

```

1  /**
2   * 각 WFrame Scope별로 공유되는 Scope 전역 변수와 공통 함수를 정의한다.
3   *
4   * com 객체는 WFrame Scope 업무 개발자가 호출해야할 공통 함수나 속성을 정의한다.
5   * com 객체는 WFrame Scope 별로 생성되기 때문에 com 객체 내에 정의된 함수에서의 선언된 $p 객체는
6   * 해당 함수를 호출한 화면의 WFrame Scope 내의 $p를 참조하게 된다.
7   *
8   * @version 3.0
9   * @author InswaveSystems
10  * @type com
11  * @class com
12  * @namespace com
13  */
14  var com = {
15      // Message Box ID 생성을 위한 순번
16      MESSAGE_BOX_SEQ : 1
17  };
18
19  //-----
20  // MDI
21  //-----
22  /**
23   * main.xml의 windowContainer에 메뉴를 open한다.
24   *
25   * @date 2020.06.10
26   * @memberOf com
27   * @param {String} psMenuId 메뉴번호
28   * @param {JSON} poParam 화면에 넘길 JSON파라미터
29   * @return {void}
30   * @author 전기완
31   * @example
32   */
33  com.mdi = {
34      openMenu : function(param, callback)
35      {
36          console.log('open menu (' + JSON.stringify(param) + ') ...');
37
38          $p.top().com.startBizLayer();
39          com.session.check();
40
41          gcm['menuId'] = param.menuId;
42          gcm['athrtId'] = param.athrtId;
43          //console.log('..... : ' + JSON.stringify(param));
44          /*
45          if(com.data.isEmpty(psMenuId)) return;
46
47          var vsMenuId = psMenuId;
48          var voMenuInfo = com.getMenuInfo(vsMenuId);
49          if(com.data.isEmpty(voMenuInfo))
50          {
51              com.alert("", "메뉴목록정보가 없습니다.");
52              return;
53          }
54          if(com.data.isEmpty(voMenuInfo["url"]))
55          {
56              com.alert("", "메뉴URL정보가 없습니다.");
57              return;
58          }
59          */
60          var voTabOpts = {};
61          voTabOpts["title"] = param["menuNm"];
62          voTabOpts["windowId"] = param["menuId"];
63          voTabOpts["src"] = param["url"];
64          voTabOpts["windowTitle"] = param["menuId"];
65          voTabOpts['menuInfo'] = param;
66
67          var voParam = param || {};
68

```

```

69 // 이미 여러있으면 패스.
70 if($p.top().wdcMain.getWindow(param.menuId) != null)
71 {
72     // 열려있는 페이지에 Focus
73     $p.top().wdcMain.getWindowById(param.menuId, true);
74 }
75 // 열려있지 않으면
76 else
77 {
78     // 페이지 추가
79     com.mdi.addWindow($p.top().wdcMain, voTabOpts, voTabOpts);
80 }
81 },
82 /**
83  * windowContainer에 Tab을 add한다.
84  *
85  * @date 2020.06.10
86  * @memberOf com
87  * @param {Object} poTabObj windowContainer Object
88  * @param {JSON} poOpts windowContainer Options
89  * @param {JSON} poParam windowContainer에 전달할 파라미터
90  * @return {void}
91  * @author 전기완
92  * @example
93  */
94 addWindow : function(poTabObj, poOpts, poParam)
95 {
96
97     if(com.data.isEmpty(poTabObj) || com.data.isEmpty(poOpts) || com.data.isEmpty(poOpts["src"]))
98         com.alert("필수 Tab정보가 없습니다.");
99     return;
100 }
101 var voParam = poParam || {};
102
103 var vsTabId = poOpts["windowId"];
104 var voTabOpts = {};
105     voTabOpts["title"] = poOpts["title"];
106     voTabOpts["windowId"] = vsTabId;
107     voTabOpts["userData"] = vsTabId;
108     voTabOpts["frameMode"] = "wframe";
109     voTabOpts["src"] = poOpts["src"];
110     //voTabOpts["windowTitle"] = (com.data.isEmpty(poOpts["windowTitle"], "")).replace(/>/,
111
112     if(com.data.isNotEmpty(voParam)){
113         voTabOpts["dataObject"]={ "type" : "json",
114                                     "name" : "param",
115                                     "data" : voParam};
116     }
117
118     if(com.data.isEmpty(poTabObj.getWindowById(vsTabId, false)))
119     {
120         var vnMaxNum = poTabObj.getAttribute("windowMaxNum");
121         if(poTabObj.windows.length >= vnMaxNum)
122         {
123             com.popup.alert("더 이상 열 수 없습니다.");
124             return;
125         }
126         else
127         {
128             poTabObj.createWindow(voTabOpts);
129         }
130     }
131 },
132
133 closeMenu : function(windowId)
134 {
135     $p.top().wdcMain.closeWindow(windowId);
136 },

```

```

137
138
139
140
141
142
143
144
145
146
147
148 /**
149  * main.xml의 컨텐츠영역의 화면을 이동시킨다.
150  *
151  * @date 2020.06.10
152  * @memberOf com
153  * @param {String} psTitle 화면타이틀
154  * @param {String} psSrc 화면 src
155  * @param {JSON} poParam 화면에 넘길 JSON파라미터
156  * @return {void}
157  * @author 전기완
158  * @example
159  *
160  * 메뉴가 목록화면일때 등록화면으로 이동시
161  * com.changeMenu("프로그램등록", "/ui/sample/programRegister.xml", {"AAA":"aaa", "BBB":"bbb"});
162  *
163  * 등록화면에서 목록화면으로 이동시
164  * com.changeMenu(null, null, {"DDD":"ddd", "CCC":"ccc"});
165  */
166 changeMenu : function(title, src, param)
167 {
168
169     var vnTabIdx = $p.top().wdcMain.getSelectedIndex();
170
171     // chageMenu는 권한처리 하지 않는 가정하에 권한처리 일단 제외.
172     /*if(com.data.isEmpty(vnTabIdx) || vnTabIdx < 0){
173         com.alert("", "현재 탭의 정보가 존재하지 않습니다.");
174         return;
175     }
176     var vnTabId = $p.top().wdcMain.getSelectedWindowId();
177     var voTabObj = $p.top().wdcMain.getWindowByWindowId(vnTabId, false);
178     if(com.data.isEmpty(voTabObj)){
179         com.alert("", "현재 탭의 정보가 존재하지 않습니다.");
180         return;
181     }
182     var voMenuInfo = voTabObj["scwin"]["_menuInfo"];
183     if(com.data.isEmpty(voMenuInfo)){
184         com.alert("", "현재 탭의 메뉴정보가 존재하지 않습니다.");
185         return;
186     }
187
188     psSrc = com.data.isEmpty(psSrc, voMenuInfo["url"]);
189
190     var voTabOpts = {};
191     voTabOpts["title"] = com.data.isEmpty(psTitle, voMenuInfo["menuNm"]);
192     voTabOpts["windowId"] = voMenuInfo["menuId"];
193     voTabOpts["src"] = psSrc;
194     voTabOpts["windowTitle"] = voMenuInfo["menuTree"];
195
196     //com.addWindow($p.top().wdcMain, voTabOpts, voParam);
197     */
198
199     $p.top().wdcMain.getFrame($p.top().wdcMain.getSelectedWindowId()).setSrc(src, {"dataObject":
200 },
201
202 /**
203  * 메뉴번호에 해당하는 메뉴정보를 return
204  *

```

```

205 * @date 2020.06.11
206 * @memberOf com
207 * @param {String} psMenuId 메뉴번호
208 * @return {JSON} 메뉴번호에 해당하는 첫번째 메뉴정보
209 * @author 전기완
210 * @example
211 */
212 getMenuInfo : function(psMenuId)
213 {
214     var voMenuList = {};
215     var voMenuInfo = {};
216     if(sessionStorage.getItem("mode") === "N")
217     {
218         voMenuList = $p.top().wfm_mainLeft.getWindow().dlMenuList;
219     }
220     else if(sessionStorage.getItem("mode") === "I")
221     {
222         voMenuList = $p.top().wfm_mainLeft.getWindow().tacLeftInsaMenu.getWindow(0).dlMenuList;
223     }
224
225     if(com.data.isNotEmpty(voMenuList))
226     {
227         voMenuInfo = voMenuList.getMatchedJSON("menuId", psMenuId, true);
228         if(voMenuInfo.length > 0) voMenuInfo = voMenuInfo[0];
229     }
230     return voMenuInfo;
231 },
232
233 /**
234 * 현재 화면의 웹스퀘어 page 경로를 반환한다.
235 *
236 * @date 2016.07.19
237 * @memberOf com
238 * @author InswaveSystems
239 * @return {String} 현재 페이지의 경로
240 * @example com.getPageUrl(); // return 예시) "/ui/BM/BM001M01.xml"
241 */
242 getPageUrl : function()
243 {
244     var pArr = document.location.href.split("w2xPath=");
245     var oArr = pArr[1].split("&");
246     return oArr[0];
247 },
248
249 /**
250 * 실행 프레임 정보 조회
251 *
252 * @date 2019.03.13
253 * @memberOf com
254 * @author InswaveSystems
255 * @returns {Object} 현재 Active Window 정보 반환
256 * @returns {String} activeinfo.type : 액티브 윈도우 타입 [P : 팝업, T : 탭컨텐츠, W : 윈도우컴포넌트]
257 * @returns {Object} activeinfo.window : 액티브 윈도우 객체
258 * @returns {String} activeinfo.programCd : 액티브 윈도우 프로그램 코드
259 */
260 getActiveWindowInfo : function()
261 {
262     var activeInfo = {
263         "type" : "", // T:Tabcontrol, W>windowContainer, P:popup
264         "window" : "", // Window객체
265         "programCd" : "" // 프로그램 코드 (팝업인경우는 예외)
266     };
267     try {
268         var domActiveElement = document.activeElement;
269         var isPopup = $p.top().$.getComponentById(domActiveElement.id);
270         var popupList = WebSquare.uiplugin.popup.popupList;
271         if (popupList.length > 0) {
272             for (var i = popupList.length - 1; i > -1; i--) {

```

```

273         if (WebSquare.uiplugin.popup.popupList[i].options.modal) {
274             isPopup = WebSquare.uiplugin.popup.popupList[i].popupWin;
275         }
276     }
277 }
278
279 if (isPopup && popupList.length > 0) {
280     // 팝업이 현재 열려있는상태.
281     activeInfo["type"] = "P";
282     activeInfo["window"] = isPopup.frame.getWindow();
283 } else {
284     activeInfo["type"] = $p.top().scwin.getLayoutId();
285     if (activeInfo["type"] == "T") {
286         var selectedTabId = $p.top().tac_layout.getSelectedTabID();
287         var findProgramList = com.menuComp.getMatchedJSON("MENU_CD", selectedTabId, true
288         if (findProgramList.length > 0) {
289             activeInfo["programCd"] = findProgramList[0].PROGRAM_CD;
290         }
291         activeInfo["window"] = $p.top().tac_layout.getWindow(selectedTabId);
292     } else if (activeInfo["type"] == "M") {
293         var selectedWindowId = $p.top().wdcMain.getSelectedWindowId();
294         var findProgramList = com.menuComp.getMatchedJSON("MENU_CD", selectedWindowId, t
295         if (findProgramList.length > 0) {
296             activeInfo["programCd"] = findProgramList[0].PROGRAM_CD;
297         }
298         activeInfo["window"] = $p.top().wdcMain.getWindow(selectedWindowId);
299     }
300 }
301 }
302 } catch (e) {
303 }
304 }
305 return activeInfo;
306 },
307
308 /**
309  * wframe의 src를 변경한다.
310  * wframe scope true일 경우만 파라미터를 넘길 수 있다.
311  *
312  * @date 2020.06.15
313  * @memberOf com
314  * @param {Object} poWframe wframe Object
315  * @param {String} psSrc open할 UI소스 경로
316  * @param {JSON} poParam wframe에 넘겨줄 파라미터 JSON Object
317  * @return {void}
318  * @author 전기완
319  * @example
320  *
321  * com.wfmSetSrc(wfm_sample, "/ui/sample.xml", {"AAA":"aaa","BBB","bbb"});
322  *
323  */
324 wfmSetSrc : function(poWframe, psSrc, poParam)
325 {
326     poWframe.setSrc(psSrc, {"dataObject": {"type":"json", "name":"param", "data":poParam}});
327 },
328
329 addHistory : function(menuNm, url, menuId)
330 {
331     var gen_history = $p.top().wfm_mainLeft.getWindow().gen_history;
332
333     // 맨 마지막에 열린 화면과 동일한 화면을 다시 오픈하는 경우에는 최근작업목록에 저장하지 않도록 반환한다.
334     if (gen_history.getLength() >= 1) {
335         var lastMenuId = gen_history.getChild(0, "btn_History").getUserData("tmpMenuInfo").split
336         if (menuId === lastMenuId) {
337             return;
338         }
339     }
340 }

```

```

341     gen_history.insertChild(0);
342     var curObj = gen_history.getChild(0, "btn_History");
343
344     var tmpHH = $p.getFormattedDate(new Date(), "yyyy-MM-dd HH:mm");
345     curObj.setValue(menuNm);
346     curObj.setTitle "[" + tmpHH + "]" + menuNm;
347     curObj.setUserData("tmpMenuInfo", menuId + "|" + url);
348 },
349
350 historyResize : function()
351 {
352     var gen_history = $p.top().wfm_mainLeft.getWindow().gen_history;
353     var swl_lnbbox = $p.top().wfm_mainLeft.getWindow().swl_lnbbox;
354
355     var listHeight = gen_history.getSize("height");
356     var swlInBoxId = swl_lnbbox.getID();
357     $("##" + swlInBoxId + ".w2switch_case").animate({ "bottom" : listHeight + 44 }, 700);
358     $(".box_history").animate({ "height" : listHeight + 24 }, 700);
359 }
360 };
361
362
363 //-----
364 // Submission
365 //-----
366 /**
367  * Submission를 실행합니다.
368  *
369  * @date 2017.01.19
370  * @param {Object} options com.createSubmission의 options 참고
371  * @param {Object} requestData 요청 데이터
372  * @param {Object} obj 전송중 disable시킬 컴퍼넌트
373  * @memberOf com
374  * @author InswaveSystems
375  * @example
376  * var searchCodeGrpOption = {
377  *     id : "sbm_searchCodeGrp",
378  *     action : "serviceId=CD0001&action=R",
379  *     target : 'data:json,{"id":"dlt_codeGrp","key":"data"}',
380  *     submitDoneHandler : scwin.searchCodeGrpCallback, isShowMeg : false };
381  * com.executeSubmission_dynamic(searchCodeGrpOption);
382  */
383 com.sbm = {
384     executeSubmission : function(options, requestData, obj)
385     {
386         var submissionObj = $p.getSubmission(options.id);
387         if(submissionObj === null)
388         {
389             com.sbm.createSubmission(options);
390             submissionObj = $p.getSubmission(options.id);
391         }
392         else
393         {
394             $p.deleteSubmission(options.id);
395             com.sbm.createSubmission(options);
396             submissionObj = $p.getSubmission(options.id);
397         }
398         com.sbm.executeBasicSubmission(submissionObj, requestData, obj);
399     },
400
401     /**
402      * Submission 객체를 동적으로 생성한다.
403      *
404      * @date 2017.11.30
405      * @param {Object} options Submission 생성 옵션 JSON 객체
406      * @param {String} options.id submission 객체의 ID. 통신 모듈 실행 시 필요.
407      * @param {String} options.ref 서버로 보낼(request) DataCollection의 조건 표현식.(조건에 따라 표현식이 복잡
408      * @param {String} options.target 서버로 응답(response) 받은 데이터가 위치 할 DataCollection의 조건 표현식

```

```

409 * @param {String} options.action 통신 할 서버 측 URI.(브라우저 보안 정책으로 crossDomain은 지원되지 않는다.
410 * @param {String} options.method [default: get, post, urlencoded-post]
411 * - get : 파라메타를 url에 붙이는 방식 (HTML과 동일).
412 * - post : 파라메타를 body 구간에 담는 방식 (HTML과 동일)
413 * - urlencoded-post : urlencoded-post.
414 * @param {String} options.mediatype [default: application/xml, text/xml, application/json, appli
415 * application/x-www-form-urlencoded 웹 form 방식(HTML방식). application/json : json 방식. applicati
416 * (두 개 차이는 http://stackoverflow.com/questions/4832357 참조)
417 * @param {String} options.mode [default: synchronous, synchronous] 서버와의 통신 방식. asynchronous
418 * @param {String} options.encoding [default: utf-8, euc-kr, utf-16] 서버 측 encoding 타입 설정 (euc
419 * @param {String} options.replace [default: none, all, instance] action으로부터 받은 response data를
420 * - all : 문서 전체를 서버로부터 온 응답데이터로 교체.
421 * - instance : 해당되는 데이터 구간.
422 * - none : 교체안함.
423 * @param {String} options.processMsg submission 통신 중 보여줄 메시지.
424 * @param {String} options.errorHandler submission오류 발생 시 실행 할 함수명.
425 * @param {String} options.customHandler submission호출 시 실행 할 함수명.
426 * @param {requestCallback} options.submitHandler {script type="javascript" ev:event="xforms-subm
427 * @param {requestCallback} options.submitDoneHandler {script type="javascript" ev:event="xforms-
428 * @param {requestCallback} options.submitErrorHandler {script type="javascript" ev:event="xforms
429 * @memberOf com
430 * @author InswaveSystems
431 * @example
432 * com.sbm.createSubmission(options);
433 */
434 createSubmission : function(options)
435 {
436     var ref = options.ref || ""; // request dataCollection("dl_list" or ["dl_list", "dl_l
437     var target = options.target || ""; // response dataCollection("dl_list" or ["dl_list", "dl_
438     var action = gcm.CONTEXT_PATH + gcm.SERVICE_URL + options.action; // ajax 요청주소
439     var mode = options.isSync === true ? 'synchronous' : 'asynchronous'; // sync방식으로 처리 할
440     var mediatype = options.mediatype || gcm.DEFAULT_OPTIONS_MEDIATYPE; // application/x-www
441     var method = (options.method || "post").toLowerCase(); // get/post/put/delete
442     var processMsg = options.processMsg || "처리중입니다.";
443     var instance = options.instance || "none";
444     var append = options.append || false;
445
446     var isShowMeg = options.isShowMeg === false ? false : true; // 처리중 메시지를 숨길 경우 options.
447
448     if (isShowMeg === false) processMsg = "";
449
450     // request dataCollection처리(수정된것만 보낼 경우 ("dl_list:m" or ["dl_list:m", "dl_list2"....])
451     var newRef = "data:json,";
452     if(com.data.isEmpty(ref)){
453         // 단건일 경우
454         if(WebSquare.util.isString(ref)){
455             ref = [ref];
456         }
457         if(WebSquare.util.isArray(ref)){
458             ref.push("_sbmInfo");
459             var refArray = [];
460             var tempRef;
461             var vsId;
462             var vsKey;
463             for(var i = 0; i < ref.length; i++){
464                 vsId = ref[i];
465                 if(WebSquare.util.isString(ref[i])) {
466                     vsKey = vsId.replace(/:m/gi, "");
467                 } else {
468                     vsKey = vsId["key"];
469                     vsId = vsId["id"];
470                 }
471                 // 수정된것만 보낼 경우
472                 if(vsId.indexOf(":m") < 0){
473                     tempRef = '{"id":"' + vsId + '","key":"' + vsKey + '"}';
474                 } else {
475                     tempRef = '{"id":"' + vsId.replace(/:m/gi, "") + '","key":"' + vsKey + '","'
476                 }

```

```

477         refArray.push(tempRef);
478     }
479     newRef = newRef + JSON.stringify(refArray).replace(/"/gi, "").replace(/\\\/gi, "\\");
480 }
481 } else {
482     //newRef = newRef + '{"id": "_sbmInfo", "key": "_sbmInfo"}';
483     newRef = null;
484 }
485
486 // response dataCollection처리
487 var newTarget = "";
488 if(com.data.isNotEmpty(target)){
489     // data:json,dm_search
490     // data:json, [{"id": "dlt_commonGrp", "action": "modified"}, {"id": "dlt_commonCode", "action":
491     // data:json, {"id": "dlt_menuSearchItem", "key": "dlt_menu"}
492     newTarget = "data:json,";
493     // 단건일 경우
494     if(WebSquare.util.isString(target)){
495         newTarget = newTarget + '{"id":"' + target + '", "key":"' + target + '"';
496         if(append) {
497             newTarget += ', "action": "append"';
498         }
499         newTarget += '}'';
500         // n건일 경우
501     } else if(WebSquare.util.isArray(target)){
502         var targetArray = [];
503         var tempTarget;
504         for(var i = 0; i < target.length; i++){
505             tempTarget = '{"id":"' + target[i]["id"] + '", "key":"' + target[i]["key"] + '"';
506             if(append) {
507                 tempTarget += ', "action": "append"';
508             }
509             tempTarget += '}'';
510             targetArray.push(tempTarget);
511         }
512         newTarget = newTarget + JSON.stringify(targetArray).replace(/"/gi, "").replace(/\\\/g
513     }
514 }
515
516 var submissionObj = {
517     "id" : options.id, // submission 객체의 ID. 통신 모듈 실행 시 필요.
518     "ref" : com.data.isNotEmpty(newRef) ? newRef : "data:json,{ }", // 서버로 보낼(request) Data
519     "target" : newTarget, // 서버로 응답(response) 받은 데이터가 위치 할 DataCollection의 조건 표현식
520     "action" : action, // 통신 할 서버 측 URI.(브라우저 보안 정책으로 crossDomain은 지원되지 않는다.)
521     "method" : method, // [default: post, get, urlencoded-post] get:파라메타를 url에 붙이는 방식
522     "mediatype" : mediatype, // application/json
523     "encoding" : "UTF-8", // [default: utf-8, euc-kr, utf-16] 서버 측 encoding 타입 설정 (euc-kr
524     "mode" : mode, // [default: synchronous, asynchronous] 서버와의 통신 방식. asynchronous:비동기
525     "processMsg" : processMsg, // submission 통신 중 보여줄 메시지.
526     // "submitHandler" : com.submitHandler,
527     "submitDoneHandler" : function(sbmObj){
528         var submitDoneHandler = window[$p.id + "scwin"]["fnCallbackHandler"];
529         if(typeof submitDoneHandler === "function"){
530             var sbmId = (sbmObj["id"]).replace($p.id, "");
531             var resJson = com.data.getJSON(sbmObj["responseText"]);
532             submitDoneHandler(sbmId, resJson);
533         }
534     },
535     "submitErrorHandler" : function(sbmObj){
536         var sbmId = (sbmObj["id"]).replace($p.id, "");
537         var submitErrorHandler = window[$p.id + "scwin"]["fnErrorHandler"];
538         if(typeof submitErrorHandler === "function"){
539             var sbmId = (sbmObj["id"]).replace($p.id, "");
540             var resJson = com.data.getJSON(sbmObj["responseText"]);
541             submitErrorHandler(sbmId, resJson);
542         }
543     }
544 };

