



Application 개발 시 유의 사항

The Best UI·UX platform for Business User eXperience

투비소프트
교육서비스팀

TOBESOFT

Application 개발 시 유의사항 - 소스관리

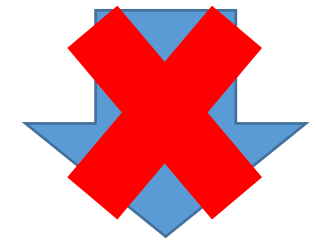
원본 소스 파일과 Generate 파일(.js) 관리



```
1 (function()
2 {
3   return function()
4   {
5     if (!this._is_form)
6       return;
7
8     var obj = null;
9
10    this.on_create = function()
11    {
12      this.set_name("Hello");
13      this.set_titletext("New Form");
14      if (Form == this.constructor)
15      {
16        this._setFormPosition(1280,720);
17      }
18
19      // Object(Dataset, ExcelExportObject) Initialize
20
21
22      // UI Components Initialize
23      obj = new Button("Button00", "39", "7", "133", "78", n
24      obj.set_taborder("0");
25      obj.set_text("Button00");
26      this.addChild(obj.name, obj);
27
```

✓ **Generate (.js)**

Generate 과정을 거쳐 js로
변환된 파일로 운영 시 사용



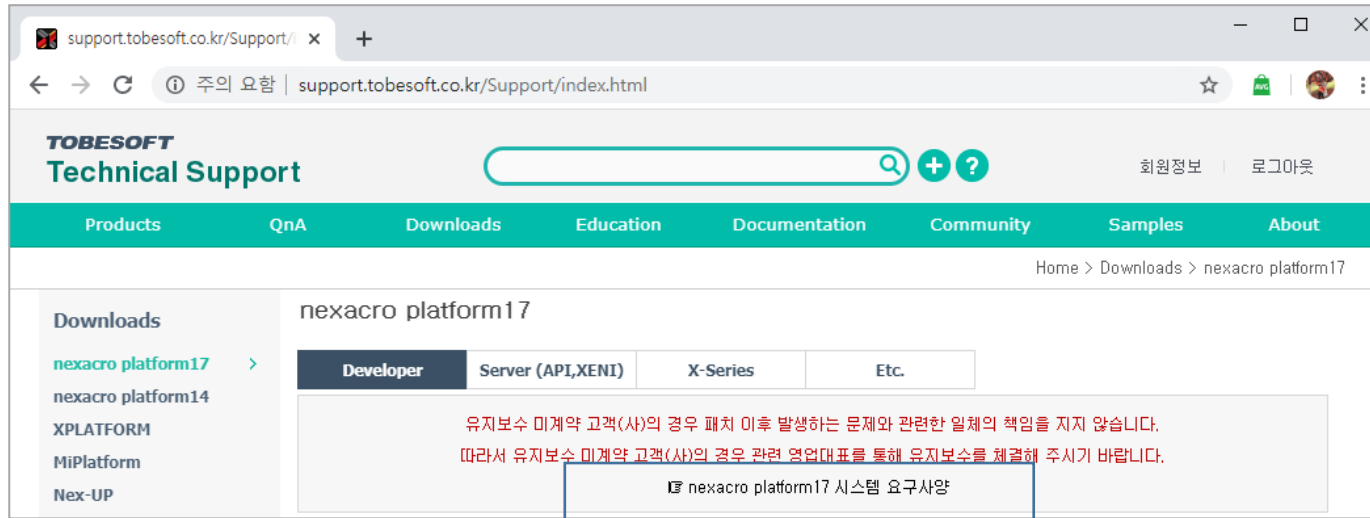
✓ **원본 소스(.xfdl)**

원본 XML 파일

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FDL version="2.0">
3   <Form id="Hello" width="1280" height=
4     <Layouts>
5       <Layout height="720" mobileorient
6         <Button id="Button00" taborder=
7       </Layout>
8     </Layouts>
9     <Script type="xscript5.1"><![CDATA[
10 this.Button00_onclick = function(obj:ne
11 {
12   this.alert("Hello World");
13
14   var objApp = nexacro.getApplication
15   objApp.trace("Hello2");
16
17   this.Button00.set_width(200); //
18
19   obj.set_height(200);
20
21   e.altkey;
22
23   nexacro.round(23.5678, 2);
24
25 };
26 ]]></Script>
27 </Form>
28 </FDL>
29
```

Application 개발 시 유의사항 - 브라우저 최소사양

HTML(WRE) 환경 사용 시 사용자 브라우저 최소사양



- ✓ 넥사크로플랫폼17의 사용자 브라우저 최소사양 결정
- ✓ 외부모듈 연동 시 해당 제품의 최소사양 확인
(레포트, 차트, 기타 연동모듈 체크)

Application 개발 시 유의사항 - 도움말(F1)

도움말에서 제공하는 다양한 기능 중 개발환경에 따른 지원/미지원 범위 참고

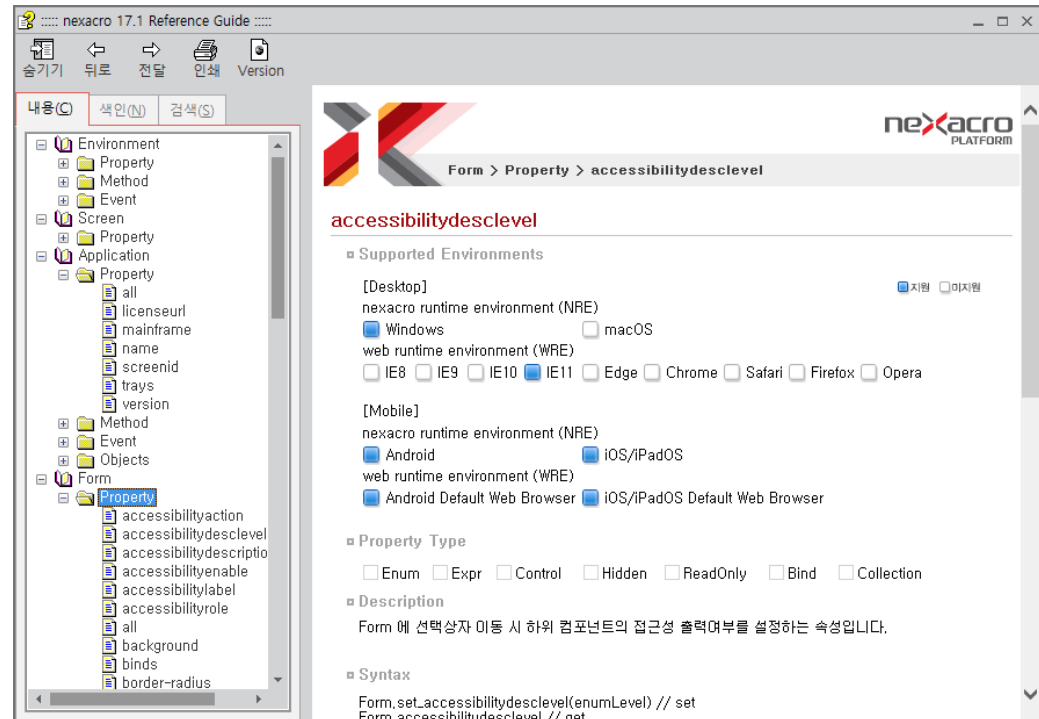
- 지원 / 미지원

- 사용하고자 하는 기능에 대한 지원여부 확인방법



- 기본 자바스크립트 도움말 미포함

- 자바스크립트 기본 스펙에 대한 도움말은 포함되어 있지 않음



Standard ECMA-262 5th Edition

Application 개발 시 유의사항 - History

웹 브라우저 기본기능 History 지원여부 결정

- 넥사크로플랫폼 WRE 사용시 History 기능

- 넥사크로플랫폼은 [SPA\(Single-page application\)](#)로 index.html 하나로 구동
- 일반 html과 같이 하이퍼링크를 통해 여러 페이지를 이동하지 않기 때문에 브라우저에 기록이 남지 않아 뒤로가기, 앞으로 가기 기능이 사용할 수 없습니다.

- Playnexacro.com 예제

- playnexacro.com 사이트 관련 샘플 제공
(해당 기능이 필요한 경우 application단에서 개발)
LocalStorage를 이용하여 구현

<https://www.playnexacro.com/#show:techtip:1199>



History 기능을 적용하기 위해서는 초기 설계 시 의사결정 중요

