td><td>0</td><td>문서의 내용이 변경되었을 때 사용자에게 저장할지 묻는 대화상자를 띄운다.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>hwpDiscard</td><td>1</td><td>문서 내용을 버린다.</td></tr>
<tr><td>hwpSaveIfDirty</td><td>2</td><td>문서가 변경된 경우 저장 한다.</td></tr>
<tr><td>hwpSave</td><td>3</td><td>무조건 저장한다.</td></tr>
</table>
Remark
format, arg에 대해서는 Open 참조. hwpSaveIfDirty, hwpSave가 지정된 경우 현재 문서 경로가 지정되어 있지 않으면 "새이름으로 저장" 대화상자를 띄워 사용자에게 경로를 묻는다.
</td></tr>
<tr><td>RegisterModule(Method)</td><td>Remark
보안 모듈 관련 문서 참조</td></tr>
<tr><td>Application(Property)</td><td>Description
최상위 오브젝트 (IHwpObject interface)를 얻는다.</td></tr>
<tr><td>XHwpMessageBox(Property)</td><td>Description
메시지 박스를 제어하는 XHwpMessageBox Object를 얻는다.</td></tr>
<tr><td>XHwpDocuments(Property)</td><td>Description
도큐먼트를 관리하는 XHwpDocuments Object를 얻는다.</td></tr>
<tr><td>XHwpWindows(Property)</td><td>Description
윈도우를 관리하는 XHwpWindows Object를 얻는다.</td></tr>
<tr><td>HParameterSet(Property)</td><td>Description
파라메터셋 오브젝트를 관리하는 HParameterSet Object를 얻는다.</td></tr>
<tr><td>HAction(Property)</td><td>Description
Action을 제어하는 HAction Object를 얻는다.</td></tr>
<tr><td>XHwpODBC(Property)</td><td>Description
ODBC로 제어할 수 있는 Object를 얻는다.</td></tr>
<tr><td>Version(Property)</td><td>Description
한글 과 한글 OCX의 버젼 정보를 구한다.읽기 전용.
byte 3 = 한글의 major version.
byte 2 = 한글의 minor version.
byte 1 = 한글 OCX의 major version.
byte 0 = 한글 OCX의 minor version.</td></tr>
<tr><td>ExportStyle(Method)</td><td>Description
HStyleTemplate 파라메터셋 오브젝트에 지정된 스타일을 Export한다.
Declaration
BOOL ExportStyle(LPDISPATCH param)
Parameters
param : HStyleTemplate</td></tr>
<tr><td>ImportStyle(Method)</td><td>Description
HStyleTemplate 파라메터셋 오브젝트에 지정된 스타일을 Import한다.
Declaration
BOOL ImportStyle(LPDISPATCH param)
Parameters
param : HStyleTemplate</td></tr>
<tr><td>FindCtrl(Method)</td><td>Description
현재 캐럿의 위치에서 Ctrl을 찾는다.
Declaration
BSTR FindCtrl()
Parameters
return : 컨트롤을 찾은 경우 CtrlID를 return 한다.</td></tr>
<tr><td>UnSelectCtrl(Method)</td><td>Description
현재 Select된 Ctrl의 Selection을 해제한다.</td></tr>
</table>

IHwpObjectEvents(_DIHwpObjectEvents) : 한/글에서부터 발생되는 이벤트

Item Name	Return	Description
Quit	없음	한/글을 종료할 때 발생
CreateXHwpWindow	없음	한/글에서 새 문서 창을 열었을 때 발생
CloseXHwpWindow	없음	한/글에서 문서 창을 닫았을 때 발생
NewDocument	long	새 문서를 생성할 경우 발생(Document ID를 반환)
DocumentBeforeClose	long	문서를 닫기 직전에 발생(Document ID를 반환)
DocumentBeforeOpen	long	문서를 열기 직전에 발생(Document ID를 반환)
DocumentAfterOpen	long	문서를 열고 난 후에 발생(Document ID를 반환)
DocumentBeforeSave	long	문서를 저장하기 직전에 발생(Document ID를 반환)
DocumentAfterSave	long	문서를 저장한 후에 발생(Document ID를 반환)

IXHwpDocuments: IXHwpDocument(도큐먼트) 오브젝트를 관리하는 오브젝트 (Document를 관리하는 Collection 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject)
Item(Property)	Description 지정한 원소의 도큐먼트 오브젝트를 얻어온다. Declaration LPDISPATCH Item(long index) Parameters index : 원소의 인덱스 return : 원소의 값(IXHwpDocument)
Count(Property)	Description 원소의 총개수
Active_XHwpDocument(Property)	Description 현재 활성화 상태인 도큐먼트 Object를 얻어온다.(IXHwpDocument)
Add(Method)	Description 도큐먼트 오브젝트를 추가한다. Declaration LPDISPATCH Add(BOOL isTab) Parameters isTab : TRUE = 새탭으로 열리는 도큐먼트, FALSE = 새창으로 열리는 도큐먼트 return : 열리게 되는 도큐먼트(IXHwpDocument)
Close(Method)	Description 관리하고 있는 도큐먼트 오브젝트를 삭제한다. Declaration void Close(BOOL isDirty) Parameters isDirty : TRUE이면 변경된 문서는 닫지 않는다. FALSE이면 변경된 문서도 닫는다.
FindItem(Method)	Description 도큐먼트 아이디로 지정된 도큐먼트 오브젝트를 얻는다. Declaration

	LPDISPATCH FindItem(long Docid) **Parameters** Docid : 도큐먼트의 고유 ID return : 도큐먼트 ID에 해당하는 도큐먼트 오브젝트(IXHwpDocument)

