

신기술/기능 요구사항 지원 보고서

[제목: IE7.0 호환성 이슈]

한국소프트웨어진흥원
공개SW기술지원센터

<Revision 정보>

일자	VERSION	변경내역	작성자
2007.02.23	0.1	초기 작성	김상운

목 차

1. 문서 개요	4
가. 문서의 목적	4
나. 본 문서의 사용방법	4
2. 신기술/기능 요구사항	5
가. 대상 업체	5
나. 요구 사항	5
3. 요구사항 내용	6
가. 요구사항 테스트 환경	6
나. 특이사항	6
4. 호환성 이슈	7
가. 호환성 문제 유형	7
나. IE7.0 자체의 변화	7
다. Vista 와 연계된 IE 7.0 의 보안 강화	7
5. 호환성 문제 유형 정리 및 해결책	8
가. Windows XP/2003/Vista 모두에 적용되는 호환성 이슈	8
1) 표준 모드(Strict Mode)와 유연 모드(Quirks Mode)	8
2) <SELECT> 요소를 windowless 동작방식으로 변경	10
3) 혼합된(mixed) 페이지에서의 확인 창	13
4) userAgent 값 변경	14
5) DHTML Scriptlet(스크립틀릿) 보안 강화	14
나. Windows Vista 에만 적용되는 호환성 이슈	15

1. 문서 개요

본 문서는 한국소프트웨어진흥원 공개SW사업단의 요청으로 인해 마이크로소프트 인터넷 익스플로러 7.0의 출시로 인한 호환성 문제와 해결방안을 제시하여 향후 리눅스 기반의 웹브라우저와 비교할 수 있는 기초로 활용한다.

가. 문서의 목적

다음과 같은 세부적인 목적을 달성하기 위하여 작성되었다.

- IE 7.0의 변경된 점

- IE 7.0을 지원하기 위한 기존 사이트 호환성 해결 방안

나. 본 문서의 사용방법

다음과 같은 방법으로 사용할 수 있다.

- 본 문서는 순수한 윈도우즈 기반에서 해결방법을 언급하고 있다.

- IE 7.0에 호환되게 사이트를 개발함으로 인해 발생할 수 있는 리눅스 웹브라우저의 호환성 문제는 언급이 없으므로 본 내용을 토대로 비교할 수 있도록 한다.

2. 신기술/기능 요구사항

가. 대상 업체

구 분		비 고
기관명	한국소프트웨어진흥원	
담당자	이영재	
연락처	02-2141-5062	
회사 위치	서울 송파구 가락본동 79-2 KIPA빌딩 12층 공개SW사업단	

나. 요구 사항

항목	지원 내용	비고
IE7.0호환성	<ul style="list-style-type: none"> ● IE7.0의 기존 사이트 호환성 여부 ● 호환성 해결 방안 제시 	

3. 요구사항 내용

가. 요구사항 테스트 환경

항목	내역	수량	비고
IE 7.0변경사항			
IE 7.0호환성			

나. 특이사항

- 앞서 언급한 내용처럼 윈도우즈 기반에서 설명하고 있으며 직접적인 리눅스기반과의 비교는 없습니다.
- 본 문서는 향후 IE 7.0에 호환되게 개발된 사이트에서 리눅스 환경과 비교시 참고자료로 활용자는 취지에서 작성된 자료이므로 리눅스 기반에서와의 비교는 향후 테스트 완료시점에서 다시 작성하도록 할 예정입니다.

4. 호환성 이슈

이러한 주요로 인해 기존 IE 6/파이어폭스에 맞추어져 제작된 웹 사이트들은 IE 7.0을 적용할 경우, 정상적인 웹 서비스를 하는 데 있어 다소 문제가 발생할 수도 있습니다. 이 장에서는, 그러한 발생 가능한 유형들을 살펴보고 해결책을 제시해 보도록 하겠습니다.

가. 호환성 문제 유형

IE 7.0 은 2가지 유형의 버전(XP기반의 버전과 Vista기반의 버전)이 있고 이로 인해 발생할 수 있는 호환성 문제도 2가지로 나뉘게 됩니다. 따라서 웹 사이트 저작자들은 IE 7.0 적용으로 인한 호환성 문제를 해결하기 위해서는 반드시 이 2가지 환경을 함께 고려해 줘야 합니다.