Application 개발 시 유의사항 – 개인화 자료

그리드 화면의 칼럼 정렬 순서 등 개인별 UI를 제공 시 개인화 자료 저장 방법

- LocalStorage기능

- 사용자 이용 PC에 저장되며, 자리(PC) 이동 시 기록 자료가 없어 재 설정 필요
- LocalStorage는 사용자가 직접 지우지 않는 이상 절대 지워지지 않는다.
 - * 각 브라우저별로 저장할 수 있는 용량이 다름

- DB

- DB에 저장되므로 자리(PC) 이동 시에도 동일하게 적용됨.

Grid Header Column을 Drag&Drop 으로 위치를 변경 후 현재포맷을 저장하면 변경 된 포맷 정보가 유지되는 것을 확인할 수 있습니다. 그리드 초기화를 실행하였을 경우 Grid 포맷을 초기값으로 되돌려 줍니다.

<input type="checkbox"/>	그룹	성명	주소	책정비용	생성일
<input checked="" type="checkbox"/>	A	이승환	인천시 서구 가좌동 118	3,000	2017-01-01 일
<input type="checkbox"/>	A	이승한	경기도 시흥시 정왕동 104 (2가)	4,000	2017-02-01 수
<input type="checkbox"/>	A	강공작	서울특별시 강동구 명일동 257	216,218	2017-11-11 토
<input type="checkbox"/>	A	김명기	서울특별시 강서구 등촌동 631-6	954,561	2017-01-01 일
<input type="checkbox"/>	A	강공작	서울특별시 강동구 명일동 257	216,218	2017-11-11 토
<input type="checkbox"/>	A	김명기	서울특별시 강서구 등촌동 631-6	954,561	2017-01-01 일
<input type="checkbox"/>	A	강공작	서울특별시 강동구 명일동 257	216,218	2017-11-11 토
<input type="checkbox"/>	A	김명기	서울특별시 강서구 등촌동 631-6	954,561	2017-01-01 일

Application 개발 시 유의사항 – Cache Level (1/5)

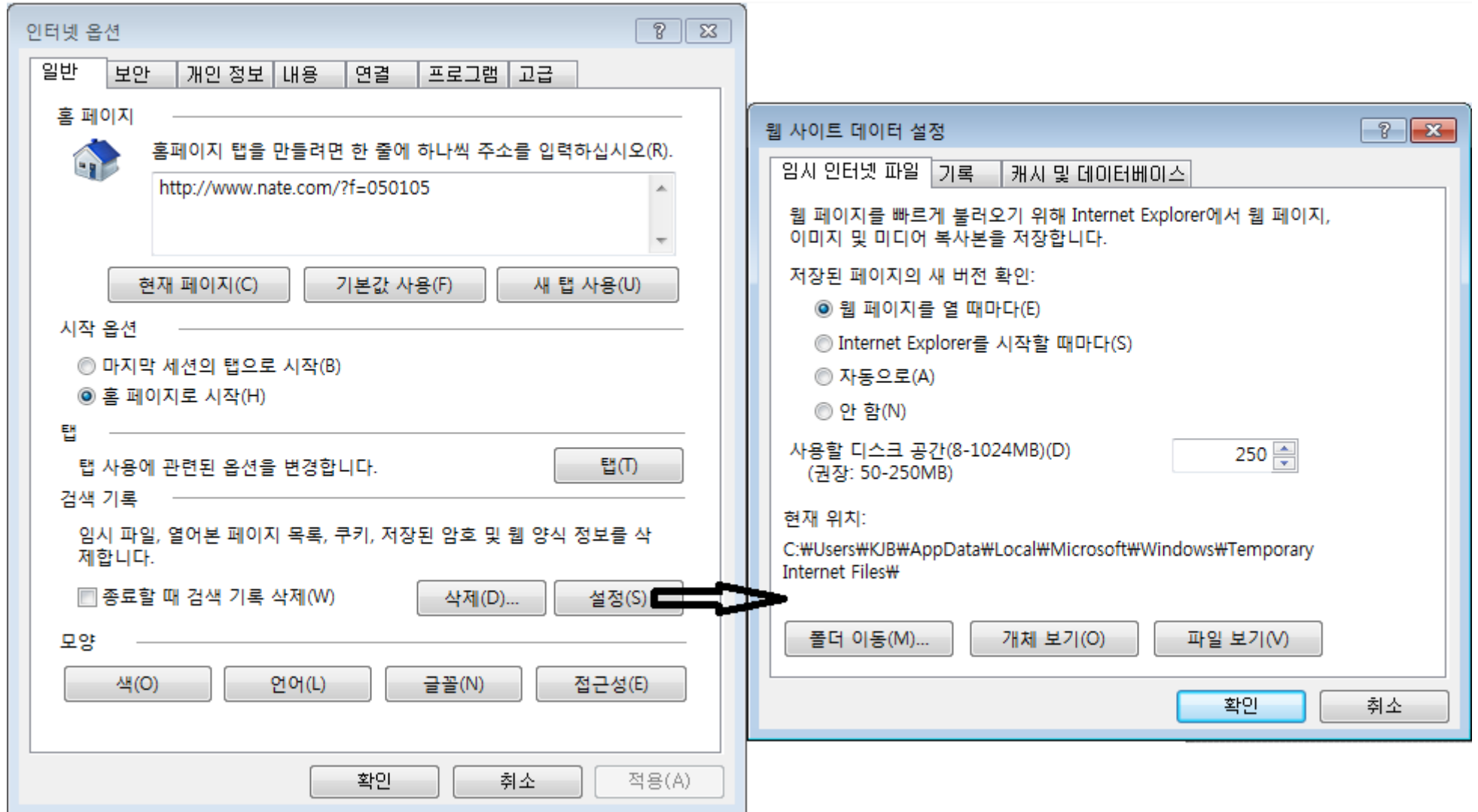
NEXACRO에서 제공하는 Cache 기능은 다음과 같습니다.

종류	설명
None	캐시 기능을 사용하지 않음. 매번 서버로부터 Data를 수신 시 설정 (주로 Dataset 관련 통신 시 사용 함)
Dynamic [기본값]	서버로부터 내려 받은 파일이 갱신되었을 때만 수신 서버 상의 파일이 갱신되지 않은 경우는 로컬 캐시(Local Cache) 파일을 재사용
Session	엔진을 기동할 때, 단 한번만 수신하고 해당 엔진이 종료될 때까지 로컬 캐시 파일만을 사용 (엔진 기동 시 로컬 캐시 파일과 일치하면 재사용)
Static	서버로부터 한번이라도 내려 받은 파일은 엔진 재실행과 상관없이 로컬 캐시 파일만 (단, Type Definition에 지정된 서비스 그룹의 버전이 변경된 경우에는 다시 내려 받음)

WRE 웹 브라우저로 확인 시 'none' 설정 이외에는 브라우저 캐시 정책을 따름

Application 개발 시 유의사항 – Cache Level (2/5)

인터넷 옵션

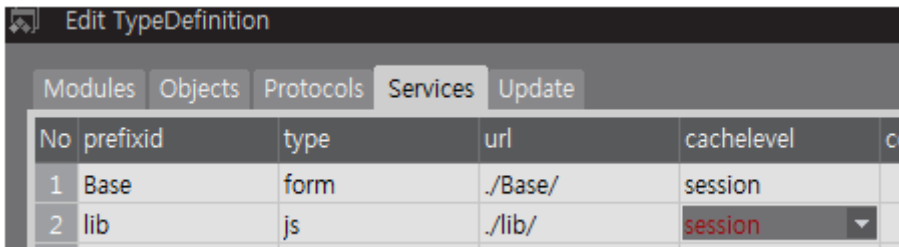


Application 개발 시 유의사항 – Cache Level (3/5)

Cache Level 기본값은 dynamic으로 설정 됨

최종 운영환경에서는 가급적 화면, 공통함수, 이미지 영역은 'session'을 권장

- 개발 중에는 잦은 소스 수정 등이 이루어지므로 "dynamic이 적절할 수 있으나 사용자 서버 테스트용 환경에서는 가능한 운영 환경과 같이 맞춤
- 최종 운영서버 반영 시 화면, 공통함수, 이미지 영역은 session 사용 권장
 - 화면 별 다수의 공통함수를 include는 권장하지 않으며,
만약 사용이 필요한 경우 성능에 영향을 끼치므로 "session을 사용 한다.



No	prefixid	type	url	cachelevel	cd
1	Base	form	./Base/	session	
2	lib	js	./lib/	session	

데이터 통신을 위한 User Service는 반드시 'none'으로 설정해야 함.

Cache Level 중 'static'은 NRE 환경에서만 사용 가능하며,
해외와 같이 네트워크 상황이 좋지 못한 경우에 유용하게 활용 할 수 있다.
다만, static은 버전관리를 수작업으로 하기 때문에 관리비용이 증가한다.

Application 개발 시 유의사항 – Cache Level (4/5)

TypeDefinition User Service Cache Level 설정

User Service						
Default						
PrefixID	Type	URL	CacheLevel	Version	Commu...	
Base	form	./Base/	session	0	0	