IXHwpDocument: 도큐먼트 오브젝트 (Document 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	**Description** 최상위 오브젝트를 얻어옴(IHwpObject - 읽기 전용)
Path(Property)	**Description** 도큐먼트의 Path를 얻어옴(읽기 전용)
FullName(Property)	**Description** 도큐먼트의 전체 경로를 얻어옴(읽기 전용)
EditMode(Property)	**Description** 도큐먼트의 에디트 모드를 설정하거나/얻어옴
Modified(Property)	**Description** 도큐먼트의 변경 여부를 설정하거나/얻어옴
Format(Property)	**Description** 도큐먼트의 저장된 포맷을 얻어옴(읽기 전용)
Password(Property)	**Description** 도큐먼트의 패스워드를 설정(쓰기 전용)
XHwpSummaryInfo(Property)	**Description** IXHwpSummaryInfo 문서 요약 정보 오브젝트 (읽기 전용)
XHwpDocumentInfo(Property)	**Description** IXHwpDocumentInfo 문서 정보 오브젝트 (읽기 전용)
XHwpPrint(Property)	**Description** IXHwpPrint 프린트 오브젝트 (읽기 전용)
XHwpRange(Property)	**Description** IXHwpRange Range 오브젝트 (읽기 전용)
XHwpFind(Property)	**Description** IXHwpFind 찾기 오브젝트 (읽기 전용)
XHwpSelection(Property)	**Description** IXHwpSelection 블록 선택 오브젝트 (읽기 전용)
XHwpFormPushButtons(Property)	**Description** IXHwpFormPushButtons 양식개체 푸쉬버튼을 관리하는 오브젝트(읽기 전용)
XHwpFormCheckButtons(Property)	**Description** IXHwpFormCheckButtons 양식개체 체크박스를 관리하는 오브젝트(읽기 전용)
XHwpFormRadioButtons(Property)	**Description** IXHwpFormRadioButtons 양식개체 라디오버튼을 관리하는 오브젝트(읽기 전용)
XHwpFormComboBoxs(Property)	**Description**

<table>
<tr><td></td><td>IXHwpFormComboBoxs 양식개체 콤보박스를 관리하는 오브젝트(읽기 전용)</td></tr>
<tr><td>XHwpFormEdits(Property)</td><td>Description
IXHwpFormEdits 양식개체 에디트를 관리하는 오브젝트(읽기 전용)</td></tr>
<tr><td>XHwpCharacterShape(Property)</td><td>Description
IXHwpCharacterShape 글자 모양 속성 오브젝트(읽기 전용)</td></tr>
<tr><td>XHwpParagraphShape(Property)</td><td>Description
IXHwpParagraphShape 문단 모양 속성 오브젝트(읽기 전용)</td></tr>
<tr><td>XHwpSendMail(Property)</td><td>Description
IXHwpSendMail 메일 보내기 오브젝트 (읽기 전용)</td></tr>
<tr><td>DocumentID(Property)</td><td>Description
도큐먼트의 고유 ID(읽기 전용)</td></tr>
<tr><td>Close(Method)</td><td>Description
문서를 닫는다.
Declaration
BOOL Close(BOOL isDirty)
Parameters
isDirty : TRUE = 문서 내용이 변경된 상태면 문서를 닫지 않는다./ FALSE = 문서 내용이 변경되었어도 강제로 문서를 닫는다.</td></tr>
<tr><td>Save(Method)</td><td>Description
문서를 저장한다.
Declaration
BOOL Save(Variant save_if_dirty)
Parameters
save_if_dirty : True를 지정하면 문서가 변경된 경우에만 저장한다. False를 지정하면 변경 여부에 관계없이 무조건 저장한다. 생략하면 True가 지정된다.
Remark
문서의 경로가 지정되어 있지 않으면 "새이름으로 저장" 대화상자가 떠서 사용자에게 경로를 묻는다.</td></tr>
<tr><td>SaveAs(Method)</td><td>Description
문서를 지정한 이름으로 저장한다.
Declaration
BOOL Save(BSTR path, VARIANT format, VARIANT arg)
Parameters
path : 문서 파일의 경로
format : 문서 형식. 별도 설명 참조. 생략하면 "HWP"가 지정된다.
arg : 세부 옵션. 의미는 format에 지정한 파일 형식에 따라 다르다. 생략하면 빈 문자열이 지정된다.
Remark
format, arg의 일반적인 개념에 대해서는 Open 참조.
"HWP" 형식으로 파일 저장시 arg에 지정할 수 있는 옵션은 다음과 같다.
<table>
<tr><th>함수</th><th>인자 타입</th><th>기본 값 (Default)</th><th>설명</th></tr>
<tr><td>lock</td><td>boolean</td><td>TRUE</td><td>저장한 후 해당 파일을 계속 오픈한 상태로 lock을 걸지 여부</td></tr>
<tr><td>backup</td><td>boolean</td><td>FALSE</td><td>백업 파일 생성 여부</td></tr>
<tr><td>compress</td><td>boolean</td><td>TRUE</td><td>압축 여부</td></tr>
<tr><td>fullsave</td><td>boolean</td><td>FALSE</td><td>스토리지 파일을 완전히 새로 생성하여 저장</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>prvimage</td><td>int</td><td>2</td><td>미리보기 이미지
(0=off, 1=BMP, 2=GIF)</td></tr>
<tr><td>prvtext</td><td>int</td><td>1</td><td>미리보기 텍스트(0=off, 1=0)</td></tr>
</table>
</td></tr>
<tr><td>Undo(Method)</td><td>Description
문서에 기록된 Undo Item을 실행한다.
Declaration
BOOL Undo(long count)
Parameters
count : 아이템의 count까지 Undo한다.</td></tr>
<tr><td>Redo(Method)</td><td>Description
문서에 기록된 Redo Item을 실행한다.
Declaration
BOOL Redo(long count)
Parameters
count : 아이템의 count까지 Redo한다.</td></tr>
<tr><td>Open(Method)</td><td>Description
문서 파일을 연다.
Declaration
Open(BSTR path, BSTR format, BSTR arg)
Parameters
path : 문서 파일의 경로(URL 사용 가능)
format : 문서 형식. 별도 설명 참조. 빈 문자열을 지정하면 자동으로 인식한다. 생략하면 빈 문자열이 지정된다.
arg : 세부 옵션. 의미는 format에 지정한 파일 형식에 따라 다르다. 생략하면 빈 문자열이 지정된다.
Remark
format에 지정할 수 있는 문서 형식은 현재 시스템에 설치된 문서 필터(*.dft)의 종류에 따라 달라진다. 일반적으로 설치되는 형식들에는 다음과 같은 종류가 있다.
<table>
<tr><td>HWP</td><td>워디안 native format</td></tr>
<tr><td>HWP30</td><td>한글 3.X/96/97</td></tr>
<tr><td>HTML</td><td>인터넷 문서</td></tr>
<tr><td>TEXT</td><td>아스키 텍스트 문서</td></tr>
<tr><td>UNICODE</td><td>유니코드 텍스트 문서</td></tr>
<tr><td>HWP20</td><td>한글 2.0</td></tr>
<tr><td>HWP21</td><td>한글 2.1/2.5</td></tr>
<tr><td>HWP15</td><td>한글 1.X</td></tr>
<tr><td>HWPML1X</td><td>HWPML 1.X 문서 (Open만 가능)</td></tr>
<tr><td>HWPML2X</td><td>HWPML 2.X 문서 (Open / SaveAs 가능)</td></tr>
<tr><td>RTF</td><td>서식있는 텍스트 문서</td></tr>
<tr><td>DBF</td><td>DBASE II/III 문서</td></tr>
<tr><td>HUNMIN</td><td>훈민정음 3.0/2000</td></tr>
<tr><td>MSWORD</td><td>마이크로소프트 워드 문서</td></tr>
<tr><td>HANA</td><td>하나워드 문서</td></tr>
<tr><td>ARIRANG</td><td>아리랑 문서</td></tr>
<tr><td>ICHITARO</td><td>一太郎 문서 (일본 워드프로세서)</td></tr>
<tr><td>WPS</td><td>WPS 문서</td></tr>
<tr><td>DOCIMG</td><td>인터넷 프레젠테이션 문서(SaveAs만 가능)</td></tr>
<tr><td>SWF</td><td>Macromedia Flash 문서(SaveAs만 가능)</td></tr>
</table>
arg에 지정할 수 있는 옵션의 의미는 필터가 정의하기에 따라 다르지만, 신텍스는 다음과 같이 공통된 형식을 사용한다.
key:value;key:value;...</td></tr>
</table>

