



# 성능향상 가이드

The Best UI·UX platform for Business User eXperience

투비소프트  
교육서비스팀

TOBESOFT



# CONTENTS



**01. 공통 영역**

**02. 환경**

**03. 메모리**



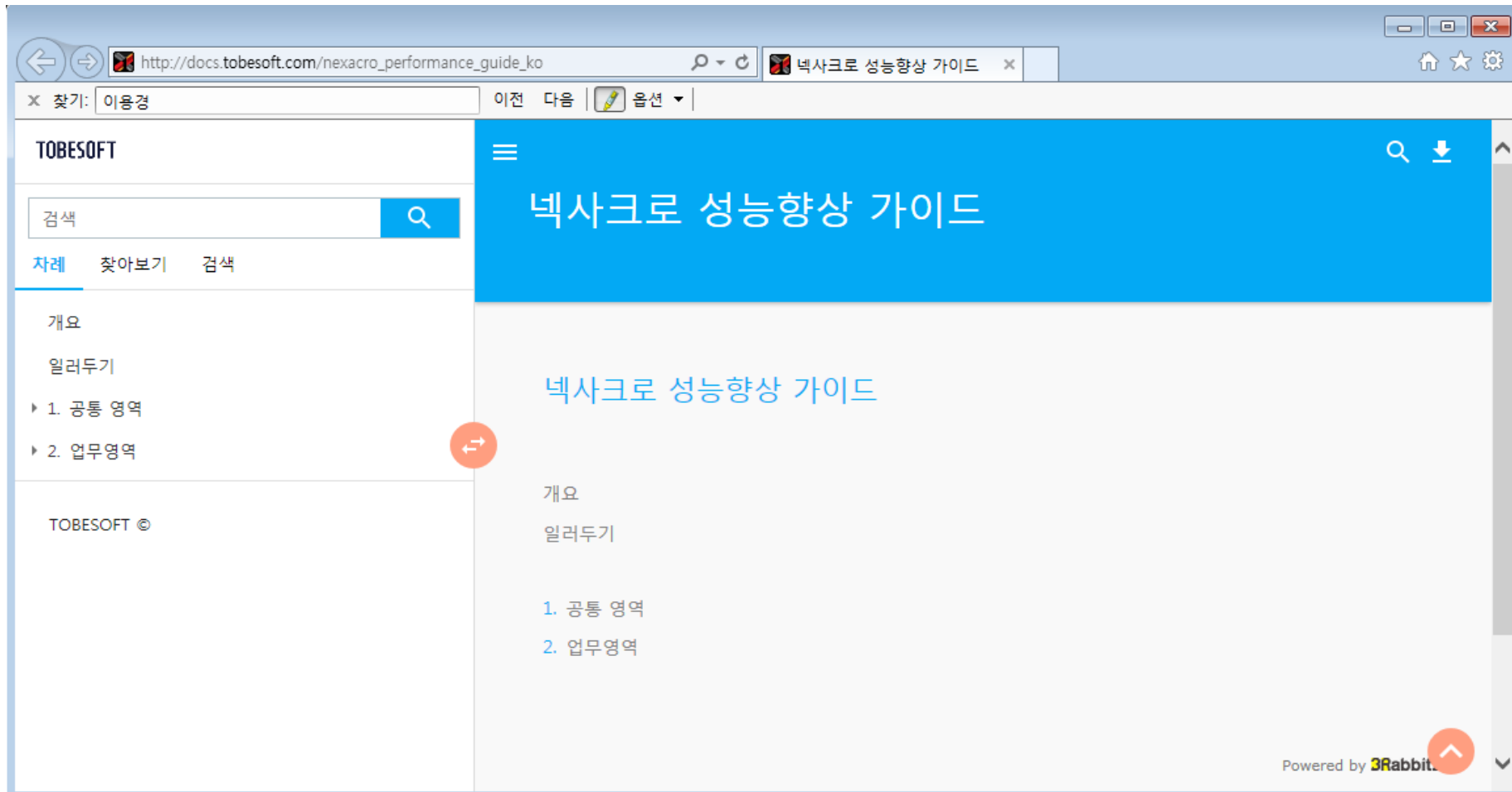
---

# 01. 공통 영역

# 성능향상 가이드 - 공통

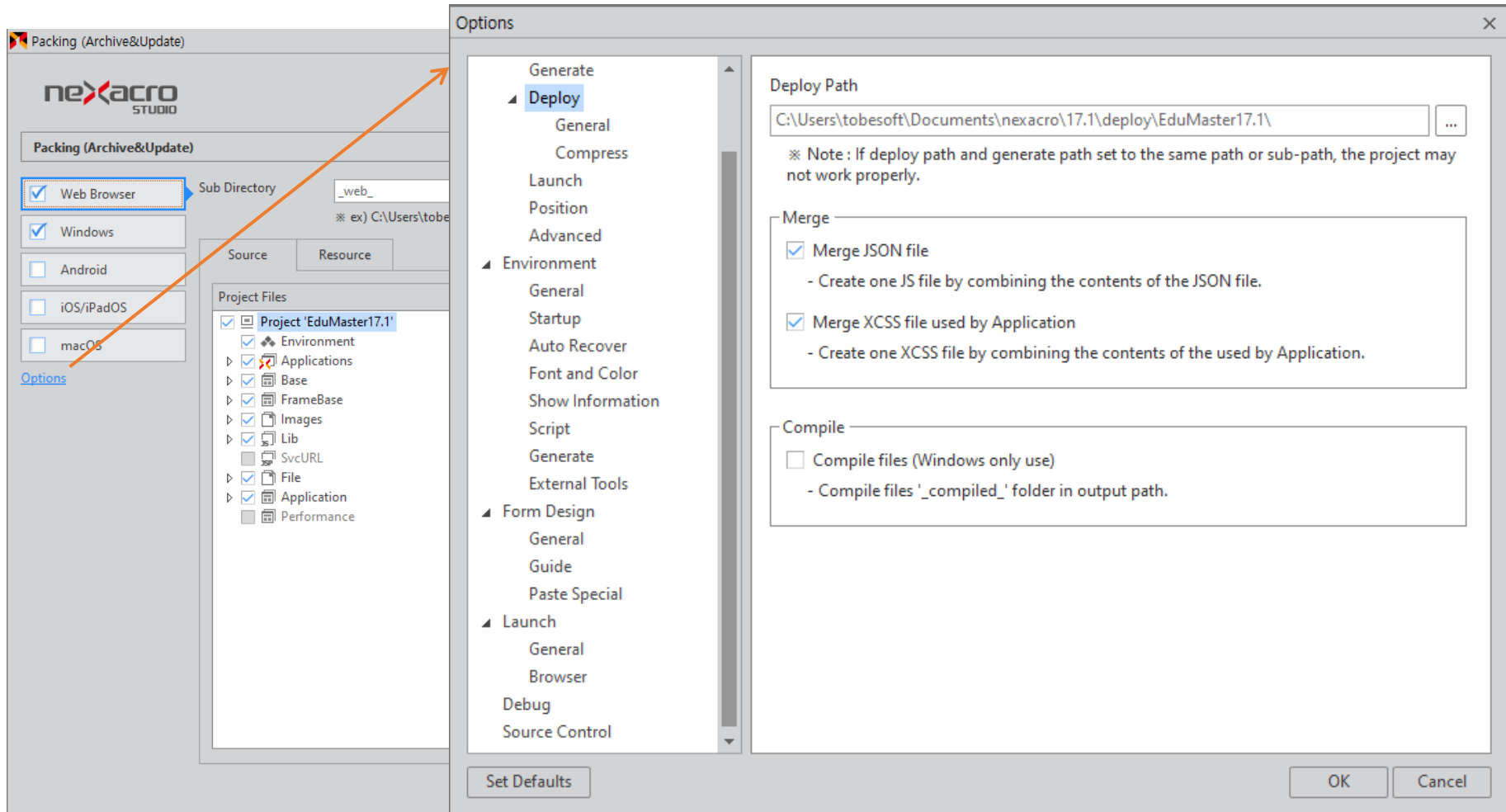
해당 내용은 투비소프트 고객지원센터에 Open 된 자료를 기준으로 정리됨.

[http://docs.tobesoft.com/nexacro\\_performance\\_guide\\_ko](http://docs.tobesoft.com/nexacro_performance_guide_ko)



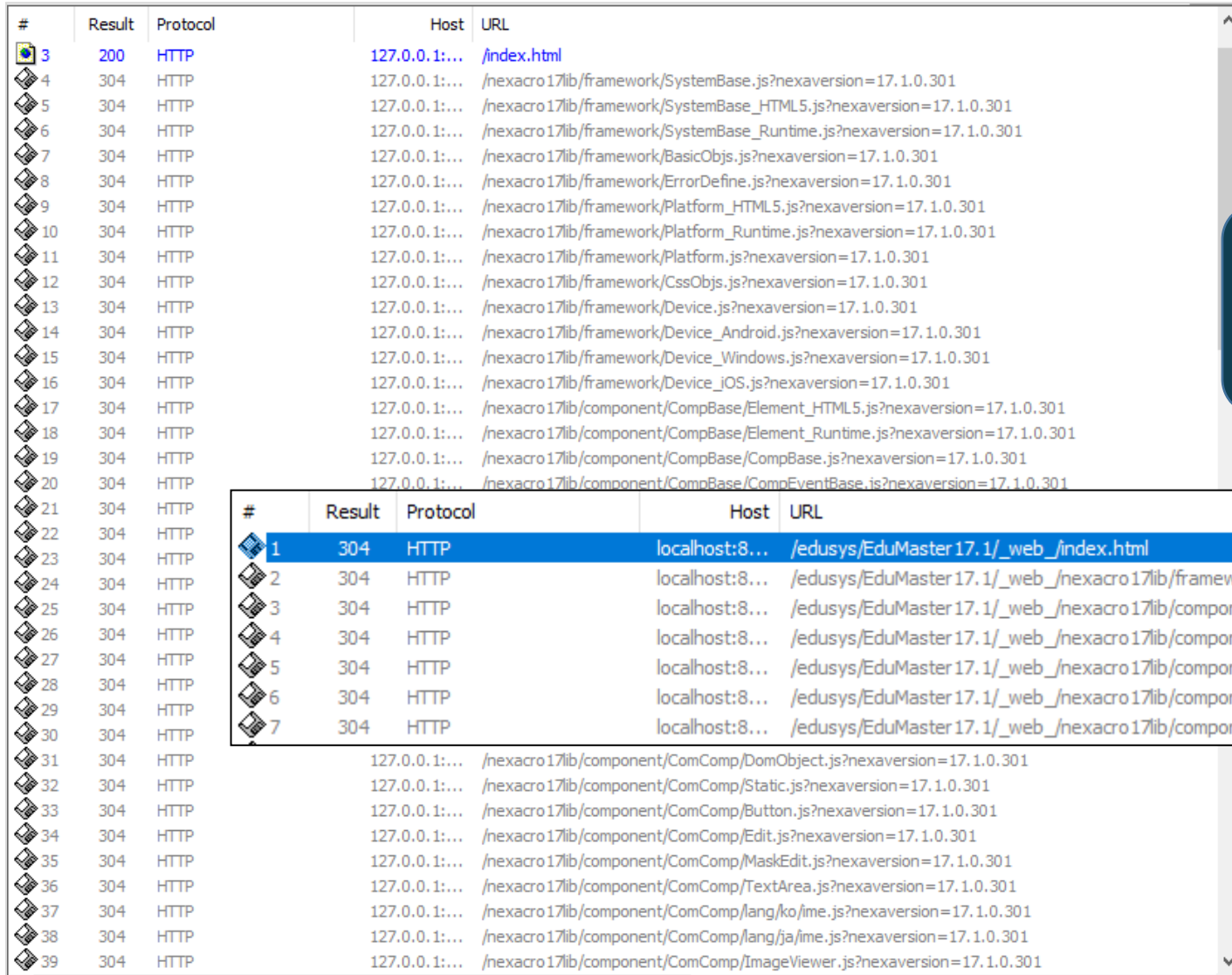
# 성능향상 가이드 – 공통 – JS Merge, Compress (1/4)