FrameBase	form	./FrameBase/	session	0	0	
Form	form	./Form/	session	0	0	
Popup	form	./Popup/	session	0	0	
Comp	form	./Comp/	session	0	0	
CompCom	form	./CompCom/	session	0	0	
CompExt	form	./CompExt/	session	0	0	
Exe	form	./Exe/	session	0	0	
Smpl	form	./Smpl/	session	0	0	
EnvApp	form	./EnvApp/	session	0	0	
Images	file	./Images/	session	0	0	
Lib	js	./Lib/	session	0	0	
LocalURL	JSP	http://localhost...	session	0	0	
SvcURL	JSP	http://demo.ne...	none	0	0	
Mobile	form	./Mobile/	session	0	0	
File	file	./File/	session	0	0	

- TypeDefinition의 User Service 중 Type이 JSP 즉 Data 통신을 담당하는 Service는 Cache Level를 "none"으로 설정
- nexacroStudio에서 실행 시 Cache동작은 none으로 실행이 되나, 실행 환경에서는 Cache Level 설정 값을 따름

Application 개발 시 유의사항 – Cache Level (5/5)

정리

- **Cache level**를 지정하지 않으면 "**dynamic**"으로 동작
- **nexacro studio**를 이용하여 Transaction을 수행 할 때 cache level **설정**에 **관계없이 "none"**으로 동작
- resource의 경로를 prefix를 이용하지 않고 URL로 지정할 경우 사용 브라우저의 캐시 설정에 따름
- **resource가 이미지인 경우** 성능이나 이미지 표시 관련 기능을 위해 사용 **브라우저의 캐시 설정을 따름 (WRE)**
- **WRE(HTML5)**환경에서는 cache level 설정은 무시가 되며, 브라우저 옵션을 따름

이때 브라우저의 옵션과 무관하게 전체 수정내용을 반영하기 위해서는 **CheckVersion** 기능을 사용해야 함

(**NP14 2017.7월** 정기버전 : 기능추가)

Application 개발 시 유의사항 – CheckVersion

WRE 운영 환경에서는 CheckVersion 사용 권장

- WRE 배포

- 필요한 모든 파일을 웹 브라우저에서 직접 로딩하는 방식으로 실행한다.
- 사용하는 웹 브라우저에 따라 성능 차이가 있을 수 있다.
- 별도의 설치 파일이 없으며, js 등 로드 파일에는 별도의 파일 버전을 설정이 필요 없다.
- 해당 Web Server에 파일을 업로드만 하면 된다.

- 단, 페이지 및 js 파일의 Cache동작으로 인해 이전 파일을 가지고 오는 것을 방지하기 위해서 소스 호출 url에 버전정보를 포함 (CheckVersion이용)하여 cache로 인해 오동작을 최소화 한다.

version 정보는 "Module", "Environment", "Application", "Service" 영역별로 다르게 정의 함

ex) 17.1.0.301

15	304	HTTP	127.0.0.1:...	/nexacro17lib/component/CompBase/Element_HTML5.js?nexaversion=17.1.0.301
16	304	HTTP	127.0.0.1:...	/nexacro17lib/component/CompBase/Element_Runtime.js?nexaversion=17.1.0.301
17	304	HTTP	127.0.0.1:...	/nexacro17lib/component/CompBase/CompBase.js?nexaversion=17.1.0.301

Application 개발 시 유의사항 – 세션처리

NEXACRO에서는 Cookie 세션변수를 정의할 뿐 값을 설정하지 않는다.

- **Cookie에 세션변수 정의**

- 넥사크로플랫폼은 Cookie에 세션변수를 선언할 뿐 별도 세션 값을 만들지 않음
- 개발자가 임의로 값을 설정하지 않으면, 서버에서 세팅 된 값을 송/수신만 담당
즉 제품상에서는 별도 세션을 관리하지 않는다.

- **개발자 스크립트에 의한 변경은 가능**

- 넥사크로플랫폼 엔진 Lib에서는 Cookie를 제어하지 않으나, 개발자 스크립트에 의해 강제적으로 코딩 한 부분은 제외

Application 개발 시 유의사항 - 보안 - eval

NEXACRO Script 작성 시 eval 함수는 사용하지 않는다.

왜 eval를 사용하면 안 될까?

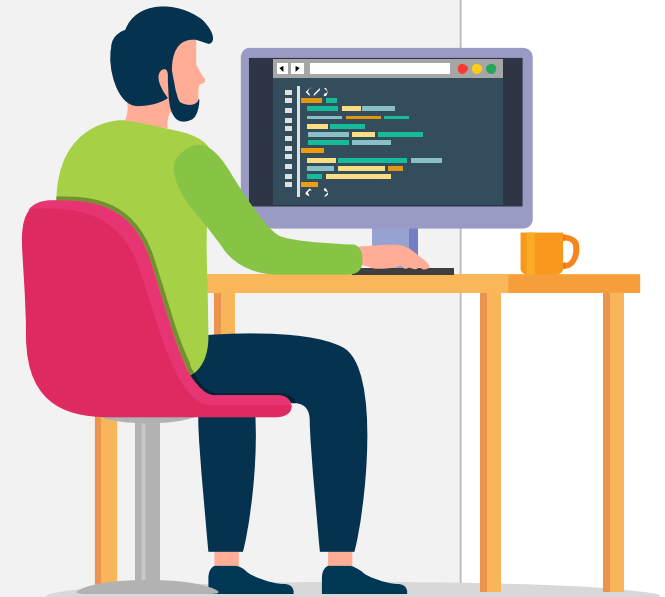
- 보안이슈 - eval은 어떤 코드라도 무조건 실행하기 때문에 성능, 보안, 버그 문제를 일으킴
- 99.9%는 eval 없이도 만들 수 있다.