<table>
<tr><td></td><td>* 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
<table>
<tr><td>"모든 파일"</td><td>setcurdir(boolean, FALSE) : 로드한 후 해당 파일이 존재하는 폴더로 현재 위치를 변경한다. hyperlink 정보가 상대적인 위치로 되어 있을 때 유용하다.</td></tr>
<tr><td>"HWP"</td><td>- lock (boolean, TRUE) : 로드한 후 해당 파일을 계속 오픈한 상태로 lock을 걸지 여부
- notext (boolean, FALSE) : 텍스트 내용을 읽지 않고 헤더 정보만 읽을지 여부. (스타일 로드 등에 사용)
- template (boolean, FALSE): 새로운 문서를 생성하기 위해 템플릿 파일을 오픈한다. 이 옵션이 주어지면 lock은 무조건 FALSE로 처리된다.
- suspendpassword (boolean, FALSE): TRUE로 지정하면, 암호가 있는 파일일 경우 암호를 묻지 않고 무조건 읽기에 실패한 것으로 처리한다.
- forceopen (boolean, FALSE): TRUE로 지정하면, 읽기 전용으로 읽어야 하는 경우 대화상자를 띄우지 않는다.</td></tr>
<tr><td>"HTML"</td><td>- code(string, codepage) : 문서 변환시 사용되는 코드 페이지를 지정할 수 있이며 code키가 존재할 경우 필터 사용시 사용자 다이얼로그를 띄우지 않는다.
- textunit(boolean, pixel) : Export될 Text의 크기의 단위 결정(pixel, point, mili 지정 가능.)
- formatunit(boolean, pixel) : Export될 문서 포맷 관련 (마진, Object 크기 등) 단위 결정. pixel, point, mili 지정 가능</td></tr>
<tr><td>"DOCIMG"</td><td>- asimg(boolean, FALSE) : 저장할 때 페이지를 image로 저장
- ashtml(boolean, FALSE) : 저장할 때 페이지를 html로 저장

[codepage 종류]
ks : 한글 KS 완성형 | kssm : 한글 조합형 | sjis : 일본 | utf8 : UTF8 | unicode : 유니코드 | gb : 중국 간체 | big5 : 중국 번체 | acp : Active Codepage 현재 시스템의 코드 페이지</td></tr>
<tr><td>"TEXT"</td><td>- code(string, codepage): 문서 변환시 사용되는 코드 페이지를 지정할 수 있이며 code키가 존재할 경우 필터 사용시 사용자 다이얼로그를 띄우지 않는다.

[codepage 종류]
- ks : 한글 KS 완성형 | kssm : 한글 조합형 | sjis : 일본 | gb : 중국 간체 | big5 : 중국 번체 | acp : Active Codepage 현재 시스템의 코드 페이지</td></tr>
</table></td></tr>
<tr><td>SendBrowser(Method)</td><td>Description
문서를 브라우저로 내보내기 기능
Declaration
BOOL SendBrowser(void)</td></tr>
<tr><td>SetActive_XHwpDocument(Method)</td><td>Description</td></tr>
</table>

<table>
<tr><td></td><td>문서를 활성화 상태로 하기
Declaration
void SetActive_XHwpDocument(void)</td></tr>
<tr><td>Clear(Method)</td><td>Description
문서의 내용을 닫고 빈문서 편집 상태로 돌아간다.
Declaration
void Clear(Variant option)
Parameters
option : 편집중인 문서의 내용에 대한 처리 방법. 생략하면 hwpAskSave가 선택된다.
<table>
<tr><th>ID</th><th>값</th><th>설명</th></tr>
<tr><td>hwpAskSave</td><td>0</td><td>문서의 내용이 변경되었을 때 사용자에게 저장할지 묻는 대화상자를 띄운다.</td></tr>
<tr><td>hwpDiscard</td><td>1</td><td>문서 내용을 버린다.</td></tr>
<tr><td>hwpSaveIfDirty</td><td>2</td><td>문서가 변경된 경우 저장 한다.</td></tr>
<tr><td>hwpSave</td><td>3</td><td>무조건 저장한다.</td></tr>
</table>
Remark
format, arg에 대해서는 Open 참조. hwpSaveIfDirty, hwpSave가 지정된 경우 현재 문서 경로가 지정되어 있지 않으면 "새이름으로 저장" 대화상자를 띄워 사용자에게 경로를 묻는다.</td></tr>
</table>