```

```

545     $p.createSubmission(submissionObj);
546 },
547
548 _setRequestheader : function(obj)
549 {
550     /*var _csrf_header = $("meta[name='_csrf_header']").attr('content');
551     var _csrf = $("meta[name='_csrf']").attr('content');
552
553     obj.setRequestHeader(''+_csrf_header, ''+_csrf);*/
554 },
555
556 _sbm_defaultPreSubmission : function(options)
557 {
558     var dsIdPrefix = $p.top().wdcMain.getFrame($p.top().wdcMain.getSelectedWindowId()).id + '_';
559     var dsList = JSON.parse(options.ref.substring(10));
560     for(var i = 0; i < dsList.length; i++) {
561         var dataSet = WebSquare.util.getComponentById(dsIdPrefix + dsList[i].id);
562         var recordVal = dataSet.get('RECORD_COUNT');
563         var pageVal = dataSet.get('CURRENT_PAGE_NO');
564         if(!recordVal || recordVal === '') {
565             dataSet.set('RECORD_COUNT', 10);
566         }
567         if(!pageVal || pageVal === '') {
568             dataSet.set('CURRENT_PAGE_NO', 1);
569         }
570     }
571     console.log('global pre-submit is called.....');
572 },
573
574 //com._sbm_defCallbackSubmission = function(resObj, subObj)
575 _sbm_defCallbackHandler : function(options)
576 {
577     console.log('global submit-callback is called.....');
578     $p.top().ifrMsg.getWindow().document.getElementById('divInfoMsg').innerText = '정상처리 되었습니다'
579
580     // 정상처리후 Local 처리
581     try {
582         //var menuId = $p.top().wdcMain.getSelectedWindowId();
583         // $p.top().wdcMain.getWindow(menuId).scwin.fnCallbackHandler(options);
584     } catch(e) {
585         // welcome은 menuId가 없나? 찾아서 Callback 직접 해줘야 하네
586         //mf_wfm_mainLeft.scopeVariable.scwin.fnCallbackHandler(options);
587     }
588 },
589
590 _sbm_defaultErrorHandler : function(resObj)
591 {
592     console.log('global error-callback is called.....');
593     $p.top().ifrMsg.getWindow().document.getElementById('divInfoMsg').innerText = '오류가 발생하였습니다'
594     resObj = resObj || {};
595     var errMsg = resObj["responseBody"];
596     com.popup.alert(errMsg);
597 },
598
599 /**
600  * 서버 통신 확장 모듈, Submission를 실행합니다.
601  *
602  * @date 2017.11.30
603  * @param {Object} sbmObj submission 객체
604  * @param {Object} requestData [Default : null, JSON, XML] 요청 데이터로 submission에 등록된 ref를 무시
605  * @param {Object} compObj [Default : null] 전송중 disable시킬 컴퍼넌트
606  * @memberOf com
607  * @author InswaveSystems
608  * @example
609  * // Submission ID : sbm_init 존재할 경우
610  * com.executeSubmission(sbm_Init);
611  * // return 예시) sbm_init 통신 실행
612  *

```

```

613 * // Submission ID : sbm_init 존재하지 않을 경우
614 * com.executeSubmission(sbm_Init);
615 * // return 예시) alert - submission 객체[sbm_init]가 존재하지 않습니다.
616 */
617 executeBasicSubmission : function(sbmObj, requestData, compObj)
618 {
619
620     if (sbmObj.action.indexOf(gcm.CONTEXT_PATH) < -1) {
621         sbmObj.action = gcm.CONTEXT_PATH + sbmObj.action;
622     }
623
624     $p.executeSubmission(sbmObj, requestData, compObj);
625 }
626 };
627
628
629
630 //-----
631 // Grid
632 //-----
633 com.grid = {
634
635     /**
636      * 그리드 상수 선언
637      */
638     CHECK : '1',
639     INSERT : 'C',
640     READ : 'R',
641     UPDATE : 'U',
642     DELETE : 'D',
643
644     /**
645      * GridView와 바인딩된 DataList 객체를 반환한다.
646      *
647      * @date 2018.01.11
648      * @param {Object} gridViewObj 바인딩 된 DataList가 존재하는지 검증할 GridView 객체
649      * @memberOf com
650      * @author InswaveSystems
651      * @return {Object} 바인딩 된 DataList 객체 반환 (바인딩 객체가 없을 경우 null 반환)
652      * @example
653      * // 바인딩 되어있는 경우
654      * com.getGridViewDataList(grd_First);
655      * // return 예시) "dlt_first"
656      *
657      * // 바인딩 되어있지 않은 경우
658      * com.getGridViewDataList(grd_First);
659      * // return 예시) undefined
660      */
661     getGridViewDataList : function(gridViewObj)
662     {
663         var dataListId = gridViewObj.getDataList();
664
665         if (dataListId !== "") {
666             var dataList = $p.getComponentById(dataListId);
667             if ((typeof dataList === "undefined") || (dataList === null)) {
668                 $p.log("DataList(" + dataListId + ")를 찾을 수 없습니다.");
669                 return null;
670             } else {
671                 return dataList;
672             }
673         } else {
674             $p.log(gridViewObj.getID() + "는 DataList가 세팅되어 있지 않습니다.");
675             return null;
676         }
677     },
678 },
679
680 /**

```

```

681 * 특정 컴포넌트에 바인된 DataList나 DataMap의 정보를 반환한다.
682 *
683 * @date 2018.01.15
684 * @memberOf com
685 * @param {Object} comObj callerObj 컴포넌트 객체
686 * @returns {Object} dataCollection정보
687 */
688 getDataCollection : function(comObj)
689 {
690     try {
691         if ((typeof comObj !== "undefined") && (typeof comObj.getRef === "function")) {
692             var ref = comObj.options.ref;
693             if ((typeof ref !== "undefined") && (ref !== "")) {
694                 var refArray = ref.substring(5).split(".");
695                 if ((typeof refArray !== "undefined") && (refArray.length === 2)) {
696                     var dataObjInfo = {};
697                     dataObjInfo.runtimeDataCollectionId = comObj.getScope().id + "_" + refArray[0];
698                     dataObjInfo.dataCollectionId = refArray[0];
699                     dataObjInfo.columnId = refArray[1];
700                     return dataObjInfo;
701                 } else {
702                     return null;
703                 }
704             } else {
705                 return null;
706             }
707         }
708     } catch (e) {
709         console.log("[com.getDataCollection] Exception :: " + e.message);
710     } finally {
711         dataCollection = null;
712     }
713 },
714 /**
715 * 데이터 리스트 필터 설정 함수.
716 *
717 * @date 2019.03.13
718 * @memberOf com
719 * @author InswaveSystems
720 * @param {String} _dataComp : 필터 설정 데이터컬렉션 객체ID(dataList 혹은 linkedDataList)
721 * @param {String} _condition : 필터 상태값
722 * @param {Boolean} _append targetPgCd : 설정 필터 추가 혹은 재정의 [default:true (추가), false(재정의)]
723 * @param {String} options : 필터 상태값이 'LIKE' 조건일 경우 추가 옵션
724 * @param {String} options.colId : LIKE 검색 컬럼 ID
725 * @param {String} options.func : LIKE 검색 조건 실행 함수명
726 * @returns {Object} 필터 결과 데이터
727 */
728 dataListFilter : function(_dataComp, _condition, _append, options)
729 {
730     try {
731         var filterDs = $p.getComponentById(_dataComp);
732         var filterDsId = filterDs.getID();
733         var conditions = _condition;
734         var append = (typeof _append === "undefined") ? true : _append;
735         var cols = conditions.split(/\W+/g);
736         var uniqCols = new Array();
737         var sortArr = [];
738
739         // 필터 컬럼 추출.
740         $.each(cols, function(i, el) {
741             if (el !== "" && uniqCols.indexOf(el) == -1 && typeof filterDs.getColumnIndex(el) !=
742                 uniqCols.push(el);
743         }
744     ));
745
746     // 문자열 길이로 재정렬

```

```

749     uniqCols.sort(function(a, b) {
750         if (a.length < b.length)
751             return 1;
752         if (a.length > b.length)
753             return -1;
754         return 0;
755     });
756
757     // 필터 설정 전 sort 상태 확인.
758     if (filterDs && filterDs.sortStatusArr) {
759         sortArr = filterDs.sortStatusArr;
760     }
761
762     // Like 필터 기능.
763     if (typeof conditions == "string" && conditions.toUpperCase() == "LIKE") {
764         if (options && typeof options.colId != "undefined" && typeof options.func != "undefined") {
765             if (typeof filterDs.getColumnIndex(options.colId) == "undefined") {
766                 com.alert("like 사용자지정 필터 컬럼이 존재하지 않습니다.");
767             } else if (typeof options.func == "function") {
768                 var isAppend = WebSquare.util.getBoolean(append) ? true : false;
769                 if (!isAppend) {
770                     filterDs.removeColumnFilterAll();
771                 }
772                 var func = eval(options.func);
773                 if (typeof func == "function") {
774                     filterDs.setColumnFilter({
775                         type : 'func',
776                         colIndex : options.colId,
777                         key : func,
778                         condition : 'and'
779                     });
780                 }
781             }
782         } else {
783             com.alert("like 사용자지정 필터 값이 잘못되었습니다.");
784         }
785     } else { // 일반필터 기능.
786         var conditionArr = conditions.split(/[&]{2}|[|]{2}/);
787         var conditionArr1 = new Array();
788         var conditionArr2 = conditions.match(/[&]{2}|[|]{2}/g);
789
790         // 필터조건 [|] , & 구분.
791         for (var i = 0; i < conditionArr.length; i++) {
792             if ((conditionArr[i] + "").trim() != "") {
793                 conditionArr1.push(conditionArr[i]);
794             }
795         }
796
797         // 필터조건 정의.
798         for (var i = 0; i < conditionArr1.length; i++) {
799             var cond = conditionArr1[i];
800             // cond = cond.replace(/\s/gi, "");
801             var valArr = [ cond ];
802             var xBy = "";
803             if (cond.indexOf("==") > -1) {
804                 valArr = cond.split("==");
805                 xBy = "==" ;
806             } else if (cond.indexOf("!=") > -1) {
807                 valArr = cond.split("!=");
808                 xBy = "!=" ;
809             } else if (cond.indexOf(">") > -1) {
810                 valArr = cond.split(">");
811                 xBy = ">" ;
812             } else if (cond.indexOf("<") > -1) {
813                 valArr = cond.split("<");
814                 xBy = "<" ;
815             } else if (cond.indexOf(">=") > -1) {
816                 valArr = cond.split(">=");

```

```

817         xBy = ">=";
818     } else if (cond.indexOf("<=") > -1) {
819         valArr = cond.split("<=");
820         xBy = "<=";
821     }
822
823     for (var j = 0; j < valArr.length; j++) {
824         var chgCnt = -1;
825         for (var k = 0; k < uniqCols.length; k++) {
826             var deQ = (valArr[j] + "").trim();
827             if (deQ.indexOf(uniqCols[k]) > -1) {
828                 valArr[j] = valArr[j].replace(uniqCols[k], "####");
829                 chgCnt = k;
830                 break;
831             }
832         }
833         if (chgCnt > -1)
834             valArr[j] = valArr[j].replace(/####/g, filterDsId + ".getCellData(" + fi
835                 + uniqCols[chgCnt] + "')");
836     }
837
838     if (xBy) {
839         conditionArr1[i] = valArr[0] + xBy + valArr[1];
840     } else {
841         conditionArr1[i] = valArr[0];
842     }
843 }
844
845 var returnConditions = "";
846 if (conditionArr2 != null) {
847     for (var i = 0; i < conditionArr1.length; i++) {
848         if (typeof conditionArr2[i] !== "undefined") {
849             if (typeof conditionArr1[i + 1] === "undefined" || conditionArr1[i + 1] =
850                 || (conditionArr1[i + 1] + "").trim() === "") {
851                 returnConditions += conditionArr1[i];
852             } else {
853                 returnConditions += conditionArr1[i] + conditionArr2[i];
854             }
855         } else {
856             returnConditions += conditionArr1[i];
857         }
858     }
859 } else {
860     for (var i = 0; i < conditionArr1.length; i++) {
861         returnConditions += conditionArr1[i];
862     }
863 }
864
865 // 필터조건 새로정의 또는 필터 조건추가 여부.
866 var isAppend = WebSquare.util.getBoolean(append) ? true : false;
867 if (!isAppend) {
868     filterDs.removeColumnFilterAll();
869 }
870
871 // 필터 설정.
872 filterDs.setColumnFilter({
873     type : 'func',
874     colIndex : 0,
875     key : function(cellData, tmpParam, rowIdx) {
876         try {
877             if (eval(returnConditions)) {
878                 return true;
879             } else {
880                 return false;
881             }
882         } catch (e) {
883             console.log("[filter.setColumnFilter] Exception returnConditions :: " + i
884

```

```

885         }
886     },
887     condition : 'and'
888 });
889 }
890
891 // 정의 필터 이전 sort 재반영.
892 if (sortArr.length > 1) {
893     var sortIndex = "";
894     var sortOrder = "";
895     for (var i = 0; i < sortArr.length; i++) {
896         var colId = sortArr[i].colId;
897         var sortOrd = sortArr[i].sortOrder;
898         if (filterDs.getColumnID(colId) && sortOrd) {
899             if (sortIndex == "") {
900                 sortIndex += colId;
901             } else {
902                 sortIndex += " " + colId;
903             }
904
905             if (sortOrder == "") {
906                 sortOrder += sortOrd;
907             } else {
908                 sortOrder += " " + sortOrd;
909             }
910         }
911     }
912     filterDs.multisort({
913         "sortIndex" : sortIndex,
914         "sortOrder" : sortOrder
915     });
916 } else if (sortArr.length == 1) {
917     var sortCol = sortArr[0].colId;
918     var sortOrder = sortArr[0].sortOrder; // 1 : 오름차순, -1 : 내림차순
919     if (sortOrder == 1) {
920         sortOrder = 0;
921     } else if (sortOrder == -1) {
922         sortOrder = 1;
923     } else {
924         sortOrder = 2;
925     }
926     filterDs.sort(sortCol, sortOrder); // 0이면 오름차순 1이면 내림차순 2이면 정렬을 취소
927 }
928
929 return filterDs.getAllFilteredJSON();
930 } catch (e) {
931 }
932 }
933 },
934
935 /**
936 * 특정 컴포넌트에 바인된 DataList나 DataMap의 컬럼 이름을 반환한다.
937 *
938 * @date 2018.01.15
939 * @memberOf com
940 * @param {Object} comObj 컴포넌트 객체
941 * @return {String} 컬럼명
942 */
943 getColumnName : function(comObj)
944 {
945     try {
946         if ((typeof comObj.getRef) === "function") {
947             var ref = comObj.getRef();
948             var refArray = ref.substring(5).split(".");
949             var dataCollectionName = refArray[0];
950             var columnId = refArray[1];
951
952             if ((typeof refArray !== "undefined") && (refArray.length === 2)) {

```

```

953         var dataCollection = comObj.getScopeWindow().$.getComponentById(dataCollectionN
954         var dataType = dataCollection.getObjectType().toLowerCase();
955         if (dataType === "datamap") {
956             return dataCollection.getName(columnId);
957         } else if (dataType === 'datalist') {
958             return dataCollection.getColumnName(columnId);
959         }
960     } else {
961         return "";
962     }
963 }
964 } catch (e) {
965     $p.log("[com.getColumnName] Exception :: " + e.message);
966 } finally {
967     dataCollection = null;
968 }
969 },
970
971 /**
972  * DataCollection 객체의 변경된 데이터가 있는지 검사한다.
973  *
974  * @date 2018.01.16
975  * @memberOf com
976  * @param {Object} dcObjArr DataCollection 또는 배열
977  * @author InswaveSystems
978  * @returns {Boolean} 검사결과 (true or false)
979  */
980 checkModified : function(dcObjArr)
981 {
982     if (com.data.getObjectType(dcObjArr) === "array") {
983         for ( var dcObj in dcObjArr ) {
984             if (com.data.getObjectType(dcObj) === "object") {
985                 if (checkModified(dcObj) === false) {
986                     return false;
987                 }
988             }
989         }
990     } else if (com.data.getObjectType(dcObjArr) === "object") {
991         return checkModified(dcObj);
992     }
993
994     return true;
995
996     function checkModified(dcObj) {
997         if ((typeof dcObj !== "undefined") && (typeof dcObj !== null)) {
998             var modifiedIndex = dcObj.getModifiedIndex();
999             if (modifiedIndex.length === 0) {
1000                 com.alert("변경된 데이터가 없습니다.");
1001                 return false;
1002             } else {
1003                 return true;
1004             }
1005         } else {
1006             return true;
1007         }
1008     }
1009 },
1010
1011 /**
1012  * GridView Row 삭제용 CheckBox 동작을 세팅한다.
1013  * GridView에 삭제용 CheckBox가 있을 경우 onPageLoad 이벤트에서 com.setGridViewDelCheckBox 함수를 호출한다.
1014  * 이 함수가 정상 동작하려면 GridView의 Delete 처리용 CheckBox의 ColumnId와 Header Id를 "chk"로 설정해야 한다
1015  *
1016  * @date 2018.11.26
1017  * @memberOf com
1018  * @author InswaveSystems
1019  * @param {Array} gridViewObj GridView 객체 배열
1020  * @example

```

```

1021 * com.setGridViewDelCheckBox(grd_OrganizationBasic);
1022 * com.setGridViewDelCheckBox([grd_Menu, grd_MenuAccess]);
1023 */
1024 setGridViewDelCheckBox : function(gridViewObjArr)
1025 {
1026     try {
1027         if (com.getObjectType(gridViewObjArr) === "array") {
1028             for (idx in gridViewObjArr) {
1029                 setGridViewDelCheckBox(gridViewObjArr[idx]);
1030             }
1031         } else {
1032             setGridViewDelCheckBox(gridViewObjArr);
1033         }
1034
1035         function setGridViewDelCheckBox(gridViewObj) {
1036             gridViewObj.bind("oncellclick", function(row, col) {
1037                 if (col == 0) {
1038                     var dltObj = com.getGridViewDataList(this);
1039                     var realRowIndex = this.getRealRowIndex(row);
1040                     var newValue = dltObj.getCellData(realRowIndex, col);
1041                     com._setGridViewRowCheckBox(this, realRowIndex, newValue === "1" ? true : fa
1042                 )
1043             });
1044             gridViewObj.bind("onheaderclick", function(headerId) {
1045                 if (headerId == "chk") {
1046                     var newValue = this.getHeaderValue(headerId);
1047                     var dltObj = com.getGridViewDataList(this);
1048                     var rowCount = dltObj.getRowCount();
1049
1050                     dltObj.setBroadcast(false);
1051                     for (var rowIdx = 0; rowIdx < rowCount; rowIdx++) {
1052                         com._deleteGridViewRow(this, rowIdx, newValue);
1053                     }
1054                     dltObj.setBroadcast(true, true);
1055                 }
1056             });
1057         }
1058     } catch (e) {
1059         $p.log("[com.setGridViewDelCheckBox] Exception :: " + e.message);
1060     }
1061 },
1062
1063 _setGridViewRowCheckBox : function(gridViewObj, rowIndex, newValue)
1064 {
1065     var rowIndexArr = gridViewObj.getChildrenRowIndexArray(rowIndex);
1066     var dltObj = com.getGridViewDataList(gridViewObj);
1067
1068     for ( var idx in rowIndexArr) {
1069         var childRowIndexArr = gridViewObj.getChildrenRowIndexArray(rowIndexArr[idx]);
1070
1071         if (childRowIndexArr.length > 0) {
1072             com._setGridViewRowCheckBox(gridViewObj, rowIndexArr[idx], newValue);
1073         } else {
1074             com._deleteGridViewRow(gridViewObj, rowIndexArr[idx], newValue);
1075         }
1076     }
1077
1078     com.grid._deleteGridViewRow(gridViewObj, rowIndex, newValue);
1079 },
1080
1081 _deleteGridViewRow : function(gridViewObj, rowIndex, newValue)
1082 {
1083     gridViewObj.setCellChecked(rowIndex, "chk", newValue);
1084     var dltObj = com.getGridViewDataList(gridViewObj);
1085
1086     if (newValue) {
1087         var rowStatus = dltObj.getRowStatus(rowIndex);
1088         if (rowStatus == "C") {

```

```

1089         dltObj.removeRow(rowIndex);
1090     } else {
1091         dltObj.deleteRow(rowIndex);
1092     }
1093 } else {
1094     dltObj.undoRow(rowIndex);
1095 }
1096 },
1097
1098 /**
1099  * 엑셀 다운로드 옵션을 설정하고 확장자 별로 다른 함수(downloadCSV || downloadExcel)를 호출한다.
1100  *
1101  * @date 2016.11.16
1102  * @param {Object} grdObj GridView Object
1103  * @param {Array} options JSON형태로 저장된 그리드의 엑셀 다운로드 옵션
1104  * @param {Array} infoArr 그리드에 대한 내용을 추가로 다른 셀에 표현하는 경우 사용하는 배열
1105  * @memberOf com
1106  * @author InswaveSystems
1107  * @example
1108  * var gridId = "grd_AdvancedExcel";
1109  * var infoArr = {};
1110  * var options = {
1111  *     fileName : "gridDataDownload.xlsx" //[defalut: excel.xlsx] 다운로드하려는 파일의 이름으로 필수 입력
1112  * };
1113  * com.gridDataDownload( gridId, options, infoArr);
1114  * //return 예시) common.js의 다른 함수(downloadCSV, downloadExcel)로 이동하거나 alert 발생
1115  */
1116 gridDataDownload : function(grdObj, options, infoArr)
1117 {
1118     if (options == undefined)
1119         options = {};
1120     if (infoArr == undefined)
1121         infoArr = {};
1122     var fileName = options.fileName;
1123
1124     if (fileName == undefined || fileName.length == 0) {
1125         var fileName = "excel_download";
1126         var dataCollectionName = grdObj.getDataList();
1127         if (dataCollectionName != null && typeof dataCollectionName != "undefined" && dataCollectionName != "") {
1128             var dataCollection = $p.getComponentById(dataCollectionName);
1129             var name = dataCollection.element.getAttribute("name");
1130             if (typeof name != "undefined" && name != "")
1131                 fileName = name;
1132         }
1133         var fileType = "xlsx"
1134         if (options.fileType == undefined || options.fileType == "") {
1135             fileName = fileName + ".xlsx";
1136         } else {
1137             fileName = fileName + "." + options.fileType;
1138         }
1139         options.fileName = fileName;
1140     }
1141
1142     fileName = fileName.toLowerCase();
1143     if (options.useProvider == "true") {
1144         options.dataProvider = "com.inswave.template.provider.ExcelDown";
1145     }
1146     if (options.useSplitProvider == "true") {
1147         options.splitProvider = "com.inswave.template.provider.ExcelSplitDown";
1148     }
1149
1150     if (options.useProvider || options.useSplitProvider) {
1151         var colCnt = grdObj.getColumnCount();
1152         var columnsIDArr = new Array();
1153
1154         for (var i = 0; i < colCnt; i++) {
1155             if (grdObj.getColumnVisible(i)) {
1156                 var isRemoveCol = false;