NEXACRO Javascript framework를 Merge, Compress 기능을 통해 파일 개수 및 사이즈 최적화 운영 시에는 NEXACRO Javascript framework (Library) 에 대해서는 필수 사용권장



# 성능향상 가이드 – 공통 – JS Merge, Compress (2/4)

NEXACRO Javascript framework를 Merge, Compress 기능을 통해 파일 개수 및 사이즈 최적화 운영 시에는 NEXACRO Javascript framework (Library) 에 대해서는 필수 사용권장



#	Result	Protocol	Host	URL
3	200	HTTP	127.0.0.1...	/index.html
4	304	HTTP	127.0.0.1...	/nexacro17lib/framework/SystemBase.js?nexaversion=17.1.0.301
5	304	HTTP	127.0.0.1...	/nexacro17lib/framework/SystemBase_HTML5.js?nexaversion=17.1.0.301
6	304	HTTP	127.0.0.1...	/nexacro17lib/framework/SystemBase_Runtime.js?nexaversion=17.1.0.301
7	304	HTTP	127.0.0.1...	/nexacro17lib/framework/BasicObjs.js?nexaversion=17.1.0.301
8	304	HTTP	127.0.0.1...	/nexacro17lib/framework/ErrorDefine.js?nexaversion=17.1.0.301
9	304	HTTP	127.0.0.1...	/nexacro17lib/framework/Platform_HTML5.js?nexaversion=17.1.0.301
10	304	HTTP	127.0.0.1...	/nexacro17lib/framework/Platform_Runtime.js?nexaversion=17.1.0.301
11	304	HTTP	127.0.0.1...	/nexacro17lib/framework/Platform.js?nexaversion=17.1.0.301
12	304	HTTP	127.0.0.1...	/nexacro17lib/framework/CssObjs.js?nexaversion=17.1.0.301
13	304	HTTP	127.0.0.1...	/nexacro17lib/framework/Device.js?nexaversion=17.1.0.301
14	304	HTTP	127.0.0.1...	/nexacro17lib/framework/Device_Android.js?nexaversion=17.1.0.301
15	304	HTTP	127.0.0.1...	/nexacro17lib/framework/Device_Windows.js?nexaversion=17.1.0.301
16	304	HTTP	127.0.0.1...	/nexacro17lib/framework/Device_iOS.js?nexaversion=17.1.0.301
17	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/Element_HTML5.js?nexaversion=17.1.0.301
18	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/Element_Runtime.js?nexaversion=17.1.0.301
19	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompBase.js?nexaversion=17.1.0.301
20	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
21	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
22	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
23	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
24	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
25	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
26	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
27	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
28	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
29	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
30	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
31	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
32	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
33	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
34	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
35	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
36	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
37	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
38	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301
39	304	HTTP	127.0.0.1...	/nexacro17lib/component/CompBase/CompEventBase.js?nexaversion=17.1.0.301

#	Result	Protocol	Host	URL
1	304	HTTP	localhost:8...	/edusys/EduMaster 17.1/_web_/index.html
2	304	HTTP	localhost:8...	/edusys/EduMaster 17.1/_web_/nexacro17lib/framework/Framework.js?nexaversion=17.1.0.301
3	304	HTTP	localhost:8...	/edusys/EduMaster 17.1/_web_/nexacro17lib/component/CompBase.js?nexaversion=17.1.0.301
4	304	HTTP	localhost:8...	/edusys/EduMaster 17.1/_web_/nexacro17lib/component/ComComp.js?nexaversion=17.1.0.301
5	304	HTTP	localhost:8...	/edusys/EduMaster 17.1/_web_/nexacro17lib/component/Grid.js?nexaversion=17.1.0.301
6	304	HTTP	localhost:8...	/edusys/EduMaster 17.1/_web_/nexacro17lib/component/DeviceAPI.js?nexaversion=17.1.0.301
7	304	HTTP	localhost:8...	/edusys/EduMaster 17.1/_web_/nexacro17lib/component/ListView.js?nexaversion=17.1.0.301

파일 Merge  
네트워크 리소스 감소

## 성능향상 가이드 – 공통 – JS Merge, Compress (3/4)

---

NEXACRO Javascript framework를 Merge, Compress 기능을 통해 파일 개수 및 사이즈 최적화 운영 시에는 NEXACRO Javascript framework (Library) 에 대해서는 필수 사용권장

- 기본 Merge시 5개의 파일이 생성되나, json파일 수정을 통한 1개 파일 변경 가능
  - 1개의 json으로 파일을 생성시 사이즈가 커지므로 최초 접속 시 다수의 파일을 동시에 받는 것보다 단점을 가지고 있으나, 캐시 처리 후에는 효율적임
- **Compiled JS는 NRE 실행환경에서만 가능하며, 보안측면을 고려할 때는 Compiled JS를 권장**
  - 개발 소스 보안측면을 고려할 때는 Compiled JS 권장

## 성능향상 가이드 – 공통 – JS Merge, Compress (4/4)

NEXACRO Javascript framework를 Merge, Compress 기능을 통해 파일 개수 및 사이즈 최적화 운영 시에는 NEXACRO Javascript framework (Library) 에 대해서는 필수 사용권장

- **Deploy 방법**

- nexacro Studio 메뉴의 Deploy Application를 통해 전체 빌드 가능

- Deploy File 선택을 통한 개별 파일 진행 가능

- 또는 제품 폴더의 **nexacrogenerator.exe**, **nexacrocompressor.exe**파일을 이용하여 배치파   
(단 windows 계열 Machine에서만 가능)

- 17.1 버전 파일명 변경 : **nexacrodeploy17.exe**

[http://docs.tobesoft.com/development\\_tools\\_guide\\_nexacro\\_17\\_ko#c8f6cde87ef24491](http://docs.tobesoft.com/development_tools_guide_nexacro_17_ko#c8f6cde87ef24491)

### [ 유닉스/리눅스 제너레이터 미지원 사유 ]

nexacro제너레이터를 사용하기 위해 내부적으로 mapping table를 만들어 사용함.

mapping파일을 만들기 위해서는 nexacro platform의 dll를 로드 하여 만들게 되는데

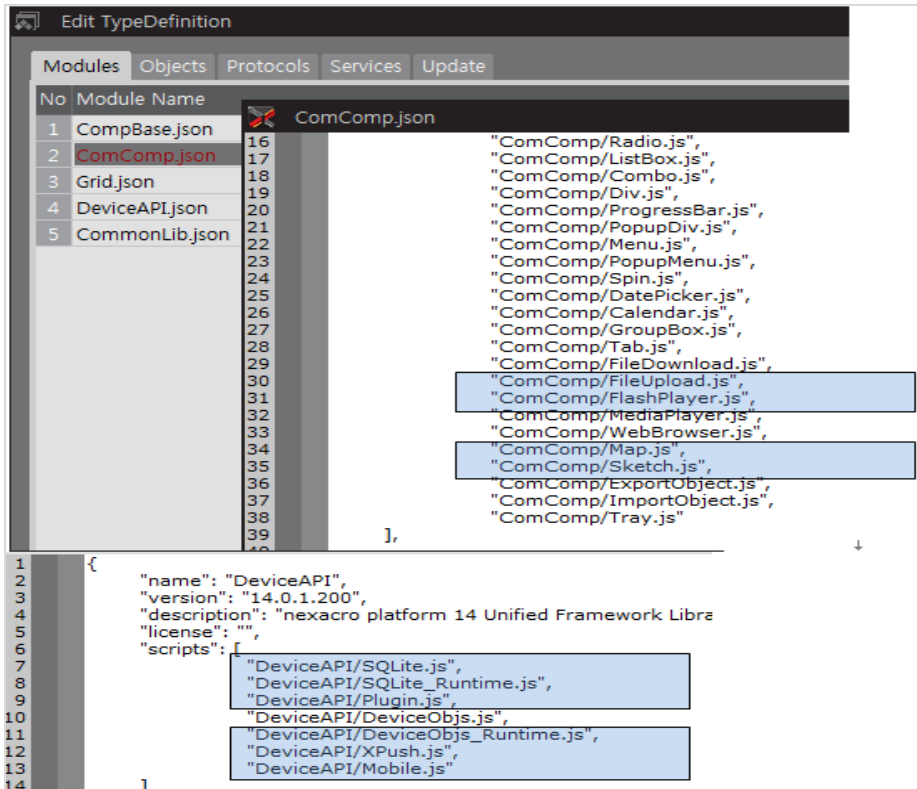
유닉스/리눅스용 미제공

또한 프로젝트 상에서 ActiveX 관련된 object를 선언되어 있을 경우 ActiveX정보는 Window에서만 가능

# 성능향상 가이드 - 공통 - 초기로딩 Js 최적화

NEXACRO 엔진모듈 중 사용하지 않는 Js파일을 제거하여 Js 최소화 한다.

일부 기능 또는 몇 개의 화면에서 사용하는 경우라면 필요한 요소만 대상 화면에 적용



소스 merge를 하게 되면, 파일사이즈 등은 크게 차이가 없어 몇 개의 js파일을 빼는 것이 성능을 개선할 수 있는 큰 팩트는 아닐 수 있으나 이를 파싱하는 단계들을 줄이고, 작지만 요소 요소의 튜닝 팩트를 모우는 것이 필요함.

# 성능향상 가이드 – 공통 – 최초 접속페이지 이미지 처리 (1/3)

초기 화면 로딩 시까지 시간이 지연되는 것을 시각적인 효과로 체감속도를 줄인다.