```
/* eval함수를 이용하여 object 화 하는 방법 */  
trace(eval("this.Grid3"));
```

```
/* eval 함수를 대체하는 방법 */  
var sGridID = this.Grid3.name;  
trace(sGridID + " : " + typeof sGridID);
```

```
var objGrid = this.components[sGridID];  
trace(objGrid + " : " + typeof objGrid);
```

```
/* 결과 */  
Grid3 : string  
[object Grid] : object
```



※ Application::Application_eval.xfdl

Application 개발 시 유의사항 - 보안 - 데이터

NEXACRO 제품에서는 기본 데이터 보안을 제공하지 않는다.

- **통신 보안가능 ?** (transaction 이용한 데이터 송/수신 시 보안이 적용되는가?)
 - 데이터 통신시 compress 및 gzip을 속성을 제공하나 이는 데이터 사이즈를 줄이기 위한 방법이지
 - 데이터 보안을 제공하는 것은 아님 (상당히 많은 분들이 오해)

- **보안모듈 연동**

- 데이터 보안은 사용제품을 도입하여 연동해야 함

- [암호화 모듈]**

이니텍, 소프트포럼 : 데이터 구간암호화 지원

(투비소프트 기본제품에서는 암호화 모듈을 제공하지 않으며, 외부 상용제품을 이용하여

구간암호화 방식형태로 Adaptor를 추가하여 사용이 가능)

Application 개발 시 유의사항 - 보안 - https

HTTP->HTTPS 통신 프로토콜 변경 후 속도 저하 원인

- **https 통신**

https는 http와 다르게 데이터 암호화를 하여 구간 통신을 하게 되며, 약 2배 이상 데이터 사이즈가 증가됨

따라서 http통신에 비해 데이터 전송 속도에 영향을 미침

Application 개발 시 유의사항 – FAQ – httpretry / httptimeout (1/2)

HTTP->HTTPS 통신 시 오류가 발생시 재시도 회수 (NER/WRE 적용)

- **httpretry**

기본 횟수 1번은 retry 숫자에 **포함되지 않기 때문에** retry를 하지 않기 위해서는 **0** 으로 설정

- **httptimeout**

- 서비스(Transaction)을 호출 후 서버로 부터 응답을 대기시간을 설정

(Timeout설정은 Application 서비스 중 서버로 부터 응답을 받는 최대 시간을 설정해야 함)

- **설정위치**

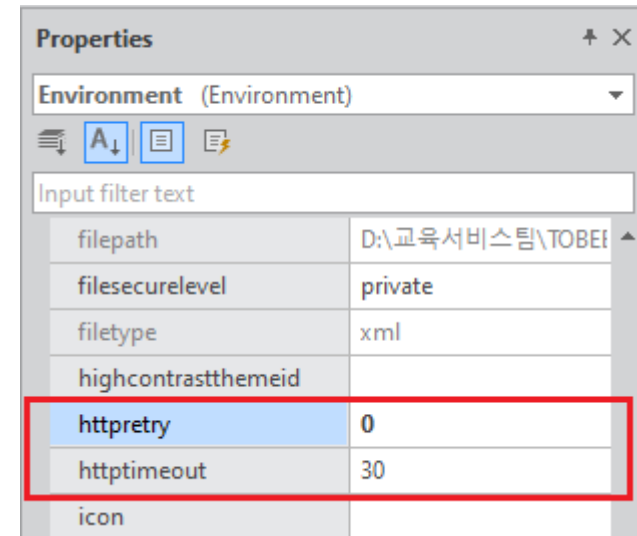
- Environment -> httpretry 속성 설정

- **참고사항**

- nexacro platform 설정

- WAS 설정

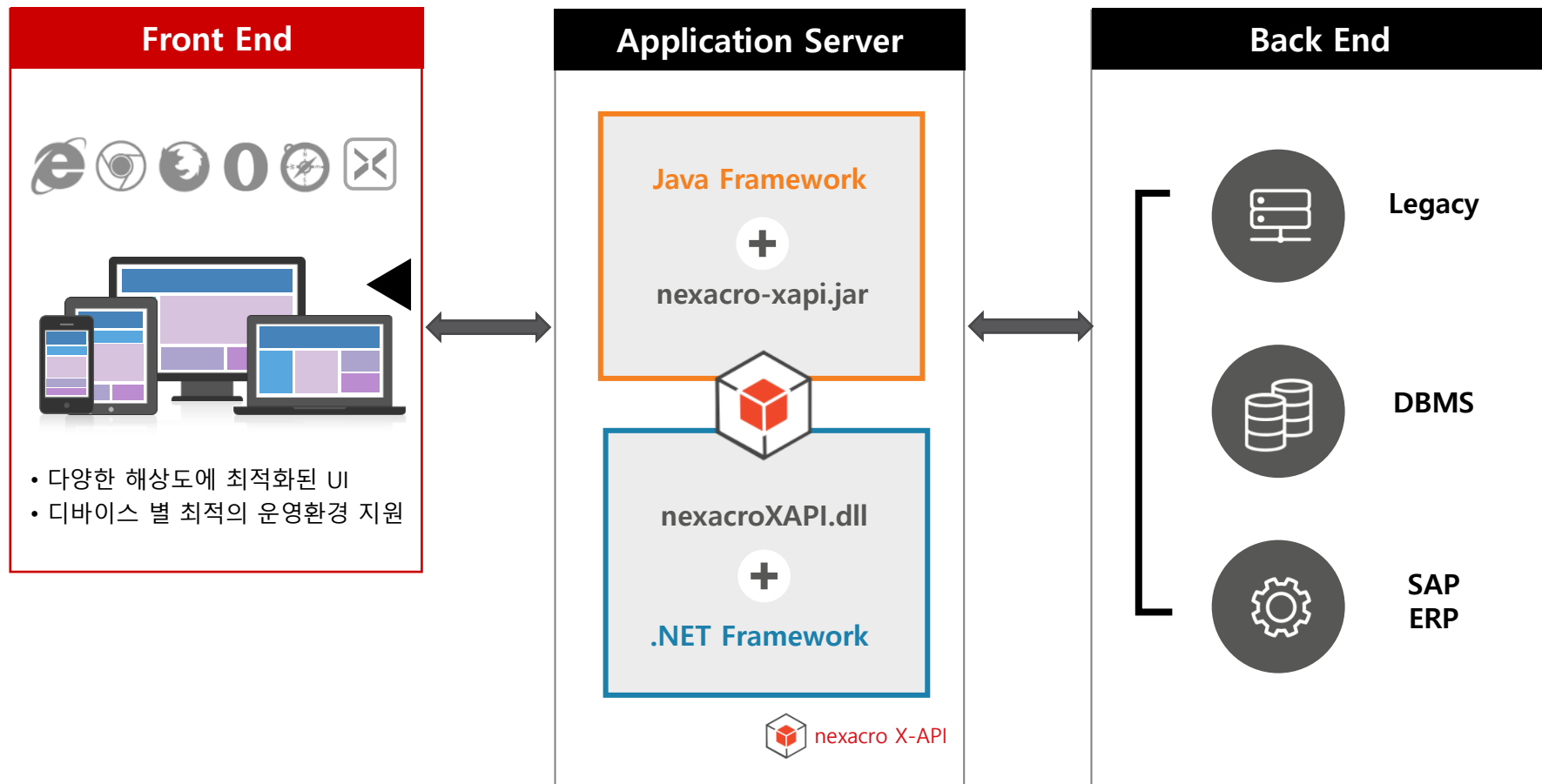
- DB connect 설정



Application 개발 시 유의사항 – FAQ – httpretry / httptimeout (2/2)

Timeout 설정은 Front End(NEXACRO, Application Server(WAS), Back End(DB) 중 최대시간 설정

- Timeout 설정 : 서비스 호출 후 결과를 받는데 까지 필요한 **최대시간** 을 설정



Application 개발 시 유의사항 – 변수 Scope

NEXACRO에서 var, this로 정의되지 않는 변수는 global 영역으로 처리됨

- 선언되지 않는 변수는 Global Scope 적용됨