```

```

1157         if (typeof options.excludeColumns != "undefined" && options.excludeColumns != nu
1158             for (var k = 0; k < options.excludeColumns.length; k++) {
1159                 if (grdObj.getColumnID(i) == options.excludeColumns[k]) {
1160                     isRemoveCol = true;
1161                     break;
1162                 }
1163             }
1164         }
1165         if (isRemoveCol) {
1166             continue;
1167         }
1168         columnsIDArr.push(grdObj.getColumnID(i));
1169     }
1170 }
1171
1172 var xmlDoc = WebSquare.xml.parse(options.providerRequestXml, false);
1173 WebSquare.xml.setValue(xmlDoc, "data/keyMap", columnsIDArr.join(", "));
1174 options.providerRequestXml = WebSquare.xml.serialize(xmlDoc);
1175 delete options.useProvider;
1176 delete options.useSplitProvider;
1177 }
1178
1179 if (fileName.indexOf(".csv") > -1) {
1180     com.grid.downloadCSV(grdObj, options);
1181 } else if (fileName.indexOf(".xlsx") > -1 || fileName.indexOf(".xls") > -1) {
1182     com.grid.downloadExcel(grdObj, options, infoArr);
1183 } else {
1184     com.popup.alert("알수없는 파일 형식입니다");
1185 }
1186 },
1187
1188 /**
1189  * 설정된 옵션으로 엑셀을 다운로드 한다.
1190  *
1191  * @date 2017.11.30
1192  * @param {Object} grdObj GridView 객체
1193  * @param {Object} options JSON형태로 저장된 그리드의 엑셀 다운로드 옵션
1194  * @param {String} options.fileName [defalut: excel.xlsx] 다운로드하려는 파일의 이름으
1195  * @param {String} [options.sheetName] [defalut: sheet] excel의 sheet의 이름
1196  * @param {String} [options.type] [defalut: 0] type 0 => 실제 데이터 <br>1 => 눈0
1197  * @param {String} [options.removeColumns] [defalut: 없음] 다운로드시 excel에서 삭제하려는 열0
1198  * @param {String} [options.removeHeaderRows] [defalut: 없음] 다운로드시 excel에서 삭제하려는 헤
1199  * @param {String} [options.foldColumns] [defalut: 없음] 다운로드시 excel에서 fold하려는 열
1200  * @param {Number} [options.startRowIndex] excel 파일에서 그리드의 데이터가 시작되는 행의 번호(헤
1201  * @param {Number} [options.startColumnIndex] excel 파일에서 그리드의 데이터가 시작되는 열의 번호(헤
1202  * @param {String} [options.headerColor] excel 파일에서 그리드의 header부분의 색
1203  * @param {String} [options.headerFontName] [defalut: 없음] excel파일에서 그리드의 header부분:
1204  * @param {String} [options.headerFontSize] excel 파일에서 그리드의 header부분의 font size
1205  * @param {String} [options.headerFontColor] excel 파일에서 그리드의 header부분의 font색
1206  * @param {String} [options.bodyColor] excel 파일에서 그리드의 body부분의 색
1207  * @param {String} [options.bodyFontName] [defalut: 없음] excel파일에서 그리드의 body부분의
1208  * @param {String} [options.bodyFontSize] excel 파일에서 그리드의 body부분의 font size
1209  * @param {String} [options.bodyFontColor] excel 파일에서 그리드의 body부분의 font색
1210  * @param {String} [options.subTotalColor] [defalut: #CCFFCC] excel파일에서 그리드의 subtot
1211  * @param {String} [options.subTotalFontName] [defalut: 없음] excel파일에서 그리드의 subtotal부
1212  * @param {String} [options.subTotalFontSize] [defalut: 10] excel파일에서 그리드의 subtotal부분
1213  * @param {String} [options.subTotalFontColor] [defalut: 없음] excel파일에서 그리드의 subtotal부
1214  * @param {String} [options.footerColor] [defalut: #008000] excel파일에서 그리드의 footer
1215  * @param {String} [options.footerFontName] [defalut: 없음] excel파일에서 그리드의 footer부분:
1216  * @param {String} [options.footerFontSize] [defalut: 10] excel파일에서 그리드의 footer부분:
1217  * @param {String} [options.footerFontColor] [defalut: 없음] excel파일에서 그리드의 footer부분:
1218  * @param {String} [options.oddRowBackgroundColor] excel 파일에서 그리드 body의 홀수줄의 배경색
1219  * @param {String} [options.evenRowBackgroundColor] [defalut: 없음] excel파일에서 그리드 body의 짝수줄
1220  * @param {String} [options.rowNumHeaderColor] [defalut: 없음] rowNumVisible 속성이 true인 경우
1221  * @param {String} [options.rowNumHeaderFontName] [defalut: 없음] rowNumVisible 속성이 true인 경우
1222  * @param {String} [options.rowNumHeaderFontSize] [defalut: 없음] rowNumVisible 속성이 true인 경우
1223  * @param {String} [options.rowNumHeaderFontColor] [defalut: 없음] rowNumVisible 속성이 true인 경우
1224  * @param {String} [options.rowNumBodyColor] [defalut: 없음] rowNumVisible 속성이 true인 경우

```

```

1225 * @param {String} [options.rowNumBodyFontName] [defalut: 없음] rowNumVisible 속성이 true인 경우
1226 * @param {String} [options.rowNumBodyFontSize] [defalut: 없음] rowNumVisible 속성이 true인 경우
1227 * @param {String} [options.rowNumBodyFontColor] [defalut: 없음] rowNumVisible 속성이 true인 경우
1228 * @param {String} [options.rowNumFooterColor] [defalut: 없음] rowNumVisible 속성이 true인 경우
1229 * @param {String} [options.rowNumFooterFontName] [defalut: 없음] rowNumVisible 속성이 true인 경우
1230 * @param {String} [options.rowNumFooterFontSize] [defalut: 없음] rowNumVisible 속성이 true인 경우
1231 * @param {String} [options.rowNumFooterFontColor] [defalut: 없음] rowNumVisible 속성이 true인 경우
1232 * @param {String} [options.rowNumSubTotalColor] [defalut: 없음] rowNumVisible 속성이 true인 경우
1233 * @param {String} [options.rowNumSubTotalFontName] [defalut: 없음] rowNumVisible 속성이 true인 경우
1234 * @param {String} [options.rowNumSubTotalFontSize] [defalut: 없음] rowNumVisible 속성이 true인 경우
1235 * @param {String} [options.rowNumSubTotalFontColor] [defalut: 없음] rowNumVisible 속성이 true인 경우
1236 * @param {String} [options.rowNumHeaderValue] [defalut: 없음] rowNumVisible 속성이 true인 경우
1237 * @param {String} [options.rowNumVisible] [defalut: false] 순서출력 유무
1238 * @param {String} [options.showProcess] [defalut: true] 다운로드 시 프로세스 창을 보여줄지
1239 * @param {String} [options.massStorage] [defalut: true] 대용량 다운로드 여부 <br>(default
1240 * @param {String} [options.showConfirm] [defalut: false] 다운로드 확인창을 띄울지 여부 <br>
1241 * @param {String} [options.dataProvider] [defalut: 없음] 대량데이터 처리 및 사용자 데이터를
1242 * @param {String} [options.providerRequestXml] [defalut: 없음] Provider 내부에서 사용할 XML 문자
1243 * @param {String} [options.userDataXml] [defalut: 없음] 사용자가 서버모듈 개발 시 필요한 데
1244 * @param {String} [options.bodyWordwrap] [defalut: false] 다운로드시 바디의 줄 바꿈 기능
1245 * @param {String} [options.useEuroLocal] [defalut: false] 다운로드시 유로화 처리 기능(,와 .
1246 * @param {String} [options.useHeader] [defalut: true] 다운로드시 Header를 출력 할지 여부
1247 * @param {String} [options.useSubTotal] [defalut: false] 다운로드시 SubTotal을 출력 할지
1248 * @param {String} [options.useFooter] [defalut: true] 다운로드시 Footer를 출력 할지 여부
1249 * @param {String} [options.separator] [defalut: ,] 다운로드시 서버로 데이터 전송할때, 데이
1250 * @param {Number} [options.subTotalScale] [defalut: -1] 다운로드시 subTotal 평균계산시 소수?
1251 * @param {String} [options.subTotalRoundingMode] [defalut: 없음] 다운로드시 subTotal 평균계산시 Ro
1252 * @param {String} [options.useStyle] [defalut: false] 다운로드시 css를 제외한, style을
1253 * @param {String} [options.freezePane] [defalut: "" ] 롤고정을 위한 좌표값 및 좌표값의 오픈
1254 * @param {String} [options.autoSizeColumn] [defalut: false] 너비자동맞춤 설정 유무 - 2016.08
1255 * @param {String} [options.displayGridlines] [defalut: false] 엑셀 전체 셀의 눈금선 제거 유무
1256 * @param {String} [options.colMerge] [defalut: false] colMerge된 컬럼을 Merge해서 출
1257 * @param {String} [options.useDataFormat] [defalut: 없음] 그리드 dataType이 text인 경우, 엑
1258 * @param {String} [options.indent] [defalut: 0] 그리드 dataType이 drilldown인 경우
1259 * @param {String} [options.columnMove] [defalut: false] 그리드 컬럼이동시 이동된 상태로 다
1260 * @param {String} [options.columnOrder] [defalut: 없음] 엑셀 다운로드시 다운로드되는 컬럼 순
1261 * @param {String} [options.fitToPage] [defalut: false] 엑셀 프린터 출력시 쪽맞춤 사용 유
1262 * @param {String} [options.landScape] [defalut: false] 엑셀 프린터 출력시 가로 방향 출력
1263 * @param {String} [options.fitWidth] [defalut: 1] 엑셀 프린터 출력시 용지너비
1264 * @param {String} [options.fitHeight] [defalut: 1] 엑셀 프린터 출력시 용지높이
1265 * @param {String} [options.scale] [defalut: 100] 엑셀 프린터 출력시 확대/축소 배율<br>
1266 * @param {String} [options.pageSize] [defalut: A4] 엑셀 프린터 출력시 인쇄용지 설정 <br>
1267 * @param {Object[]} [infoArr] subTotalFontName 그리드에 대한 내용을 추가로 다른
1268 * @param {Number} [infoArr.rowIndex] [defalut: "" ] 내용을 표시할 행번호
1269 * @param {Number} [infoArr.colIndex] [defalut: "" ] 내용을 표시할 열번호
1270 * @param {Number} [infoArr.rowSpan] [defalut: "" ] 병합할 행의 수
1271 * @param {Number} [infoArr.colSpan] [defalut: "" ] 병합할 열의 수
1272 * @param {String} [infoArr.text] [defalut: "" ] 표시할 내용
1273 * @param {String} [infoArr.textAlign] [defalut: "right"] 표시할 내용의 정렬 방법 <br> ]
1274 * @param {String} [infoArr.fontSize] [defalut: "10px"] font size 설정 20px, 10px,
1275 * @param {String} [infoArr.fontName] [defalut: "" ] font name 설정
1276 * @param {String} [infoArr.color] [defalut: "" ] font color 설정 red, blue, gree
1277 * @param {String} [infoArr.fontWeight] [defalut: "" ] font weight 설정 bold
1278 * @param {String} [infoArr.drawBorder] [defalut: "true"] cell의 border지정 true, fals
1279 * @param {String} [infoArr.wordWrap] [defalut: "" ] cell의 줄 바꿈 기능 true, false
1280 * @param {String} [infoArr.bgColor] [defalut: "" ] cell의 배경 color 설정 red, blue,
1281 * @memberOf com
1282 * @return {file} <b>Excel file</b>
1283 * @author InswaveSystems
1284 * @example
1285 * var gridId = "grd_AdvancedExcel";
1286 * var infoArr = {};
1287 * var options = {
1288 *   fileName : "downloadExcel.xlsx" //[default : excel.xlsx] options.fileName 값이 없을 경우 defau
1289 * };
1290 * com.downloadExcel(grdObj, options, infoArr );
1291 */
1292 downloadExcel : function(grdObj, options, infoArr)

```

```

1293 {
1294     if (typeof options === "undefined") {
1295         options = {
1296             hiddenVisible : false,
1297             fileName : "excel.xlsx"
1298         };
1299     }
1300
1301     if (typeof infoArr === "undefined") {
1302         infoArr = {};
1303     }
1304
1305     // excel 다운로드시 기본 설정으로 화면내의 hidden컬럼을 removeColumns에 포함시킨다.
1306     // 이를 원치 않을 경우 options.hiddenVisible = 'true' 로 설정한다.
1307     if (!options.hiddenVisible) {
1308         var grdCnt = grdObj.getTotalCol();
1309
1310         var hiddenColIndex = [];
1311         for (var idx = 0; idx < grdCnt; idx++) {
1312             if (!grdObj.getColumnVisible(idx)) {
1313                 hiddenColIndex.push(idx);
1314             }
1315         }
1316         // hidden 컬럼이 있는 경우만 추가할 수 있도록 (2016.10.28 추가)
1317         if (hiddenColIndex.length > 0) {
1318             if (options.removeColumns != null)
1319             {
1320                 if (options.removeColumns.length > 0) {
1321                     options.removeColumns = options.removeColumns + "," + hiddenColIndex[0];
1322                 } else {
1323                     options.removeColumns = hiddenColIndex.join(',');
1324                 }
1325
1326                 // 중복 요소 제거
1327                 var _removeColumnArr = options.removeColumns.split(",");
1328                 options.removeColumns = _removeColumnArr.reduce(function(a, b) {
1329                     if (a.indexOf(b) < 0) {
1330                         a.push(b);
1331                     }
1332                     return a;
1333                 }, []).join(',');
1334             }
1335         }
1336     }
1337
1338     var options = {
1339         fileName : options.fileName || "excel.xlsx", // String, [defalut: excel.xlsx]
1340         sheetName : options.sheetName || "sheet", // String, [defalut: sheet] excel의 s
1341         type : options.type || "0", // String, [defalut: 0] type이 0인 경우 실제 데이터 1인
1342         // 셀은 나오지 않음)
1343         removeColumns : options.removeColumns || "", // String, [defalut: 없음] 다운로드시
1344         removeHeaderRows : options.removeHeaderRows || "", // String, [defalut: 없음] 다
1345         foldColumns : options.foldColumns || "", // String, [defalut: 없음] 다운로드시 exc
1346         startRowIndex : options.startRowIndex || 0, // Number, excel파일에서 그리드의 데이
1347         startColumnIndex : options.startColumnIndex || 0, // Number, excel파일에서 그리드
1348         headerColor : options.headerColor || "#33CCCC", // String, excel파일에서 그리드의
1349         headerFontName : options.headerFontName || "", // String, [defalut: 없음] excel
1350         headerFontSize : options.headerFontSize || "10", // String, excel파일에서 그리드의
1351         headerFontColor : options.headerFontColor || "", // String, excel파일에서 그리드의
1352         bodyColor : options.bodyColor || "#FFFFFF", // String, excel파일에서 그리드의 body
1353         bodyFontName : options.bodyFontName || "", // String, [defalut: 없음] excel파일
1354         bodyFontSize : options.bodyFontSize || "10", // String, excel파일에서 그리드의 bod
1355         bodyFontColor : options.bodyFontColor || "", // String, excel파일에서 그리드의 bod
1356         subTotalColor : options.subTotalColor || "#CCFFCC", // String, [defalut: #CCFF
1357         subTotalFontName : options.subTotalFontName || "", // String, [defalut: 없음] e
1358         subTotalFontSize : options.subTotalFontSize || "10", // String, [defalut: 10]
1359         subTotalFontColor : options.subTotalFontColor || "", // String, [defalut: 없음]
1360         footerColor : options.footerColor || "#008000", // String, [defalut: #008000]

```

```

1361 footerFontName : options.footerFontName || "", // String, [defalut: 없음] excel파일에서 그리
1362 footerFontSize : options.footerFontSize || "10", // String, [defalut: 10] excel파일에서 그리
1363 footerFontColor : options.footerFontColor || "", // String, [defalut: 없음] excel파일에서 그리
1364 oddRowBackgroundColor : options.oddRowBackgroundColor || "", // String, excel파일에서 그리드
1365 evenRowBackgroundColor : options.evenRowBackgroundColor || "", // String, [defalut: 없음]
1366 rowNumHeaderColor : options.rowNumHeaderColor || "", // String, [defalut: 없음] rowNumVis
1367 rowNumHeaderFontName : options.rowNumHeaderFontName || "", // String, [defalut: 없음] row
1368 rowNumHeaderFontSize : options.rowNumHeaderFontSize || "", // String, [defalut: 없음] row
1369 rowNumHeaderFontColor : options.rowNumHeaderFontColor || "", // String, [defalut: 없음] r
1370 // 폰트색상
1371 rowNumBodyColor : options.rowNumBodyColor || "", // String, [defalut: 없음] rowNumVisible
1372 rowNumBodyFontName : options.rowNumBodyFontName || "", // String, [defalut: 없음] rowNumV
1373 rowNumBodyFontSize : options.rowNumBodyFontSize || "", // String, [defalut: 없음] rowNumV
1374 rowNumBodyFontColor : options.rowNumBodyFontColor || "", // String, [defalut: 없음] rowNu
1375 rowNumFooterColor : options.rowNumFooterColor || "", // String, [defalut: 없음] rowNumVis
1376 rowNumFooterFontName : options.rowNumFooterFontName || "", // String, [defalut: 없음] row
1377 rowNumFooterFontSize : options.rowNumFooterFontSize || "", // String, [defalut: 없음] row
1378 rowNumFooterFontColor : options.rowNumFooterFontColor || "", // String, [defalut: 없음] r
1379 // 폰트색상
1380 rowNumSubTotalColor : options.rowNumSubTotalColor || "", // String, [defalut: 없음] rowNu
1381 // 배경색
1382 rowNumSubTotalFontName : options.rowNumSubTotalFontName || "", // String, [defalut: 없음]
1383 // 영역의 폰트이름
1384 rowNumSubTotalFontSize : options.rowNumSubTotalFontSize || "", // String, [defalut: 없음]
1385 // 영역의 폰트크기
1386 rowNumSubTotalFontColor : options.rowNumSubTotalFontColor || "", // String, [defalut: 없음]
1387 // Subtotal 영역의 폰트색상
1388 rowNumHeaderValue : options.rowNumHeaderValue || "", // String, [defalut: 없음] rowNumVis
1389 rowNumVisible : options.rowNumVisible || "false", // String, [defalut: false] 순서출력 유무
1390 showProcess : WebSquare.util.getBoolean(options.showProcess) || true, // Boolean, [defalut: true]
1391 massStorage : WebSquare.util.getBoolean(options.massStorage) || true, // Boolean, [defalut: true]
1392 // true로 하고 showConfirm을 false로 한 경우에 IE에서 신뢰할만한 사이트를 체크하는
1393 // 옵션이 뜬다.)
1394 showConfirm : WebSquare.util.getBoolean(options.showConfirm) || false, // Boolean, [defalut: false]
1395 // advancedExcelDownload를 호출후 사용자가 window의 버튼을 한번더 클릭해야 한다.
1396 // massStorage는 자동으로 true가 된다)
1397 dataProvider : options.dataProvider || "", // String, [defalut: 없음] 대량데이터 처리 및 사용
1398 splitProvider : options.splitProvider || "", // String, [defalut: 없음] 대량데이터 처리 및 사
1399 providerRequestXml : options.providerRequestXml || "", // String, [defalut: 없음] Provide
1400 userDataXml : options.userDataXml || "", // String, [defalut: 없음] 사용자가 서버모듈 개발 시
1401 bodyWordwrap : WebSquare.util.getBoolean(options.bodyWordwrap) || false, // Boolean, [defalut: false]
1402 useEuroLocale : options.useEuroLocale || "false", // String, [defalut: false] 다운로드시 유
1403 useHeader : options.useHeader || "true", // String, [defalut: true] 다운로드시 Header를 출력
1404 useSubTotal : options.useSubTotal || "false", // String, [defalut: false] 다운로드시 SubTotal
1405 // expression을 지정한 경우 avg,sum,min,max,targetColValue,숫자를 지원 함.
1406 useFooter : options.useFooter || "true", // String, [defalut: true] 다운로드시 Footer를 출력
1407 separator : options.separator || ",", // String, [defalut: ,] 다운로드시 서버로 데이터 전송할때
1408 subTotalScale : options.subTotalScale || -1, // Number, [defalut: -1] 다운로드시 subTotal
1409 subTotalRoundingMode : options.subTotalRoundingMode || "", // String, [defalut: 없음] 다운
1410 // ("CEILING","FLOOR","HALF_UP")
1411 useStyle : options.useStyle || "", // String, [defalut: false] 다운로드시 css를 제외한, styl
1412 freezePane : options.freezePane || "", // String, [defalut: ""] 틀고정을 위한 좌표값 및 좌표값
1413 // freezePane="0,1,0,5" X축 0, Y축 1에서 X축으로 0, Y축으로 5로 틀고정 )
1414 autoSizeColumn : options.autoSizeColumn || "true", // String, [defalut: false] 너비자동맞춤
1415 displayGridlines : options.displayGridlines || "", // String, [defalut: false] 엑셀 전체 셀
1416 colMerge : options.colMerge || "", // String, [defalut: false] colMerge된 컬럼을 Merge해서
1417 useDataFormat : options.useDataFormat || "", // String, [defalut: 없음] 그리드 dataType이 t
1418 // 표시형식 텍스트, "false"인 경우 표시형식 일반 출력)
1419 indent : options.indent || "", // String, [defalut: 없음] 그리드 dataType이 drilldown인 경우
1420 columnMove : options.columnMove || "", // String, [defalut: false] 그리드 컬럼이동시 이동된 순
1421 columnOrder : options.columnOrder || "", // String, [defalut: 없음] 엑셀 다운로드시 다운로드되
1422 // 순서로 다운로드된다 )
1423 fitToPage : options.fitToPage || "false", // String, [defalut: false] 엑셀 프린터 출력시 쪽맞
1424 landscape : options.landscape || "false", // String, [defalut: false] 엑셀 프린터 출력시 가로
1425 fitWidth : options.fitWidth || "1", // String, [defalut: 1] 엑셀 프린터 출력시 용지너비
1426 fitHeight : options.fitHeight || "1", // String, [defalut: 1] 엑셀 프린터 출력시 용지높이
1427 scale : options.scale || "100", // String, [defalut: 100] 엑셀 프린터 출력시 확대/축소 배율, s
1428 pageSize : options.pageSize || "A4" // String, [defalut: A4] 엑셀 프린터 출력시 인쇄용지 설정