- 체감속도 감소

- 네트워크 상황이 좋지 못한 PC나, PC사양이 낮은 경우 최초 접속 시 로딩 지연시간이 발생함  
로딩 시간은 동일하나, 대기 시간 동안 이미지를 보여줌으로써 접속중임을 인지할 수 있도록 한다.

index.html

```
<div id="loadingImg" style="width:100%;height:100%;POSITION:absolute;z-index:1">  
<div style="position:absolute;top:50%;left:50%;margin:-83px 0 0 -110px">  
    
</div>  
</div>
```

```
function oninitframework()
```

```
{  
  nexacro._initHTMLSysEvent(window, document);  
  nexacro._prepareManagerFrame();  
  application._globalvalue = "gv_val1=a";  
  application.load("myproj", "test.xadl.js");  
}
```

```
function hideImage()
```

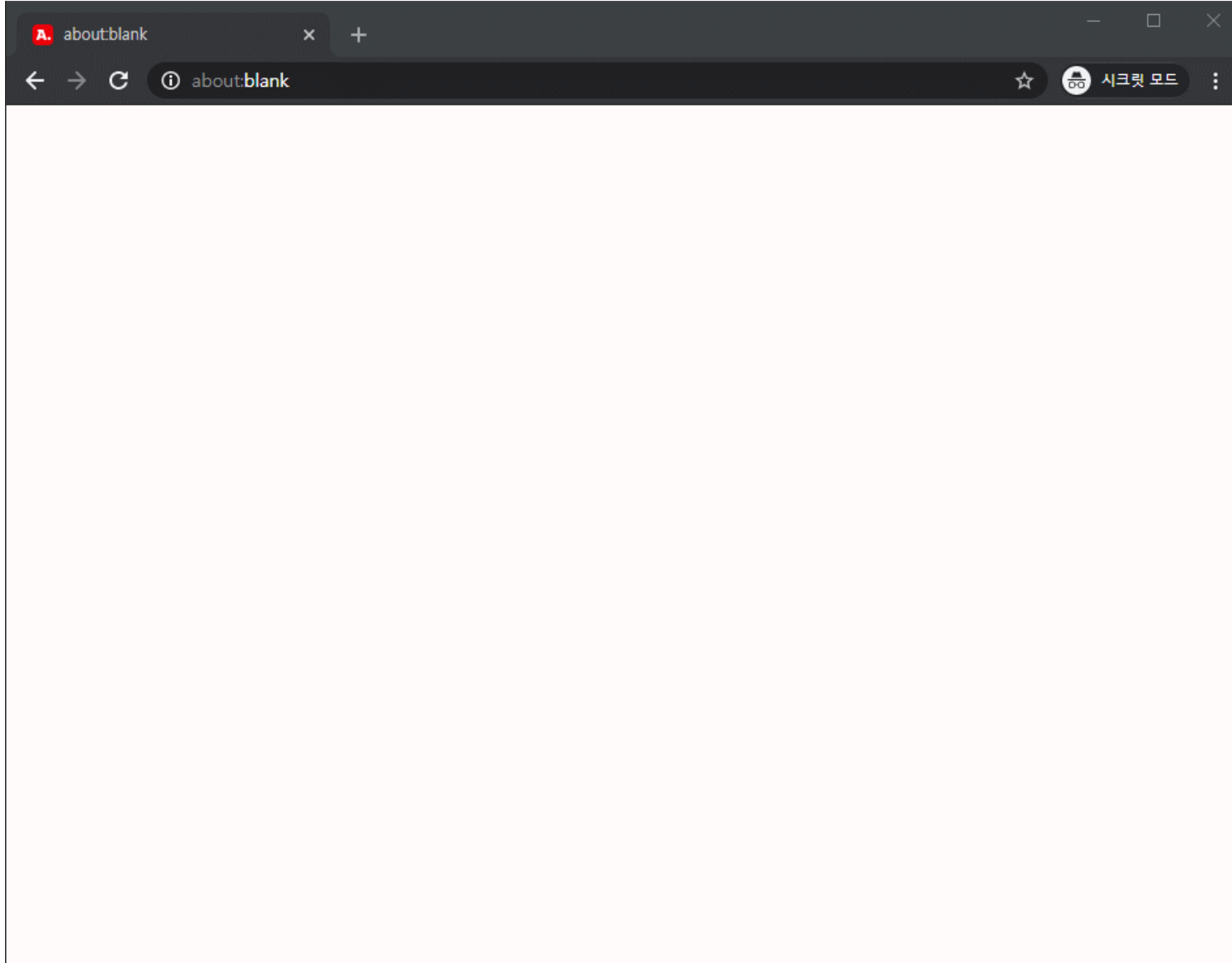
```
{  
  document.getElementById('loadingImg').style.display="none";  
}
```

```
// xadl application onload event에서 이미지 숨기기
```

```
this.application_onload =  
function(obj:Application, e:nexacro.LoadEventInfo)  
{  
  hideImage();  
}
```

# 성능향상 가이드 – 공통 – 최초 접속페이지 이미지 처리

<https://www.playnexacro.com/#show:learn:2817>



## 성능향상 가이드 – 공통 – defer (1/2)

---

Javascript defer 옵션을 통해 이미지를 먼저 보여준다.

- **defer**

- 초기 제품을 실행하기 위해서는 내부 엔진파일(js)를 다운로드 받게 되는데 이때 pc사용에 따라 다소 시간이 지연되나 defer옵션을 통해 이미지를 먼저 보여준다.

```
<script type="text/javascript" src="./nexacro14lib/framework/SystemBase.js" defer > </script>
```

“defer ” 속성이 설정된 경우 브라우저는 해당 요소를 만나는 즉시 코드를 내려 받지만 실행은 지연

## 성능향상 가이드 – 공통 – defer (2/2)

Javascript defer 옵션을 통해 이미지를 먼저 보여준다.

### ❖ 일반적인 실행

```
<script type="text/javascript" src="./nexacro17lib/framework/SystemBase.js" ></script>
```



### ❖ defer 옵션 추가 실행

```
<script type="text/javascript" src="./nexacro17lib/framework/SystemBase.js" defer ></script>
```



# 성능향상 가이드 – 공통 – 화면 개수 제한

Application 에서 사용할 수 있는 최대 화면 개수를 제한한다.

- **메모리**

- Application 실행 시 한계 메모리가 있음 이를 초과시 비정상 종료 및 이상동작  
(대표 현상 : 화면이 움직이지 않거나, 백지화 현상, 화면 깜박임 등)

- **오픈 화면 개수 제한**

- 화면 구성에 따라 메모리 사용량이 다르므로, Application개발 시 최대 오픈 할 수 있는 화면(MDI) 오픈 창 등의 개수를 제한한다.

**이 부분은 Application 설계 단계에서 업무 특성을 고려하여 최대 오픈 가능한 화면개수를 제한.**

# 성능향상 가이드 – 공통 – 로딩 페이지 단순화

---

최초 로딩 페이지는 가급적 단순하게 구성한다.

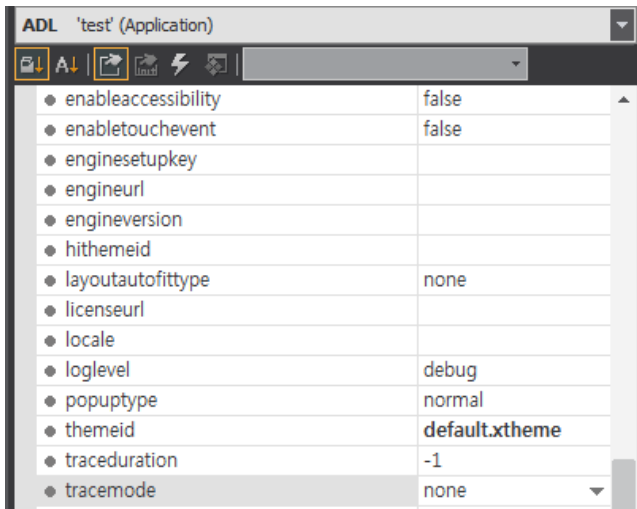
- 로딩 페이지 단순화

- 최초로 로딩되고 보여지는 페이지는 가급적 단순화하여 구성
- 메인 페이지에 많은 수의 데이터셋과 링크된 페이지, 많은 include등으로 구성하게 되면 이로 인해 로딩 시간이 길어지며, 초기 로딩속도 이슈가 원인이 됨

# 성능향상 가이드 – 공통 – tracemode

tracemode는 개발 시 유용하나 운영 시에는 성능에 영향을 미침  
운영 환경에서는 반드시 tracemode 설정을 off 및 trace문 주석처리

- tracemode는 개발 시 디버깅이 용이하나, 운영 시에는 성능에 영향을 줍니다.
  - 운영서버에는 trace를 off 하는 것을 권장합니다.
  - tracemode를 off 하더라도 trace문 내부의 성능에 영향을 주는 구문이 있다면 내부 코드가 실행되므로 trace문 자체를 주석 처리하도록 합니다.
- 개발 완료 후 운영 단계에서는 trace off 및 불필요한 trace문은 제거합니다.



```
// var _pApplication = nexacro.Application ;  
_pApplication.trace = function (log, level) {  
    alert("trace");  
};  
if (nexacro._Browser == "Runtime") {  
    nexacro._trace = function () {};  
    trace = nexacro._trace;  
} else {  
    window.trace = function () {}  
}
```

# 성능향상 가이드 - 공통 - 스타일 최적화

많은 이미지와 다양한 스타일을 적용하게 되면 응답시간 및 렌더링 등의 영향을 받음  
이미지는 손실률이 없고, 높은 압축률을 보이는 png사용 권장

- 성능을 고려하여 페이지 구성 시 이미지는 최소화
  - 컴포넌트의 스타일로 구성이 가능하다면 이로 대체
  - 컴포넌트의 다양한 스타일을 구성할 수 있으나 스타일이 복잡 할수록 렌더링에는 영향을 줌  
즉 성능에 영향을 미침

## 렌더링?

개발자가 만든 JS를 제품에 파싱 및 처리과정을 통해 화면을 그리는 과정

Column0	Column1	Column2	Column3	Column4	Column5
안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.
안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.
안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.
안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.
안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.

Column0	Column1	Column2	Column3	Column4	Column5
안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.
안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.
안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.
안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.
안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.
안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.	안녕하세요.

상단 이미지는 기본 형태이나 하단 이미지는 그리드의 edittype과 padding이 적용된 경우입니다.

하단의 그리드를 HTML 형태로 만들기 위해서는 더 많은 node가 필요할 것입니다.

- **Padding 줄 경우 node가 추가적으로 생성**
  - padding을 줄 경우 node가 추가적으로 생성되므로 성능향상을 위해서는 권장하지 않음

## 성능향상 가이드 - 공통 - 공통함수 구성 (1/3)

모든 폼에서 모든 공통함수를 include하는 방식보다는 필요한 부분만을 include하여 최적화  
공통영역 성능은 담당자의 구현 및 컨설팅 역량에 따라 상당한 영향을 줌

- **공통으로 사용하는 include함수 최소화**

- 모든 폼에서 사용하는 공통함수를 include하는 방식보다는 필요한 부분만을 include하여 최적화 함
- 특정 업무에서만 사용하는 함수는 공통에서 제외하고 해당 폼에서 처리

- **공통영역에 대한 부분은 담당자의 역량에 따라 성능좌우**

- 공통영역에 대한 부분은 Application의 전체 성능에 영향을 주므로 개별 화면에 해당하는 부분은 해당 폼에서만 구현해 사용하도록 하고, Application전체에 해당하는 부분에 대해서만 공통영역으로 만들어 사용을 해야 함.