```
this.Button00_onclick = function(obj:nexacro.Button,e:nexacro.ClickEventInfo)
{
    glovalVal    = 2; // Scope var 미지정시 global변수
    var localVal = 3; // Scope 미지정 variant선언은
                    // Form의 Member가 아님
                    // (다른 폼에서 값 참조불가)
    this.test = function()
    {
        value     = 3; //Global Member 처리
        obj.prop  = 3; //Scope미정 object는 Global Member처리
        this.value = 3; //정상적인 방법
    }
};
```

MiPlatform 소스를 컨버전 하는 경우 자주 놓치는 부분

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

Application 개발 시 유의사항 - 통신방식

Transaction 송/수신 데이터 통신방식

nDataType
0 : XML default
1 : Binary
2 : SSV

bCompress (NRE)
True : 압축
False : 압축하지 않음

[Client]

```
var nDataType = 0; // (0: XML타입, 1: 이진 타입, 2: SSV 타입)
scope.transaction(sRealSvcID, sURL, sInputDsNm, sOutputDsNm, sArgument, sCallbackFunc,
    [,bAsync[,nDataType[,bCompress]]]);
```

[Server]

```
HttpPlatformResponse platformResponse = new HttpPlatformResponse(response,
PlatformType.CONTENT_TYPE_XML, strCharset); //PlatformType.CONTENT_TYPE_SSV
```

Application 개발 시 유의사항 – 통신방식 (XML/SSV/Binary)

```
<Root xmlns="http://www.nexacro.com/platform/dataset" ver="5000">
  <Parameters>
    <Parameter id="ErrorCode" type="int">0</Parameter>
    <Parameter id="ErrorMsg" type="string">SUCC</Parameter>
  </Parameters>
  <Dataset id="out_emp">
    <ColumnInfo>
      <Column id="EMPL_ID" type="string" size="10"/>
      <Column id="FULL_NAME" type="string" size="50"/>
      <Column id="DEPT_CD" type="string" size="10"/>
      <Column id="POS_CD" type="string" size="10"/>
      <Column id="GENDER" type="string" size="10"/>
      <Column id="HIRE_DATE" type="date" size="10"/>
      <Column id="MARRIED" type="string" size="10"/>
      <Column id="SALARY" type="int" size="10"/>
      <Column id="MEMO" type="string" size="10"/>
    </ColumnInfo>
    <Rows>
      <Row>
        <Col id="EMPL_ID"/>
        <Col id="FULL_NAME"/>
        <Col id="DEPT_CD"/>
        <Col id="POS_CD"/>
        <Col id="GENDER"/>
        <Col id="HIRE_DATE">20201117</Col>
        <Col id="MARRIED"/>
        <Col id="SALARY">0</Col>
        <Col id="MEMO"/>
      </Row>
      <Row>
        <Col id="EMPL_ID"/>
        <Col id="FULL_NAME"/>
        <Col id="DEPT_CD"/>
        <Col id="POS_CD"/>
        <Col id="GENDER"/>
        <Col id="MARRIED"/>
        <Col id="SALARY">0</Col>
        <Col id="MEMO"/>
      </Row>
    </Rows>
  </Dataset>
</Root>
```

```
SSV:utf-
8▲ErrorCode:int=0▲ErrorMsg:string=SUCC▲Dataset:out_emp▲_RowType_▼EMPL_ID:string(10)▼FULL_N
AME:string(50)▼DEPT_CD:string(10)▼POS_CD:string(10)▼GENDER:string(10)▼HIRE_DATE:date(10)▼M
ARRIED:string(10)▼SALARY:int(10)▼MEMO:string(10)▲N?????20201117▼Q▲N?????0▲N?????20
201117▼Q▲N?????20201117▼Q▲N?????20201117▼Q▲N?????20201117▼Q▲NAAA159김태
경▼Q2▼Q4▼M▼false▼Q▼▲N▼ZZ007▼박정호▼Q3▼Q3▼M▼20201102▼false▼321321536▼red test▲NAAA152안
녕▼Q3▼M▼20201116▼true▼3524132▼green▲N▼DD005▼Ji▼Q3▼Q3▼M▼20201117▼Q▼60000▼yellow▲
```