```

```

1429     };
1430
1431     var infoArr = {
1432         rowIndex : infoArr.rowIndex || 0, // Number, 내용을 표시할 행번호
1433         colIndex : infoArr.colIndex || 0, // Number, 내용을 표시할 열번호
1434         rowSpan : infoArr.rowSpan || 0, // Number, 병합할 행의 수
1435         colSpan : infoArr.colSpan || 0, // Number, 병합할 열의 수
1436         text : infoArr.text || "", // String, 표시할 내용
1437         textAlign : infoArr.textAlign || "right", // String, 표시할 내용의 정렬 방법 left, center, right
1438         fontSize : infoArr.fontSize || "10px", // String, font size 설정 20px, 10px, 5px
1439         fontName : infoArr.fontName || "", // String, font name 설정
1440         color : infoArr.color || "", // String, font color 설정 red, blue, green
1441         fontWeight : infoArr.fontWeight || "", // String, font weight 설정 bold
1442         drawBorder : infoArr.drawBorder || "true", // String, cell의 border 지정 true, false
1443         wordWrap : infoArr.wordWrap || "", // String, cell의 줄 바꿈 기능 true, false
1444         bgColor : infoArr.bgColor || "" // String, cell의 배경 color 설정 red, blue, green
1445     };
1446
1447     grdObj.advancedExcelDownload(options, infoArr);
1448 },
1449
1450 /**
1451  * 설정된 옵션으로 CSV파일을 다운로드 한다.
1452  *
1453  * @date 2017.11.30
1454  * @param {Object} grdObj GridView Object
1455  * @param {Object[]} options JSON형태로 저장된 그리드의 엑셀 다운로드 옵션
1456  * @param {String} options.fileName [default: excel.csv] 저장 될 파일 이름
1457  * @param {String} [options.type] [default: 1] Grid 저장 형태 (0이면 데이터 형태, 1이면 표시 방식)
1458  * @param {String} [options.delim] [default: ,] CSV 파일에서 데이터를 구분할 구분자
1459  * @param {String} [options.removeColumns] [default: 없음] 저장 하지 않을 columns index, 여러컬럼인 경우
1460  * @param {String} [options.header] [default: 1] Grid의 숨겨진 Column에 대한 저장 여부(0이면 저장 하지 않
1461  * @param {String} [options.hidden] [default: 0] Grid의 숨겨진 Column에 대한 저장 여부(0이면 저장 하지 않
1462  * @param {String} [options.checkBox] [default: 1] Grid의 Control(Check, Radio, Button) Column
1463  * @param {String} [options.saveList] [default: 없음] hidden에 관계없이 저장할 column id들의 array
1464  * @memberOf com
1465  * @return {file} CSV file
1466  * @author InswaveSystems
1467  * var gridId = "grd_AdvancedExcel";
1468  * var options = {
1469  *     fileName : "downloadCSV.csv" //[default : excel.csv] options.fileName 값이 없을 경우 default값
1470  * };
1471  * com.downloadCSV(grdObj, options);
1472  * //return 예시) 엑셀 파일 다운로드
1473  */
1474 downloadCSV : function(grdObj, options)
1475 {
1476     var options = {
1477         fileName : options.fileName || "excel.csv", // [default: excel.csv] 저장 될 파일 이름
1478         type : options.type || "1", // [default: 1] Grid 저장 형태 (0이면 데이터 형태, 1이면 표시 방식)
1479         delim : options.delim || ",", // [default: ,] CSV 파일에서 데이터를 구분할 구분자
1480         removeColumns : options.removeColumns || "", // [default: 없음] 저장 하지 않을 columns index
1481         header : options.header || "1", // [default: 1] Grid의 숨겨진 Column에 대한 저장 여부(0이면 저
1482         hidden : options.hidden || 0, // [default: 0] Grid의 숨겨진 Column에 대한 저장 여부(0이면 저장
1483         checkBox : options.checkBox || "1", // [default: 1] Grid의 Control(Check, Radio, Bu
1484         // Column히든, 1이면 보여줌)
1485         saveList : options.saveList || "" // [default: 없음] hidden에 관계없이 저장할 column id들의 ar
1486     }
1487     grdObj.saveCSV(options);
1488 },
1489
1490 /**
1491  * 엑셀 업로드 옵션을 설정하고 확장자 별로 다른 함수(uploadCSV || uploadExcel)를 호출한다.
1492  *
1493  * @date 2017.11.30
1494  * @param {Object} grdObj 그리드뷰 아이디
1495  * @param {Array} options JSON형태로 저장된 그리드의 엑셀 업로드 옵션
1496  * @param {String} type 타입(xls, xlsx, csv)을 구분 후, 적합한 API를 사용하여 업로드 한다.

```

```

1497 * @memberOf com
1498 * @author InswaveSystems
1499 * @example
1500 * var gridId = "grd_AdvancedExcel";
1501 * var type = "xlsx";
1502 * var options = {
1503 *   fileName : "gridDataUpload.xlsx" // default값이 존재하지 않으므로 꼭 fileName 값을 넣어야 한다.
1504 * };
1505 * com.gridDataUpload(grdObj, type, options);
1506 * //return 예시) com.js의 다른 함수(uploadCSV, uploadExcel)로 이동하거나 alert 발생
1507 */
1508 gridDataUpload : function(grdObj, type, options)
1509 {
1510   type = type.toLowerCase();
1511
1512   if (type == "csv") {
1513     com.grid.uploadCSV(grdObj, options);
1514   } else if (type == "xls" || type == "xlsx") {
1515     com.grid.uploadExcel(grdObj, options);
1516   } else {
1517     com.popup.alert("지원하지 않는 파일 형식입니다.");
1518   }
1519 },
1520
1521 /**
1522 * 엑셀 xls, xlsx 업로드
1523 *
1524 * @date 2017.11.30
1525 * @param {Object} grdObj GridView Object
1526 * @param {Object} options JSON형태로 저장된 그리드의 엑셀 업로드 옵션
1527 *
1528 * @param {String} [options.type] 1 => 엑셀 파일이 그리드의 보이는 결과로 만들어져있을때 <br>
1529 * @param {Number} [options.sheetNo] excel파일에서 그리드의 데이터가 있는 sheet번호
1530 * @param {Number} [options.startRowIndex] [defalut:0] excel파일에서 그리드의 데이터가 시작되는 행의
1531 * @param {Number} [options.startColumnIndex] [defalut:0] excel파일에서 그리드의 데이터가 시작되는 열의
1532 * @param {Number} [options.endColumnIndex] [defalut: 0] excel파일에서 그리드의 데이터가 끝나는 열의
1533 * (엑셀컬럼수가 그리드컬럼수 보다 작은 경우 그리드 컬럼수를 쓴
1534 * @param {String} [options.headerExist] [defalut:0] excel파일에서 그리드의 데이터에 header가 있
1535 * 1 => header 존재 <br> 0 => 없음)
1536 * @param {String} [options.footerExist] [defalut: 1] excel파일에서 그리드의 데이터에 footer가
1537 * 1 => footer 존재 <br> 0 => 없음 <br> (그리드에 foote
1538 * @param {String} [options.append] [defalut: 0] excel파일에서 가져온 데이터를 그리드에 app
1539 * 1 => 현재 그리드에 데이터를 추가로 넣어줌 <br> 0 => 현재
1540 * @param {String} [options.hidden] [defalut: 0] 읽어들이려는 엑셀파일에 hidden column이
1541 * 0 => 엑셀파일에 hidden 데이터가 없으므로 그리드 hidden
1542 * 1 => 엑셀파일에 hidden 데이터가 있으므로 엑셀 파일로부터
1543 * @param {String} [options.fillHidden] [defalut: 0] Grid에 hiddenColumn에 빈 값을 넣을지
1544 * 1 => hidden Column에 빈 값을 저장하지 않음 <br> 0 =>
1545 * hidden Column에 값을 넣어줌 <br>(hidden이 0인 경우에는
1546 * @param {String} [options.skipSpace] [defalut: 0] 공백무시 여부 <br> 1 => 무시 <br> 0 =>
1547 * @param {String} [options.insertColumns] radio, checkbox와 같은 컬럼을 엑셀에서 받아 오지 않고 <l
1548 * @param {String} [options.popupUrl] 업로드시에 호출할 popup의 url
1549 * @param {String} [options.status] [defalut: R]업로드된 데이터의 초기 상태값, 설정하지 않으면
1550 * @param {String} [options.pwd] 엑셀파일에 암호가 걸려 있는 경우, 비밀번호
1551 * @memberOf com
1552 * @author InswaveSystems
1553 * @example
1554 * var options = {
1555 *   type : "0"
1556 *   ,sheetNo : 0
1557 * };
1558 * com.uploadExcel(grd_basicInfo, options);
1559 */
1560 uploadExcel : function(grdObj, options)
1561 {
1562   var options = {
1563     type : options.type || "0", // String, 1이면 엑셀 파일이 그리드의 보이는 결과로 만들어져있을때
1564     sheetNo : options.sheetNo || 0, // Number, excel파일에서 그리드의 데이터가 있는 sheet번호

```

```

1565     startRowIndex : options.startRowIndex || 1, // Number, [default:0] excel파일에서 그리드:
1566     startColumnIndex : options.startColumnIndex || 0, // Number, [default:0] excel파일에서
1567     endColumnIndex : options.endColumnIndex || 0, // Number, [default: 0] excel파일에서 그
1568     // ( 엑셀컬럼수가 그리드컬럼수 보다 작은 경우 그리드 컬럼수를 설정)
1569     headerExist : options.headerExist || "0", // String, [default:0] excel파일에서 그리드의
1570     // (1이면 header 존재 0이면 없음)
1571     footerExist : options.footerExist || "1", // String, [default: 1] excel파일에서 그리드의
1572     // (1이면 footer 존재 0이면 없음 기본값은 1 그리드에 footer가 없으면 적용되지 않음)
1573     append : options.append || "0", // String, [default: 0] excel파일에서 가져온 데이터를 그리
1574     // (1이면 현재 그리드에 데이터를 추가로 넣어줌 0이면 현재 그리드의 데이터를 삭제하고 넣음)
1575     hidden : options.hidden || "0", // String, [default: 0] 읽어들이려는 엑셀파일에 hidden co
1576     // 엑셀파일에 hidden 데이터가 없으므로 그리드 hidden column에 빈 데이터를 삽입
1577     // 1 : 엑셀파일에 hidden 데이터가 있으므로 엑셀 파일로부터 hidden 데이터를 삽입 )
1578     fillHidden : options.fillHidden || "0", // String, [default: 0] Grid에 hiddenColumn에
1579     // 위한 int형 숫자(1이면 hidden Column에 빈 값을 저장하지 않음, 0이면 hidden
1580     // column이 저장되어있지 않은 Excel File이라 간주하고 hidden Column에 빈
1581     // 값을 넣어줌)(hidden이 0인 경우에는 fillHidden은 영향을 끼치지 않음)
1582     skipSpace : options.skipSpace || "0", // String, [default: 0] 공백무시 여부(1이면 무시 0<
1583     insertColumns : options.insertColumns || "", // Array, radio, checkbox와 같은 컬럼을 엑셀
1584     // 사용자 컬럼 설정 으로 업로드 ( 데이터 구조 : [ { columnIndex:1, columnValue:"1" } ] )
1585     popupUrl : options.popupUrl || "", // String, 업로드시에 호출할 popup의 url
1586     status : options.status || "R", // String, [default: R]업로드된 데이터의 초기 상태값, 설정하
1587     pwd : options.pwd || "" // String, 엑셀파일에 암호가 걸려 있는 경우, 비밀번호
1588 };
1589
1590     grdObj.advancedExcelUpload(options);
1591 },
1592
1593 /**
1594 * 엑셀 CSV 업로드
1595 *
1596 * @date 2017.11.30
1597 * @param {Object} grdObj GridView Object
1598 * @param {Object} options JSON형태로 저장된 그리드의 엑셀 업로드 옵션
1599 * @param {String} [options.type] [default: 1, 0] 데이터 형태 (0이면 실 데이터 형태, 1이면 display 표시 방
1600 * @param {String} [options.header] [default: 1, 0] Grid header 존재 유무 (0이면 header row수를 무시하
1601 * @param {String} [options.delim] [default: ',', ''] CSV 파일에서 데이터를 구분할 구분자
1602 * @param {String} [options.escapeChar] CSV 데이터에서 제거해야 되는 문자셋 ( ex) '\\' )
1603 * @param {Number} [options.startRowIndex] [default: 0] csv파일에서 그리드의 데이터가 시작되는 행의 번호,
1604 * @param {String} [options.append] [default: 0, 1] csv파일에서 가져온 데이터를 그리드에 append시킬지 여부
1605 * @param {Number} [options.hidden] [default: 0, 1] hidden Column에 대한 저장 여부(0이면 저장하지않음, 1<
1606 * @param {String} [options.fillHidden] [default: 0, 1] hidden Column에 빈 값을 넣을지를 결정하기 위한 :
1607 * @param {String} [options.skipSpace] [default: 0, 1] 공백무시 여부(1이면 무시 0이면 포함)
1608 * @param {String} [options.expression] [default: 1, 0] expression 컬럼 데이터를 포함하고 있는지 여부, 1
1609 * @param {String} [options.popupUrl] 업로드시에 호출할 popup의 url
1610 * @memberOf com
1611 * @author InswaveSystems
1612 * @example
1613 * var gridId = "grd_AdvancedExcel";
1614 * var options = {};
1615 * com.uploadCSV( gridId, options);
1616 * //return 메시) 엑셀 파일(.CSV) 업로드
1617 */
1618 uploadCSV : function(grdObj, options)
1619 {
1620     var options = {
1621         type : options.type || "0", // String, [default: 1, 0]데이터 형태 (0이면 실 데이터 형태, 1이면 d
1622         header : options.header || "0", // String, [default: 1, 0]Grid header 존재 유무 (0이면 head
1623         // skip한다.)
1624         delim : options.delim || ",", // String, [default: ',', '']CSV 파일에서 데이터를 구분할 구분자
1625         escapeChar : options.escapeChar || "", // String, CSV 데이터에서 제거해야 되는 문자셋 ( ex) '\\'
1626         startRowIndex : options.startRowIndex || 0, // Number, [default: 0] csv파일에서 그리드의 데0
1627         // 무시된다.
1628         append : options.append || "0", // String, [default: 0, 1]csv파일에서 가져온 데이터를 그리드에
1629         // 데이터를 삭제하고 넣음)
1630         hidden : options.hidden || 1, // Number, [default: 0, 1]hidden Column에 대한 저장 여부(0이면
1631         fillHidden : options.fillHidden || "0", // String, [default: 0, 1]hidden Column에 빈 값을
1632         // 저장하지 않음, 0이면 hidden column이 저장되어있지 않은 csv File이라 간주하고 hidden Column에 빈 값

```

```

1633 // 경우에는 fillhidden은 영향을 끼치지 않음)
1634 skipSpace : options.skipSpace || "0", // String, [defalut: 0, 1]공백무시 여부(1이면 무시 0이면
1635 expression : options.expression || "1", // String, [defalut: 1, 0]expression 컬럼 데이터를
1636 popupUrl : options.popupUrl || "" // String, 업로드시에 호출할 popup의 url
1637 }
1638 grdObj.readCSV(options);
1639 },
1640
1641 /**
1642 * dataList에 row를 추가한다.
1643 *
1644 * @date 2020.06.23
1645 * @memberOf com
1646 * @param {Object} poDlObj dataList or gridView Object
1647 * @param {String} insertRow후에 포커스를 줄 컬럼ID(default 0)
1648 * @return {Number} 추가된 행의 row index
1649 * @author 전기완
1650 * @example
1651 * var newIdx = com.insertRow(dlUserList);
1652 * var newIdx = com.insertRow(grdUserList); // gridView를 이용할 경우만 focus 적용
1653 * var newIdx = com.insertRow(grdUserList, "userNm"); // focus줄 컬럼ID를 안줄경우 첫번째 컬럼으로 fo
1654 */
1655 insertRow : function(poDlObj, psFocusColId)
1656 {
1657     var rowIdx = null;
1658     var voGrid = null;
1659     var voDlObj = poDlObj;
1660     if(com.data.isNotEmpty(voDlObj)){
1661         if(com.data.isNotEmpty(voDlObj)
1662             && typeof voDlObj.getPluginName === "function"
1663             && voDlObj.getPluginName() === "gridView"){
1664             if(com.data.isNotEmpty(voDlObj.getDataListInfo())){
1665                 voDlObj = voDlObj.getDataListInfo().id;
1666                 if(com.data.isNotEmpty(voDlObj)){
1667                     voGrid = poDlObj;
1668                     poDlObj = $p.getComponentById(voDlObj);
1669                 }
1670             }
1671         }
1672         if(typeof poDlObj.getObjectType === "function"
1673             && poDlObj.getObjectType() === "dataList"){
1674             var posIdx = poDlObj.getRowPosition();
1675             rowIdx = 0;
1676             if(com.data.isNotEmpty(posIdx)){
1677                 rowIdx = posIdx + 1;
1678             }
1679             poDlObj.insertRow(rowIdx);
1680             if(com.data.isNotEmpty(voGrid)){
1681                 voGrid.setFocusedCell(rowIdx, com.data.isEmpty(psFocusColId, 0), true);
1682             }
1683         }
1684     }
1685     return rowIdx;
1686 },
1687
1688 /**
1689 * dataList의 row를 삭제한다.
1690 *
1691 * @date 2020.06.23
1692 * @memberOf com
1693 * @param {Object} poDlObj dataList Object
1694 * @param {boolean} pbDelAllSatus 신규행이 아닌 row도 삭제할지 여부
1695 * @return {Number} 삭제된 행의 row index
1696 * @author 전기완
1697 * @example
1698 * var delIdx = com.deleteRow(dlUserList); // 추가된 행만 삭제
1699 * var delIdx = com.deleteRow(dlUserList, true); // 모든 상태의 행을 삭제
1700 */

```

```

1701 deleteRow : function(poDlObj, pbDelAllSatus)
1702 {
1703     var rowIdx = null;
1704     if(com.data.isEmpty(poDlObj)
1705         && typeof poDlObj.getObjectType() === "function"
1706         && poDlObj.getObjectType() === "datalist"){
1707         var rowCnt = poDlObj.getRowCount();
1708         if(rowCnt > 0){
1709             var rowIdx = poDlObj.getRowPosition();
1710             if(com.data.isEmpty(rowIdx) && rowIdx > -1){
1711                 var rowStat = poDlObj.getRowStatus(rowIdx) // 일반:"R", 갱신:"U", 삽입:"C", 삭제:"D"
1712                 if(rowStat === "C"){
1713                     poDlObj.removeRow(rowIdx);
1714                 } else if(rowStat === "D"){
1715                     poDlObj.undeleteRow(rowIdx);
1716                 } else {
1717                     if(pbDelAllSatus === true){
1718                         poDlObj.deleteRow(rowIdx);
1719                     } else {
1720                         com.popup.alert("", "기존 데이터는 삭제할 수 없습니다.");
1721                     }
1722                 }
1723             } else {
1724                 com.popup.alert("", "삭제할 데이터를 선택해 주세요");
1725             }
1726         } else {
1727             com.popup.alert("", "삭제할 데이터가 없습니다.");
1728         }
1729     }
1730     return rowIdx;
1731 },
1732
1733 setGridCount : function(dl, current, total) {
1734     current.setValue(dl.getTotalRow());
1735     if(dl.getTotalRow() == 0) {
1736         total.setValue(0);
1737     } else {
1738         total.setValue(dl.getCellData(dl.getTotalRow() - 1, 'TOTAL'));
1739     }
1740
1741     /*
1742     if(!WebSquare.util.getComponentById(dl.org_id + '_currentCount')) {
1743         scwin.current = WebSquare.util.dynamicCreate(dl.org_id + '_currentCount', 'span', {label
1744     } else {
1745         scwin.current.setValue(dl.getTotalRow());
1746     }
1747
1748     if(!WebSquare.util.getComponentById(dl.org_id + '_mid')) {
1749         WebSquare.util.dynamicCreate(dl.org_id + '_mid', 'span', {label : '/'}, comp);
1750     }
1751
1752     if(!WebSquare.util.getComponentById(dl.org_id + '_totalCount')) {
1753         scwin.total = WebSquare.util.dynamicCreate(dl.org_id + '_totalCount', 'span', {label : '
1754     } else {
1755         scwin.total.setValue(dl.getCellData(dl.getTotalRow() - 1, 'TOTAL'));
1756     }
1757     */
1758 },
1759
1760 isDataChanged : function(dl) {
1761     var flag = false;
1762     for(var i = 0; i < dl.getTotalRow(); i++){
1763         if(com.grid.READ !== dl.getRowStatus(i)) {
1764             flag = true;
1765             continue;
1766         }
1767     }
1768     return flag;

```

```

1769     }
1770 };
1771
1772 //-----
1773 // Code
1774 //-----
1775 com.code = {
1776
1777     getCodeData : function(params)
1778     {
1779         var codeDatas = {};
1780         var codeReqId = [];
1781         params = Array.isArray(params) ? params : [params];
1782         for(var i = 0; i < params.length; i++)
1783         {
1784             var id = params[i].id;
1785             var codeData = [];
1786             // 세션스토리지에 존재하면 그냥 가져온다.
1787             if(sessionStorage.getItem(id) != null)
1788             {
1789                 codeData = JSON.parse(sessionStorage.getItem(id));
1790             } else {
1791                 codeReqId.push(id);
1792             }
1793             codeDatas[id] = codeData;
1794         }
1795
1796         if(codeReqId.length > 0) {
1797             $.ajax({
1798                 url : '/code/selectCodeListByIds.do',
1799                 type : 'post',
1800                 async : false,
1801                 contentType : 'application/json; charset=utf-8',
1802                 data : JSON.stringify({"cmnCd":codeReqId}),
1803                 dataType: 'json',
1804                 success : function(datas)
1805                 {
1806                     var codeObj = datas.codeList;
1807                     var cd = '';
1808                     for(var i = 0 ; i < codeObj.length ; i++)
1809                     {
1810                         cd = codeObj[i].cmnCd;
1811                         if(Object.keys(codeDatas).indexOf(cd) < 0) {
1812                             codeDatas[cd] = [];
1813                         }
1814                         codeDatas[cd].push(codeObj[i]);
1815                     }
1816                 },
1817                 error: function(xhr, textStatus, errorThrown)
1818                 {
1819                     console.log(xhr.responseText);
1820                     //com.error(xhr.responseText);
1821                 }
1822             });
1823
1824             var keys = [];
1825             keys = Object.keys(codeDatas);
1826             for(var i = 0; i < keys.length; i++) {
1827                 sessionStorage.setItem(keys[i], JSON.stringify(codeDatas[keys[i]]));
1828             }
1829         }
1830
1831
1832         // 공통코드용 데이터 리스트
1833         var vsStr = '<w2:dataList baseNode="list" repeatNode="map" id="$CD_DS_ID$" saveRemovedData="
1834             + ' <w2:columnInfo>'
1835             + ' <w2:column id="cmnCd" name="공통코드" dataType="text"></w2:column>'
1836             + ' <w2:column id="cdval" name="코드값" dataType="text"></w2:column>'