(즉 공통담당자는 모든 함수를 공통모듈로 빼지 말고 타당성을 정확히 파악 후 진행)

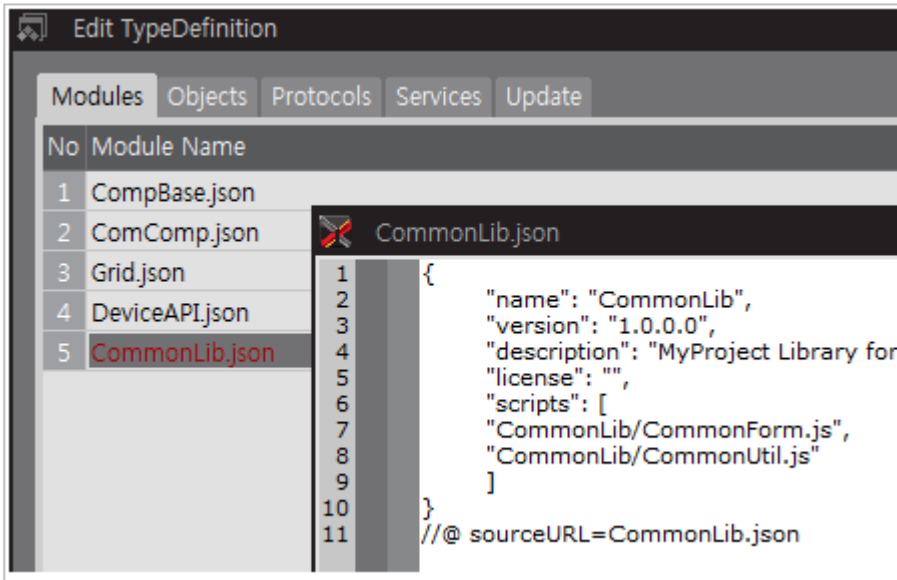
**공통영역은 단순 몇몇 화면의 이슈 해결이 아닌 Application 전체 영향을 미침**

# 성능향상 가이드 - 공통 - 공통함수 구성 (2/3)

## 공통함수를 폼이 아닌 json에 등록하여 최초 1회만 로딩하도록 구성

- 공통함수 폼이 아닌 json형태로 등록

- 공통함수를 폼마다 include 하지 않고 json에 등록하여 최초 1회만 로딩하도록 구현  
즉 업무화면에서는 include가 없도록 구성



### 공통함수 예시 (CommonForm.js)

```
var _pLoadManager = nexacro.LoadManager.prototype;
_pLoadManager.on_load_datamodule = function (svcid, errstatus,
message, fireerrorcode, returncode, requesturi, locationuri)
{
    var load_Item = this.getDataItem(svcid);
    if (load_Item)
    {
        var callback_id = load_Item._context_callback;
```


## 성능향상 가이드 – 공통 – 공통함수 구성 (3/3)

공통함수를 jon처리 후에도 include가 필요한 경우 다수의 공통함수를 include하는 것 보다는 필요한 파일만 include하여 사용한다.

- 하나의 공통함수에 여러 개의 js를 include ?

- 하단의 main.xjs를 include 하고 다수의 js가 포함되어 있는 경우 불필요한 js가 포함됨
- 해당 업무에 필요한 js만 include하도록 세분화 필요

```
include "common::main.xjs";
```



```
include "common::grid.xjs";  
include "common::util.xjs";  
include "common::report.xjs";  
include "common::menu.xjs";  
include "common::service.xjs";  
include "common::file.xjs";  
include "common::component.xjs";
```

## 성능향상 가이드 - 공통 - 공통코드 구현

---

폼에서 사용하는 공통코드는 form의 onload시 호출하게 되므로 반드시 async 통신 사용  
다수의 공통코드는 1:1 Dataset 보다는 1:N Dataset를 받도록 구성한다.

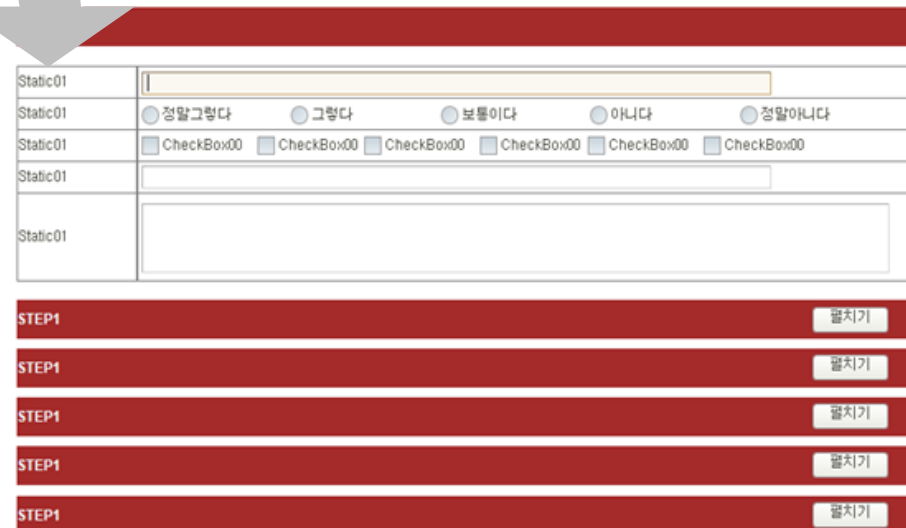
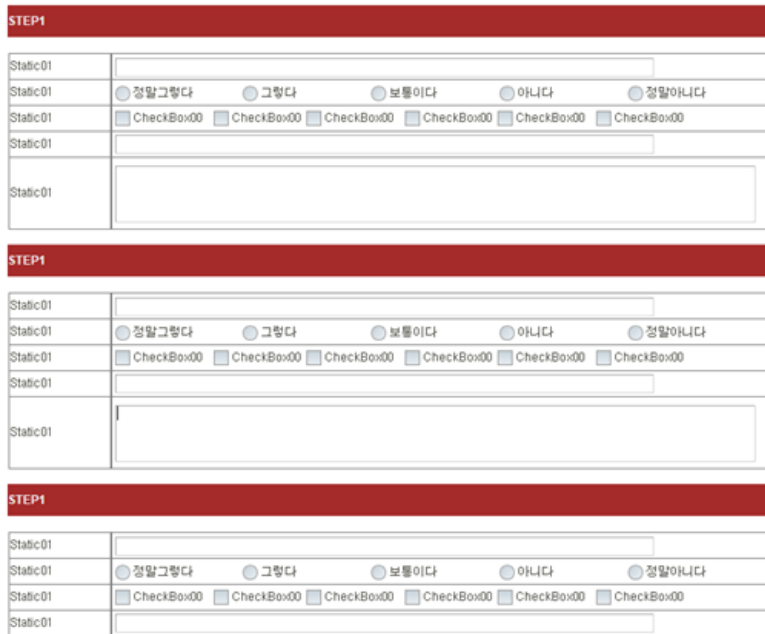
- 폼 onload 서비스는 비동기 사용
  - 공통코드는 주로 폼의 onload이벤트에서 서비스를 호출하는 경우가 많으므로, 가능한 비동기 사용을 권장한다.
  - 다수의 Dataset을 조회하는 경우에는 1:1보다는 1:N Dataset를 받아들 수 있도록 구성
- 참고
  - **공통코드**는 쉽게 변경이 되지 않기 때문에 한번 호출한 데이터는 **Global dataset**으로 관리하며, 메모리상의 데이터를 재활용하여 서비스 시간을 줄일 수 있다.

# 성능향상 가이드 - 공통 - 화면 단순화 및 분할-(1/2)

다수의 컴포넌트로 구성된 복잡한 화면의 경우 브라우저 사양에 따라 성능저하 원인이 된다.  
초기 화면에 표현되지 않아도 되는 경우 특정 영역을 Div등으로 구성하여 후 처리한다.

- Div 영역 분할

- 다수의 Component로 구성된 복잡한 형태의 화면의 경우 브라우저 사양에 따라 성능 저하 원인 초기 화면에 표현되지 않아도 되는 경우 특정 영역을 Div 등으로 구성하여 후 처리
- 주로 CS화면을 컨버전 하는 경우 다소 복잡한 형태가 많으니 초기 개발 설계 시 주의해야 한다.



## 성능향상 가이드 - 공통 - 화면 단순화 및 분할-(2/2)

다수의 컴포넌트로 구성된 복잡한 화면의 경우 브라우저 사양에 따라 성능저하 원인이 된다.  
초기 화면에 표현되지 않아도 되는 경우 특정 영역을 Div등으로 구성하여 후 처리한다.

- 기본 페이지를 보여주고 후 처리  
- Timer를 통해 후 처리과정 분리

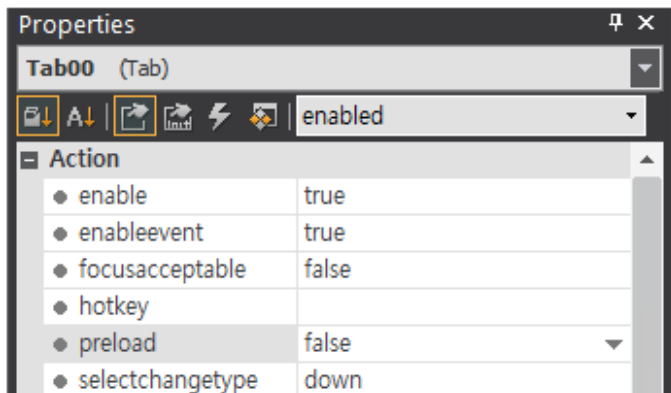
The image shows a form with several input fields: a text box, a row of radio buttons with labels '정말그렇다', '그렇다', '보통이다', '아니다', '정말아니다', a row of six checkboxes labeled 'CheckBox00', another text box, and a larger text area. Below the form, there are five identical dark red buttons, each labeled 'STEP1' and '끝치기' (End).

```
this.form_onload =  
    function(obj:nexacro.Form,e:nexacro.LoadEventInfo)  
    {  
        this.setTimer(1,1000);  
    };  
  
this.form_ontimer =  
    function(obj:nexacro.Form,e:nexacro.TimerEventInfo)  
    {  
        this.Div00.set_url("Application::Application_eval.xfdl");  
    };
```

## 성능향상 가이드 – 공통 – Tabpage

다수의 Tabpage로 구성되고 각 페이지가 url link되어 있는 경우 성능에 영향을 주는 경우 Preload 옵션을 통해 초기 로딩시간을 단축할 수 있다.