나. IE7.0 자체의 변화

IE 7.0은 그 프로그램 자체로 많은 변경 사항들을 포함하고 있습니다. 이러한 변경으로 인해 발생하는 호환성 문제는 “Windows XP/2003/Vista” 의 모든 운영체제에 설치된 IE 7.0에 적용이 되며, 특정 운영체제 환경에서 수정이 되었다면 다른 환경에서도 동일하게 적용 받을 수 있습니다.

다. Vista 와 연계된 IE 7.0 의 보안 강화

사용자를 보호하기 위해 보안을 좀더 강화시킨 윈도우즈 비스타는 주요 기능으로 사용자 계정컨트롤(UAC: User Access Control)를 구현하고 있으며, 비스타에서 실행되는 IE 7.0 은 이러한 사용자 계정 컨트롤과 맞물려서 특별히 “보호 모드(Protected Mode)”를 제공하고 있습니다. 비스타에서만의 이러한 보안 강화 기능으로 인해 “Windows XP/2003” 운영체제에 설치된 IE 7.0 과는 별개의 호환성 문제가 발생할 수 있습니다.

5. 호환성 문제 유형 정리 및 해결책

이제부터, 각 유형별로 호환성 문제가 발생할 수 있는 상황을 살펴보고 그 해결책에 대해 자세히 알아보도록 하겠습니다.

가. Windows XP/2003/Vista 모두에 적용되는 호환성 이슈

여기에 나열되는 문제들은 IE 7.0 이 설치된 모든 운영체제에서 나타날 수 있는 문제들입니다. 이러한 호환성 문제들은 현재 사용자들이 직접 "Internet Explorer 7 다운로드" 사이트로부터 설치한 경우, 해당 사용자들이 불편을 겪고 있는 사항일 수 있으므로 가장 시급하게 해결되어야 할 문제들입니다. 크게 다음과 같이 3가지 범주에서 호환성 문제가 발생할 수 있습니다.

1. IE 6 보다 강화된 보안 기능으로 인해 발생할 수 있는 문제
2. IE 6 에 비해 향상된 웹 표준 준수로 인해 발생할 수 있는 문제
3. 사용자 인터페이스 변화로 인해 발생할 수 있는 문제

각각의 환경에서 발생할 수 있는 대표적인 유형들을 하나씩 아래에서 구체적으로 알아보겠습니다.

1) 표준 모드(Strict Mode)와 유연 모드(Quirks Mode)

표준 모드와 유연(호환) 모드는 사실상 이미 인터넷 익스플로러 6.0부터 소개된 기능입니다. 해당 웹 페이지가 표준 모드인지 유연 모드인지에 대해서는 HTML 웹 페이지의 처음 부분에 명시된 <!DOCTYPE> 지시자로 구분이 됩니다. 일례로, 표준 모드를 준수하는 웹 페이지는 다음과 같은 DOCTYPE 을 포함합니다.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Strict//EN">
```

반면에 유연 모드의 웹 페이지라면 다음과 같은 DOCTYPE 을 포함하거나 또는 아예 DOCTYPE 자체를 포함하고 있지 않은 웹 페이지일 수 있습니다.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

IE 7.0 은 IE 6 에서 가지고 있던 표준 모드에서의 버그를 대폭 수정하여 보다 더 완벽하게 HTML 4.01 과 CSS 2.1 표준을 적용시키고 있습니다. (100% 완벽하게 지원하지는 못함) 이와 함께, 유연 모드는 기존과 마찬가지로 호환성을 최대한 유지하도록 해줍니다.

웹 사이트에 있는 웹 페이지들을 “표준 모드” DOCTYPE 을 가지고 HTML 4.01/CSS 2.1 표준에 맞게 제작한다면, 대부분의 웹 브라우저들이 해당 웹 페이지에 대해 “동일한 화면”을 보여주는 것이 가능합니다.

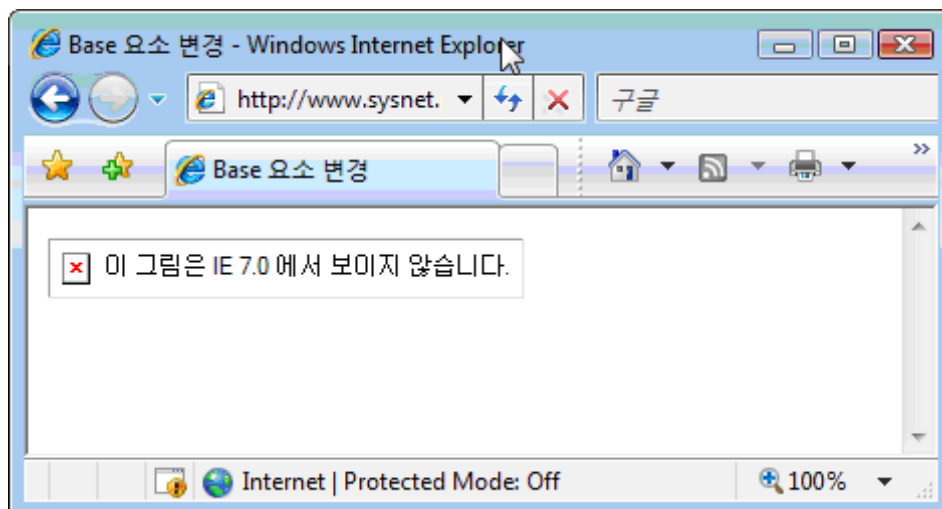
IE 7.0 에서 BASE 요소는 반드시 HEAD 요소 안에 위치해야 합니다. 예를 들어, 다음과 같은 HTML 파일이 있을 때, 기존 IE 6 이전 버전에서는 정상적으로 이미지를 보여주지만, IE 7.0 에서는 보여지지 않습니다.