```

```

1837 + ' <w2:column id="eftvStartdt" name="유효시작일자" dataType="text"></w2:column>'
1838 + ' <w2:column id="eftvEnddt" name="유효종료일자" dataType="text"></w2:column>'
1839 + ' <w2:column id="cdvalnm" name="코드값명" dataType="text"></w2:column>'
1840 + ' <w2:column id="cdvalEngnm" name="코드값영문명" dataType="text"></w2:column>'
1841 + ' <w2:column id="cdvalAbbrnm" name="코드값약어명" dataType="text"></w2:column>'
1842 + ' <w2:column id="cdvalSeqc" name="코드순서" dataType="text"></w2:column>'
1843 + ' <w2:column id="milTypeCd" name="군유형코드" dataType="text"></w2:column>'
1844 + ' <w2:column id="mdcd" name="군구분코드" dataType="text"></w2:column>'
1845 + ' <w2:column id="sc1pstdvscd" name="신분구분코드" dataType="text"></w2:column>'
1846 + ' <w2:column id="armyUseYn" name="육군사용여부" dataType="text"></w2:column>'
1847 + ' <w2:column id="navyUseYn" name="해군사용여부" dataType="text"></w2:column>'
1848 + ' <w2:column id="afUseYn" name="공군사용여부" dataType="text"></w2:column>'
1849 + ' <w2:column id="attr1Val" name="속성1값" dataType="text"></w2:column>'
1850 + ' <w2:column id="attr2Val" name="속성2값" dataType="text"></w2:column>'
1851 + ' <w2:column id="attr3Val" name="속성3값" dataType="text"></w2:column>'
1852 + ' <w2:column id="attr4Val" name="속성4값" dataType="text"></w2:column>'
1853 + ' <w2:column id="attr5Val" name="속성5값" dataType="text"></w2:column>'
1854 + ' <w2:column id="attr6Val" name="속성6값" dataType="text"></w2:column>'
1855 + ' <w2:column id="attr7Val" name="속성7값" dataType="text"></w2:column>'
1856 + ' <w2:column id="attr8Val" name="속성8값" dataType="text"></w2:column>'
1857 + ' <w2:column id="attr9Val" name="속성9값" dataType="text"></w2:column>'
1858 + ' <w2:column id="attr10Val" name="속성10값" dataType="text"></w2:column>'
1859 + ' <w2:column id="attr11Val" name="속성11값" dataType="text"></w2:column>'
1860 + ' <w2:column id="attr12Val" name="속성12값" dataType="text"></w2:column>'
1861 + ' <w2:column id="attr13Val" name="속성13값" dataType="text"></w2:column>'
1862 + ' <w2:column id="attr14Val" name="속성14값" dataType="text"></w2:column>'
1863 + ' <w2:column id="attr15Val" name="속성15값" dataType="text"></w2:column>'
1864 + ' <w2:column id="attr16Val" name="속성16값" dataType="text"></w2:column>'
1865 + ' <w2:column id="attr17Val" name="속성17값" dataType="text"></w2:column>'
1866 + ' <w2:column id="attr18Val" name="속성18값" dataType="text"></w2:column>'
1867 + ' <w2:column id="attr19Val" name="속성19값" dataType="text"></w2:column>'
1868 + ' <w2:column id="attr20Val" name="속성20값" dataType="text"></w2:column>'
1869 + ' <w2:column id="mpngObjctNm" name="매핑대상명" dataType="text"></w2:column>'
1870 + ' <w2:column id="mpngCdval" name="매핑코드값" dataType="text"></w2:column>'
1871 + ' <w2:column id="mpngCdvalnm" name="매핑코드값명" dataType="text"></w2:column>'
1872 + ' <w2:column id="rmrk" name="비고" dataType="text"></w2:column>'
1873 + ' <w2:column id="hgrnkCmnCd" name="상위공통코드" dataType="text"></w2:column>'
1874 + ' <w2:column id="hgrnkCdval" name="상위코드값" dataType="text"></w2:column>'
1875 + ' <w2:column id="useYn" name="사용여부" dataType="text"></w2:column>'
1876 + ' </w2:columnInfo>'
1877 + ' </w2:dataList>';
1878 var codeIds = Object.keys(codeDatas);
1879 var vsDlId;
1880 var voDlObj;
1881 for(var i = 0; i < codeIds.length; i++){
1882     vsDlId = "dl_" + codeIds[i];
1883     voDlObj = $p.getComponentById(vsDlId);
1884     if(com.data.isEmpty(voDlObj)){
1885         $p.data.remove(vsDlId);
1886     }
1887     $p.data.create(vsStr.replace("$CD_DS_ID$", vsDlId));
1888     voDlObj = $p.getComponentById(vsDlId);
1889     voDlObj.setJSON(codeDatas[codeIds[i]]);
1890     codeDatas[codeIds[i]] = voDlObj;
1891 }
1892
1893 return codeDatas;
1894 },
1895
1896 setComboBox : function(elId, datas)
1897 {
1898     //$('##cmb' + elId).empty();
1899
1900     var s = $("<select id='cmb" + elId + "'/>");
1901
1902     if(datas.constructor.name == 'String') datas = JSON.parse(datas);
1903     for(var i = 0; i < datas.length; i++)
1904     {

```

```

1905     var label1 = datas[i].label;
1906     var value1 = datas[i].value;
1907     $("", {value:value1, text:label1}).appendTo(s);
1908 }
1909 s.appendTo($("#" + elId));
1910
1911 if(elId.indexOf(".") > -1){
1912     var vsCompId = elId.split(".");
1913     compObj = $p.getComponentById(vsCompId[0]);
1914     if(com.data.isEmpty(compObj)){
1915         return;
1916     }
1917     if(compObj.getPluginName() === "gridView"){
1918         console.log( "data:dl_" + datas.org_id);
1919         compObj.setColumnNodeSet(vsCompId[1], "data:" + datas.org_id, "cdvalnm", "cdval");
1920     }
1921 } else {
1922     var compObj = $p.getComponentById(elId);
1923     if(typeof compObj !== undefined){
1924         var pluginName = compObj.getPluginName();
1925         if(["selectbox", "checkbox", "radio"].indexOf(compObj.getPluginName()) > -1){
1926             if(WebSquare.util.isArray(datas) && datas.length > 0){
1927                 compObj.removeAll(true);
1928                 var defaultOptCount = compObj.getItemCount();
1929                 for(var i = 0; i < datas.length; i++){
1930                     compObj.addItem(datas[i]["cdval"], datas[i]["cdvalnm"], i + defaultOptCo
1931                 }
1932             } else {
1933                 compObj.setNodeSet("data:" + datas.org_id, "cdvalnm", "cdval");
1934             }
1935         }
1936     }
1937 }
1938 }
1939 }
1940 }
1941 };
1942
1943 //-----
1944 // Data
1945 //-----
1946 com.data = {
1947     splitChar : " ",
1948     // 일반전화번호
1949     regExpTel : "/^(02|3[1-3]|4[1-4]|5[1-5]|6[1-4]))(\d{3,4})(\d{4})$/",
1950     // 휴대폰번호
1951     regExpHp : "/^(01([0|1|6|7|8|9]))([0-9]{3,4})([0-9]{4})$/",
1952     // 일반 및 휴대폰번호
1953     regExpTelHpNo : "/^(02|3[1-3]|4[1-4]|5[1-5]|6[1-4]))(^01([0|1|6|7|8|9]))(\d{3,4})(\d{4})$/",
1954     // 이메일
1955     regExpEmail : "/^[0-9a-zA-Z]([-_\.]?[0-9a-zA-Z])*@[0-9a-zA-Z]([-_\.]?[0-9a-zA-Z])*\.[0-9a-zA-Z]{2,3}$/i",
1956     // 이메일도메인
1957     regExpDomain : "/^[0-9a-zA-Z]([-_\.]?[0-9a-zA-Z])*\. [0-9a-zA-Z]{2,3}$/i",
1958     // 날짜8자리
1959     regExpDate8 : "/^(19|20)\d{2}(\d{1-9}|10|12)(\d{1-9}|12)[0-9]{3}(\d{1-9})$/",
1960     // IP
1961     regExpIp : "/^((0-9)|[1-9][0-9]|10[0-9]{2}|20[0-4][0-9]|25[0-5])\.){3}((0-9)|[1-9][0-9]|10[0-9]{2}|
1962
1963 /**
1964 * Object가 null, undefined, "", {}, []일 경우 true 이외의 경우는 false를 return한다.
1965 *
1966 * @date 2020.06.05
1967 * @memberOf com
1968 * @param {Object} poObj empty 체크할 Object
1969 * @return {boolean} true or false
1970 * @author 전기완
1971 * @example
1972 *

```

```

1973  */
1974  isEmpty : function(obj)
1975  {
1976      return (obj == null || obj == undefined) ? true : false;
1977  },
1978  /**
1979   * Object가 null, undefined, "", {}, []일 경우 false 이외의 경우는 true를 return한다.
1980   *
1981   * @date 2020.06.05
1982   * @memberOf com
1983   * @param {Object} poObj empty 체크할 Object
1984   * @return {boolean} true or false
1985   * @author 전기완
1986   * @example
1987   *
1988   */
1989  isEmpty : function(obj)
1990  {
1991      return !com.data.isEmpty(obj);
1992  },
1993  /**
1994   * Object가 null, undefined, "", {}, []일 경우 대체Object를 return한다..
1995   *
1996   * @date 2020.06.05
1997   * @memberOf com
1998   * @param {Object} [poObj] empty 체크할 Object
1999   * @param {Object} [poConvObj] 대체할 Object
2000   * @return {Object} empty일 경우 poConvObj를 아닐경우 poObj를 return한다.
2001   * @author 전기완
2002   * @example
2003   * com.data.isEmpty("", {"AAA":"aaa"});
2004   */
2005  isEmpty : function(poObj, poConvObj)
2006  {
2007      return com.data.isEmpty(poObj) ? poObj : poConvObj;
2008  },
2009
2010  /**
2011   * JSON 객체를 String 타입으로 반환한다.
2012   *
2013   * @date 2020.06.05
2014   * @memberOf com
2015   * @param {Object} object String으로 변환할 JSON 객체
2016   * @return {String} String으로 변환된 객체
2017   * @author 전기완
2018   * @example
2019   * com.data.strSerialize({"AAA":"aaa"});
2020   */
2021  strSerialize : function(object)
2022  {
2023      if (typeof object == 'string') {
2024          return object;
2025      } else if (com.data.isJSON(object)) {
2026          return JSON.stringify(object);
2027      } else {
2028          return object;
2029      }
2030  },
2031
2032  /**
2033   * JSON Object로 변환해서 반환한다.
2034   *
2035   * @date 2020.06.15
2036   * @memberOf com
2037   * @param {String} str JSON 문자열
2038   * @return {Object} JSON 객체 or null
2039   * @author 전기완
2040   * @example

```

```

2041 * var json = '{"tbx_sPrjNm":"1","tbx_sPrtLv":"2","tbx_sReqLv":"3"}';
2042 */
2043 getJSON : function(str)
2044 {
2045     try {
2046         return JSON.parse(str);
2047     } catch (e) {
2048         return str;
2049     }
2050 },
2051
2052 /**
2053  * JSON Object인지 여부를 검사한다.
2054  *
2055  * @date 2020.06.05
2056  * @memberOf com
2057  * @param {Object} jsonObj JSON Object가 맞는지 검사할 JSON Object
2058  * @return {Boolean} true or false
2059  * @author 전기완
2060  * @example
2061  * com.isJSON(""); ==> false
2062  * com.isJSON( {"tbx_sPrjNm": "1", "tbx_sPrtLv": "2", "tbx_sReqLv": "3"} ); ==> true
2063  */
2064 isJSON : function(jsonObj)
2065 {
2066     if (typeof jsonObj !== 'object')
2067         return false;
2068     try {
2069         JSON.stringify(jsonObj);
2070         return true;
2071     } catch (e) {
2072         return false;
2073     }
2074 },
2075
2076 /**
2077  * XML Document 객체인지 여부를 검사한다.
2078  *
2079  * @date 2014.12.09
2080  * @memberOf com
2081  * @param {Object} data XML Document 객체인지 여부를 검사한다.
2082  * @author InswaveSystems
2083  * @return {Boolean} true or false
2084  */
2085 isXmlDoc : function(data)
2086 {
2087     if (typeof data !== 'object')
2088         return false;
2089     if ((typeof data.documentElement !== 'undefined' && data.nodeType == 9)
2090         || (typeof data.documentElement == 'undefined' && data.nodeType == 1)) {
2091         return true;
2092     }
2093     return false;
2094 },
2095
2096 eqFilter : function(my_object, my_criteria)
2097 {
2098     if(my_object == undefined) return {};
2099
2100     var tmpData = JSON.parse(JSON.stringify(my_object));
2101     return tmpData.filter(function(obj)
2102     {
2103         return Object.keys(my_criteria).every(function(c)
2104         {
2105             return obj[c] == my_criteria[c];
2106         });
2107     });
2108 },

```

```

2109
2110 neFilter : function(my_object, my_criteria)
2111 {
2112     if(my_object == undefined) return {};
2113
2114     var tmpData = JSON.parse(JSON.stringify(my_object));
2115     return tmpData.filter(function(obj)
2116     {
2117         return Object.keys(my_criteria).every(function(c)
2118         {
2119             return obj[c] != my_criteria[c];
2120         });
2121     });
2122 },
2123
2124 /**
2125  * 날짜유효성체크(yyyyMMdd 8자리) 결과를 return한다.
2126  *
2127  * @date 2020.06.05
2128  * @memberOf com
2129  * @param {String} param 체크할 문자열
2130  * @return {boolean} 유효한 날짜 여부 return
2131  * @author 전기완
2132  * @example
2133  *
2134  * com["regExpDateFunc"]("20191018");
2135  *
2136  */
2137 regExpDateFunc : function(param)
2138 {
2139     if(com.data.isEmpty(param)){
2140         return true;
2141     }
2142
2143     param = String(param);
2144
2145     if(!com.data["regExpDate8"].test(param)){
2146         return false;
2147     }
2148
2149     try {
2150         var year = Number(param.substr(0, 4));
2151         var month = Number(param.substr(4, 2));
2152         var day = Number(param.substr(6, 2));
2153
2154         if(month < 1 || month > 12){
2155             return false;
2156         }
2157         var maxDaysInMonth = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
2158         var maxDay = maxDaysInMonth[month - 1];
2159
2160         // 윤년체크(2월이면서 (4년마다이면서 100년은 제외 또는 400년마다 한번)
2161         if( month == 2 && ( (year % 4 === 0 && year % 100 != 0) || year % 400 === 0) ){
2162             maxDay = 29;
2163         }
2164
2165         if(day <= 0 || day > maxDay){
2166             return false;
2167         }
2168     } catch (e) {
2169         return false;
2170     }
2171     return true;
2172 },
2173
2174 /**
2175  * dataCollection의 유효성 체크를 수행한다.
2176  *

```

```

2177 * @date 2020.06.05
2178 * @memberOf com
2179 * @param {JSON} poOpts 체크를 수행할 dataCollection의 체크목록
2180 * @return {void}
2181 * @author 전기완
2182 * @example
2183 *
2184 * var voOpts = {};
2185 *   voOpts["dm_dataMap"] = [
2186 *     {"colId" : "name", "compId" : "ibx_name", "mandatory" : true, "length" : "6-15"},
2187 *     {"colId" : "tel1", "compId" : "sbx_telNo1", "mandatory" : true},
2188 *     {"colId" : "tel2", "compId" : "ibx_telNo2", "mandatory" : true, "length" : "3-4", "type" : "NUM"},
2189 *     {"colId" : "tel3", "compId" : "ibx_telNo3", "mandatory" : true, "fixL" : "4", "type" : "NUM"},
2190 *     {"colId" : "hp2", "compId" : "ibx_mobileNo2", "length" : "3-4", "type" : "NUM"},
2191 *     {"colId" : "hp3", "compId" : "ibx_mobileNo3", "fixL" : "4", "type" : "NUM"},
2192 *     {"colId" : "email1", "compId" : "ibx_email1", "maxL" : 20, "type" : "exHAN"},
2193 *     {"colId" : "email2", "compId" : "ibx_email2", "maxL" : 30, "domain":true, "type" : "exHAN"},
2194 *     {"colId" : "date1", "compId" : "ica_startDate", "date8" : true, "maxDate" : ica_endDate},
2195 *     {"colId" : "date2", "compId" : "ica_endDate", "date8" : true, "minDate" : ica_startDate},
2196 *     {"colId" : "date3", "compId" : "ica_joinDate", "date8" : true},
2197 *     {"colId" : "chk", "compId" : "chk_sample", "mandatory" : true, "minCnt" : 2, "maxCnt" : 10},
2198 *     {"colId" : "rdo", "compId" : "rdo_sample", "mandatory" : true}, // 라디오
2199 *     {"colId" : "sbx", "compId" : "sbx_sample", "mandatory" : true}, // 콤보박스
2200 *     {"colId" : "edt", "compId" : "edt_sample", "mandatory" : true} // 에디터
2201 *   ];
2202 *   voOpts["dl_sample"] = [
2203 *     {"colId" : "col1", "compId" : "grd_list", "mandatory" : true, "length" : "6-15"},
2204 *     {"colId" : "col2", "compId" : "grd_list", "tel" : true},
2205 *     {"colId" : "col3", "compId" : "grd_list", "hp" : true},
2206 *     {"colId" : "col4", "compId" : "grd_list", "email" : true},
2207 *     {"colId" : "col5", "compId" : "grd_list", "date8" : true}
2208 *   ];
2209 *   com.checkValid(voOpts);
2210 *
2211 * 체크오류시 false를 return 한 후 오류 메시지 표시 후 해당 컴포넌트에 포커스를 이동한다.
2212 * 성공시 true를 return 한다.
2213 *
2214 * mandatory : true일 경우 필수입력 체크
2215 * minL : "n" 문자 길이가 최소 n byte이상인지 여부 체크
2216 * maxL : "n" 최대 길이가 최대 n byte이하인지 여부 체크
2217 * fixL : "n" 문자 길이가 n byte인지 여부 체크
2218 * length : "n-n" n byte이상 n byte이하 여부 체크
2219 * type : "NUM" or "exHAN" 오류메시지 표시에 숫자 or 한글제외의 메시지로 표현
2220 * domain : E-mail의 도메인 여부 체크(XXX.XX or XXX.XXX)
2221 * date8 : true일 경우 8자리 날짜인지 여부 체크
2222 * minDate : component Object를 추가시 입력가능 최소 날짜를 체크
2223 * maxDate : component Object를 추가시 입력가능 최대 날짜를 체크
2224 * minCnt : n 체크박스일 경우 체크 갯수가 n개 이상인지 여부 체크
2225 * maxCnt : n 체크박스일 경우 체크 갯수가 n개 이하인지 여부 체크
2226 *
2227 * tel : true일 경우 일반전화번호 여부 체크(##-###-####, ##-####-####, ###-###-####, ###-####-####)
2228 * hp : true일 경우 휴대전화번호 여부 체크(###-###-####, ###-####-####)
2229 * email : true일 경우 이메일 여부 체크
2230 */
2231 checkValid : function(poOpts)
2232 {
2233   if(com.data.isEmpty(poOpts)){
2234     return true;
2235   }
2236   var voDCIdList = Object.keys(poOpts);
2237   var voDcObj;
2238   var voResult;
2239   var vnInvalidRow;
2240   var vnRowCount;
2241   for(var i = 0; i < voDCIdList.length; i++){
2242     voDcObj = $p.getComponentById(voDCIdList[i]);
2243     if(com.data.isNotEmpty(voDcObj)
2244       && typeof voDcObj.getObjectType === "function"){

```

```
2245 if(voDcObj.getObjectType() === "dataMap"){
2246     voResult = com.data._checkValidRow(voDcObj.getJSON(), poOpts[voDCIdList[i]]);
2247     if(com.data.isNotEmpty(voResult)){
2248         break;
2249     }
2250 } else if(voDcObj.getObjectType() === "dataList"){
2251     vnRowCnt = voDcObj.getRowCount();
2252     var chkCol = com.data.isEmpty(voDcObj.getCol...
```

```

2313     }
2314     com.popup.alert("invalidFail", vsAlertMsg, {"compId":vsCompId, "row" : vnInvalidRow, "co
2315     return false;
2316 } else {
2317     return true;
2318 }
2319 },
2320
2321 /**
2322  * com.checkValid함수내에서
2323  * dataCollection의 row단위로 유효성 체크를 수행한다.
2324  *
2325  * @date 2020.06.05
2326  * @memberOf com
2327  * @param {JSON} poRowJSON 체크를 수행할 row data
2328  * @param {JSON} poInfo 체크를 수행할 컬럼의 정보
2329  * @param {Number} arguments[2] dataList일 경우 수행중인 row index
2330  * @return {JSON} 실패시 실패정보를 return
2331  * @author 전기완
2332  * @example
2333  *
2334  */
2335 _checkValidRow : function(poRowJSON, poInfo)
2336 {
2337     var vsColId;        // 컬럼ID
2338     var vsColData;      // 컬럼값
2339     var vsInvalidType;  // 실패유형
2340     var vsInvalidVal;   // 실패유형 체크에 사용된 값
2341     var vsCompId;       // 컴포넌트ID
2342     var vsCompObj;
2343     var vsCompareData;  // 비교대상
2344     var vnRow = arguments[2]; // 그리드 현재 Row
2345     for(var i = 0; i < poInfo.length; i++){
2346         vsColId = poInfo[i]["colId"];
2347         vsColData = poRowJSON[vsColId];
2348         vsCompId = poInfo[i]["compId"];
2349         vsType = poInfo[i]["type"];
2350         vsCompObj = $p.getComponentById(vsCompId);
2351
2352         // 필수
2353         if(poInfo[i]["mandatory"] === true && com.data.isEmpty(vsColData)){
2354             //return {"colId" : vsColId, "invalidType" : "mandatory", "compId" : poInfo[i]["comp
2355             vsInvalidType = "mandatory";
2356         }
2357         // 최소길이
2358         } else if(com.data.isNotEmpty(poInfo[i]["minL"])
2359             && com.data.isNotEmpty(vsColData)
2360             && $p.getStringByteSize(String(vsColData)) < Number(poInfo[i]["minL"])){
2361             vsInvalidType = "minL";
2362             vsInvalidVal = poInfo[i]["minL"];
2363         }
2364         // 최대길이
2365         } else if(com.data.isNotEmpty(poInfo[i]["maxL"])
2366             && com.data.isNotEmpty(vsColData)
2367             && $p.getStringByteSize(String(vsColData)) > Number(poInfo[i]["maxL"])){
2368             vsInvalidType = "maxL";
2369             vsInvalidVal = poInfo[i]["maxL"];
2370         }
2371         // 최소최대길이
2372         } else if(com.data.isNotEmpty(poInfo[i]["length"])
2373             && com.data.isNotEmpty(vsColData)
2374             && ( $p.getStringByteSize(String(vsColData)) < Number(poInfo[i]["length"].split(
2375                 || $p.getStringByteSize(String(vsColData)) > Number(poInfo[i]["length"].
2376             vsInvalidType = "length";
2377             vsInvalidVal = poInfo[i]["length"];
2378         }
2379         // 고정길이
2380         } else if(com.data.isNotEmpty(poInfo[i]["fixL"])
2381             && com.data.isNotEmpty(vsColData)
2382             && $p.getStringByteSize(String(vsColData)) != Number(poInfo[i]["fixL"])){
2383             vsInvalidType = "fixL";
2384             vsInvalidVal = poInfo[i]["fixL"];