- **preload 속성 false 권장**
  - Tabpage에 연결된 Form의 onload 이벤트 발생 시점을 지정하는 속성입니다.

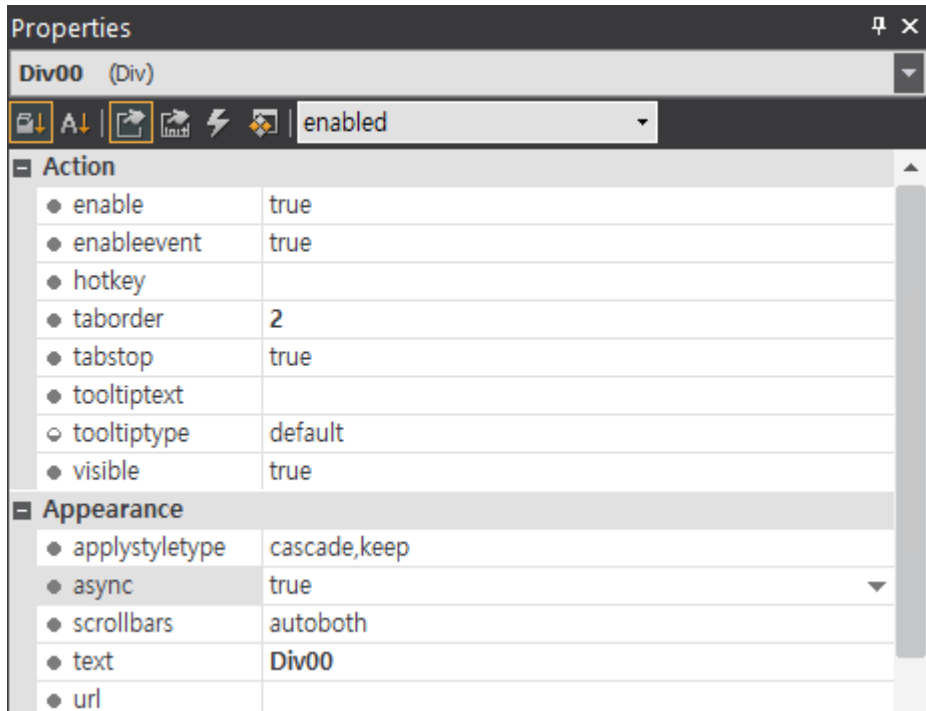


주의사항 : 폼에서 각 Page의 값을 참조하는 경우 해당 page가 로딩되어 있는지 체크해야 함

# 성능향상 가이드 – 공통 – Tabpage, Div, PopupDiv

## Tabpage url 링크 방식 구성 시 async 속성 false시 성능저하

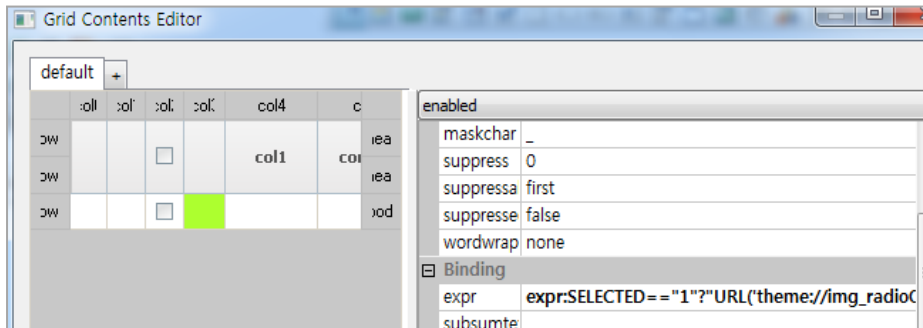
- **async 속성 true 권장 (default : true)**
  - Tabpage에 연결된 Form의 로드 시 TabPage의 child를 접근가능 한 시점을 지정하는 속성입니다.



## 성능향상 가이드 – 공통 – Grid - Expression (1/2)

그리드의 `expr` 구문이 많은 경우 성능에 많은 영향을 줌, 따라서 최소화 및 간결하게 작성해야 한다.  
데이터가 많은 경우 데이터 변경 및 스크롤 시 그리드 영역이 다시 표현되므로 성능저하 원인

- **Expr 사용 최소화 및 단순화**
  - 그리드에서 Expr 사용은 성능에 가장 큰 영향을 줌



### 그리드

`expr:comp.parent.fn_Compare(...)`

### 화면

```
this.fn_Compare = function() {  
    this.Dataset01.findRow(...);  
    this.Dataset01.filter(...);  
    return;  
}
```

\* Expression의 구문 처리가 복잡할수록 그리드 성능은 떨어짐.

**expr이 하나라도 있는 경우 페이지이동(스크롤,키 이동)등 전반적으로 영향을 미침.**

## 성능향상 가이드 – 공통 – Grid - Expression (2/2)

---

그리드의 expr 구문이 많은 경우 성능에 많은 영향을 줌, 따라서 최소화 및 간결하게 작성해야 한다.  
데이터가 많은 경우 데이터 변경 및 스크롤 시 그리드 영역이 다시 표현되므로 성능저하 원인

- **Expr-> Dataset Bind처리**

- 그리드 수식 칼럼 등을 expr형태로 표현하는 경우 Dataset의 수식 칼럼을 만들어 사용  
예) 조회 후 callback 함수에서 수식 칼럼을 만들어 주는 스크립트를 추가

# 성능향상 가이드 – 공통 – enableevent (1/2)

## Dataset enableevent 사용시 주의사항

- **Dataset의 enableevent 사용 시 주의사항 (미 사용)**

- 아래와 같이 setColumn 메소드를 사용하게 되면 Dataset의 oncolumnchanged이벤트가 발생합니다.

```
for(var i=0;....)
{
    ....
    this.Dataset01.setColumn(i,"COL01",...);
}
```

setColumn 메소드가 호출되면 데이터셋에서 다음과 같은 이벤트가 발생합니다.  
cancelcolumnchange (return = true) → oncolumnchanged

- **Dataset의 enableevent 사용 시 주의사항 (사용)**

```
this.Dataset01.set_enableevent(false);
for(var i=0;....)
{
    ....
    this.Dataset01.setColumn(i,"COL01",...);
}
this.Dataset01.set_enableevent(true);
```

enableevent를 false처리를 하게 되면 **Dataset이벤트가 중지 됨**



# 성능향상 가이드 – 공통 – enableredraw (1/2)

## Grid Contents를 Dynamic하게 생성 시 *enableredraw* 사용

- **Grid의 *enableredraw* 사용 시 참고사항**

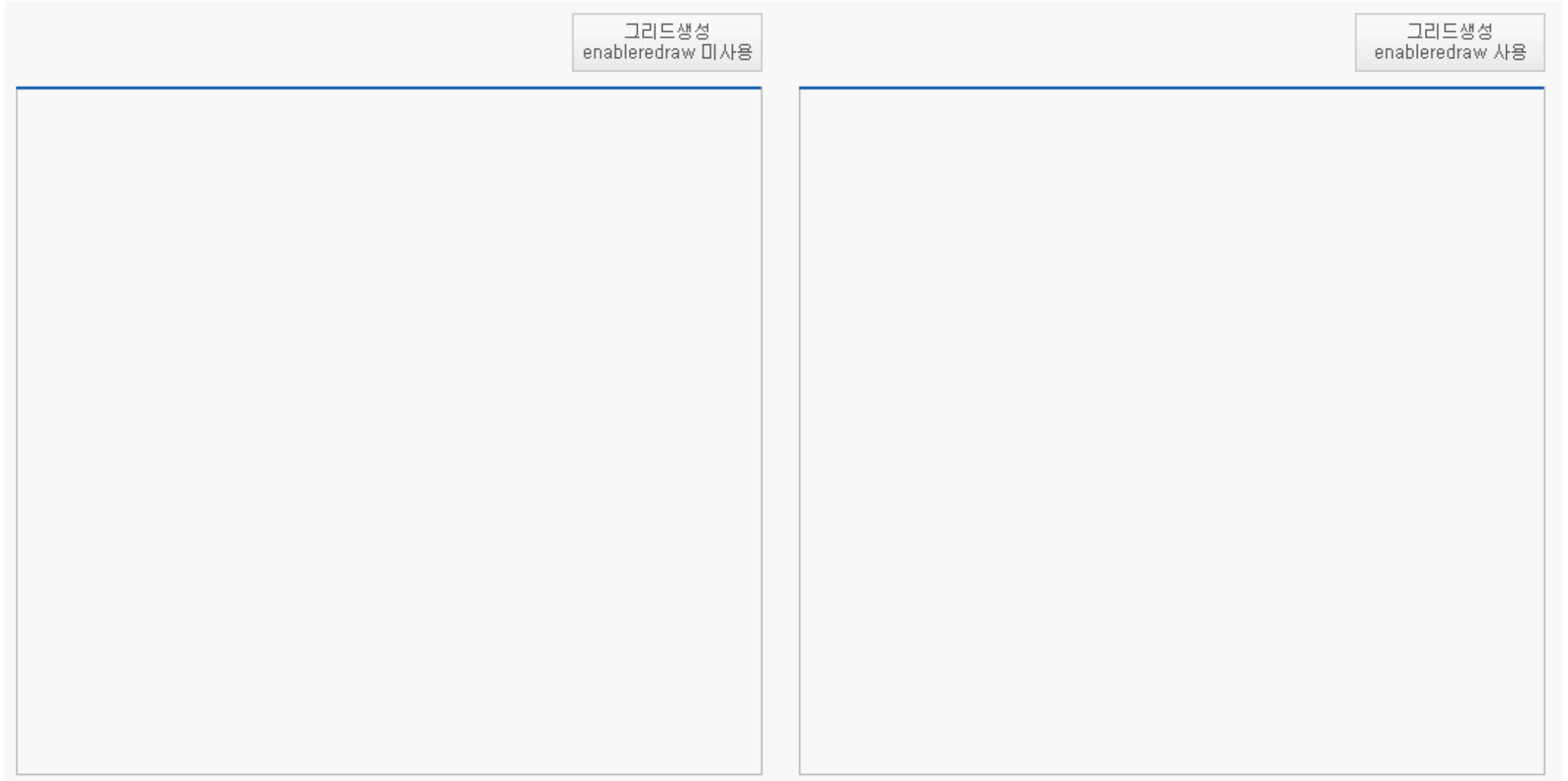
- Grid 에 변경이 발생하면 화면을 자동으로 다시 그릴 지 설정하는 속성입니다.
- *enableredraw* 사용여부에 따라 속도 차이가 많이 발생함.

```
this.Grid01.set_enableredraw(false);  
for(var i=0;.....)  
{  
    ....  
    this.Grid01.setProperty(... ..)  
}  
this.Grid01.set_enableredraw(true);
```

Grid를 **Dynamic**하게 스크립트를 이용하여  
Contents를 생성하는 경우 해당 옵션 사용 권장

## 성능향상 가이드 – 공통 – enableredraw (2/2)

Grid Contents를 Dynamic하게 생성 시 *enableredraw* 사용



# 성능향상 가이드 – 공통 – fastvscrolltype

Grid 컴포넌트에서 제공하는 속성으로 세로 스크롤 시 표시되는 데이터의 개수를 제한하여 Reflow, Repaint 횟수를 줄여 성능 개선 효과를 볼 수 있다.