IXHwpFormPushButtons: IXHwpFormPushButton 오브젝트를 관리하는 오브젝트 (양식개체 PushButton을 관리하는 Collection 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject)
Item(Property)	Description 지정한 원소의 양식개체 PushButton 오브젝트를 얻어온다. Declaration LPDISPATCH Item(long index) Parameters index : 원소의 인덱스 return : 원소의 값(IXHwpFormPushButton)
Count(Property)	Description 원소의 총개수
ItemFromName(Property)	Description 양식 개체 PushButton의 이름으로 원소를 찾는다. Declaration LPDISPATCH ItemFormName(BSTR name) Parameters name : 양식 개체 PushButton의 이름 return : 양식 개체 PushButton Object(IXHwpFormPushButton)

IXHwpFormPushButton: 양식 개체 푸쉬 버튼 오브젝트 (PushButton 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description

	최상위 오브젝트를 얻어옴(IHwpObject)
Caption(Property)	캡션
Name(Property)	이름
ForeColor(Property)	전경색
BackColor(Property)	배경색
GroupName(Property)	그룹 이름
TabStop(Property)	Tab Stop
TabOrder(Property)	탭 순서
Width(Property)	너비
Height(Property)	높이
Left(Property)	왼쪽 좌표
Top(Property)	위쪽 좌표
CharShapeID(Property)	그리기 개체 Contorl ID
FollowContext(Property)	주위의 글자 속성을 따를지의 여부
AutoSize(Property)	글자 크기에 맞게 개체 크기가 바뀜
BorderType(Property)	테두리 타입
DrawFrame(Property)	틀을 그릴지의 여부
Enabled(Property)	활성, 비활성의 여부
WordWrap(Property)	자동 줄 바꿈

IXHwpFormCheckButtons: IXHwpFormCheckButton 오브젝트를 관리하는 오브젝트 (양식개체 CheckButton을 관리하는 Collection 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject)
Item(Property)	Description 지정한 원소의 양식개체 CheckButton 오브젝트를 얻어온다. Declaration LPDISPATCH Item(long index) Parameters index : 원소의 인덱스 return : 원소의 값(IXHwpFormCheckButton)
Count(Property)	Description 원소의 총개수
ItemFromName(Property)	Description 양식 개체 CheckButton의 이름으로 원소를 찾는다. Declaration LPDISPATCH ItemFormName(BSTR name) Parameters name : 양식 개체 CheckButton의 이름

	return : 양식 개체 CheckButton Object(IXHwpFormCheckButton)

IXHwpFormCheckButton:양식 개체 체크 버튼 오브젝트 (CheckButton 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject)
Caption(Property)	캡션
Name(Property)	이름
ForeColor(Property)	전경색
BackColor(Property)	배경색
GroupName(Property)	그룹 이름
TabStop(Property)	Tap Stop
TabOrder(Property)	탭 순서
Width(Property)	너비
Height(Property)	높이
Left(Property)	왼쪽 좌표
Top(Property)	위쪽 좌표
CharShapeID(Property)	그리기 개체 Control ID
FollowContext(Property)	주위의 글자 속성을 따를지의 여부
AutoSize(Property)	글자 크기에 맞게 개체 크기가 바뀜
BackStyle(Property)	배경 투명도
BorderType(Property)	테두리 타입
DrawFrame(Property)	틀을 그릴지의 여부
Enabled(Property)	활성, 비활성의 여부
TriState(Property)	체크 상태 옵션
Value(Property)	값
WordWrap(Property)	자동 줄 바꿈

IXHwpFormRadioButtons: IXHwpFormRadioButton 오브젝트를 관리하는 오브젝트 (양식개체 RadioButton을 관리하는 Collection 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject)
Item(Property)	Description 지정한 원소의 양식개체 RadioButton 오브젝트를 얻어온다. Declaration

	LPDISPATCH Item(long index) Parameters index : 원소의 인덱스 return : 원소의 값(IXHwpFormRadioButton)
Count(Property)	Description 원소의 총개수
ItemFromName(Property)	Description 양식 개체 RadioButton의 이름으로 원소를 찾는다. Declaration LPDISPATCH ItemFormName(BSTR name) Parameters name : 양식 개체 RadioButton의 이름 return : 양식 개체 RadioButton Object(IXHwpFormCheckButton)

HwpFormRadioButton: 양식 개체 라디오 버튼 오브젝트 (RadioButton 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject)
Name(Property)	이름
ForeColor(Property)	전경색
BackColor(Property)	배경색
GroupName(Property)	그룹 이름
TabStop(Property)	Tap Stop
TabOrder(Property)	탭 순서
Width(Property)	너비
Height(Property)	높이
Left(Property)	왼쪽 좌표
Top(Property)	위쪽 좌표
CharShapeID(Property)	그리기 개체 Control ID
FollowContext(Property)	주위의 글자 속성을 따를지의 여부
AutoSize(Property)	글자 크기에 맞게 개체 크기가 바뀜
BackStyle(Property)	배경 투명도
BorderType(Property)	테두리 타입
Caption(Property)	캡션
DrawFrame(Property)	틀을 그릴지의 여부
Enabled(Property)	활성, 비활성의 여부
RadioGroupName(Property)	라디오 그룹 이름
TriState(Property)	체크 상태 옵션

Value(Property)	값
WordWrap(Property)	자동 줄 바꿈

IXHwpFormComboBoxs: IXHwpFormComboBox 오브젝트를 관리하는 오브젝트 (양식개체 ComboBox을 관리하는 Collection 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject)
Item(Property)	Description 지정한 원소의 양식개체 ComboBox 오브젝트를 얻어온다. Declaration LPDISPATCH Item(long index) Parameters index : 원소의 인덱스 return : 원소의 값(IXHwpFormComboBox)
Count(Property)	Description 원소의 총개수
ItemFromName(Property)	Description 양식 개체 ComboBox의 이름으로 원소를 찾는다. Declaration LPDISPATCH ItemFormName(BSTR name) Parameters name : 양식 개체 ComboBox의 이름 return : 양식 개체 ComboBox Object(IXHwpFormComboBox)