```

```

2381 // 이메일 도메인
2382 } else if(poInfo[i]["domain"] === true
2383     && com.data.isNotEmpty(vsColData)
2384     && !com["regExpDomain"].test(String(vsColData))){
2385     vsInvalidType = "domain";
2386     vsInvalidVal = poInfo[i]["domain"];
2387 // 날자
2388 } else if(poInfo[i]["date8"] === true
2389     && com.data.isNotEmpty(vsColData)
2390     && !com["regExpDateFunc"](String(vsColData))){
2391     vsInvalidType = "date8";
2392     vsInvalidVal = poInfo[i]["date8"];
2393 // 최대날자
2394 } else if(com.data.isNotEmpty(vsCompObj) && com.data.isNotEmpty(vsColData)
2395     && com.data.isNotEmpty(poInfo[i]["maxDate"]) && com.data.isNotEmpty(poInfo[i]["m
2396     && vsColData > poInfo[i]["maxDate"].getValue()){
2397     vsInvalidType = "maxDate";
2398     vsInvalidVal = poInfo[i]["maxDate"].getValue();
2399 // 최소날자
2400 } else if(com.data.isNotEmpty(vsCompObj) && com.data.isNotEmpty(vsColData)
2401     && com.data.isNotEmpty(poInfo[i]["minDate"]) && com.data.isNotEmpty(poInfo[i]["m
2402     && vsColData < poInfo[i]["minDate"].getValue()){
2403     vsInvalidType = "minDate";
2404     vsInvalidVal = poInfo[i]["minDate"].getValue();
2405 // 최소선택갯수
2406 } else if(com.data.isNotEmpty(poInfo[i]["minCnt"])
2407     && com.data.isNotEmpty(vsColData)
2408     && vsColData.split(com["splitChar"]).length < poInfo[i]["minCnt"]){
2409     vsInvalidType = "minCnt";
2410     vsInvalidVal = poInfo[i]["minCnt"];
2411 // 최대선택갯수
2412 } else if(com.data.isNotEmpty(poInfo[i]["maxCnt"])
2413     && com.data.isNotEmpty(vsColData)
2414     && vsColData.split(com["splitChar"]).length > poInfo[i]["maxCnt"]){
2415     vsInvalidType = "maxCnt";
2416     vsInvalidVal = poInfo[i]["maxCnt"];
2417 // 일반전화번호
2418 } else if(poInfo[i]["tel"] === true && com.data.isNotEmpty(vsColData)
2419     && !com["regExpTel"].test(vsColData)){
2420     vsInvalidType = "tel";
2421 // 휴대전화번호
2422 } else if(poInfo[i]["hp"] === true && com.data.isNotEmpty(vsColData)
2423     && !com["regExpHp"].test(vsColData)){
2424     vsInvalidType = "hp";
2425 // 일반 및 휴대전화번호
2426 } else if(poInfo[i]["regExpTelHpNo"] === true && com.data.isNotEmpty(vsColData)
2427     && !com["regExpTelHpNo"].test(vsColData.replace(/-/gi, ""))){
2428     vsInvalidType = "regExpTelHpNo";
2429 // 이메일
2430 } else if(poInfo[i]["email"] === true && com.data.isNotEmpty(vsColData)
2431     && !com["regExpEmail"].test(vsColData)){
2432     vsInvalidType = "email";
2433 } else if(poInfo[i]["ip"] === true && com.data.isNotEmpty(vsColData)
2434     && !com["regExpIp"].test(vsColData)){
2435     vsInvalidType = "ip";
2436 }
2437 if(com.data.isNotEmpty(vsInvalidType)){
2438     return {"colId" : vsColId, "invalidType" : vsInvalidType, "compId" : vsCompId, "inva
2439 }
2440 }
2441 return null;
2442 },
2443
2444 /**
2445 * com.checkValid 함수 실행중 실패시 띄운 alert의 callback
2446 *
2447 * @date 2020.06.05
2448 * @memberOf com

```

```

2449 * @param {String} psPopupId 팝업ID
2450 * @param {Object} poRtnVal 팝업 return data
2451 * @param {JSON} poEtcInfo 팝업호출시 전달한 data
2452 * @return {void}
2453 * @author 전기완
2454 * @example
2455 */
2456 invalidCallback : function(psPopupId, poRtnVal, poEtcInfo){
2457
2458     if(psPopupId === "invalidFail"){
2459         poEtcInfo = JSON.parse(poEtcInfo);
2460         var vsCompId = poEtcInfo["compId"];
2461         var voCompObj = $p.getComponentById(vsCompId);
2462
2463         if(com.data.isEmpty(voCompObj)){
2464             return;
2465         }
2466         if(voCompObj.getPluginName() === "gridView"){
2467             var voArgs = new Array();
2468             voArgs.push(voCompObj);
2469             voArgs.push(poEtcInfo["row"]);
2470             voArgs.push(poEtcInfo["colId"]);
2471             $p.setTimeout(
2472                 function(){
2473                     arguments[0].setFocusedCell(arguments[1], arguments[2], true);
2474                 },
2475                 {"args" : voArgs,
2476                  "delay" : 300,
2477                  "key" : "validCheckFail"});
2478         } else {
2479             voCompObj.focus();
2480         }
2481     }
2482 },
2483
2484 /**
2485 * com.checkValid 함수 실행중
2486 * 길이체크 실패시 띄울 alert의 메시지를 조합
2487 *
2488 * @date 2020.06.05
2489 * @memberOf com
2490 * @param {String} psType 문자유형
2491 * @param {String} psInvalidType 체크유형
2492 * @param {String} psInvalidVal 유형체크값
2493 * @return {String} 조합된 문자열
2494 * @author 전기완
2495 * @example
2496 */
2497
2498 getLengthMsg : function(psType, psInvalidType, psInvalidVal)
2499 {
2500     var vsTypeStr = "";
2501     var vsMsg = "";
2502     var vsHanMsg = "";
2503     if(vsType === "NUM"){
2504         vsTypeStr = "숫자 ";
2505     } else if(vsType === "exHAN"){
2506         vsTypeStr = "한글제외 ";
2507     }
2508
2509     // 최소길이
2510     if(psInvalidType === "minL"){
2511         var vnHanSize = Math.floor(psInvalidVal/3);
2512         vsMsg = vsTypeStr + psInvalidVal + "글자이상";
2513         vsHanMsg = "(한글 " + vnHanSize + "글자)";
2514     } // 최대길이
2515     } else if(psInvalidType === "maxL"){
2516         var vnHanSize = Math.floor(psInvalidVal/3);

```

```

2517         vsMsg = vsTypeStr + psInvalidVal + "글자이하";
2518         vsHanMsg = "(한글 " + vnHanSize + "글자)";
2519         // 범위길이
2520     } else if(psInvalidType === "length"){
2521         var vnMinSize = psInvalidVal.split("-")[0];
2522         var vnMaxSize = psInvalidVal.split("-")[1];
2523         var vnMinHanSize = Math.floor(vnMinSize/3);
2524         var vnMaxHanSize = Math.floor(vnMaxSize/3);
2525         vsMsg = vsTypeStr + vnMinSize + "글자이상 " + vnMaxSize + "글자이하"
2526         vsHanMsg = "(한글 " + vnMinHanSize + "~" + vnMaxHanSize + "글자)";
2527         // 고정길이
2528     } else if(psInvalidType === "fixL"){
2529         var vnHanSize = Math.floor(psInvalidVal/3);
2530         vsMsg = vsTypeStr + psInvalidVal + "글자";
2531         vsHanMsg = "(한글 " + vnHanSize + "글자)";
2532     }
2533     return "을(를) " + (com.data.isEmpty(vsTypeStr) ? (vsMsg + vsHanMsg) : vsMsg);
2534 },
2535
2536 /**
2537  * 그룹안에 포함된 컴포넌트의 입력 값에 대한 유효성을 검사한다.
2538  * 컴포넌트 속성 유효성 검사를 수행하고, valInfoArr 유효성 검사 옵션에 대해서 유효성 검사를 수행한다.
2539  * valInfoArr 유효성 검사 옵션 파라미터를 전달하지 않은 경우 컴포넌트 속성(mandatory, allowChar, ignoreChar,
2540  *
2541  * @date 2018.01.19
2542  * @memberOf com
2543  * @param {Object} grpObj 그룹 컴포넌트 객체
2544  * @param {Object[]} options 유효성 검사 옵션 <br/>
2545  * @param {String} options[id] : 유효성 검사 대상 DataCollection 컬럼 아이디 <br/>
2546  * @param {Boolean} options[id].mandatory : 필수 입력 값 여부 <br/>
2547  * @param {Number} options[id].minLength : 최소 입력 자리수 <br/>
2548  * @param {requestCallback} options[id].valFunc : 사용자 유효성 검사 함수 <br/>
2549  * @param {String} tacId 그룹이 포함된 TabControl 컴포넌트 아이디
2550  * @param {String} tabId 그룹이 포함된 TabControl 컴포넌트의 Tab 아이디
2551  * @returns {Boolean} 유효성 검사 결과
2552  * @since 2015.08.05
2553  * @example
2554  *
2555  * if (com.validateGroup(grp_LoginInfo)) {
2556  *     if (confirm("변경된 데이터를 저장하시겠습니까?")) {
2557  *         com.executeSubmission("WS0201U04");
2558  *     }
2559  * }
2560  *
2561  * var valInfo = [ { id : "grpCd", mandatory : true, minLength : 5 },
2562  *                 { id : "grpNm", mandatory : true } ];
2563  *
2564  * if (com.validateGroup(grp_LoginInfo, valInfo)) {
2565  *     if (confirm("변경된 데이터를 저장하시겠습니까?")) {
2566  *         com.executeSubmission("WS0201U04");
2567  *     }
2568  * }
2569  *
2570  * var valInfo = [ { id : "totWeight", mandatory : true },
2571  *                 { id : "totWeightPwr", mandatory : true },
2572  *                 { id : "totWeightPwr", mandatory : true },
2573  *                 { id : "ibxWeight1", mandatory : true,
2574  *                   valFunc : function() {
2575  *                       if (numLib.parseInt(ibxTotWeight.getValue()) < numLib.parseInt(ibxWeight
2576  *                           return "총 중량이 세부 중량보다 커야 합니다.";
2577  *                       }
2578  *                 } ],
2579  *                 { id : "winding", mandatory : true } ];
2580  *
2581  * if (com.validateGroup(grpCsInfo, valInfo, tacCsInfo, "tabCsInfo1") == false) {
2582  *     return false;
2583  * }
2584  *

```

```

2585 * var valInfo = [ { id : "prntMenuCd", mandatory : true },
2586 *                 { id : "menuCd", mandatory : true,
2587 *                   valFunc : function() {
2588 *                     if (dmaMenu.get("prntMenuCd") == dmaMenu.get("menuCd")) {
2589 *                       return "상위 메뉴 코드와 메뉴 코드가 같아서는 안됩니다.";
2590 *                     }
2591 *                   } },
2592 *                 { id : "menuNm", mandatory : true },
2593 *                 { id : "menuLevel", mandatory : true },
2594 *                 { id : "menuSeq", mandatory : true },
2595 *                 { id : "urlPath", mandatory : true },
2596 *                 { id : "isUse", mandatory : true } ];
2597 *
2598 * if (com.validateGroup(tblMenuInfo, valInfo, tacMenuInfo, "tabMenuInfo1") == false) {
2599 *   return false;
2600 * }
2601 *
2602 * @description
2603 * ※ 입력 허용 문자, 입력 허용 불가 문자, 최대 입력 문자수 설정은 컴포넌트의 속성에서 설정을 권한한다. <br/>
2604 * - allowChar : 입력 허용 문자 <br/>
2605 * - ignoreChar : 입력 허용 불가 문자 <br/>
2606 * - maxLength : 최대 입력 문자수 <br/>
2607 * - maxByteLength : 최대 입력 바이트수 <br/>
2608 */
2609 validateGroup : function(grpObj, valInfoArr, tacObj, tabId)
2610 {
2611   var objArr = WebSquare.util.getChildren(grpObj, {
2612     excludePlugin : "group trigger textbox output calendar image span anchor pageInherit wfr",
2613     recursive : true
2614   });
2615
2616   var valStatus = {
2617     isValid : true,
2618     message : "",
2619     error : []
2620   };
2621   // { columnId: "", comObjId: "", columnNam : "", message: "" }
2622   };
2623   try {
2624     for ( var objIdx in objArr) {
2625       var obj = objArr[objIdx];
2626
2627       if ((typeof objArr[objIdx].validate === "function") && (objArr[objIdx].validate() ==
2628         return false;
2629       )
2630
2631       var dataObjInfo = com.getDataCollection(obj);
2632       var dataCollection = null;
2633       var columnId = null;
2634       var value = null;
2635
2636       if ((dataObjInfo !== undefined) && (dataObjInfo !== null)) {
2637         dataCollection = WebSquare.util.getComponentById(dataObjInfo.runtimeDataCollecti
2638         columnId = dataObjInfo.columnId;
2639       }
2640
2641       if ((dataCollection !== null) && (dataCollection.getObjectType() === "dataMap")) {
2642         value = dataCollection.get(dataObjInfo.columnId).trim();
2643       } else {
2644         var tempIdArr = obj.getID().split("_");
2645         if (obj.getPluginName() !== "editor") {
2646           if ((typeof obj.getValue === "function") && (typeof obj.getValue().trim ===
2647             value = obj.getValue().trim();
2648           ) else {
2649             continue;
2650           }
2651         } else {
2652           value = obj.getText().trim();

```

```

2653     }
2654 }
2655
2656 for ( var valIdx in valInfoArr ) {
2657     var valInfo = valInfoArr[valIdx];
2658     if ((typeof valInfo.id !== "undefined") && (valInfo.id === columnId)) {
2659         if ((typeof valInfo.mandatory !== "undefined") && (valInfo.mandatory === true)) {
2660             _setResult(dataCollection, valInfo.id, obj.getID(), "필수 입력 항목입니다.");
2661         } else if ((typeof valInfo.minLength !== "undefined") && (valInfo.minLength
2662             _setResult(dataCollection, valInfo.id, obj.getID(), "최소 길이 " + valInfo
2663         } else if (typeof valInfo.valFunc === "function") {
2664             var resultMsg = valInfo.valFunc(value);
2665             if ((typeof resultMsg !== "undefined") && (resultMsg !== "")) {
2666                 _setResult(dataCollection, valInfo.id, obj.getID(), resultMsg);
2667             }
2668         }
2669     }
2670 }
2671 }
2672
2673 if (valStatus.error.length > 0) {
2674     valStatus.isValid = false;
2675     valStatus.message = "유효하지 않은 값이 입력 되었습니다";
2676
2677     if ((typeof tacObj !== "undefined") && (typeof tabId !== "undefined") && (tabId !==
2678         var tabIndex = tacObj.getTabIndex(tabId);
2679         tacObj.setSelectedTabIndex(tabIndex);
2680     }
2681
2682     com.valStatus.objectType = "group";
2683     com.valStatus.isValid = false;
2684     com.valStatus.objectName = valStatus.error[0].comObjId;
2685
2686     com.alert(valStatus.error[0].message, "com._groupValidationCallback");
2687 }
2688
2689 return valStatus.isValid;
2690
2691 function _setResult(dataCollection, columnId, comObjId, message) {
2692     var scope = com._getScope(dataCollection);
2693
2694     var errIdx = valStatus.error.length;
2695     valStatus.error[errIdx] = {};
2696     valStatus.error[errIdx].columnId = columnId;
2697     valStatus.error[errIdx].comObjId = comObjId;
2698
2699     if (dataCollection !== null) {
2700         var comObj = scope.$p.getComponentById(comObjId);
2701         valStatus.error[errIdx].columnName = scope.com.getColumnName(comObj);
2702     } else {
2703         valStatus.error[errIdx].columnName = comObj.getInvalidMessage();
2704     }
2705     valStatus.error[errIdx].message = scope.com.attachPostposition(valStatus.error[errId
2706 }
2707 } catch (e) {
2708     console.log("[com.validateGroup] Exception :: Object Id : " + obj.getID() + ", Plug-in N
2709         + e.message);
2710 } finally {
2711     objArr = null;
2712 }
2713 },
2714
2715 /**
2716  * GridView를 통해서 입력된 데이터에 대해서 유효성을 검증한다.
2717  *
2718  * @date 2018.01.19
2719  * @memberOf com
2720  * @param {Object} gridViewObj GridView 객체

```

```

2721 * @param {Object[]} options 데이터 유효성 검증 옵션
2722 * @param {String} options[].id 유효성 검사 대상 DataCollection 컬럼 아이디
2723 * @param {Boolean} options[].mandatory 필수 입력 값 여부
2724 * @param {Number} options[].minLength 최소 입력 자리수
2725 * @param {requestCallback} options[].valFunc 사용자 유효성 검사 함수
2726 * @param {Object} tacObj GridView가 포함된 TabControl 컴포넌트 객체
2727 * @param {String} tabId GridView가 포함된 TabControl 컴포넌트의 Tab 아이디
2728 * @returns {Boolean} 유효성검사 결과
2729 * @since 2015.08.05
2730 * @example
2731 * var valInfo = [ {id: "grpCd", mandatory: true, minLength: 5},
2732 *                 {id: "grpNm", mandatory: true} ];
2733 *
2734 * if (com.validateGridView(grd_MenuAuthority, valInfo)) {
2735 *     if (confirm("변경된 데이터를 저장하시겠습니까?")) {
2736 *         scwin.saveGroup();
2737 *     }
2738 * }
2739 *
2740 * var valInfo = [ { id : "prntMenuCd", mandatory : true },
2741 *                 { id : "menuCd", mandatory : true,
2742 *                   valFunc : function() {
2743 *                       if (dmaMenu.get("prntMenuCd") == dmaMenu.get("menuCd")) {
2744 *                           return "상위 메뉴 코드와 메뉴 코드가 같아서는 안됩니다.";
2745 *                       }
2746 *                   } },
2747 *                 { id : "menuNm", mandatory : true },
2748 *                 { id : "menuLevel", mandatory : true },
2749 *                 { id : "menuSeq", mandatory : true },
2750 *                 { id : "urlPath", mandatory : true },
2751 *                 { id : "isUse", mandatory : true } ];
2752 *
2753 * if (com.validateGridView(grd_MenuAuthority, valInfo, tacMenuInfo, "tabMenuInfo1") == false) {
2754 *     return false;
2755 * }
2756 * @description
2757 * * 입력 허용 문자, 입력 허용 불가 문자, 최대 입력 문자수 설정은 GridView의 Column의 속성에서 설정한다. <br/>
2758 * * - allowChar : 입력 허용 문자 <br/>
2759 * * - ignoreChar : 입력 허용 불가 문자 <br/>
2760 * * - maxLength : 최대 입력 문자수 <br/>
2761 * * - maxByteLength : 최대 입력 바이트수 <br/>

```

```

2789         continue;
2790     }
2791     for ( var valIdx in valInfoArr) {
2792         var valInfo = valInfoArr[valIdx];
2793         if ((typeof valInfo.id !== "undefined") && (typeof modifiedData[valInfo.id] !==
2794             var value = modifiedData[valInfo.id].trim();
2795             if ((typeof valInfo.mandatory !== "undefined") && (valInfo.mandatory === true
2796                 _setResult(modifiedIdx[dataIdx], dataList, gridViewObj.getID(), valInfo.
2797             } else if ((typeof valInfo.minLength !== "undefined") && (valInfo.minLength
2798                 _setResult(modifiedIdx[dataIdx], dataList, gridViewObj.getID(), valInfo.
2799                     + "자리 이상으로 입력해야 합니다.");
2800             } else if (typeof valInfo.valFunc === "function") {
2801                 var resultMsg = valInfo.valFunc(value, modifiedData);
2802                 if ((typeof resultMsg !== "undefined") && (resultMsg !== "")) {
2803                     _setResult(modifiedIdx[dataIdx], dataList, gridViewObj.getID(), valI
2804                 }
2805             }
2806         }
2807     }
2808     if (valStatus.error.length > 0) {
2809         break;
2810     }
2811 }
2812 }
2813
2814 if (valStatus.error.length > 0) {
2815     valStatus.isValid = false;
2816     valStatus.message = "유효하지 않은 값이 입력 되었습니다";
2817
2818     if ((typeof tacObj !== "undefined") && (typeof tabId !== "undefined") && (tabId !==
2819         var tabIndex = tacObj.getTabIndex(tabId);
2820         tacObj.setSelectedTabIndex(tabIndex);
2821     }
2822
2823     com.valStatus.isValid = false;
2824     com.valStatus.objectType = "gridView";
2825     com.valStatus.objectName = valStatus.error[0].comObjId;
2826     com.valStatus.columnId = valStatus.error[0].columnId;
2827     com.valStatus.rowIndex = valStatus.error[0].rowIndex;
2828
2829     com.popup.alert(valStatus.error[0].message, "com._groupValidationCallback");
2830
2831 }
2832
2833 return valStatus.isValid;
2834
2835 function _setResult(rowIndex, dataList, gridViewObjId, columnId, message) {
2836     var errIdx = valStatus.error.length;
2837     valStatus.error[errIdx] = {};
2838     valStatus.error[errIdx].columnId = columnId;
2839     valStatus.error[errIdx].comObjId = gridViewObjId;
2840     valStatus.error[errIdx].columnName = dataList.getColumnName(columnId);
2841     valStatus.error[errIdx].rowIndex = rowIndex;
2842     valStatus.error[errIdx].message = com.attachPostposition(valStatus.error[errIdx].col
2843 }
2844 } catch (e) {
2845     $p.log("[com.validateGridView] Exception :: " + e.message);
2846 } finally {
2847     modifiedData = null;
2848     modifiedIdx = null;
2849     dataList = null;
2850     gridViewObj = null;
2851 }
2852 },
2853
2854 /**
2855  * 문자열을 날짜포맷으로 변환
2856  *

```

```

2857 * @date 2020.06.05
2858 * @memberOf com
2859 * @param {String} psData 문자
2860 * @return {String} 포맷 변형된 문자열
2861 * @author 전기완
2862 * @example
2863 */
2864 strToDateFormat : function(psData)
2865 {
2866     if(com.data.isEmpty(psData)){
2867         return psData;
2868     }
2869     vsRtnData = String(psData);
2870
2871     if(vsRtnData.length == 8){
2872         vsRtnData = vsRtnData.replace(/(\d{4})(\d{2})(\d{2})/, '$1-$2-$3');
2873     } else if(psData.length == 12){
2874         vsRtnData = vsRtnData.replace(/(\d{4})(\d{2})(\d{2})(\d{2})(\d{2})/, '$1-$2-$3 $4:$5');
2875     } else if(psData.length == 14 || (vsClassNames.indexOf("_Date14") > -1 && psData.length > 14)
2876     {
2877         vsRtnData = vsRtnData.replace(/(\d{4})(\d{2})(\d{2})(\d{2})(\d{2})(\d{2})/, '$1-$2-$3 $4
2878     }
2879     return vsRtnData;
2880 },
2881 /**
2882 * 일반전화번호 및 휴대전화번호를 형태에 맞게 "-"을 넣어서 변경
2883 *
2884 *
2885 * @date 2020.06.05
2886 * @memberOf com
2887 * @param {String} psData 문자
2888 * @return {String} 포맷 변형된 문자열
2889 * @author 전기완
2890 * @example
2891 * 컴포넌트에서 사용시
2892 * displayFormatter 속성에 "com.telFormat" 기입
2893 *
2894 * 스크립트에서 사용시
2895 * com.telFormat("0212345678")
2896 * com.telFormat("01012345678")
2897 */
2898 telFormat : function(psData)
2899 {
2900     var vsRtnData = psData
2901     if(com["RegExpTelHpNo"].test(vsRtnData)){
2902         vsRtnData = vsRtnData.replace(/(^02.{0}|^01.{1}|[0-9]{3})([0-9]+)([0-9]{4})/, "$1-$2-$3"
2903     }
2904     return vsRtnData;
2905 },
2906 /**
2907 * 일반전화번호를 유효성 체크 후 형태에 맞게 "-"을 넣어서 변경
2908 * 유효성 체크 실패시 원본데이터 return
2909 *
2910 *
2911 * @date 2020.06.05
2912 * @memberOf com
2913 * @param {String} psData 문자
2914 * @return {String} 포맷 변형된 문자열
2915 * @author 전기완
2916 * @example
2917 * 컴포넌트에서 사용시
2918 * displayFormatter 속성에 "com.telChkFormat" 기입
2919 *
2920 * 스크립트에서 사용시
2921 * com.telFormat("0212345678")
2922 * com.telFormat("01012345678")
2923 */
2924 telChkFormat : function(psData)