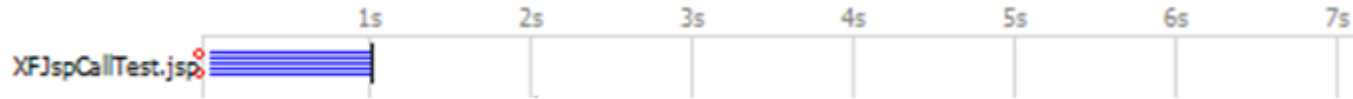
- Grid의 fastvscrolltype 사용 시 참고사항

- Grid의 경우 복잡한 DOM Tree로 이루어져 있으므로 이 또한 성능에 영향을 주게 됩니다. fastvscrolltype은 스크롤 시 일부 데이터만 갱신하는 형태로 성능을 개선할 수 있습니다.

chk	Column0	Column1	Column2	Column3	Column4	Column5	Column6
0	강감찬	조회	서울특별시 중구	321,458	20170101	321,458	고려시대 명장, 본
0	김부식	입력	경상북도 경주시	425,862	20170201	425,862	고려시대 문신, 본
0	정몽주	수정	경기도 수원시 경	57,786	20170601	57,786	고려 말 문신, 본
0	김유신	삭제	경상북도 영주시	51,651	20170931	51,651	삼국통일을 이룩
0	최무선	조회	충청남도 부여군	216,548	20171005	216,548	고려 말, 조선 초
0	정중부	입력	경기도 여주시 북	216,218	20171111	216,218	고려시대 무신, 본
0	김부식	수정	서울특별시 종로	954,561	20170101	954,561	고려시대 문신, 본
0	안중근	삭제	경상북도 경주시	2,121,684	20170201	2,121,684	한말의 독립운동
0	안창호	조회	경기도 이천시 권	6,549,845	20170601	6,549,845	한말의 독립운동
0	유관순	입력	충청남도 부여군	65,150	20170931	65,150	한말의 독립운동
0	이종무	조회	전라북도 익산시	158,886	20170429	158,886	고려 말, 조선 초
0	한명회	입력	부산광역시 연제	258,624	20170111	258,624	조선 전기의 문신
0	정몽주	수정	제주특별자치도	4,569,985	20170829	4,569,985	고려 말 문신, 본
0	문익점	삭제	경상남도 합천군	1,245,568	20170605	1,245,568	고려시대 학자, 본
0	최영	조회	서울특별시 종로	336,985	20170614	336,985	고려 말 무신, 시
0	장영실	입력	전라북도 남원시	578,962	20170231	578,962	조선 시대의 과학

# 성능향상 가이드 - 공통 - 통신 후 간결화

데이터를 조회 및 저장 후 실제 서비스 수행시간 대비 응답속도가 느린 경우 점검사항  
서비스 시작, 끝 callback 이후 수행 문장 별 수행시간 등을 로그로 남겨 원인 추적



```
this.transaction( "MyService01" ,... ", "Dataset03=output1", "a=b", "callbackFunction");
```

```
this.callbackFunction = function (sid, nErrorCode, nErrorMsg)
```

```
{  
  if (nErrorCode >= 0)  
  {  
    this.dstaset.set_enableevent(false);  
    for(var i=0;i<this.dataset;i++)  
    {  
      //데이터셋 재가공 작업 등 후처리  
    }  
    this.dstaset.set_enableevent(true);  
  }  
}
```

Required Time (Seconds) : **3**      The number of queries: **100,000**

Seq	Address	Company Name	HomePage	Technical Support HomePage
1	5F INTOPS Building 108-7 Samsung-Dong	TOBESOFT Co Ltd	www.tobesoft.com	www.xplatform.com
2	5F INTOPS Building 108-7 Samsung-Dong	TOBESOFT Co Ltd	www.tobesoft.com	www.xplatform.com
3	5F INTOPS Building 108-7 Samsung-Dong	TOBESOFT Co Ltd	www.tobesoft.com	www.xplatform.com
4	5F INTOPS Building 108-7 Samsung-Dong	TOBESOFT Co Ltd	www.tobesoft.com	www.xplatform.com

# 성능향상 가이드 - 공통 - 대량 데이터 처리

대량의 데이터는 처리에 따른 대기 시간으로 사용자의 불편함과, 다량의 데이터 메모리 적재 한계, 대량 데이터는 페이징, 분할전송, 조회조건 제한 등 사전에 문제가 되지 않도록 개발방향 검토

No	책이름
1	1등등의 비밀노트(조선일보 교육팀장 양근만 기자가 공개한)
2	공부의 비결(누구나 알지만 아무도 모르는)
3	모성혁명
4	비포비포 119 이유식
5	우리 아이 공부비결 체계에 숨어있다
6	초등 독서가 대학을 결정한다(초등 1,2학년)
7	현명한 부모들이 꼭 알아야 할 대화법
8	5분 뇌호흡
9	건강검진(12000원으로 받는)
10	과자 내 아이를 해치는 알콜한 유족

특정 서비스 분할 전송으로 변경

대량 데이터 화면의 조건 제약

기준일 2008-06-12 - 2013-06-29

```
mainRs = pstmt.executeQuery();
HttpPartPlatformResponse res =
    new HttpPartPlatformResponse(response, PlatformType.CONTENT_TYPE_BINARY);
while (mainRs.next())
{
    row = out_ds0.newRow();
    ..
    if(..)
    {
        res.sendDataSet(out_ds0);
    }
    ..
}
```

Alert

! 기준일은 3개월을 초과 할 수 없습니다.

확인

# 성능향상 가이드 – 공통 – 데이터셋을 이용한 파일전송

---

파일 전송은 File up/down 컴퍼넌트 사용 권장

데이터셋을 이용하는 경우 일괄 처리에 따른 PC 및 서버 비용 증가.

- **File up/down 기능**
  - 파일 전송은 Filter up/down 컴포넌트 사용 권장
- **Dataset 이용 시 주의사항**
  - 데이터셋을 이용하여 파일 전송하는 경우 데이터셋 처리과정에서 PC 및 서버 비용증가
  - 사이즈 제약에 따른 오류발생
- **상용제품 사용**
  - 파일 전송 프로그램 도입 사용

대용량의 파일 사이즈를 전송하는 경우 서버 리소스 부족에 따른 장애 발생가능, 즉 파일사이즈 제약필요

## 성능향상 가이드 – 공통 – Open창 제약사항

---

WRE 실행환경의 경우 application.open 즉 HTML5의 window.open을 사용할 경우 엔진 재로딩 및 파싱  
단순 open창은 브라우저 기능(html)실행 권장

가급적, 브라우저 내부에서 실행 되도록 창을 구성하거나,  
MDI 형태로 구성하도록 권장하며 Open창은 최소로 사용하도록 합니다.

## 성능향상 가이드 – 공통 – resize, visible

---

화면 size변경, visible 속성 변경 또는 동적으로 콘텐츠를 생성하는 경우 브라우저는 해당 시점마다 redraw 및 reflow 발생, 즉 과도한 콘텐츠의 경우 이런 동작이 반복되는 경우 성능에 영향을 미침

```
// form loading 시  
this.form_onload = function(...)  
{  
  this.div00.set_visible(false); // reflow , redraw  
  this.div00.set_height(0); // reflow , redraw ....  
}
```

## 성능향상 가이드 – 공통 – 이벤트 내 과도한 스크립트

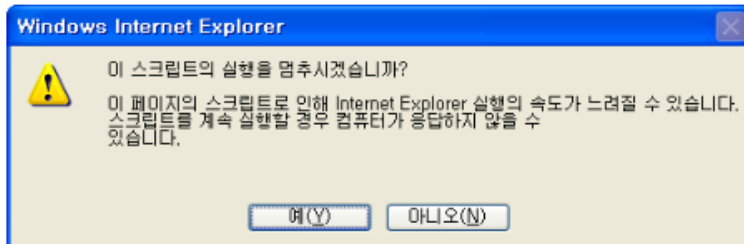
---

이벤트 내에서 과도하게 많은 스크립트가 수행되면, 이로 인한 체감 성능이 떨어짐

특히, 그리드 및 데이터셋 등의 이벤트 내에서 점검이 필요하며, 다수의 이벤트가 중복되지 않게 주의

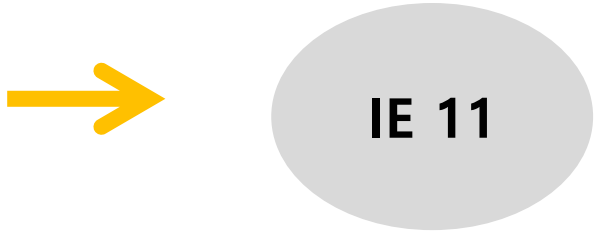
# 성능향상 가이드 - 공통 - IE8 스크립트 중단 메시지

IE8 웹 브라우저의 경우 한번의 스크립트 수행 시 5백만 개 이상의 명령을 수행하는 경우 경고 메시지 출력, 메시지 출력을 방지하기 위해서는 스크립트가 과도하게 수행되지 않도록 처리 필요



```
for(var i= .....) { .... }  
  
timer를 이용하여 분리  
  
for(var i= .....)  
{ ....  
  if(i == 1000)  
  {  
    tnis.tempLoop = i;  
    this.setTimer(1,10); return;  
  }  
}
```

HTML5 환경으로 개발 하고자 하는 경우  
**운영환경 고려 시**  
사용 가능한 브라우저 정보를 정할 것을 권장



# 성능향상 가이드 – 공통 – 디자인 처리(IE9)

---

IE9 이하 사용자를 위한 디자인 참고사항

WRE 실행환경에 해당됨.

- **Round 처리 최소화**

- IE9이하에서 CSS 처리시 bordertype의 round로 설정 시 성능에 영향을 줄 수 있으므로 round처리는 최소화 한다.

- **Alpha 사용 최소화 (XPLATFORM, NEXACRO 14 해당)**

- IE9이하에서 CSS 처리시 Alpha값을 처리할 경우 성능에 영향을 줄 수 있으므로 가급적 사용을 제한한다.