IXHwpFormComboBox: 양식 개체 콤보 박스 오브젝트 (ComboBox 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject)
Name(Property)	이름
ForeColor(Property)	전경색
BackColor(Property)	배경색
GroupName(Property)	그룹 이름
TabStop(Property)	Tap Stop
TabOrder(Property)	탭 순서
Width(Property)	너비
Height(Property)	높이
Left(Property)	왼쪽 좌표
Top(Property)	위쪽 좌표
CharShapeID(Property)	그리기 개체 Control ID

FollowContext(Property)	주위의 글자 속성을 따를지의 여부
AutoSize(Property)	글자 크기에 맞게 개체 크기가 바뀜
BorderType(Property)	테두리 타입
DrawFrame(Property)	틀을 그릴지의 여부
EditEnable(Property)	에디트 상태 활성, 비활성의 여부
Enabled(Property)	활성, 비활성의 여부
ListBoxRows(Property)	리스트 박스 열
ListBoxWidth(Property)	리스트 박스 너비
Text(Property)	선택된 값
WordWrap(Property)	자동 줄 바꿈
Count(Property)	아이템 개수
CurSel(Property)	현재 선택된 인덱스
LBText(Property)	인덱스에 해당하는 값
InsertString(Method)	Description 양식 개체 ComboBox에 문자열을 채워넣는다. Declaration void InsertString(BSTR itemvalue, long index) Parameters itemvalue : 리스트에 채워넣기 위한 값 index : 리스트의 특정 위치
DeleteString(Method)	Description 리스트의 지정한 위치에 있는 값을 지운다. Declaration void DeleteString(unsigned long index) Parameters index : 리스트의 특정 위치
FindStringExact(Method)	Description 지정한 문자열이 리스트에 있는지를 찾는다. Declaration long FindStringExact(long index, BSTR itemvalue) Parameters index : 리스트의 특정 위치 itemvalue : 리스트에서 찾기 위한 값
ResetContent(Method)	Description 리스트 내용을 초기화 한다. Declaration void ResetContent(void)

IXHwpFormEdits: IXHwpFormEdit 오브젝트를 관리하는 오브젝트 (양식개체 Edit을 관리하는 Collection 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject)

Item(Property)	Description 지정한 원소의 양식개체 Edit 오브젝트를 얻어온다. Declaration LPDISPATCH Item(long index) Parameters index : 원소의 인덱스 return : 원소의 값(IXHwpFormEdit)
Count(Property)	Description 원소의 총개수
ItemFromName(Property)	Description 양식 개체 Edit의 이름으로 원소를 찾는다. Declaration LPDISPATCH ItemFormName(BSTR name) Parameters name : 양식 개체 Edit의 이름 return : 양식 개체 Edit Object(IXHwpFormEdit)

IXHwpFormEdit: 양식 개체 에디트 오브젝트 (Edit 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject)
Name(Property)	이름
ForeColor(Property)	전경색
BackColor(Property)	배경색
GroupName(Property)	그룹 이름
TabStop(Property)	Tap Stop
TabOrder(Property)	탭 순서
Width(Property)	너비
Heigh(Property)	높이
Left(Property)	왼쪽 좌표
Top(Property)	위쪽 좌표
CharShapeID(Property)	그리기 개체 Control ID
FollowContext(Property)	주위의 글자 속성을 따를지의 여부
AutoSize(Property)	글자 크기에 맞게 개체 크기가 바뀜
BorderType(Property)	테두리 타입
DrawFrame(Property)	틀을 그릴지의 여부
Enabled(Property)	활성, 비활성의 여부
MaxLength(Property)	에디트 가능한 총 길이
MultiLine(Property)	멀티 라인 지원

Number(Property)	숫자만 입력 가능
PasswordChar(Property)	패스워드 표시에 사용할 글자
ReadOnly(Property)	읽기만 가능
ScrollBars(Property)	스크롤바 표시
TabKeyBehavior(Property)	탭 키를 눌렀을 때 반응
Text(Property)	에디트 텍스트
WordWrap(Property)	자동 줄 바꿈
LineCount(Property)	에디트 라인 줄 수

IXHwpWindows: IXHwpWindow오브젝트를 관리하는 오브젝트 (Window를 관리하는 Collection 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject - 읽기 전용)
Active_XHwpWindow(Property)	Description 현재 활성화 상태인 윈도우 Object를 얻어온다.(IXHwpWindow)
Item(Property)	Description 지정한 원소의 윈도우 오브젝트를 얻어온다. Declaration LPDISPATCH Item(long index) Parameters index : 원소의 인덱스 return : 원소의 값(IXHwpWindow)
Count(Property)	Description 원소의 총개수
Add(Method)	Description 윈도우를 하나 추가한다.(새창으로 열기 기능과 동일) Declaration LPDISPATCH Add(void) Parameters return : 추가된 윈도우 오브젝트(IXHwpWindow)
Close(Method)	Description 윈도우를 모두 닫는다. Declaration BOOL Close(BOOL isDirty) Parameters isDirty : TURE이면 문서 내용이 변경된 경우 닫지 않는다. FALSE이면 문서 내용이 변경된 것과 상관없이 강제로 닫는다.

IXHwpWindow: 윈도우 오브젝트 (Window 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject - 읽기 전용)

XHwpDocuments(Property)	Description 도큐먼트 관리 오브젝트를 얻어옴(IXHwpDocuments - 읽기 전용)
XHwpTabs(Property)	Description 탭 관리 오브젝트를 얻어옴(IXHwpTabs - 읽기 전용)
Left(Property)	Description 윈도우의 좌측 위치 좌표를 설정/얻음
Top(Property)	Description 윈도우의 맨위 위치 좌표를 설정/얻음
Width(Property)	Description 윈도우의 너비를 설정/얻음
Height(Property)	Description 윈도우의 높이를 설정/얻음
Visible(Property)	Description 윈도우 보이기/보이지 않기 설정/얻음
Close(Method)	Description 윈도우를 닫음 Declaration void Close(BOOL isDirty) Parameters isDirty : TURE이면 문서 내용이 변경된 경우 닫지 않는다. FALSE이면 문서 내용이 변경된 것과 상관없이 강제로 닫는다.