```

<!DOCTYPE html          PUBLIC "-//W3C//DTD          XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
</head>
<body>
<base href="http://www.w3.org/Icons/" />

</body>
</html>
    
```

[예제: BASE 태그가 BODY 요소 안에 있기 때문에 IE 7.0 에서는 무효]



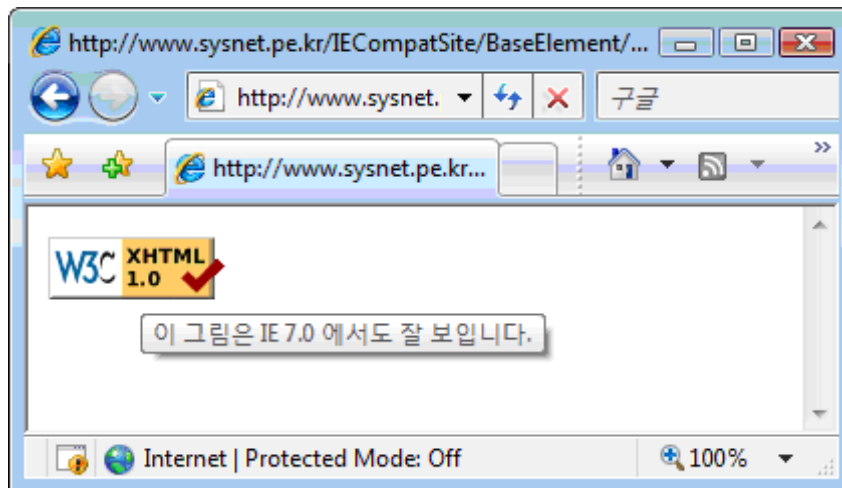
[그림: BODY 내부에 BASE 태그가 위치 - IE 7.0 에서 보이지 않습니다.]

이와 같은 경우 정상적으로 그림이 보이도록 하기 위해서는 다음과 같이 BASE 요소를 HEAD 요소 안에 있도록 이동시켜야 합니다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<base href="http://www.w3.org/Icons/" />
</head>
<body>

</body>
</html>
```

[예제: BASE 태그가 HEAD 요소 안에 있어서 그림이 보이게 됩니다.]



[그림: HEAD 내부에 BASE 태그가 위치 - IE 7.0에서 정상적으로 보입니다.]

2) <SELECT> 요소를 windowless 동작방식으로 변경

별도의 윈도우로 구현된 ActiveX 등을 제외하고 표준 HTML 태그의 구현 중에서 유일하게 “Window” 자원을 가진 요소가 바로 SELECT 였습니다. “Window” 자원이라는 것은, Windows 운영체제에서 제공되는 윈도우 핸들(Window Handle)을 가졌다는 것을 의미하며 이로 인해 SELECT 요소가 가진 Zoom, Z-Order 속성이 다른 HTML 요소들의 동일한 속성들과 별개로 동작하게 됩니다. 대표적인 사례가 HTML 의 DIV 태그 등으로 구성된 풀다운 메뉴들이 SELECT 요소를 덮지 못하는 문제를 들 수 있는데, 이로 인해 다른 윈도우 자원을 가진 요소를 사용해서 이런 문제점을 해결하기도 하였습니다.