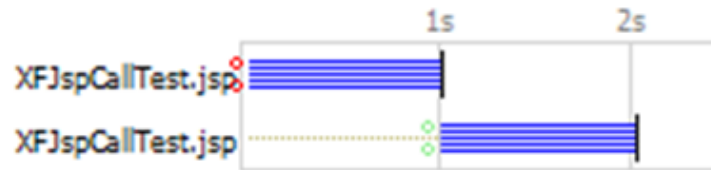
예) #11111111 => #111111ff

사용자의 브라우저 버전은 IE 10이상 사용을 권장

# 성능향상 가이드 - 공통 - 통신방식

통신방식은 동기/비동기 방식을 선택하여 사용 가능하나, 비동기 방식을 권장함.

- 동기(transaction) 통신은 통신 후 메인 thread가 locking되고 응답 후 처리를 함
  - 이로 인해 통신 요청 후 응답이 오기까지 어플리케이션이 대기상태에 빠짐
  - 화면 로딩 시 동기통신을 사용하여 서비스 호출 시 응답시간이 지연되는 경우 백지현상 발생
  - 폼의 onload이벤트에서 서비스 호출 시 비동기 통신 권장
  - 동기/비동기를 동시에 호출하는 경우 동기통신의 응답결과에 영향을 받음
- 통신의 경우 움직이는 이미지 등 대기 이미지가 보여지지 않는 현상이 발생하므로 비동기 권장
- 1초가 소요되는 동기통신을 5개 호출하는 것 보다 5개를 비동기로 요청하는 것이 더 빠른 응답결과를 보임
  - 1개의 요청으로 1개의 Dataset이 아닌 1:N 형태로 서비스를 구성



주로 개발 시에는 테스트 데이터 량이 적어 동기통신의 경우에도 큰 영향이 없으나,  
실제 운영환경에 적용 시 주요이슈 문제 중 하나

# 성능향상 가이드 – 공통 – 통신방식(SSV 1/2)

## NEXACRO는 XML, SSV, Binary(NRE 전용) 통신포맷 지원

- 개발자 도구 또는 네트워크 분석 툴 등으로 확인을 위해서는 XML통신이 식별이 쉬울 수 있으나, 사용자 테스트 서버 및 운영서버에서는 SSV통신을 권장.
- SSV 통신은 CSV통신과 유사한 형태로 XML이 불필요한 TAG를 제거 하므로 보다 적은 네트워크 비용 사용
  - 또한 XML파싱 보다 빠른 성능

요청 헤더	요청 본문	응답 헤더	응답 본문	쿠키	시작자
1	SSV:utf-8	ErrorCode:int=0	ErrorMsg:string=SUCC	Dataset:output1	_Const_const1:

요청 헤더	요청 본문	응답 헤더	응답 본문	쿠키	시작자
1	<?xml version="1.0" encoding="utf-8"?>				
2	<Root xmlns="http://www.nexacro.com/platform/dataset" ver="5000">				
3	<Parameters>				
4	<Parameter id="ErrorCode" type="int">0</Parameter>				
5	<Parameter id="ErrorMsg" type="string">SUCC</Parameter>				
6	</Parameters>				
7	<Dataset id="output1">				
8	<ColumnInfo>				
9	<ConstColumn id="const1" type="string" size="32">				
0	<ConstColumn id="const2" type="string" size="32">				
1	<ConstColumn id="const3" type="string" size="32">				
2	<Column id="column_1" type="string" size="255"/>				
3	<Column id="column_2" type="string" size="255"/>				
4	<Column id="column_3" type="string" size="255"/>				
5	<Column id="column_4" type="string" size="255"/>				
6	<Column id="column_5" type="string" size="255"/>				
7	<Column id="column_6" type="string" size="255"/>				
8	<Column id="column_7" type="string" size="255"/>				

송신은 transaction API, 수신은 서버 API인 PlatformResponse에서 통신 방식을 지정합니다.

서버 framework 구성 시 request 방식에 따라 response방식을 지정하면 효율적입니다.

(즉, client가 SSV이면 server도 SSV, client가 XML이면 server도 XML)

# 성능향상 가이드 – 공통 – 통신방식(SSV 2/2)

SSV 통신 포맷인 X-API의 설정 값 변경이 아닌 String연산을 통해 response해야 성능향상 효과를 볼 수 있다.

- X-API를 통해 데이터 포맷을 설정할 수 있음
  - X-API를 이용하여 SSV포맷을 만들게 되면 내부 처리과정에서 Dataset 등을 생성하게 되는데 이 과정에서 많은 시간이 소요 됨
- SSV 통신을 통한 성능 개선을 하기 위해서는 기본 Java의 String연산으로 출력 하도록 로직 구현

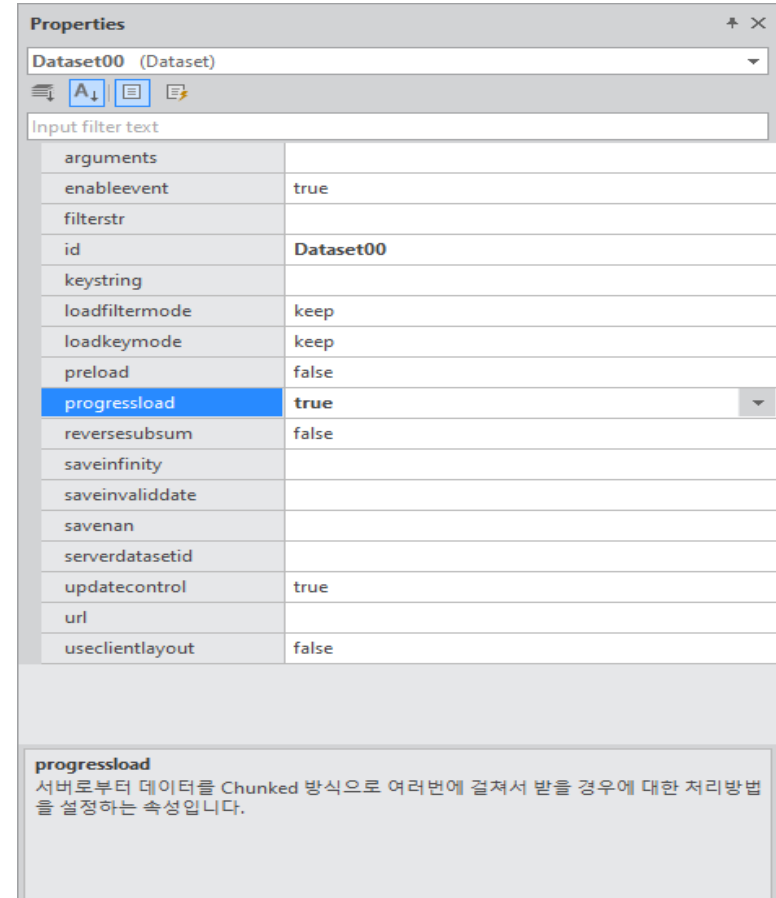
```
PlatformResponse platformResponse = new  
PlatformResponse(response.getOutputStream(),  
PlatformType.CONTENT_TYPE_SSV, strCharset);
```

```
<%@ page contentType="text/html;charset=euc-kr" %>  
<%@ page language="java"%>  
<%@ page import="java.io.*" %>  
<%@ page import="java.sql.*" %>  
<%@ page import="java.util.*" %>  
<%  
    char a = (char) 0x1e;  
    char b = (char) 0x1f;  
    String RS = String.valueOf(a);  
    String US = String.valueOf(b);  
    out.print("SSV:utf-8"+RS);  
    out.print("ErrorCode=0"+US+"ErrorMsg= Succ!!!!!!"+RS);  
    out.print("Dataset:output"+RS);  
  
    out.print("_RowType_"+US+"no:String(6)" +US+"empName:String(32)" +US+  
        "zipCode:String(6)" +US+"tel:String(11)" +RS);  
    for(int idx=0; idx<10; idx++)  
    {  
        out.print("N"+US+Integer.toString(idx)+US+"Mason"+US+"123456"  
            +US+"01045678901"+RS);  
    }  
    out.print(RS);  
>%>
```

# 성능향상 가이드 - 공통 - 대용량 데이터

## 대용량 데이터 처리시 Chunked 방식 적용

- Server Service구현을 FirstRow방식 즉 Chunked방식 구현
- Client의 Dataset 속성에서 progressload를 true설정  
단(IE8,9는 해당 옵션 동작 안함)



Chunked 된 데이터를 수신중에 onload 이벤트가 발생하면 DSLoadEventInfo 의 reason 속성값에 1 이 전달됩니다.  
데이터를 모두 수신하여 onload 이벤트가 발생하면 DSLoadEventInfo 의 reason 속성값에 0 이 전달됩니다.

**nexacro17 Education - New Form** \_ □ ×

First  Total   
 Next  Sleep

No	col_0	col_1	col_2	col_3	col_4	col_5	col_6

TIME	MES



---

## 02. 환경

## 성능향상 가이드 – 환경 – gzip(압축) (1/3)

브라우저에서 제공하는 gzip방식을 이용하여 패킷 사이즈 절감  
약 70%의 압축률을 보이거나 이미지 등은 제외한 텍스트 형식을 js만 적용

```
<Tomcat 기준 : Server.xml]
Connector compressableMimeType="application/javascript,text/xml,text/html,text/javascript,application/xml ,
text/css,application/x-javascript,text/json,application/json,text/plain"
compression="off"
URIEncoding="UTF-8"
useSendfile="false"
compressionMinSize="2048"
connectionTimeout="20000"
port="8080"

...중략 ...

compression="off" 를
on으로 설정 시 압축되어서 속도 향상이 가능합니다.
```

설정  
방법

WAS별 설정방법은 해당 벤더사에 문의하시기 바랍니다.