```
00000000h: FE 10 13 88 00 25 00 02 00 09 45 72 72 6F 72 43 ; 3..?%....ErrorC
00000010h: 6F 64 65 00 03 00 00 00 00 00 08 45 72 72 6F 72 ; ode.....Error
00000020h: 4D 73 67 00 15 00 04 53 55 43 43 FE 01 13 88 00 ; Msg....SUCC??.
00000030h: 98 00 07 6F 75 74 5F 65 6D 70 FE 10 13 88 00 02 ; ?.out_emp??.
00000040h: 00 00 00 09 00 07 45 4D 50 4C 5F 49 44 00 01 00 ; .....EMPL_ID...
00000050h: 0A 00 01 00 09 46 55 4C 4C 5F 4E 41 4D 45 00 01 ; .....FULL_NAME...
00000060h: 00 32 00 01 00 07 44 45 50 54 5F 43 44 00 01 00 ; .2....DEPT_CD...
00000070h: 0A 00 01 00 06 50 4F 53 5F 43 44 00 01 00 0A 00 ; .....POS_CD.....
00000080h: 01 00 06 47 45 4E 44 45 52 00 01 00 0A 00 01 00 ; ...GENDER.....
00000090h: 09 48 49 52 45 5F 44 41 54 45 00 05 00 0A 00 01 ; .HIRE_DATE.....
000000a0h: 00 07 4D 41 52 52 49 45 44 00 01 00 0A 00 01 00 ; ..MARRIED.....
000000b0h: 06 53 41 4C 41 52 59 00 02 00 0A 00 01 00 04 4D ; .SALARY.....M
000000c0h: 45 4D 4F 00 01 00 0A 00 01 00 30 00 01 00 09 00 ; EMO.....0.....
000000d0h: 15 00 00 00 15 00 00 00 15 00 00 00 15 00 00 00 ; .....
000000e0h: 15 00 00 00 29 42 77 5D 38 04 40 00 00 00 15 00 ; .....)Bw]8.@.....
000000f0h: 00 00 03 00 00 00 00 00 15 00 00 00 28 00 01 00 ; .....(.....
00000100h: 09 00 15 00 00 00 15 00 00 00 15 00 00 00 15 00 ; .....
00000110h: 00 00 15 00 00 00 00 00 15 00 00 00 03 00 00 00 ; .....
00000120h: 00 00 15 00 00 00 30 00 01 00 09 00 15 00 00 00 ; .....0.....
00000130h: 15 00 00 00 15 00 00 00 15 00 00 00 15 00 00 00 ; .....
00000140h: 29 42 77 5D 38 04 40 00 00 00 15 00 00 00 03 00 ; )Bw]8.@.....
00000150h: 00 00 00 00 15 00 00 00 30 00 01 00 09 00 15 00 ; .....0.....
00000160h: 00 00 15 00 00 00 15 00 00 00 15 00 00 00 15 00 ; .....
00000170h: 00 00 29 42 77 5D 38 04 40 00 00 00 15 00 00 00 ; ..)Bw]8.@.....
00000180h: 03 00 00 00 00 00 15 00 00 00 30 00 01 00 09 00 ; .....0.....
00000190h: 15 00 00 00 15 00 00 00 15 00 00 00 15 00 00 00 ; .....
000001a0h: 15 00 00 00 29 42 77 5D 38 04 40 00 00 00 15 00 ; .....)Bw]8.@.....
000001b0h: 00 00 03 00 00 00 00 00 15 00 00 00 30 00 01 00 ; .....0...
000001c0h: 09 00 15 00 00 00 15 00 00 00 15 00 00 00 15 00 ; .....
```

Application 개발 시 유의사항 – PrefixID

서비스 경로 사용시 Full경로가 아닌 PrefixID 사용

```
this.transaction("svcSelect","http://demo.nexacro.com/edu/nexacro17/
select_emp.jsp","", "ds_emp=out_emp","", "fn_call");
```



```
this.transaction("svcSelect","SvcURL::select_emp.jsp", "", "ds_emp=out_emp","", "fn_call");
```

	변경 전	변경 후
Cache	웹 브라우저 Cache기능 사용	Prefix Cache 기능 사용 * WRE이 경우 웹 브라우저 캐시사용
URL 변경 시	모든 소스 변경대상	Prefix URL주소 변경

Application 개발 시 유의사항 - 서비스 호출

한번의 서비스 호출로 n개의 Dataset를 가져온다.

```
this.transaction("svcSelect1","SvcURL::select_code.jsp", "", "ds_emp=out_emp", "", "fn_call");  
this.transaction("svcSelect2","SvcURL::select_code.jsp", "", "ds_pos=out_pos", "", "fn_call");
```

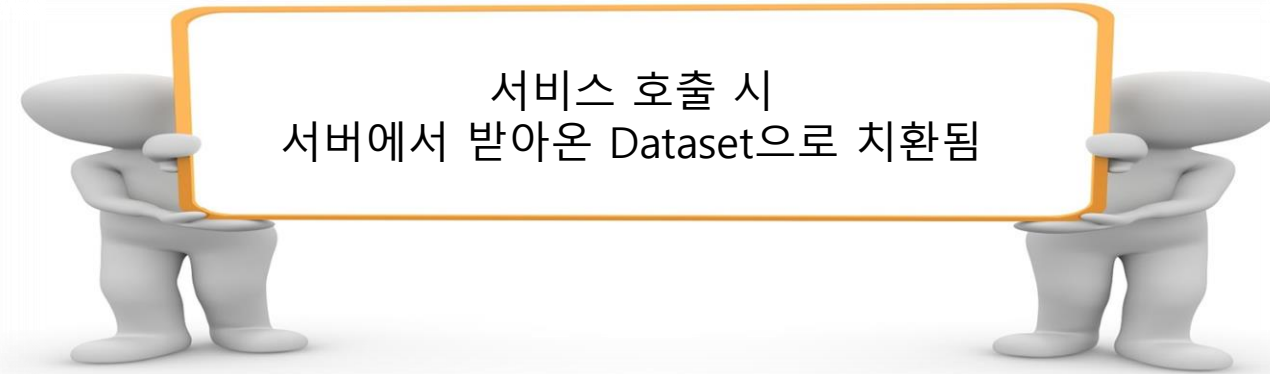


```
var strParameter = "var1=" + nexacro.wrapQuote("홍길동") +  
    " var2=" + nexacro.wrapQuote("이순신");  
this.transaction("svcSelect","SvcURL::select_code.jsp",  
    "", "ds_dept=out_dept ds_pos=out_pos", strParameter, "fn_call");
```

- n개의 Dataset 전달 시 구분은 " " 공백으로 처리됨
- 전달하는 변수 또한 N개를 보낼 때는 " "으로 처리됨 따라서 전달하는 파라미터는 `nexacro.wrapQuote(strParameter)`로 전달

Application 개발 시 유의사항 – Dataset Layout

서비스 호출 시 데이터가 없는 경우에도 반드시 Layout을 리턴 해 준다.



```
<?xml version="1.0" encoding="utf-8"?>
<Root xmlns="http://www.nexacro.com/platform/dataset" ver="5000">
  <Parameters>
    <Parameter id="ErrorCode" type="int">0</Parameter>
    <Parameter id="ErrorMsg" type="string">SUCC</Parameter>
  </Parameters>
  <Dataset id="out_emp">
  </Dataset>
</Root>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Root xmlns="http://www.nexacro.com/platform/dataset" ver="5000">
  <Parameters>
    <Parameter id="ErrorCode" type="int">0</Parameter>
    <Parameter id="ErrorMsg" type="string">SUCC</Parameter>
  </Parameters>
  <Dataset id="out_emp">
    <ColumnInfo>
      <Column id="EMPL_ID" type="string" size="10"/>
      <Column id="FULL_NAME" type="string" size="50"/>
      <Column id="DEPT_CD" type="string" size="10"/>
      <Column id="POS_CD" type="string" size="10"/>
      <Column id="GENDER" type="string" size="10"/>
      <Column id="HIRE_DATE" type="date" size="10"/>
      <Column id="MARRIED" type="string" size="10"/>
      <Column id="SALARY" type="int" size="10"/>
      <Column id="MEMO" type="string" size="10"/>
    </ColumnInfo>
    <Rows>
      <Row>
        <Col id="EMPL_ID">AS123</Col>
      </Row>
    </Rows>
  </Dataset>
</Root>
```

Application 개발 시 유의사항 – useclientlayout

서비스 호출 시 서버에서 받아온 Dataset으로 치환

단) useclientlayout 속성 true 변경 시 기존 layout 유지

- **Dataset Layout은 서버에서 받아오는 것을 추천함**
 - (Script를 통해 Type 등 변경은 권장하지 않음)
- **Dataset Layout은 한 곳에서 기준을 잡고 만들어 주는 것이 유지보수에 효과적임**

Application 개발 시 유의사항 – Column Type (1/2)

Dataset Column을 이용한 수식 계산 시 Column Type를 이용

❖ 그룹별 소계

CORP	DEPT		NAME	ID	SALARY	BONUS
Korea Corp.	Education	0	John	KR101	6,000	300
		0	Jackson	KR102	6,000	200
		0	Adam	KR103	3,000	50
		0	Melon	KR104	9,000	1,000
		0	Kate	KR105	8,000	200
	Education Sum	1			32,000	1,750
	Marketing	0	Kein	KR110	1,000	100
		0	Ivy	KR111	9,000	300
		0	Kara	KR112		
		0	Twice	KR113		
	Marketing Sum	1			10,000	400
	Sales	0	Bts	KR106	1,000	500
		0	Andrew	KR107	8,000	30
		0	Belle	KR108	6,000	50
		0	Elsa	KR109	7,000	100
	Sales Sum	1			22,000	680
Korea Corp. Sum		2			64,000	2,830

Type :
number

❖ 그리드 합계