```

```

2925 {
2926     var vsRtnData = psData
2927     if(com["regExpTel"].test(vsRtnData)){
2928         vsRtnData = com.telFormat(vsRtnData);
2929     }
2930     return vsRtnData;
2931 },
2932
2933 /**
2934  * 휴대전화번호를 유효성 체크 후 형태에 맞게 "-"을 넣어서 변경
2935  * 유효성 체크 실패시 원본데이터 return
2936  *
2937  * @date 2020.06.05
2938  * @memberOf com
2939  * @param {String} psData 문자
2940  * @return {String} 포맷 변형된 문자열
2941  * @author 전기완
2942  * @example
2943  * 컴포넌트에서 사용시
2944  * displayFormatter 속성에 "com.moblChkFormat" 기입
2945  *
2946  * 스크립트에서 사용시
2947  * com.telFormat("0212345678")
2948  * com.telFormat("01012345678")
2949  */
2950 mobilChkFormat : function(psData)
2951 {
2952     var vsRtnData = psData
2953     if(com["regExpHp"].test(vsRtnData)){
2954         vsRtnData = com.telFormat(vsRtnData);
2955     }
2956     return vsRtnData;
2957 },
2958
2959 /**
2960  * 메일주소 체크한다.
2961  *
2962  * @date 2014. 12. 10.
2963  * @memberOf com
2964  * @param {String} str 메일주소
2965  * @return {Boolean} 정상이면 공백("")을 반환, 그외는 에러 메시지 반환
2966  * @example com.isEmail("emailTest@email.com")
2967  */
2968 checkEmail : function(str)
2969 {
2970     if (typeof str != "undefined" && str != "") {
2971         var format = /^[a-zA-Z0-9_\.\-]+\@((\[[a-zA-Z0-9\-\]]+\.)+|([a-zA-Z0-9]{2,4})+)$/;
2972
2973         if (format.test(str)) {
2974             return true;
2975         } else {
2976             return false;
2977         }
2978     }
2979     return true;
2980 },
2981
2982 /**
2983  * 숫자 3자리마다 ",", 표시 후 return
2984  *
2985  * @date 2020.06.05
2986  * @memberOf com
2987  * @param {String or Number} psStr 문자 or 숫자
2988  * @param {String or Number} pnDecimalLength 소수점 자리수(정수일 경우 미입력)
2989  * @return {String} 포맷 변형된 문자열
2990  * @author 전기완
2991  * @example
2992  * 컴포넌트에서 사용시

```

```

2993 * displayFormatter 속성에 "com.toFormatNum" 기입
2994 *
2995 * 스크립트에서 사용시
2996 * com.toFormatNum(10.123) ==> 10
2997 * com.toFormatNum(10.123, 2) ==> 10.12
2998 */
2999 toFormatNum : function(psStr, pnDecimalLength)
3000 {
3001     psStr = Number(com.data.isEmpty(psStr, "0"));
3002     if(isNaN(psStr)){
3003         return "";
3004     }
3005
3006     var vnPow = Math.pow(10, Number(com.data.isEmpty(pnDecimalLength, 0)));
3007     psStr = Math.round(psStr * vnPow) / vnPow;
3008     psStr = String(psStr);
3009     var vsNum1 = "";
3010     var vsNum2 = "";
3011     if(psStr.indexOf(".") > -1){
3012         vsNum1 = psStr.split(".")[0];
3013         vsNum2 = "." + psStr.split(".")[1];
3014     } else {
3015         vsNum1 = psStr;
3016     }
3017     return vsNum1.replace(/(\d)(?=(?:\d{3})+(?! \d))/g, '$1,') + vsNum2;
3018 },
3019
3020 /**
3021 * 내외국인 주민등록번호 유효성을 검사한다.
3022 *
3023 * @date 2014. 12. 10.
3024 * @memberOf com
3025 * @param {String} str 문자열
3026 * @returns {Boolean} 올바른 번호가 아닌경우 false
3027 * @example com.checkPersonID("9701011234567");
3028 */
3029 checkPersonID : function(str)
3030 {
3031     var checkID = new Array(2, 3, 4, 5, 6, 7, 8, 9, 2, 3, 4, 5);
3032     var i = 0, sum = 0;
3033     var temp = 0;
3034     var yy = "";
3035
3036     if (str.length != 13) {
3037         return false;
3038     }
3039     for (i = 0; i < 13; i++) {
3040         if (str.charAt(i) < '0' || str.charAt(i) > '9') {
3041             return false;
3042         }
3043     }
3044
3045     // foreigner PersonID Pass
3046     if (str.substring(6, 13) == "5000000" || str.substring(6, 13) == "6000000" || str.substring(
3047         || str.substring(6, 13) == "8000000") {
3048         return true;
3049     }
3050     for (i = 0; i < 12; i++) {
3051         sum += str.charAt(i) * checkID[i];
3052     }
3053     temp = sum - Math.floor(sum / 11) * 11;
3054     temp = 11 - temp;
3055     temp = temp - Math.floor(temp / 10) * 10;
3056
3057     // 나이 (-) 체크
3058     if (str.charAt(6) == '1' || str.charAt(6) == '2' || str.charAt(6) == '5' || str.charAt(6) ==
3059         yy = "19";
3060     } else {

```

```

3061     yy = "20";
3062 }
3063
3064 if (parseInt(common_util.getCurrentDate('yyyy')) - parseInt(yy + str.substring(0, 2)) < 0) {
3065     return false;
3066 }
3067
3068 // 외국인 주민번호 체크로직 추가
3069 if (str.charAt(6) != '5' && str.charAt(6) != '6' && str.charAt(6) != '7' && str.charAt(6) !=
3070     if (temp == eval(str.charAt(12))) {
3071         return true;
3072     } else {
3073         return false;
3074     }
3075 } else {
3076     if ((temp + 2) % 10 == eval(str.charAt(12))) {
3077         return true;
3078     } else {
3079         return false;
3080     }
3081 }
3082 return false;
3083 },
3084
3085
3086
3087 /**
3088  * 주민번호 문자열에 Formatter(#####-#####)를 적용하여 반환한다.
3089  *
3090  * @date 2016.08.02
3091  * @param {String} str 주민번호 문자열
3092  * @memberOf com
3093  * @author InswaveSystems
3094  * @return {String} 포맷터가 적용된 주민번호 문자열
3095  * @example
3096  * com.transIdNum("1234561234567");
3097  * // return 예시) "123456-1234567"
3098  */
3099 transIdNum : function(str)
3100 {
3101     var front = String(str).substr(0, 6);
3102     var back = String(str).substr(6, 7);
3103     var output = front + "-" + back;
3104
3105     return output;
3106 },
3107
3108 /**
3109  * 전화번호, setDisplayFormat("###-###-###") - 입력된 str에 포맷터를 적용하여 반환한다.
3110  *
3111  * @date 2016.08.02
3112  * @param {String} str 포맷터를 적용할 문자열
3113  * @memberOf com
3114  * @author InswaveSystems
3115  * @return {String} 포맷터가 적용된 문자열
3116  * @example
3117  * com.transPhone("0212345678");
3118  * // return 예시) "02-1234-5678"
3119  * com.transPhone("021234567");
3120  * // return 예시) "02-123-4567"
3121  * com.transPhone("03112345678");
3122  * // return 예시) "031-1234-5678"
3123  * com.transPhone("0311234567");
3124  * // return 예시) "031-123-4567"
3125  */
3126 transPhone : function(str)
3127 {
3128     return str.replace(/^(01[016789]{1}|02|0[3-9]{1}|0[0-9]{1})-?([0-9]{3,4})-?([0-9]{4})$/, "$1-$2-$3");

```

```

3129     },
3130
3131     /**
3132     * 시간 - 입력된 String 또는 Number에 포맷터를 적용하여 반환한다.
3133     *
3134     * @date 2016.08.02
3135     * @param {String} value 시간 Formatter를 적용한 값 (String 또는 Number 타입 지원)
3136     * @memberOf com
3137     * @author InswaveSystems
3138     * @return {String} 포맷터가 적용된 문자열
3139     * @example
3140     * com.transTime("123402");
3141     * // return 예시) "12:34:02"
3142     */
3143     transTime : function(value)
3144     {
3145         var hour = String(value).substr(0, 2);
3146         var minute = String(value).substr(2, 2);
3147         var second = String(value).substr(4, 2);
3148         var output = hour + ":" + minute + ":" + second;
3149
3150         return output;
3151     },
3152
3153     /**
3154     * 세번째자리마다 콤마 표시, 금액, setDisplayFormat("#,###&#46##0", "fn_userFormatter2") - 입력된 str에 1
3155     *
3156     * @date 2016.08.02
3157     * @param {String} value String or Number 포맷터를 적용할 값 (String 또는 Number 타입 지원)
3158     * @param {String} type 적용할 포맷터 형식(Default:null,dollar,plusZero,won)
3159     * @memberOf com
3160     * @author InswaveSystems
3161     * @return {String} 포맷터가 적용된 문자열
3162     * @example
3163     * com.transComma("12345");
3164     * // return 예시) 12,345
3165     * com.transComma("12345", "dollar");
3166     * // return 예시) $12,345
3167     * com.transComma("12345", "plusZero");
3168     * // return 예시) 123,450
3169     * com.transComma("12345", "won");
3170     * // return 예시) 12,345원
3171     */
3172     transComma : function(value, type)
3173     {
3174         var amount;
3175
3176         if (type == "plusZero") {
3177             amount = new String(value) + "0";
3178         } else {
3179             amount = new String(value);
3180         }
3181
3182         amount = amount.split(".");
3183
3184         var amount1 = amount[0].split("").reverse();
3185         var amount2 = amount[1];
3186
3187         var output = "";
3188         for (var i = 0; i <= amount1.length - 1; i++) {
3189             output = amount1[i] + output;
3190             if ((i + 1) % 3 == 0 && (amount1.length - 1) != i)
3191                 output = ',' + output;
3192         }
3193
3194         if (type == "dollar") {
3195             if (!amount2) {
3196                 output = "$ " + output;

```

```

3197         } else {
3198             output = "$ " + output + "." + amount2;
3199         }
3200     } else if (type == "won") {
3201         if (!amount2) {
3202             output = output + "원";
3203         } else {
3204             output = output + "." + amount2 + "원";
3205         }
3206     } else {
3207         if (!amount2) {
3208             output = output;
3209         } else {
3210             output = output + "." + amount2;
3211         }
3212     }
3213
3214     return output;
3215 },
3216
3217 /**
3218  * 텍스트 - 입력된 str에 포맷터를 적용하여 반환한다.
3219  *
3220  * @date 2016.08.02
3221  * @param {String} str String or Number 포맷터를 적용할 값
3222  * @memberOf com
3223  * @author InswaveSystems
3224  * @return {String} 포맷터가 적용된 문자열
3225  * @example
3226  * com.transText("1");
3227  * // return 예시) 1
3228  * com.transText("12");
3229  * // return 예시) 12
3230  * com.transText("123");
3231  * // return 예시) 1.23
3232  * com.transText("1234");
3233  * // return 예시) 12.34
3234  * com.transText("12345");
3235  * // return 예시) 123.345
3236  * com.transText("123456");
3237  * // return 예시) 1234.56
3238  * com.transText("1234567");
3239  * // return 예시) 12345.67
3240  */
3241 transText : function(str)
3242 {
3243     var amount = new String(str);
3244     var result;
3245
3246     if (amount.length < 3) {
3247         result = amount.substr(0, amount.length);
3248     } else {
3249         result = amount.substr(0, amount.length - 2) + "." + amount.substr(amount.length - 2, am
3250     }
3251     return result;
3252 },
3253
3254 /**
3255  * 날짜 - 입력된 str에 포맷터를 적용하여 반환한다.
3256  *
3257  * @date 2016.08.02
3258  * @param {String} str 포맷터를 적용할 파라미터 (String 또는 Number 타입 지원)
3259  * @param {String} type 적용할 포맷터 형식 Default:null,slash,date
3260  * @memberOf com
3261  * @author InswaveSystems
3262  * @return {String} 포맷터가 적용된 문자열
3263  * @example
3264  * com.transDate(20120319, "slash");

```

```

3265 * // return 예시) 12/03/19
3266 * com.transDate(20120319, "date");
3267 * // return 예시) 2012/03/19
3268 * com.transDate(20120319, "colon");
3269 * // return 예시) 2012:03:19
3270 * com.transDate(20120319);
3271 * // return 예시) 2012년 03월 19일
3272 */
3273 transDate : function(str, type)
3274 {
3275     var output = "";
3276     var date = new String(str);
3277
3278     if (type == "slash") {
3279         date = date.substring(2, date.length);
3280         for (var i = 0; i <= date.length - 1; i++) {
3281             output = output + date[i];
3282             if ((i + 1) % 2 == 0 && (date.length - 1) !== i)
3283                 output = output + "/";
3284         }
3285     } else if (type == "date") {
3286         if (date.length == 8) {
3287             output = date.substr(0, 4) + "/" + date.substr(4, 2) + "/" + date.substr(6, 2);
3288         }
3289     } else if (type == "colon") {
3290         if (date.length == 8) {
3291             output = date.substr(0, 4) + ":" + date.substr(4, 2) + ":" + date.substr(6, 2);
3292         }
3293     } else {
3294         var year = date.substr(0, 4);
3295         var month = date.substr(4, 2);
3296         var day = date.substr(6, 2);
3297         var output = year + "년 " + month + "월 " + day + "일";
3298     }
3299     return output;
3300 },
3301
3302 /**
3303 * 문자열 왼쪽에 일정길이(maxLen) 만큼 '0'으로 채우기
3304 *
3305 * @date 2014.12.10
3306 * @param {String} str 포맷터를 적용할 문자열
3307 * @param {Number} maxLen 0 으로 채울 길이
3308 * @memberOf com
3309 * @author InswaveSystems
3310 * @return {String} 일정길이 만큼 0 으로 채워진 문자열
3311 * @example
3312 * com.fillZero("24", 4);
3313 * // return 예시) "0024"
3314 *
3315 * com.fillZero("11321", 8);
3316 * // return 예시) "00011321"
3317 */
3318 fillZero : function(str, maxLen)
3319 {
3320     var len = str;
3321     var zero = "";
3322
3323     if (typeof str == 'number')
3324         len = '' + str;
3325
3326     if (len.length < maxLen) {
3327         for (var i = len.length; i < maxLen; i++) {
3328             zero += "0";
3329         }
3330     }
3331     return (zero + '' + str);
3332 },

```

```

3333
3334 /**
3335  * 사업자번호 유효성을 검사한다.
3336  *
3337  * @date 2014.12.10
3338  * @memberOf com
3339  * @param {String} str 사업자번호 문자열
3340  * @returns {Boolean} 올바른 번호가 아닌경우 false
3341  * @example com.checkBizID("1102112345");
3342  */
3343 checkBizID : function(str)
3344 {
3345     var sum = 0;
3346     var aBizID = new Array(10);
3347     var checkID = new Array("1", "3", "7", "1", "3", "7", "1", "3", "5");
3348
3349     for (var i = 0; i < 10; i++) {
3350         aBizID[i] = str.substring(i, i + 1);
3351     }
3352     for (var i = 0; i < 9; i++) {
3353         sum += aBizID[i] * checkID[i];
3354     }
3355     sum = sum + parseInt((aBizID[9] * 5) / 10);
3356     temp = sum % 10;
3357     temp1 = 0;
3358
3359     if (temp != 0) {
3360         temp1 = 10 - temp;
3361     } else {
3362         temp1 = 0;
3363     }
3364     if (temp1 != aBizID[9]) {
3365         return false;
3366     }
3367     return true;
3368 },
3369
3370 /**
3371  * 법인등록번호 유효성을 검사한다.
3372  *
3373  * @date 2014. 12. 10.
3374  * @memberOf com
3375  * @param {String} str 문자열
3376  * @returns {Boolean} 올바른 번호가 아닌경우 false
3377  * @example com.checkCorpID("110211234567");
3378  */
3379 checkCorpID : function(str)
3380 {
3381     var checkID = new Array(1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2);
3382     var sCorpNo = str;
3383     var lV1 = 0;
3384     var nV2 = 0;
3385     var nV3 = 0;
3386
3387     for (var i = 0; i < 12; i++) {
3388         lV1 = parseInt(sCorpNo.substring(i, i + 1)) * checkID[i];
3389
3390         if (lV1 >= 10) {
3391             nV2 += lV1 % 10;
3392         } else {
3393             nV2 += lV1;
3394         }
3395     }
3396     nV3 = nV2 % 10;
3397
3398     if (nV3 > 0) {
3399         nV3 = 10 - nV3;
3400     } else {

```

```

3401         nv3 = 0;
3402     }
3403     if (sCorpNo.substring(12, 13) != nv3) {
3404         return false;
3405     }
3406     return true;
3407 },
3408
3409 /**
3410  * 입력받은 문자열이 한글이면 true, 아니면 false를 리턴한다.
3411  *
3412  * @date 2016.08.02
3413  * @param {String} str 한글 유형인지 검증할 문자열
3414  * @memberOf com
3415  * @author InswaveSystems
3416  * @return {Boolean} true or false
3417  * @example
3418  * com.isKorean("");
3419  * // return 예시) false
3420  * com.isKorean("무궁화꽃이");
3421  * // return 예시) true
3422  */
3423 isKorean : function(str)
3424 {
3425     if (str != null && str.length > 0) {
3426         var locale = 0;
3427         for (var i = 0; i < str.length; i++) {
3428             locale = com.getLocale(str.charAt(i));
3429         }
3430         if ((locale & ~0x3) == 0) {
3431             return true;
3432         }
3433     }
3434     return false;
3435 },
3436
3437 /**
3438  * 입력받은 문자열에 한글이 포함되어 있으면 true, 아니면 false를 리턴한다.
3439  *
3440  * @date 2016.08.02
3441  * @param {String} str 한글이 포함되어 있는지 검증 받을 문자열
3442  * @memberOf com
3443  * @author InswaveSystems
3444  * @return {Boolean} true or false
3445  * @example
3446  * com.isKoreanWord("abcd무궁화"); //return 예시) true
3447  * com.isKoreanWord("abcd"); //return 예시) false
3448  */
3449 isKoreanWord : function(str)
3450 {
3451     var c;
3452     for (var i = 0; i < str.length; i++) {
3453         c = str.charAt(i);
3454         if (com.isKorean(c)) {
3455             return true;
3456         }
3457     }
3458     return false;
3459 },
3460
3461 /**
3462  * 종성이 존재하는지 여부를 검사한다.
3463  *
3464  * @date 2016.08.02
3465  * @param {String} str 종성의 여부를 검사할 문자열
3466  * @memberOf com
3467  * @author InswaveSystems
3468  * @return {Boolean} true or false

```

```

3469 * @example
3470 * com.isFinalConsonant("종서")
3471 * // return 예시) false
3472 * com.isFinalConsonant("종성")
3473 * // return 예시) true
3474 * com.isFinalConsonant("입니다")
3475 * // return 예시) false
3476 * com.isFinalConsonant("입니다")
3477 * // return 예시) true
3478 */
3479 isFinalConsonant : function(str)
3480 {
3481     var code = str.charCodeAt(str.length - 1);
3482     if ((code < 44032) || (code > 55197)) {
3483         return false;
3484     }
3485     if ((code - 16) % 28 == 0) {
3486         return false;
3487     }
3488     return true;
3489 },
3490
3491 /**
3492 * 단어 뒤에 '은'이나 '는'을 붙여서 반환한다.
3493 *
3494 * @date 2016.08.02
3495 * @param {String} str 은, 는 붙일 문자열
3496 * @memberOf com
3497 * @author InswaveSystems
3498 * @return {String} 변환된 문자열
3499 * @example
3500 * com.attachPostposition("나");
3501 * // return 예시)"나는"
3502 * com.attachPostposition("나와 너");
3503 * // return 예시)"나와 너는"
3504 * com.attachPostposition("그래서");
3505 * // return 예시)"그래서는"
3506 * com.attachPostposition("그랬습니다만");
3507 * // return 예시)"그랬습니다만은"
3508 */
3509 attachPostposition : function(str)
3510 {
3511     if (com.isFinalConsonant(str)) {
3512         str = str + "은 ";
3513     } else {
3514         str = str + "는 ";
3515     }
3516     return str;
3517 },
3518
3519 /**
3520 * 특정 컴포넌트가 속한 WFrame Scope을 반환한다.
3521 *
3522 * @date 2018.06.11
3523 * @private
3524 * @param {Object} 컴포넌트 객체 또는 아이디(WFrame Scope 경로를 포함한 Full Path Id)
3525 * @memberOf
3526 * @author InswaveSystems
3527 */
3528 _getScope : function(comObj)
3529 {
3530     if (typeof comObj === "string") {
3531         var scopeObj = $p.getComponentById(comObj);
3532         if (scopeObj !== null) {
3533             return scopeObj.getScopeWindow();
3534         }
3535     } else {
3536         return comObj.getScopeWindow();
3537     }
3538 }

```

```

3537     }
3538 },
3539
3540 /**
3541  * 유효성 검사 실패에 대한 Alert 메시지 창이 닫힌 후에 수행되는 콜백 함수이다.
3542  *
3543  * @date 2018.04.15
3544  * @private
3545  * @memberOf
3546  * @author InswaveSystems
3547  * @returns
3548  */
3549 _groupValidationCallback : function()
3550 {
3551     if ((com.valStatus.objectName !== "") && (com.valStatus.isValid === false)) {
3552         var obj = $p.getComponentById(com.valStatus.objectName);
3553         if (com.valStatus.objectType === "gridView") {
3554             obj.setFocusedCell(com.valStatus.rowIndex, com.valStatus.columnId);
3555         } else if (com.valStatus.objectType === "group") {
3556             obj.focus();
3557         }
3558     }
3559 },
3560
3561 /**
3562  * 객체의 typeof 값을 반환하며 typeof의 값이 object인 경우 array, json, xml, null로 체크하여 반환한다.
3563  *
3564  * @date 2016.12.20
3565  * @param {Object} obj type을 반환 받을 객체(string,boolean,number,object 등)
3566  * @author InswaveSystems
3567  * @return {String} 객체의 타입으로 typeof가 object인 경우 array, json, xml, null로 세분화하여 반환한다. 그
3568  * @example
3569  * com.getObjectType("WebSquare");
3570  * // return 예시) "string"
3571  * com.getObjectType({"name":"WebSquare"});
3572  * // return 예시) "json"
3573  * com.getObjectType(["1","2"]);
3574  * // return 예시) "array"
3575  */
3576 getObjectType : function(obj)
3577 {
3578     var objType = typeof obj;
3579     if (objType !== 'object') {
3580         return objType;
3581     } else if (obj.constructor === {}.constructor) {
3582         return 'json';
3583     } else if (obj.constructor === [].constructor) {
3584         return 'array';
3585     } else if (obj === null) {
3586         return 'null';
3587     } else {
3588         var tmpDoc = WebSquare.xml.parse("<data></data>");
3589         if (obj.constructor === tmpDoc.constructor || obj.constructor === tmpDoc.firstElementChi
3590             return 'xml';
3591         } else {
3592             return objType;
3593         }
3594     }
3595 }
3596 };
3597
3598
3599
3600
3601
3602
3603
3604