IXHwpTabs: IXHwpTab오브젝트를 관리하는 오브젝트 (Tab를 관리하는 Collection 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Count(Property)	Description 윈도우에 열려있는 탭의 개수(읽기 전용)
Item(Property)	Description 지정한 원소의 탭 오브젝트를 얻어온다. Declaration LPDISPATCH Item(long index) Parameters index : 원소의 인덱스 return : 원소의 값(IXHwpTab)
Add(Method)	Description 지정한 원소의 탭 오브젝트를 추가한다.(문서를 새 탭으로 열기) Declaration LPDISPATCH Add(void) Parameters return : 추가된 원소의 값(IXHwpTab)
Close(Method)	Description 탭을 모두 닫는다. Declaration BOOL Close(BOOL isDirty) Parameters isDirty : TURE이면 문서 내용이 변경된 경우 닫지 않는다. FALSE이면 문서 내용이 변경된 것과 상관없이 강제로 닫는다.

IXHwpTab: 탭 오브젝트
(Tab 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Close(Method)	Description 탭을 닫는다. Declaration void Close(BOOL isDirty) Parameters isDirty : TURE이면 문서 내용이 변경된 경우 닫지 않는다. FALSE이면 문서 내용이 변경된 것과 상관없이 강제로 닫는다.

HSet : ParameterSet Item 데이터들의 집합

예) HParameterSet.HSecDef.HSet - HSecDef에 저장된 모든 Item 데이터의 집합을 얻음

Item Name	Description
SetItem(Method)	Description 지정된 아이템이름을 가진 데이터에 VARIANT값을 대입한다. Declaration SetItem(BSTR itemid, VRIANT value) Parameters itemid : 아이템 이름 value : 데이터

HAction : 한/글에서 특정 기능을 수행하기 위한 액션 오브젝트

예) 아래와 같은 형태로 사용하는 것이 HAction을 사용하는 올바른 사용법이다.
HAction.GetDefault("Print", HParameterSet.HPrint.HSet); // 액션 초기화
HParameterSet.HPrint.NumCopy = 3; //인쇄 매수를 3장으로 지정
HAction.Execute("Print", HParameterSet.HPrint.HSet); // 액션 수행

Item Name	Description
GetDefault(Method)	Description 지정된 액션이름과 LPDISPATCH로 HSet을 받아 액션을 초기화 한다. Declaration BOOL GetDefault(BSTR actname, LPDISPATCH object) Parameters actname : 액션 이름 object : HSet Object
Execute(Method)	Description 지정된 액션이름과 LPDISPATCH로 HSet을 받아 액션을 수행 한다. Declaration BOOL Execute(BSTR actname, LPDISPATCH object) Parameters actname : 액션 이름 object : HSet Object
PopupDialog(Method)	Description 지정된 액션이름과 LPDISPATCH로 HSet을 받아 액션의 다이얼로그를 생성한다. Declaration BOOL PopupDialog(BSTR actname, LPDISPATCH object) Parameters

	actname : 액션 이름 object : HSet Object
Run(Method)	Description GetDefault, PopupDialog, Execute를 동시에 수행하도록 한다. Declaration BOOL Run(BSTR actname) Parameters actname : 액션 이름

HArray : ParameterArray의 데이터 집합

Item Name	Description
Count(Property)	Description Array에 크기를 지정하거나 얻을 수 있다.
Item(Property)	Description Array에 지정된 index에 해당하는 값을 VARIANT로 지정하거나 얻을 수 있다.

IXHwpMessageBox: OLE Automation standard object (메세지박스 MessageBox 개체 - 사용 편이를 위해 제공됨)

Item Name	Description
Application(Property)	Description 최상위 오브젝트를 얻어옴(IHwpObject - 읽기 전용)
String(Property)	Description 메시지 박스에 넣을 문자열
Flag(Property)	Description 메시지 박스에 사용할 Flag
Result(Property)	Description 메시지 박스의 리턴값
DoModal(Method)	Description 메시지 박스 보이기 Declaration void DoModal(void)

IDHwpAction: 한/글에서 특정 기능을 수행하기 위한 액션 오브젝트 (호환 유지를 위해 제공됨 - HwpCtrl에서 사용되는 DHwpAction과 동일)

Item Name	Description
ActID(Property)	Description 액션 ID를 나타낸다. 읽기 전용.
SetID(Property)	Description 액션이 사용하는 parameter set ID를 나타낸다. 읽기 전용.
GetDefault(Method)	Description 현재 상태에 따라 액션 실행에 필요한 인수를 구한다. Declaration void GetDefault(LPDISPATCH param) Parameters param : 인수를 저장할 Parameter Set

<table>
<tr><td>CreateSet(Method)</td><td>Description
액션과 대응하는 Parameter Set을 생성한다.
Declaration
LPIDSPATCH CreateSet(void)
Parameters
return : ParameterSet Object (IDHwpParameterSet)</td></tr>
<tr><td>Execute(Method)</td><td>Description
지정한 인수로 액션을 실행한다.
Declaration
void Execute(LPDISPATCH param)
Parameters
parma : 액션의 실행을 제어할 인수, ParameterSet의 종류와 아이템의 의미는 액션이 정의한 바에 따라 다르다.(IDHwpParameterSet)</td></tr>
<tr><td>PopupDialog(Method)</td><td>Description
액션의 대화상자를 띄운다.
Declaration
long PopupDialog(LPDISPATCH param)
Parameters
parma : 여기에 지정된 아이템의 값에 따라 대화상자의 컨트롤의 초기값이 결정되고, 대화상자가 닫힌 후에는 사용자가 지정한 값들이 담겨 돌아온다.
return : 액션이 정의하기에 따라 다르지만, 일반적으로 다음과 같은 modal dialog result를 리턴한다.
<table>
<tr><th>ID</th><th>값</th><th>설명</th></tr>
<tr><td>hwpOK</td><td>IDOK</td><td>다이얼로그 박스의 확인버튼을 눌렀을 경우 리턴 되는 값</td></tr>
<tr><td>hwpCancel</td><td>IDCANCEL</td><td>다이얼로그 박스의 취소버튼을 눌렀을 경우 리턴 되는 값</td></tr>
<tr><td>hwpError</td><td>-1</td><td>실행시 에러가 발생 하였을 경우 리턴 되는 값</td></tr>
</table></td></tr>
<tr><td>Run(Method)</td><td>Description
액션을 실행한다.
Declaration
void Run(void)
Remarks
CreateSet, GetDefault, PopupDialog, Execute를 차례대로 부른 것과 같다.</td></tr>
</table>