이러한 문제를 근본적으로 해결하기 위해 IE 7.0부터는 SELECT 요소가 “Windowless 요소”로 구

현되었고, 더 이상 다른 요소들과 어울리지 않고 독립적으로 작용했던 문제가 발생하지 않게 되었습니다. 예를 들어, 아래의 웹 페이지는 SELECT 태그 위에 DIV 영역을 보이려고 DIV.zIndex 값을 SELECT.zIndex 값보다 더 높게 설정하고 있습니다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<body>

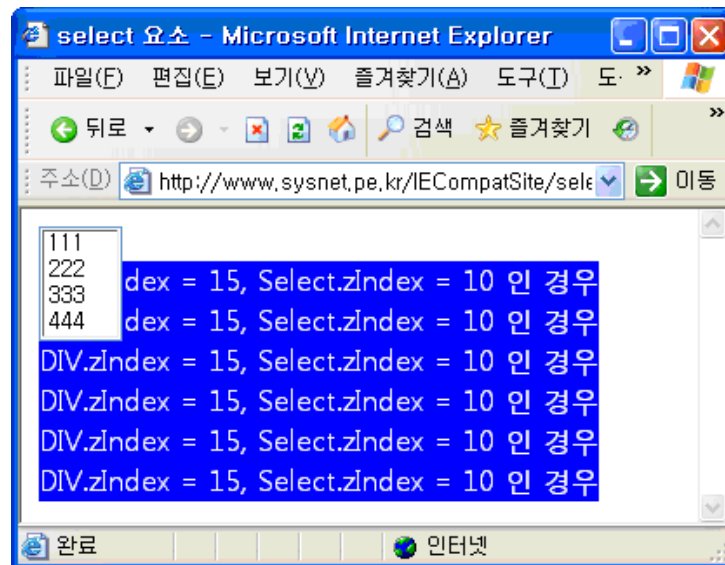
<select size="4" style="position:absolute; top: 10px; left: 10px; z-index: 10">
<option>111</option>
<option>222</option>
<option>333</option>
<option>444</option>
</select>

<div style="position:absolute; top: 30px; left: 10px; z-index: 15; background-color: blue">
DIV.zIndex = 15, Select.zIndex = 10 인 경우<br />
DIV.zIndex = 15, Select.zIndex = 10 인 경우<br />
DIV.zIndex = 15, Select.zIndex = 10 인 경우<br />
DIV.zIndex = 15, Select.zIndex = 10 인 경우<br />
DIV.zIndex = 15, Select.zIndex = 10 인 경우<br />
DIV.zIndex = 15, Select.zIndex = 10 인 경우<br />
</div>