```
getSum("SALARY") -> getSum("nexacro.toNumber(SALARY)")
```

Application 개발 시 유의사항 – Column Type (2/2)

Dataset Column을 이용한 수식 계산 시 Column Type를 이용

❖ 기능

Sort는 Column의 Type이 기준이 됨

즉) 형 변환 불가

The screenshot shows a window titled "nexacro17 Education - Employees" with a subtitle "Dataset Column Type". It displays two tables side-by-side, each with columns for No, Name, ID, and Salary. The left table's Salary column is labeled "Salary(String)" and the right table's is "Salary(Int)". Both tables contain the same data: Olivia (AA-001, 83,000), John (AA-002, 76,000), Jackson (BB-001, 5,000), Maia (BB-002, 50,000), Adam (CC-001, 48,000), and Ray (DD-001, 60,000). Below the tables are two buttons: "+오름차순" and "-내림차순".

No	Name	ID	Salary(String)
1	Olivia	AA-001	83,000
2	John	AA-002	76,000
3	Jackson	BB-001	5,000
4	Maia	BB-002	50,000
5	Adam	CC-001	48,000
6	Ray	DD-001	60,000

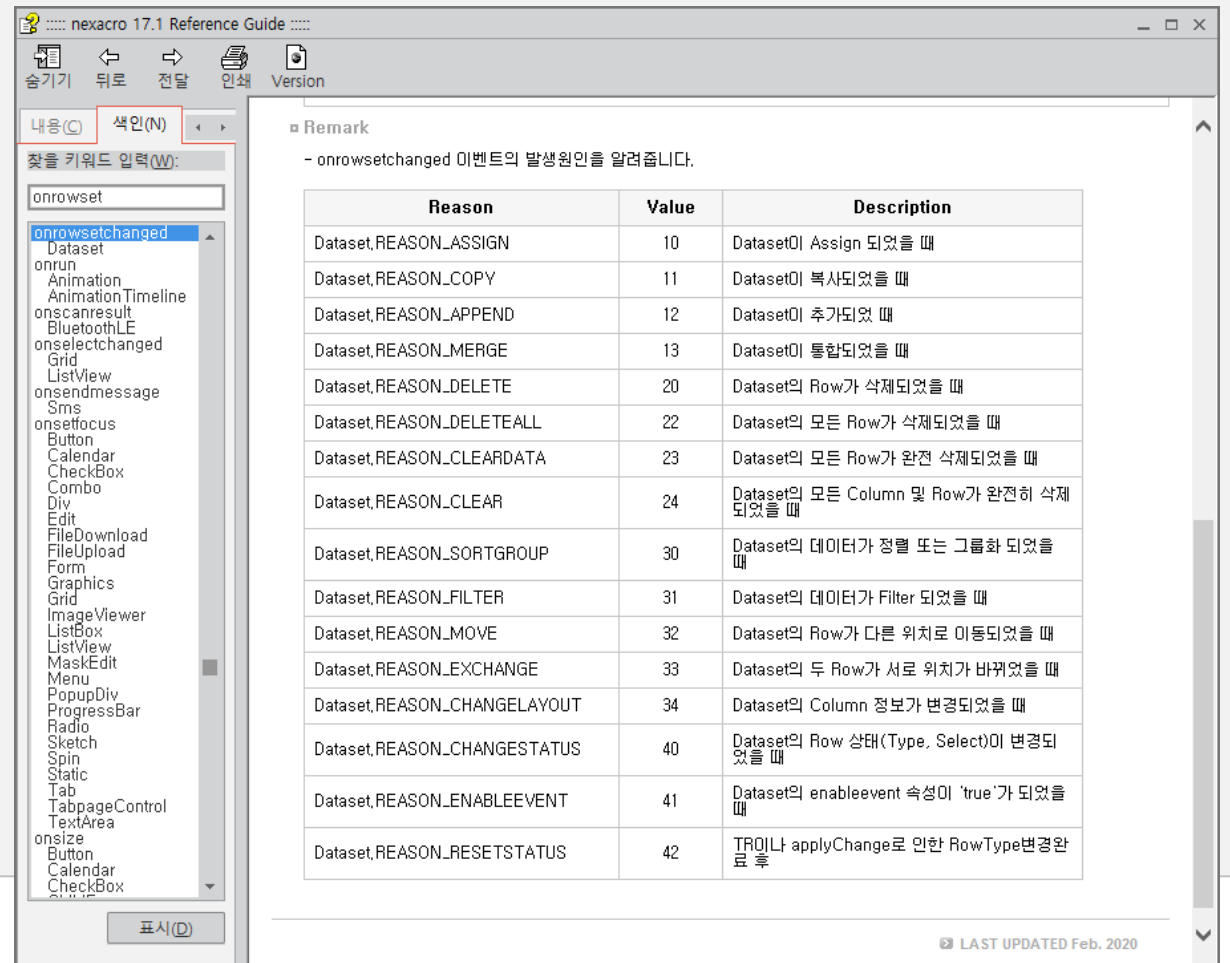
No	Name	ID	Salary(Int)
1	Olivia	AA-001	83,000
2	John	AA-002	76,000
3	Jackson	BB-001	5,000
4	Maia	BB-002	50,000
5	Adam	CC-001	48,000
6	Ray	DD-001	60,000

Application 개발 시 유의사항 – Dataset Event

Dataset Event 중 onload, onrowsetchanged, onrowposchanged 등 Event객체에 e.reason이 포함된 경우 스크립트 작성 시 반드시 분기 처리한다.

```
this.ds_emp_onrowsetchanged = function(obj:nexacro.NormalDataset, e:nexacro.DSRowPosChangeEventInfo)
```