IDHwpParameterSet: 오브젝트 또는 액션의 실행에 필요한 정보를 주고 받을 수 있도록 하기 위한 오브젝트 (호환 유지를 위해 제공됨 - HwpCtrl에서 사용되는 DHwpParameterSet과 동일)

Item Name	Description
Count(Property)	Description 현재 존재하는 아이템의 개수를 나타낸다. (읽기 전용)
IsSet(Property)	Description Parameter Set인지 연부를 나타낸다. (읽기 전용) Remarks 임의의 IDispatch 포인터로부터 Parameter set / parameter array를 구분하기 위해 동일한 이름의 Property를 가지고 종류에 따라

	TRUE/FALSE를 돌려준다. (Prameter set은 TRUE를 리턴한다.)
GetSetID(Property)	Description Parameter Set의 ID를 나타낸다. (읽기 전용)
Clone(Method)	Description 동일한 데이터를 가진 Parameter Set을 복사하여 리턴한다. Declaration LPDISPATCH Clone(void) Parameters return : ParameterSet을 리턴한다. (IDHwpParameterSet)
CreateItemArray(Method)	Description 아이템으로 Parameter Array 타입의 배열을 생성한다. Declaration LPDISPATCH CreateItemArray(BSTR itemid, long count) Parameters itemid : 아이템 ID count : 생성할 배열의 초기 크기 return : 생성된 parameter array 오브젝트 (IDHwpParameterArray) Remarks 동일한 ID를 가진 기존의 아이템은 삭제된다.
CreateItemSet(Method)	Description 아이템으로 Parameter Set을 생성한다. Declaration LPDISPATCH CreateItemSet(BSTR itemid, BSTR setid) Parameters itemid : 아이템 ID setid : 생성할 Parameter Set ID return : 생성된 서브 parameter Set 오브젝트 (IDHwpParameterSet) Remarks ParameterSet 내부에 아이템으로 또 다른 Parameter Set을 가지는 서브셋의 개념이다.
GetIntersection(Method)	Description 두 Parameter Set에 공통적으로 존재하고, 값도 동일한 아이템만으로 구성된 intersection Set을 구한다. Declaration void GetIntersection(LPDISPATCH srcset) Parameters srcset : this와 srcset의 intersection이 this에 저장된다.
IsEquivalent(Method)	Description 두 Parameter Set의 내용이 동일한 값을 가지고 있는지 검사한다. Declaration BOOL IsEquivalent(LPDISPATCH srcset) Parameters srcset : this와 srcset의 비교한 결과를 리턴한다. return : 동일하면 TRUE, 다르면 FALSE
Item(Method)	Description 지정한 아이템의 값을 리턴한다. Declaration Variant Item(BSTR itemid) Parameters itemid : 아이템 ID return : 아이템의 값 Remarks

	만약 지정한 아이템이 존재하지 않으면 아이템의 포맷에 따라 0 또는 빈 문자열을 리턴한다.
ItemExsit(Method)	Description 지정한 아이템이 존재하는지 검사한다. Declaration BOOL ItemExist(BSTR itemid) Parameters itemid : 아이템 ID return : 존재하면 TRUE, 존재하지 않으면 FALSE
Merge(Method)	Description 두 Parameter Set의 내용을 병합한다. Declaration void Merge(LPDISPATCH srcset) Parameters srcset : this와 srcset이 병합되어 this에 저장된다. Remarks 결과는 "this의 모든 아이템 + srcset에만 존재하는 아이템"이다.
RemoveAll(Method)	Description Parameter Set을 초기화 한다. Declaration void RemoveAll(BSTR setid) Parameters setid : 새로 적용할 Set ID Remarks 이미 존재하는 Parameter Set 오브젝트를 이용해 새로운 타입의 Parameter Set으로 초기화하여 재사용하는 목적에 사용된다.
RemoveItem(Method)	Description 지정한 아이템을 삭제한다. Declaration void RemoveItem(BSTR itemid) Parameters itemid : 아이템 ID
SetItem(Method)	Description 지정한 아이템의 값을 설정한다. Declaration void SetItem(BSTR itemid, Variant value) Parameters itemid : 아이템 ID value : 설정할 값 Remarks 이미 동일한 ID의 아이템이 존재하면 지정한 값으로 바뀌고, 존재하지 않으면 아이템이 생성된다.

IDHwpParameterArray: Parameter Set의 아이템으로 배열을 표현하는데 사용된다. 일반적인 Method의 독립적인 인수로 사용되는 일은 없고, Parameter Set의 아이템으로만 사용된다.
(호환 유지를 위해 제공됨 - HwpCtrl에서 사용되는 DHwpParameterArray와 동일)

Item Name	Description
Count(Property)	Description 배열의 크기를 나타낸다. Remarks

	배열의 크기는 runtime에 dynamic하게 조절 할 수 있다.
IsSet(Property)	Description Parameter Set인지 여부를 나타낸다.(읽기 전용) Remarks 임의의 IDispatch 포인터로부터 Parameter Set / Parameter Array를 구분하기 위해 동일한 이름의 Property를 가지고 종류에 따라 TRUE/FALSE를 돌려준다.(ParameterArray는 FALSE를 리턴한다.)
Clone(Method)	Description 동일한 크기와 데이터를 갖는 ParameterArray 개체를 복사하여 돌려준다 Declaration LPDISPATCH Clone(void)
Copy(Method)	Description 배열을 복사한다. Declaration void Copy(LPDISPATCH srcarray) Parameters srcarray : srcarray의 내용이 그대로 this로 복사된다.
Item(Method)	Description 지정한 원소의 값을 리턴한다. Declaration Variant Item(long index) Parameters index : 원소의 인덱스. 1부터 시작한다.
SetItem(Method)	Description 지정한 원소의 값을 설정한다. Declaration void SetItem(long index, Variant value) Parameters index : 원소의 인덱스, 1부터 시작 value : 원소의 값

IDHwpCtrlCode: 문서 내부의 표, 각주 등의 컨트롤(특수 문자 포함)를 나타내는 오브젝트이다. (호환 유지를 위해 제공됨 - HwpCtrl에서 사용되는 DHwpCtrlCode와 동일)