</body>
</html>
```

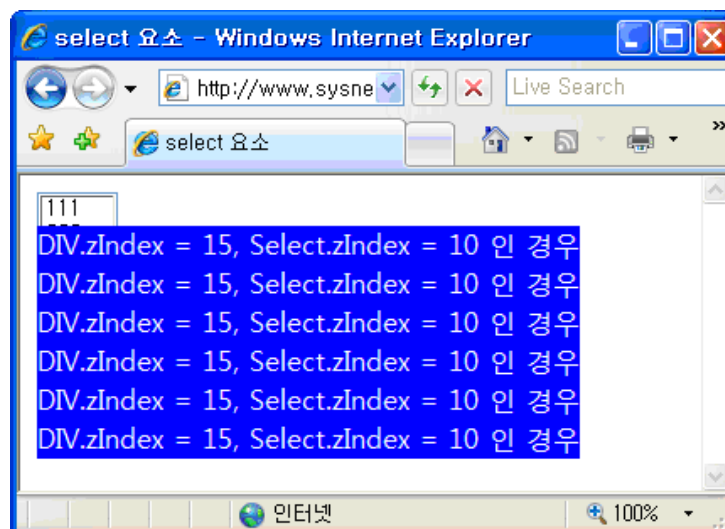
[예제: SELECT 태그에 z-index 적용 테스트]

위의 HTML 파일을 IE 6 이전 버전에서 보게 되면 아래 화면과 같이 DIV 영역이 SELECT 요소에 의해 가려지는 것을 확인할 수 있습니다. 물론, 이것은 HTML 저작자가 원한 결과가 아닙니다. 하지만, SELECT 요소가 “웹 브라우저 윈도우”의 “자식 윈도우”로 구현되어 있기 때문에 “웹 브라우저 윈도우” 내부의 텍스트 영역의 요소들로는 가려질 수가 없었던 것입니다.



[그림: IE 6 에서 본 SELECT / DIV 예제]

반면, IE 7.0 에서 보게 되면 아래 화면과 같이 SELECT 영역이 정상적으로 DIV 영역에 의해 가려지는 것을 확인할 수 있습니다.



[그림: IE 7.0 에서 본 SELECT / DIV 예제]

이러한 SELECT 요소의 변화는 바람직하게 받아들여질 수 있지만, 한편으로는 기존에 SELECT 를 덮기 위해 사용했던 우회 방법들이 IE 7.0 에서도 여전히 정상적으로 보여지는 지를 확인해 볼 필요가 있게 되었습니다.

3) 혼합된(mixed) 페이지에서의 확인 창

이 옵션은 “인터넷 영역”, “로컬 인트라넷 영역”, “신뢰된 사이트 영역” 모두에서 기본적으로 “확인”으로 설정되어 있습니다. IE 6 이전 버전에서도 이와 유사한 옵션이 있었는데, “인터넷 옵션” -> “고급 탭” 에서만 설정할 수 있었던 “보안과 비보안 모드 사이를 전환할 때 경고” 기능으로 “영역(Zone)” 별 설정이 가능하지 않았던 단점이 있었습니다. (참고로, 이 값은 IE 7.0 에서는 기본적으로 해제되어 있습니다.)

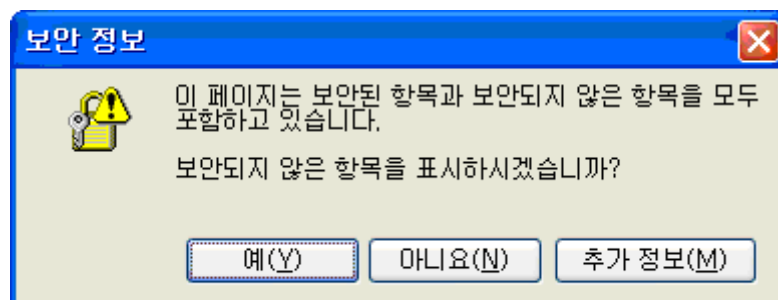
여기서 혼합된(mixed) 페이지란, 사용자가 안전하다고 선택한 HTTPS로 접근한 리소스에서, 안전성 여부를 검증할 수 없는 HTTP 리소스를 참조하고 있는 것을 말합니다. 예를 들어, 다음과 같은 웹 페이지가 “https://yoursite/httpsContents.htm” 이라는 경로를 통해 서비스되고 있다고 가정해 보면,

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

페이지 내부의 IMG 요소가 (안전하지 않은)HTTP 프로토콜 자원을 포함하고 있기 때문에, “혼합된 페이지”로 간주되어 사용자로 하여금 “확인”을 받도록 아래와 같은 경고창을 띄워주게 됩니다.

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<body>
경고 확인창: 
</body>
</html>
```

페이지 내부의 IMG 요소가 (안전하지 않은)HTTP 프로토콜 자원을 포함하고 있기 때문에, “혼합된 페이지”로 간주되어 사용자로 하여금 “확인”을 받도록 아래와 같은 경고창을 띄워주게 됩니다.



[그림: 보안을 요하는 페이지에서 보안되지 않은 자원을 포함하고 있다는 경고]

만약, 사용자가 “아니오”를 선택한다면 해당 HTTP 자원에 대해서는 웹 브라우저가 더 이상 처리를 진행하지 않습니다. 따라서, 위에서처럼 인 경우에는 해당 이미지가 화면에 보이지 않게

됩니다.

이와 같은 경고창을 원치 않는다면 다음과 같은 2가지 방법을 통해서 해결이 가능합니다.

1.HTTPS콘텐츠에서는 HTTPS 콘텐츠만을 포함: 해당 HTTPS 자원에서 포함하는 외부 자원 역시 HTTPS 경로로 포함시켜 주도록 수정할 수 있습니다. 만약 대상 사이트에서 HTTPS를 지원하지 않는다면, 이미지 같은 경우 별도로 다운로드 받아서 자신의 웹 사이트 폴더에 추가시킨 후, 내부 상대 경로를 통해서 지정해 줄 수 있습니다.