## 성능향상 가이드 – 환경 – gzip(압축) (2/3)

브라우저에서 제공하는 gzip방식을 이용하여 패킷 사이즈 절감

약 70%의 압축률을 보이거나 이미지 등은 제외한 텍스트 형식을 js만 적용

- gzip 파일을 미리 생성시 서버에서 gzip으로 압축하는 비용을 줄일 수 있음
  - 다만, 파일 변경 시 변환 작업이 필요하므로 관리 팩터가 늘어남
- gzip 파일 생성은 unix machine에서 제공하는 gzip명령 또는 Window Util(7zip) 가능
  - gzip설정 시 서버 비용이 증가되므로, 미리 gzip된 파일을 다운로드 할 수 도 있음
  - 이 경우 응답헤더에 Content-Encoding type을 gzip으로 명시되어야 함
  - 응답코드에 명시하는 방법은 서버에서 설정하는 방법과 url filter를 이용

참고 :

WebToB 미리 압축된 파일에 대한 환경설정파일

\*HEADERS

```
gzip action="AddResponse",  
FieldName="Content-Encoding", FieldValue = "gzip",  
RegExp="W.(js.gz)$" <-- 정규식을 이용하여 처리
```

web.xml

```
<filter-mapping>  
<filter-name>gzipFilter</filter-name>  
<url-pattern>*.gz</url-pattern>  
</filter-mapping>  
<filter>  
<filter-name>gzipFilter</filter-name>  
<filter-class>com.gzip_nc.gzip_nc</filter-class>  
</filter>
```

**주의 : gzip을 중복해서 사용하는 경우 포맷오류 발생**

# 성능향상 가이드 – 환경 – gzip(압축) (3/3)

브라우저에서 제공하는 gzip방식을 이용하여 패킷 사이즈 절감

약 70%의 압축률을 보이거나 이미지 등은 제외한 텍스트 형식을 js만 적용

```
package com.gzip_nc; import java.io.IOException; import
javax.servlet.Filter; import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class gzip_nc implements Filter{ @Override public void destroy()
{
// TODO Auto-generated method stub } @Override public void
doFilter(ServletRequest arg0, ServletResponse arg1, FilterChain arg2)
throws IOException, ServletException {

HttpServletRequest httpRequest = (HttpServletRequest) arg0;
HttpServletResponse httpResponse = (HttpServletResponse) arg1;
httpResponse.addHeader("Content-Encoding", "gzip");
arg2.doFilter(httpRequest, httpResponse); } @Override public void
init(FilterConfig arg0) throws ServletException
{
// TODO Auto-generated method stub }
}
```

## ▼ General

Request URL: http://localhost:9999/nexacro/tt/nexacro141i  
b/framework/SystemBase.js2.gz  
Request Method: GET  
Status Code: ● 200 OK  
Remote Address: [::1]:9999

## ▼ Response Headers [view source](#)

Accept-Ranges: bytes  
Content-Encoding: gzip  
Content-Length: 70261

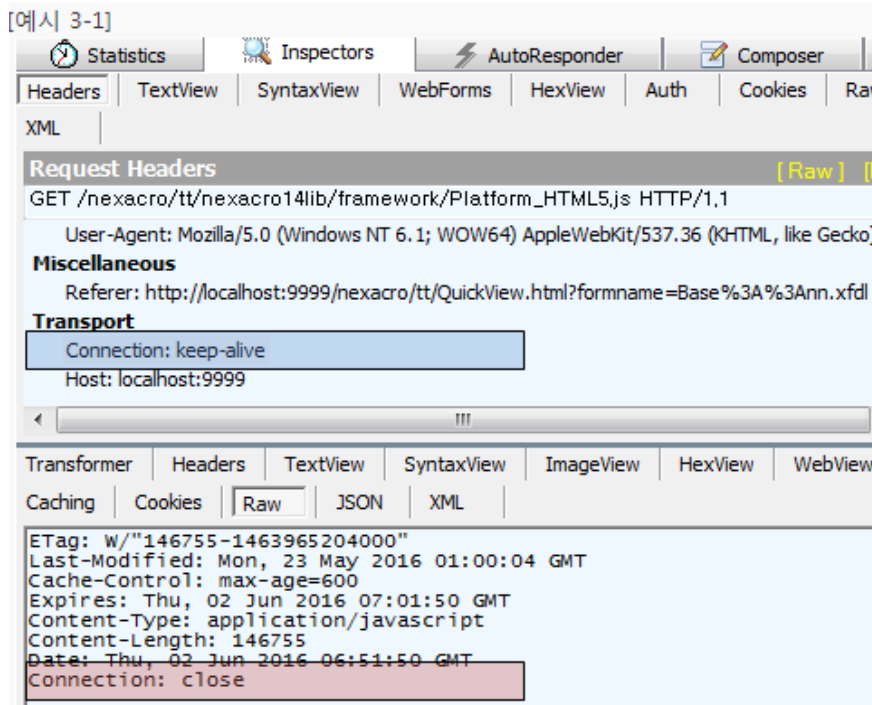
· 응답 헤더에 encoding type 처리 확인

# 성능향상 가이드 – 환경 – KeepAlive (1/2)

일정 개수의 서버 세션(Connect)를 유지하여, 커넥션 시간을 단축

네트워크 성능이 좋은 경우에는 의미가 없으나, 성능 저하 및 SSL통신과 같은 인증이 필요한 경우 필요

- KeepAlive는 HTTP 1.1 스펙으로 서버와 한번 맺은 세션을 지속할지 여부를 설정
  - NRE 및 WRE는 Request 시 기본 KeepAlive를 사용



요청(request)시에는  
**Connection:keep-alive**로 하나  
**close**로 응답(response)하고 있습니다. -

이 경우는 서버 설정으로 **keep-alive**를 **off**한  
상태라고 볼 수 있습니다.

즉 **KeepAlive**를 사용한다고 하더라도 서버  
세팅 결과에 따라 결정됨

## 성능향상 가이드 – 환경 – KeepAlive (2/2)

---

일정 개수수의 서버 세션(Connect)를 유지하여, 커넥션 시간을 단축

네트워크 성능이 좋은 경우에는 의미가 없으나, 성능 저하 및 SSL통신과 같은 인증이 필요한 경우 필요

- 이미 열려있는 세션을 활용하므로 세션 오픈 종료에 따른 시간 단축
  - 사용하지 않는 세션을 유지해야 하므로 서버 측 관리 비용이 증가
  - 상당수의 웹 서버는 KeepAlive를 On하여 사용.
- 네트워크 성능이 좋은 경우 크게 의미가 없으나, 그렇지 못한 경우 KeepAlive 사용여부 확인 필요
- SSL통신을 사용할 경우 커넥션 시 마다 인증처리를 하므로 KeepAlive를 사용은 성능향상 효과 기대

NRE의 경우 xadl의 usehttpkeepalive 속성으로 변경이 가능합니다.

다만 framework js등은 기본적으로 keepalive로 동작됩니다.

## 성능향상 가이드 – 환경 – Etag (1/2)

---

Etag는 Last-modified와 같이 브라우저 캐시에 저장된 파일과 웹 서버의 파일과 일치 여부를 식별 (Object의 동일성을 식별하는데 사용하는 값)

- HTTP1.1에서 파일 캐시의 비교 대상은 Etag 또는 Last-modified
- Etag는 Last-modified와 같이 브라우저의 캐시에 저장된 파일과 웹 서버의 파일이 서로 일치하는지 식별하는 방법 중 하나
- Etag는 **node, time, size** 등의 조합으로 구성
  - 서버 측의 변경을 통해 조합의 변경 가능
- Etag를 사용하는 **여러 개의 웹 서버를 운영 중이라면 Etag 값은 동일해야 함**
  - 만약 웹 서버의 Etag값이 다른 경우 다시 다운로드 됨

# 성능향상 가이드 - 환경 - Etag (2/2)

## Etag 구성

```
Request Headers
GET /nexacro/tt/nexacro14lib/component/ComComp/Div.js HTTP/1.1
Cache
  If-Modified-Since: Mon, 23 May 2016 00:59:50 GMT
  If-None-Match: W/"35327-1463965190000"
Client
  Accept: */*
  Accept-Encoding: gzip, deflate, sdch
Transformer | Headers | TextView | SyntaxView | ImageView | HexView
Caching | Cookies | Raw | JSON | XML
HTTP/1.1 304 Not Modified
Server: Apache-Coyote/1.1
Etag: W/"35327-1463965190000"
Date: Thu, 02 Jun 2016 08:11:36 GMT
Connection: close
```

Etag는 Header 정보에 포함되어 송수신됨

각 서버 별 동일한 파일에 대하여 동일한 ETag 값을 보장 할 수 없다면 ETag를 제거하고 Last-modified만을 사용하여도 됩니다.

Etag를 사용하지 않는 윈도우 서버의 경우 최신파일에서 이전파일로 변경(원복)되는 경우 일자가 이전으로 변경되어 브라우저로 접속 시 갱신이 되지 않을 수 있으므로 주의가 필요하며 이와 같은 문제는 반대로 ETag설정으로 해결될 수 있습니다



---

## 03. 메모리

# Application 개발 시 유의사항 - 메모리 누수 (1/4)

## 사례로 본 누수현상 및 해결방법

- 의도치 않은 전역변수 (변수 scope)
  - 선언되지 않은 변수는 **Global**영역에 만들어짐

예) sTemp = "TobeSoft"; //Global 변수  
this.sTemp = "Tobesoft"; //폼 변수

<http://tobetong.com/?p=3147>

- 동적 생성된 객체 해제방법

### [ 객체 리소스 해제 - 기존 ]

```
this.removeChild("Bind01");  
this.removeChild("Bind02");
```



### [ 객체 리소스 해제 - 수정 ]

```
var objRemoved;  
objRemoved = this.removeChild("Bind01");  
objRemoved.destroy();  
objRemoved = null;  
  
objRemoved = this.removeChild("Bind02");  
objRemoved.destroy();  
objRemoved = null;
```

# Application 개발 시 유의사항 - 메모리 누수 (2/4)

---

## Window 7 IE11에서의 메모리 누수현상

- **IE브라우저 Reload 반복 시 메모리 증가**

- Window 7에서 IE11 사용시 reload를 반복하는 경우 메모리 증가 현상

- 예) Application개발 시 메뉴 이동을 reload기능으로 호출하는 방식으로 구현

- \* 메뉴 이동은 html page(application) 의 reload기능이 아닌 Application 내에서 이동하도록 구현

결론적으로 MS의 공식적인 답변은

“웹 브라우저 특성상 스크립트나 HTML 코드를 100% 완벽하게 관리해 주지는 않습니다.” 입니다

# Application 개발 시 유의사항 – 메모리 누수 (3/4)

## 외부모듈 사용을 위해 WebBrowser 컴포넌트를 사용한 경우

- **WebBrowser 컴포넌트 초기화**

- 넥사크로 제품에서 외부모듈을 연동하기 위해 WebBrowser 모듈을 이용하게 되는데 WebBrowser 자체 버그로 인해 메모리가 누적되는 현상
- 문제 해결을 위해 폼 unload시 추가 스크립트 작성 필요

```
this.WorkFrame_onbeforeclose = function(obj:Form, e)
{
    var frameAll = document.querySelectorAll('iframe');
    var frame;
    for (var i=0 ; i<frameAll.length ; i++)
    {
        frame = frameAll[i];
        if (frame.id.indexOf(this.getOwnerFrame().winId) > -1 && frame.src.indexOf(".html") > -1) {
            frame.src = "about:blank";
            frame.parentNode.removeChild(frame);
        }
    }
    window.CollectGarbage && window.CollectGarbage();
}
```

# Application 개발 시 유의사항 - 메모리 누수 (4/4)

---

## rMateChart 사용 시 Div방식으로 구현

- **rMateChart 사용시 참고**

- rMateChart는 크게 WebBrowser과 Div방식으로 구현이 가능함
- WebBrowser의 경우 nexacroBrowser과 HTML5환경에서 모두 동작 가능함  
(단 모바일 및 메모리를 많이 차지하며, WebBrowser 가지고 있는 메모리 누수현상 발생)
- Div방식으로 개발을 하는 경우 메모리 사이즈가 작고, 모바일 환경에서도 잘 표현됨  
(단 WRE 환경만 지원함)

\* NRE의 경우 rMateChart 최신버전 패치 및 WebBrowser 실행버전을 IE11(최신버전) 실행