Item Name	Description
CtrlCh(Property)	Description 컨트롤 문자. (읽기전용) Remarks 일반적으로 컨트롤 ID를 사용해 컨트롤의 종류를 판별하지만, 이보다 더 포괄적인 범주를 나타내는 컨트롤 문자로 판별할 수도 있다. 예를 들어 각주와 미주는 ID는 다르지만, 컨트롤 문자는 17로 동일하다. 컨트롤 문자는 1-31 사이의 값을 사용한다. Ch / 설명 1 / 예약 2 / 구역/단 정의 3 / 필드 시작

	<table><tr><th>Ch</th><th>설명</th></tr><tr><td>4</td><td>필드 끝</td></tr><tr><td>5</td><td>예약</td></tr><tr><td>6</td><td>예약</td></tr><tr><td>7</td><td>예약</td></tr><tr><td>8</td><td>예약</td></tr><tr><td>9</td><td>탭</td></tr><tr><td>10</td><td>강제 줄 나눔</td></tr><tr><td>11</td><td>그리기 개체 / 표</td></tr><tr><td>12</td><td>예약</td></tr><tr><td>13</td><td>문단 나누기</td></tr><tr><td>14</td><td>예약</td></tr><tr><td>15</td><td>주석</td></tr><tr><td>16</td><td>머리말 / 꼬리말</td></tr><tr><td>17</td><td>각주 / 미주</td></tr><tr><td>18</td><td>자동 번호</td></tr><tr><td>19</td><td>예약</td></tr><tr><td>20</td><td>예약</td></tr><tr><td>21</td><td>쪽바뀜</td></tr><tr><td>22</td><td>책갈피 / 찾아보기 표시</td></tr><tr><td>23</td><td>덧말 / 글짜 겹침,,,</td></tr><tr><td>24</td><td>하이픈</td></tr><tr><td>25</td><td>예약</td></tr><tr><td>26</td><td>예약</td></tr><tr><td>27</td><td>예약</td></tr><tr><td>28</td><td>예약</td></tr><tr><td>29</td><td>예약</td></tr><tr><td>30</td><td>묶음 빈칸</td></tr><tr><td>31</td><td>고정 폭 빈칸</td></tr></table>
CtrlID(Property)	Description 컨트롤 ID. (읽기 전용) Remarks 컨트롤 ID는 컨트롤의 종류를 나타내기 위해 할당된 ID로서, 최대 4개의 문자로 구성된 문자열이다. 예를 들어 표는 "tbl", 각주는 "fn"이다. 워디안에서 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
atno	AutoNum	AutoNum	번호넣기
nwno	AutoNum	AutoNum	새번호로
pgct	PageNumCtrl	PageNumCtrl	페이지 번호 제어 (97의 홀수쪽에서 시작)
pghd	PageHiding	PageHiding	감추기
pgnp	PageNumPos	PageNumPos	쪽번호 위치
head	HeaderFooter	HeaderFooter	머리말
foot	HeaderFooter	HeaderFooter	꼬리말
%dte	FieldCtrl	FieldCtrl	현재의 날짜/시간 필드
%ddt	FieldCtrl	FieldCtrl	파일 작성 날짜/시간 필드
%pat	FieldCtrl	FieldCtrl	문서 경로 필드
%bmk	FieldCtrl	FieldCtrl	블럭 책갈피
%mmg	FieldCtrl	FieldCtrl	메일 머지
%xrf	FieldCtrl	FieldCtrl	상호 참조
%fmu	FieldCtrl	FieldCtrl	계산식
%clk	FieldCtrl	FieldCtrl	누름틀
%smr	FieldCtrl	FieldCtrl	문서 요약 정보 필드
%usr	FieldCtrl	FieldCtrl	사용자 정보 필드
%hlk	FieldCtrl	FieldCtrl	하이퍼링크
bokm	TextCtrl	TextCtrl	책갈피
idxm	IndexMark	IndexMark	찾아보기
tdut	Dutmal	Dutmal	덧말
tcmt	없음	없음	주석

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

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

HasList(Property)	**Description** 글상자를 지원하는지의 여부(읽기 전용)
Next(Property)	**Description**

<table>
<tr><td></td><td>다음 컨트롤.(읽기 전용)
Remarks
문서 중의 모든 컨트롤(표, 그림등의 특수 문자들)은 linked list로 서로 연결되어 있는데, list중 다음 컨트롤을 나타낸다.</td></tr>
<tr><td>Prev(Property)</td><td>Description
앞 컨트롤.(읽기 전용)
Remarks
문서 중의 모든 컨트롤(표, 그림등의 특수 문자들)은 linked list로 서로 연결되어 있는데, list중 다음 컨트롤을 나타낸다.</td></tr>
<tr><td>Properties(Property)</td><td>Description
컨트롤의 속성을 나타낸다.
Parameters
모든 컨트롤은 대응하는 parameter set으로 속성을 읽고 쓸 수 있다.</td></tr>
<tr><td>UserDesc(Property)</td><td>Description
컨트롤의 종류를 사용자에게 보여줄 수 있는 localize된 문자열로 나타낸다. (읽기 전용)</td></tr>
<tr><td>GetAnchorPos(Method)</td><td>Description
컨트롤의 anchor의 위치를 리턴한다.
Declaration
LPDISPATCH GetAnchorPos(long type)
Parameters
type : 기준 위치
<table>
<tr><th>값</th><th>설명</th><th>비고</th></tr>
<tr><td>0</td><td>바로 상위 리스트에서의 anchor position</td><td>default</td></tr>
<tr><td>1</td><td>탑레벨 리스트에서의 anchor position</td><td></td></tr>
<tr><td>2</td><td>루트 리스트에서의 anchor position</td><td></td></tr>
</table>
value : 성공했을 경우 LispParaPos Parameter Set이 리턴 된다. 실패했을 경우 NULL이 리턴된다.</td></tr>
</table>

– Action Object

Action Object 매뉴얼 참고(ActionObject.hwp)

– ParameterSet Object

ParameterSet Object 매뉴얼 참고(ParameterSetObject.hwp)

– Add-On Object(메뉴, 툴바 제어 기능, 사용자 정의 액션 기능)