2.“혼합된 콘텐츠 표시” 옵션 설정: 아래 화면과 같이 인터넷 익스플로러의 “도구” -> “인터넷 옵션” 대화창에서 “보안” -> “인터넷 영역”에 포함된 설정의 “혼합된 콘텐츠 표시” 옵션을 “사용”으로 선택합니다.

4) userAgent 값 변경

IE 7.0 은 userAgent 값이 기존 “MSIE 6.0”에서 “MSIE 7.0”으로 바뀌었습니다. 예서 설명한 것처럼 IE 7.0 은 Vista 용과 XP/2003 용으로 나뉘는 데, 만약 이러한 차이를 구분하고 싶다면 userAgent 의 Windows NT 값을 검색해서 판단을 해야 합니다.

예를 들어, Windows XP 에 설치된 IE 7.0 의 userAgent 값은 다음과 같습니다.

Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; ...)

반면에 Windows Vista 에 설치된 IE 7.0 의 userAgent 값은 다음과 같습니다.

Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; ...)

5) DHTML Scriptlet(스크립틀릿) 보안 강화

DHTML Scriptlet은 HTC(HTML Component)와 함께 인터넷 익스플로러에서만 구현할 수 있는 재사용 가능한 HTML 컴포넌트의 한 유형입니다. HTC 와는 달리 구현이 OBJECT 태그를 통해서 되기 때문에 “Windowed 개체”로 구현이 되어, 기존에 SELECT 요소의 Windowed 제한을 넘어서기 위한 수단으로 사용되기도 했습니다. 하지만, IE 7에 와서는 DHTML Scriptlet에 대한 사용을 기본적으로 “사용 안함”으로 설정되어 있기 때문에 OBJECT 태그에서 정상적으로 보여지지 않게 되었습니다.

예를 들어, 여러분들의 웹 페이지에 다음과 같은 유형의 DHTML Scriptlet을 포함하고 있는 경우,

[예제: DHTML Scriptlet을 포함하고 있는 웹 페이지]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

기존의 IE 6에서는 정상적으로 “HtmlComponent.htm” 파일의 내용이 화면에 출력되었지만, IE 7.0에서는 아래 화면과 같이 OBJECT 영역이 “X”자로 표시되어 비활성화 됩니다.

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
  <title>DHTML Scriptlet</title>
</head>
<body>
<object width="300" height="170" type="text/x-scriptlet" data="HtmlComponent.htm">
</object>
</body>
</html>
```

기존의 IE 6에서는 정상적으로 “HtmlComponent.htm” 파일의 내용이 화면에 출력되었지만, IE 7.0에서는 아래 화면과 같이 OBJECT 영역이 “X”자로 표시되어 비활성화 됩니다.

나. Windows Vista 에만 적용되는 호환성 이슈

이미 수많은 매체를 통해서 알려졌듯이, 윈도우즈 비스타 운영체제는 “보안” 측면에서 많은 개선을 이룬 제품입니다. 사용자 계정 컨트롤(UAC: User Access Control) 자체로도 이미 상당한 보안 효과를 이루어 내었지만, 그에 한걸음 더 나아가서 IE 7.0에서는 “보호 모드(Protected mode)”라는 층을 하나 더 없음으로써 보다 더 강력한 보안 기능을 구현하고 있습니다.

사실, 이러한 보안 강화는 기존에 제작되어 배포되고 있던 ActiveX 컨트롤 입장에서는 많은 제약으로 다가올 수 있습니다. 따라서, 윈도우즈 비스타에 설치된 IE 7.0에서 발생하는 호환성 문제는 지금까지 살펴본 문제에 비해 좀더 어려운 문제일 수 있으며, 심한 경우 해당 ActiveX 의 소스 코드 자체를 수정해서 다시 빌드/배포해야 할 수도 있습니다.

웹 사이트 저작자들은 반드시 개발된 사이트 또는 개발중인 사이트에 대해서 Windows XP SP2 환경과 함께 Windows Vista 에서의 IE 7.0 보호 모드로도 호환성 테스트를 진행할 것을 강력히 권장합니다.

참고로, “Windows Vista IE7 에서 ActiveX 컨트롤 개발 방법론”에 대한 기술 백서도 한국 마이크로소프트에 의해서 배포될 예정이므로 ActiveX 개발과 관계된 분들은 해당 백서를 읽어 보시면 많은 도움을 받을 수 있을 것입니다.