```
{  
  
  if(e.reason == 12) // append  
  {  
  
  } else if(e.reason == 20) // delete  
  {  
  
  } else if(e.reason == 30) // sort  
  {  
  
  }  
  
};
```



The screenshot shows the 'nexacro 17.1 Reference Guide' window. The search bar contains 'onrowset', and the search results list various events, with 'onrowsetchanged' selected. The 'Remark' section states: '- onrowsetchanged 이벤트의 발생원인을 알려줍니다.' Below this is a table with the following data:

Reason	Value	Description
Dataset,REASON_ASSIGN	10	Dataset이 Assign 되었을 때
Dataset,REASON_COPY	11	Dataset이 복사되었을 때
Dataset,REASON_APPEND	12	Dataset이 추가되었을 때
Dataset,REASON_MERGE	13	Dataset이 통합되었을 때
Dataset,REASON_DELETE	20	Dataset의 Row가 삭제되었을 때
Dataset,REASON_DELETEALL	22	Dataset의 모든 Row가 삭제되었을 때
Dataset,REASON_CLEARDATA	23	Dataset의 모든 Row가 완전 삭제되었을 때
Dataset,REASON_CLEAR	24	Dataset의 모든 Column 및 Row가 완전히 삭제되었을 때
Dataset,REASON_SORTGROUP	30	Dataset의 데이터가 정렬 또는 그룹화 되었을 때
Dataset,REASON_FILTER	31	Dataset의 데이터가 Filter 되었을 때
Dataset,REASON_MOVE	32	Dataset의 Row가 다른 위치로 이동되었을 때
Dataset,REASON_EXCHANGE	33	Dataset의 두 Row가 서로 위치가 바뀌었을 때
Dataset,REASON_CHANGELAYOUT	34	Dataset의 Column 정보가 변경되었을 때
Dataset,REASON_CHANGESTATUS	40	Dataset의 Row 상태(Type, Select)이 변경되었을 때
Dataset,REASON_ENABLEEVENT	41	Dataset의 enableevent 속성이 'true'가 되었을 때
Dataset,REASON_RESETSTATUS	42	TRQI나 applyChange로 인한 RowType변경완료 후

LAST UPDATED Feb. 2020

Application 개발 시 유의사항 – Grid

Grid의 Properties 의 cellmovingtype=col인 경우 Event에서 스크립트 작성시 e.col, e.cell 값을 이용하여 스크립트 작성시 유의사항

```
this.grd04_onclick= function(obj:nexacro.Grid,e:nexacro.GridClickEventInfo)
{
    if(e.col == 1)
    {
        this.alert("첫 번째 Coloumn");
    }
};
```

Grid의 속성 cellmovingtype을 col로 변경하는 경우 마우스를 이용하여 Column을 변경가능. 이때 col index가 재 설정되므로 의도치 않는 이벤트 실행이 동작됩니다.

Application 개발 시 유의사항 - Dataset 중심의 바인딩/스크립트

NEXACRO 제품을 이용하여 업무화면 개발 시 Dataset 중심으로 코딩 한다.

가능한 Dataset 바인딩을 활용하고, 이벤트 작성시 Dataset의 이벤트를 먼저 사용한다.

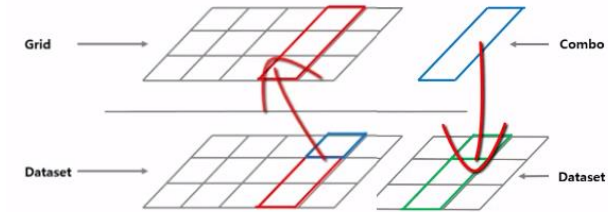
nexacro17 Education - Employees

dataset 중심의 스크립트 작성

DEPT_CD	DEPT_NAME	No	Name	ID	Department	Married
01	Accounting Team	1	Olivia	AA-001	Accounting Team	<input checked="" type="checkbox"/>
02	HR Team	2	Adam	CC-001	Accounting Team	<input checked="" type="checkbox"/>
03	Sales Team					
04	Design Team					
05	Education Team					

Count: 2

Name	Olivia	Hire Date	2010-10-03 일
Emp ID	AA-001	Annual Salary	83,000
Gender	<input type="radio"/> Male <input checked="" type="radio"/> Female	Memo	ivory
Married	<input checked="" type="checkbox"/>		



Properties

div_detail.form.msk_salary (MaskEdit)

Input filter text

tooltiptype	
trimtype	
type	
value	ds_emp SALARY

Application 개발 시 유의사항 - 산출물 - jsDoc (1/3)

Javascript소스코드 파일에 주석을 달기 위해 사용되는 마크업 언어

❖ JsDoc?

Javascript소스코드 파일에 주석을 달기 위해 사용하는 마크업 언어로 JsDoc에 포함하는 주석을 사용하여

코드를 작성하고 인터페이스를 설명하는 문서를 생성할 수 있다.

Ex)

```
/**
 * @class Grid에 기능 추가(addCol..)
 * @param {Object} objGrid - 대상 그리드
 * @return N/A
 * @example
 * this._gfnGridAddProp(this.grdMain);
 */
```

* jsDoc 작성 가이드 문서는 별도 제공하지 않음

jsDoc 산출물을 출력하기 위해서는 js 작성 시 **jsDoc 포맷**에 맞추어 작성되어야 함.

Application 개발 시 유의사항 – 산출물 – jsDoc (2/3)

프로젝트 공통Script lib모듈로 작성된 소스코드를 jsDoc 문서로 산출물을 작성한다.

이름

- jsFile
- symbols
- tool
- Lib_Comm.js.bak
- Lib_Comm.xjs.bak
- make_jsdoc.bat.bak
- files.html
- index.html**
- make_jsdoc.bat

← → ↻ 🏠 | file:///D:/교육서비스팀/유틸리티/jsdoc/index.html

Class Index

[Class Index](#) | [File Index](#)

Classes

- [global](#)
- [pForm.qfn_closeDLL](#)
- [pForm.qfn_columnHidCallba](#)
- [pForm.qfn_excelExport](#)
- [pForm.qfn_excelImport](#)
- [pForm.qfn_excelImportAll](#)
- [pForm.qfn_exportOnerror](#)
- [pForm.qfn_exportOnsuccess](#)
- [pForm.qfn_findGridText](#)
- [pForm.qfn_formOnLoad](#)
- [pForm.qfn_getBindColumnT](#)
- [pForm.qfn_getCommCode](#)
- [pForm.qfn_getCompld](#)
- [pForm.qfn_getDLLPath](#)

[global](#)

[pForm.qfn_closeDLL](#)

컴포넌트 DLL 을 로딩하고 그 결과 Object를 리턴

[pForm.qfn_columnHidCallback](#)

그리드 우클릭 POPUPMENU 내부함수
그리드 컬럼숨기기/보이기

[pForm.qfn_excelExport](#)

excel export

[pForm.qfn_excelImport](#)

excel import(데이터 헤더제외)

[pForm.qfn_excelImportAll](#)

excel import(데이터 헤더포함)

Application 개발 시 유의사항 – 산출물 – jsDoc (3/3)

jsDoc 을 활용한 프로젝트 공통Script 산출물

[Class Index](#) | [File Index](#)

Classes

[_global](#)

[pForm.qfn_closeDLL](#)

[pForm.qfn_columnHidCallba](#)

[pForm.qfn_excelExport](#)

[pForm.qfn_excelImport](#)

[pForm.qfn_excelImportAll](#)

[pForm.qfn_exportOnerror](#)

[pForm.qfn_exportOnsuccess](#)

[pForm.qfn_findGridText](#)

[pForm.qfn_formOnLoad](#)

[pForm.qfn_getBindColumnT](#)

[pForm.qfn_getCommCode](#)

[pForm.qfn_getCompld](#)

[pForm.qfn_getDLLPath](#)

[pForm.qfn_getExtCommon](#)

[pForm.qfn_grid_onheadclick](#)

[pForm.qfn_grid_onkeydown](#)

Class pForm.qfn_excelExport

excel export

Defined in: [Excel.js](#).

Class Summary

[pForm.qfn_excelExport](#)(objGrid, sSheetName, sFileName)

Class Detail

[pForm.qfn_excelExport](#)(objGrid, sSheetName, sFileName)

```
this.gfnExcelExport(this.grid_export, "SheetName", "");
```

Parameters:

{Object} **objGrid**
- Grid Object
{String} **sSheetName** *Optional*
- sheet name
{String} **sFileName** *Optional*
- file name

Returns:

N/A