```

```

3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617 //-----
3618 // Event
3619 //-----
3620 /**
3621  * scwin.onpageload 함수 이전에 실행되는 함수
3622  * 개발자 직접 호출 금지
3623  * scwin.onpageload 호출 전 처리가 필요할 경우
3624  * scwin.doInit을 호출(공통처리외에 개발자 사용을 지양)
3625  * 상위부모 페이지의 scwin.onpageload를 실행하기 전 공통처리
3626  *
3627  * @date 2020.06.05
3628  * @memberOf com
3629  * @return {void}
3630  * @author 전기원
3631  * @example
3632  *
3633  */
3634 com.event = {
3635
3636     wfInit : function(e)
3637     {
3638         console.log('com.wfInit...');
3639         if(udc_obj.id.startsWith('mf_wdcMain_subWindow') && udc_obj.id.endsWith('buttonset'))
3640         {
3641             var allMenu = JSON.parse(sessionStorage.getItem('menuList'));
3642             var menuInfo = allMenu.filter(function(obj){ return (obj.menuId == gcm['menuId']) && obj..
3643
3644             if(menuInfo[0]['navi'] != null) $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).repl.
3645             $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('span[id$=spaMenuTitle]').text
3646         }
3647
3648         if(udc_obj.id.startsWith('mf_wdcMain_subWindow') && !udc_obj.id.endsWith('buttonset'))
3649         {
3650             //WebSquare.layer.showProcessMessage("처리중");
3651
3652             var allMenu = JSON.parse(sessionStorage.getItem('menuList'));
3653             var menuInfo = allMenu.filter(function(obj){ return (obj.menuId == gcm['menuId']) && obj..
3654             if(allMenu != null && allMenu != 'undefined' && menuInfo.length > 0)
3655             {
3656                 com.session.adjustAuthority();
3657             }
3658         }
3659     },
3660
3661     /**
3662     * scwin.onpageload 함수 이후에 실행되는 함수
3663     * 개발자 직접 호출 금지
3664     * scwin.onpageload 호출 후 처리가 필요할 경우
3665     * scwin.doPageInit을 사용
3666     * 상위부모 페이지의 scwin.onpageload를 먼저 실행한 후 wframe 선언 순서대로 scwin.onpageload를 실행이 완료되면
3667     * wframe의 scwin.doPostInit을 먼저 실행 후 부모의 scwin.doPostInit을 실행한다.
3668     *
3669     * @date 2020.06.05
3670     * @memberOf com
3671     * @return {void}
3672     * @author 전기원

```

```

3673 * @example
3674 *
3675 */
3676 wfPost : function()
3677 {
3678     console.log('com.wfPost...');
3679     //WebSquare.layer.hideProcessMessage();
3680 },
3681
3682
3683 /**
3684  * 컴포넌트 설정 이벤트 중지 함수.
3685  *
3686  * @date 2019.03.13
3687  * @memberOf com
3688  * @author InswaveSystems
3689  * @param {String} _targetComp : 설정 컴포넌트 객체ID
3690  * @param {Boolean} _flag : 이벤트 설정 여부 값 [default: false(실행), true(중지)]
3691  * @param {Object} _eventList : 중지 이벤트 리스트 값(배열) [default:null (모든 이벤트)]
3692  */
3693 setEventPause : function(_targetComp, _flag, _eventList)
3694 {
3695     try {
3696         var comp = $p.getComponentById(_targetComp);
3697         var flag = WebSquare.util.getBoolean(_flag);
3698         var eventList = typeof _eventList != "undefined" ? _eventList : null;
3699         if (typeof comp == "undefined")
3700             return -1;
3701
3702         if (comp.options.pluginName == "dataList") {
3703             comp.setBroadcast(flag);
3704
3705             if (flag) {
3706                 comp.broadcast({
3707                     "linkedDataList" : [ "notifyDataChanged" ],
3708                     "gridView" : [ "notifyDataChanged" ],
3709                     "generalComp" : "both"
3710                 });
3711             }
3712
3713             comp.setEventPause(eventList, flag);
3714             for ( var i in comp.childCompHash ) {
3715                 var childComp = ngmf.object(comp.childCompHash[i].id);
3716                 if (typeof childComp != "undefined") {
3717                     childComp.setEventPause(eventList, flag);
3718                 }
3719             }
3720
3721             for ( var i in comp.refCompHash ) {
3722                 var refComp = ngmf.object(comp.refCompHash[i].id);
3723                 if (typeof refComp != "undefined") {
3724                     refComp.setEventPause(eventList, flag);
3725                 }
3726             }
3727
3728             if (flag) {
3729                 WebSquare.event.fireEvent(comp, "ondataload");
3730             }
3731             } else {
3732                 comp.setEventPause(eventList, !flag);
3733             }
3734         } catch (e) {
3735
3736         }
3737     },
3738
3739     /**
3740     * 해당 그룹 안의 컴포넌트에서 엔터키가 발생하면 해당 컴포넌트의 값을 DataCollection에 저장하고 objFunc 함수를 실행

```

```

3741 *
3742 * @date 2018.02.15
3743 * @param {Object} grpObj 그룹 객체
3744 * @param {Object} objFunc 함수 객체
3745 * @param {Number} rowIndex DataList가 바인딩된 gridView인 경우 ==> 현재 포커스된 focusedRowIndex [ex.
3746 * <br/>아닌 경우 ==> rowIndex는 생략
3747 * @memberOf com
3748 * @author InswaveSystems
3749 * @example
3750 * com.setEnterKeyEvent(grp_AuthorityDetail, scwin.search);
3751 * // return 예시) "엔터키가 발생 -> 해당 함수 실행 및 DataCollection에 UI 컴포넌트에 입력된 데이터를 DataColl
3752 */
3753 setEnterKeyEvent : function(grpObj, objFunc)
3754 {
3755     var objArr = WebSquare.util.getChildren(grpObj, {
3756         excludePlugin : "group trigger textbox output calendar image span",
3757         recursive : true
3758     });
3759     try {
3760         for (var i = 0; i < objArr.length; i++) {
3761             try {
3762                 if (typeof objFunc === "function") {
3763                     objArr[i].bind("onkeyup", function(e) {
3764                         if (e.keyCode === 13) {
3765                             if (typeof this.getRef === "function") {
3766                                 var ref = this.getRef();
3767                                 var refArray = ref.substring(5).split(".");
3768                                 if ((typeof refArray !== "undefined") && (refArray.length === 2))
3769                                     var dataCollectionName = refArray[0];
3770                                     var columnId = refArray[1];
3771                                     var dataCollection = this.getScopeWindow().$p.getComponentBy
3772                                     var dataType = dataCollection.getObjectType().toLowerCase();
3773                                     if (dataType === "datamap") {
3774                                         dataCollection.set(columnId, this.getValue());
3775                                     } else if ((dataType === 'datalist') && (typeof rowIndex !==
3776                                         dataCollection.setCellData(dataCollection.getRowPosition
3777                                     )
3778                                 }
3779                             }
3780                             objFunc();
3781                         }
3782                     });
3783                 }
3784             } catch (e) {
3785                 $p.log("[com.setEnterKeyEvent] Exception :: " + e.message);
3786             } finally {
3787                 dataCollection = null;
3788             }
3789         }
3790     } catch (e) {
3791         $p.log("[com.setEnterKeyEvent] Exception :: " + e.message);
3792     } finally {
3793         objArr = null;
3794     }
3795 }
3796 };
3797 };
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808

```

```

3809
3810
3811
3812
3813
3814
3815
3816
3817
3818 //-----
3819 // Popup
3820 //-----
3821
3822 /**
3823  * 검색 폼, 팝업을 생성한다.
3824  *
3825  * @date 2020.08.10
3826  * @memberOf com
3827  * @param {String} [formId] [필수]생성 대상의 xf:group 태그 id
3828  * @param {String} [viewString] [필수]생성할 팝업 화면 xml path
3829  * @param {JSON} [popupOption] [필수]각 팝업별 Option 참조
3830  * @param {String} [extraInputId] 팝업내에서 선택된 정보의 항목을 2개 이상 출력할 때 제한적으로 사용
3831  * @author 김동혁
3832  * @example
3833  */
3834
3835 com.popup =
3836 {
3837   open : function(url, opt, data)
3838   {
3839     let basicWidth = 995;
3840     let basicHeight = 735;
3841     console.log(opt);
3842     if(com.data.isEmpty(opt)) opt = { id:url.replace(/\.\/gi, '').replace(/\\/gi, ''), popupName:
3843     if(com.data.isEmpty(opt.width)) opt.width = basicWidth;
3844     if(com.data.isEmpty(opt.height)) opt.height = basicHeight;
3845     if(com.data.isEmpty(data)) data = {};
3846
3847
3848     let _dataObj = {
3849       type : "json",
3850       data : data,
3851       name : "param"
3852     };
3853
3854     try {
3855       let deviceWidth = parseFloat($(".body").css("width"));
3856       let deviceHeight = parseFloat($(".body").css("height"));
3857       if (!opt.notMinSize) {
3858         if (deviceWidth > 0 && width > deviceWidth) {
3859           opt.width = deviceWidth - 4; // 팝업 border 고려
3860         }
3861
3862         if (deviceHeight > 0 && opt.height > deviceHeight) {
3863           opt.height = deviceHeight - 4; // 팝업 border 고려
3864         }
3865       }
3866     } catch (e) {
3867
3868     }
3869
3870     let top = ((document.body.offsetHeight / 2) - (parseInt(opt.height) / 2) + $(document).scroll
3871     let left = ((document.body.offsetWidth / 2) - (parseInt(opt.width) / 2) + $(document).scroll
3872
3873     if (typeof _dataObj.data.callbackFn === "undefined") {
3874       _dataObj.data.callbackFn = "";
3875     } else if (_dataObj.data.callbackFn.indexOf("gcm") !== 0) {
3876       _dataObj.data.callbackFn = $p.id + _dataObj.data.callbackFn;

```

```

3877     }
3878     console.log(opt.width + ', ' + opt.height);
3879
3880     let options = {
3881         id : opt.id,
3882         popupName : opt.popupName || "",
3883         type : opt.type || "wframePopup",
3884         width : opt.width + "px",
3885         height : opt.height + "px",
3886         top : opt.top || top || "140px",
3887         left : opt.left || left || "500px",
3888         modal : (opt.modal == false) ? false : true,
3889         dataObject : _dataObj,
3890         alwaysOnTop : opt.alwaysOnTop || false,
3891         useModalStack : (opt.useModalStack == false) ? false : true,
3892         resizable : (opt.resizable == false) ? false : true,
3893         useMaximize : opt.useMaximize || false
3894     };
3895
3896     $p.openPopup(url, options);
3897 },
3898 param : function()
3899 {
3900     return $p.getPopupParam();
3901 },
3902 close : function(callbackFnStr, retObj, callbackYn, selectedIdx)
3903 {
3904     let popId = $p.getPopupId();
3905     let retStr = com.data.strSerialize(retObj);
3906
3907     if ((typeof callbackFnStr !== "undefined") && (callbackFnStr !== "")) {
3908         var func;
3909         if (callbackFnStr.indexOf("__close_callback_func__") > -1) {
3910             func = gcm.CB_FUNCTION_MANAGER["cbFuncSave"][callbackFnStr];
3911             delete gcm.CB_FUNCTION_MANAGER["cbFuncSave"][callbackFnStr];
3912         } else {
3913             func = window.WebSquare.util.getGlobalFunction(callbackFnStr);
3914         }
3915
3916         if (func) {
3917             $p.closePopup(popId);
3918             func(com.data.getJSON(retStr));
3919         } else {
3920             var parentObj = opener || parent;
3921             if (window.$p.getParameter("w2xPath") !== parentObj.$p.getParameter("w2xPath")) {
3922                 com.popup._closePopup(popId, callbackFnStr, retStr, parentObj);
3923                 return;
3924             }
3925             $p.closePopup(popId);
3926         }
3927     } else {
3928         $p.closePopup(popId);
3929     }
3930 },
3931
3932 tmpopen : function()
3933 {
3934     $.magnificPopup.open(
3935     {
3936         type : 'inline',
3937         // items : { src : '/main.do?w2xPath=' + url + '&' + $.param(opt.param) },
3938         items : { src : "<div align='center' onclick='com.popup.tmpclose()'><img src='/images/n"
3939         markup : "<div class='mfp-iframe-scaler'><div class='mfp-close'></div><iframe class='mfp"
3940         modal : false,
3941         verticalFit : true,
3942         //alignTop : false,
3943         //fixedContentPos : false,
3944         //fixedBgPos : true,

```

```

3945 midClick : true,
3946 closeMarkup : "<button title='asdfsas' type='button' class='mfp-close popup-close-button'
3947 callbacks :
3948 {
3949     beforeOpen : function()
3950     {
3951         /*console.log('before open...');
3952         //WebSquare.util.dynamicCreate('wf' + opt.id, "wframe", { src:url + '?' + $.para
3953         WebSquare.util.dynamicCreate('wf' + opt.id, "wframe", { src:url }, WebSquare.getI
3954     },
3955     open : function()
3956     {
3957         //debugger;
3958         //WebSquare.util.getComponentById('wf' + opt.id).scope.scwin.param = {};
3959         //WebSquare.util.getComponentById('wf' + opt.id).scope.scwin.param = $p.top().wd
3960
3961         /*$('#wf' + opt.id).appendTo($('#div' + opt.id));
3962         /*$('#.mfp-content').css({'width':opt.width, 'height':opt.height, 'background-col
3963     },
3964     change : function()
3965     {
3966         // console.log('changed...');
3967     },
3968     beforeClose : function()
3969     {
3970         /*console.log('beforeClose...');
3971         if(callback !=null && callback != undefined) callback(WebSquare.util.getComponen
3972
3973         // 모든 Event들 Detach
3974         $('#.mfp-content').find('*').off();
3975         // 모든 Element 삭제
3976         $('#wf' + opt.id).empty();
3977         $('#div' + opt.id).empty();
3978         $('#.mfp-content').empty();
3979         WebSquare.util.getComponentById('wf' + opt.id).destroy();*/
3980     },
3981     close : function()
3982     {
3983         //console.log('close...');
3984     },
3985     resize : function()
3986     {
3987         //console.log('resize...');
3988     }
3989 }
3990 });
3991 },
3992 tmpclose : function()
3993 {
3994     $.magnificPopup.close();
3995 },
3996
3997 alert : function(messageStr, closeCallbackFncName)
3998 {
3999     com.popup.msgBox("alert", messageStr, closeCallbackFncName);
4000 },
4001
4002 confirm : function(messageStr, closeCallbackFncName)
4003 {
4004     com.popup.msgBox("confirm", messageStr, closeCallbackFncName);
4005 },
4006
4007 msgBox : function(messageType, messageStr, closeCallbackFncName, isReturnValue, title)
4008 {
4009     var messageStr = messageStr || "";
4010     var messageType = messageType || "alert";
4011     var defaultTitle = null;
4012     var popId = messageType || "Tmp";

```

```

4013
4014     popId = popId + com.MESSAGE_BOX_SEQ++;
4015
4016     if (messageType === "alert") {
4017         defaultTitle = "Alert";
4018     } else {
4019         defaultTitle = "Confirm";
4020     }
4021
4022     if (typeof isReturnValue === "undefined") {
4023         isReturnValue = false;
4024     }
4025
4026     // closeCallbackFnc 정보관리
4027     if (typeof closeCallbackFncName === "function") {
4028         var cbFuncIdx = ++gcm.CB_FUNCTION_MANAGER["cbFuncIdx"];
4029         var idx = "__close_callback_Func_" + new Date().getTime() + "_" + cbFuncIdx;
4030         gcm.CB_FUNCTION_MANAGER["cbFuncSave"][$p.id + idx] = closeCallbackFncName;
4031         closeCallbackFncName = idx;
4032     }
4033
4034     var data = {
4035         "message" : messageStr,
4036         "callbackFn" : closeCallbackFncName,
4037         "isReturnValue" : isReturnValue,
4038         "messageType" : defaultTitle
4039     };
4040     var options = {
4041         id : popId,
4042         popupName : defaultTitle,
4043         title : title || defaultTitle,
4044         width : 340,
4045         height : 242
4046     };
4047     com.popup.open(gcm.CONTEXT_PATH + "/ui/com/message_box.xml", options, data);
4048 },
4049
4050 /*resize : function(mutations)
4051 {
4052     console.log('s-----');
4053     console.log(mutations.length);
4054     mutations.forEach(function(mutation)
4055     {
4056         console.log(mutation.addedNodes[0]);
4057         if($(mutation.addedNodes[0]).hasClass('w2popup_window'))
4058         {
4059             console.log('..... : ' + $(mutation.addedNodes[0]));
4060         }
4061     });
4062     console.log('e-----');
4063 },*/
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079 simpleForm : function(divId, url, opt)
4080 {

```

```

4081 //popupData["userData"] = option;
4082
4083
4084 var button1 = $('<button/>', {id:'btn' + divId, name:'btn' + divId, text:'submit', css:{wid
4085 click : function(e)
4086 {
4087     var options = {
4088         id : opt.id,
4089         dataObject : {type:'json', name:'param', data:opt},
4090         type : 'wframePopup'/*,
4091         modal : true,
4092         alwaysOnTop : true,
4093         useModalStack : true,
4094         resizable : false,
4095         useMaximize : true,
4096         status : false,
4097         menubar : false,
4098         scrollbars : true*/
4099     };
4100     //$.extend(options, opt);
4101     $p.openPopup(url, options);
4102 });
4103 //$('#.w2window_selected_header').siblings('.w2window_body').find('#' + divId).append(button1
4104 $('#' + divId).data(opt);
4105 $('#' + divId).append(button1);
4106 },
4107 singleForm : function(formId, viewString, popupOption, extraInputId)
4108 {
4109     var inputId = "";
4110     if(com.data.isEmpty(formId) || com.data.isEmpty(viewString)) {
4111         return false;
4112     }
4113     inputId = formId + '_input';
4114
4115     var anchor = $('<a>', {'class':'w2anchor2', 'href':'javascript:void(0);'});
4116     $(anchor).click(function () {
4117         if(extraInputId) {
4118             $('#' + extraInputId).val('');
4119         }
4120         com.openCommonPopup(viewString, popupOption , inputId);
4121     });
4122     $('#' + $p.id + formId).addClass('input_search');
4123     WebSquare.util.dynamicCreate($p.id + inputId, 'input', {}, WebSquare.util.getComponentById($
4124     var element = WebSquare.util.getComponentById($p.id + inputId);
4125     element.bind('onkeyup', function (e) {
4126         if(e.keyCode === 13 && com.data.isNotEmpty(element.getValue())) {
4127             $(anchor).trigger('click');
4128         }
4129     });
4130
4131     $('#' + $p.id + formId).append(anchor);
4132     if(extraInputId) {
4133         var input2 = $('<input>', {'class':'w2input w2input_disabled', 'disabled':'true','type':
4134         $('#' + $p.id + formId).append(input2);
4135         $('#' + $p.id + formId).addClass('type01');
4136         popupOption.extraInput = extraInputId;
4137     }
4138 },
4139 multiForm : function(formId, viewString, popupOption, extraInputId)
4140 {
4141     var inputId = "";
4142     if(com.data.isEmpty(formId) || com.data.isEmpty(viewString)) {
4143         return false;
4144     }
4145
4146     inputId = formId + '_input';
4147
4148     var input = $('<input>', {'class':'w2input', 'type':'text', 'id' : inputId});

```

```

4149 var anchor = $("<a>", {'class': 'w2anchor2', 'href': 'javascript:void(0);'});
4150 $(input).keyup(function (e) {
4151     if(e.keyCode === 13 && com.data.isNotEmpty($(this).val())) {
4152         $(anchor).trigger('click');
4153     }
4154 });
4155 $(anchor).click(function () {
4156     com.openCommonPopup(viewString, popupOption, inputId);
4157 });
4158 $('# + $p.id + formId).addClass('input_search');
4159 $('# + $p.id + formId).append(input);
4160 $('# + $p.id + formId).append(anchor);
4161 if(extraInputId) {
4162     var input2 = $("<input>", {'class': 'w2input w2input_disabled', 'disabled': 'true', 'type':
4163     $('# + $p.id + formId).append(input2);
4164     $('# + $p.id + formId).addClass('type01');
4165     popupOption.extraInput = extraInputId;
4166 }
4167 }
4168 };
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180 //-----
4181 // File
4182 //-----
4183 var callmap = {};
4184 com.file = {
4185
4186     // formid : 어느곳에 위치, fileId : 파일ID, formId : 다른파일업로드폼과 구분하기 위한 숫자
4187     simpleForm : function(formId, fileId, fileSn, init, callback)
4188     {
4189         if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4190         {
4191             $('# + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('# + formId').empty();
4192         }
4193         else if($('# + formId')[0] != undefined)
4194         {
4195             $('# + formId').empty();
4196         }
4197
4198         var initid = 'init' + formId;
4199         var callid = 'callback' + formId;
4200         callmap[initid] = init;
4201         callmap[callid] = callback;
4202
4203         $.ajax(
4204         {
4205             url : '/file/simpleForm.do',
4206             type : 'get',
4207             async : false,
4208             data : 'formId=' + formId + '&fileId=' + fileId + '&fileSn=' + fileSn,
4209             success: function(data, textStatus, xhr)
4210             {
4211                 if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4212                 {
4213                     $('# + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('# + formId').append(d
4214                 }
4215                 else if($('# + formId')[0] != undefined)
4216                 {

```

```

4217         $('#' + formId).append(data);
4218     },
4219     },
4220     error: function(xhr, textStatus, errorThrown)
4221     {
4222         console.log(xhr.responseText);
4223     }
4224     });
4225 },
4226 sfSelectFile : function(formId, fileId, fileSn)
4227 {
4228     $.ajax({
4229         url : '/file/selectFile.do',
4230         type : 'post',
4231         async : false,
4232         contentType : 'application/x-www-form-urlencoded; charset=utf-8',
4233         data : 'fileId=' + fileId + '&fileSn=' + fileSn,
4234         success: function(data)
4235         {
4236             if(data.file != null)
4237             {
4238                 if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4239                 {
4240                     $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + form
4241                     $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + form
4242                 }
4243                 else if($('#' + formId)[0] != undefined)
4244                 {
4245                     $('#fileInfo' + formId).append("<span class='upload_file_txt' onClick=\"\");
4246                     $('#fileInfo' + formId).append("<button class='btn_file_del' onClick=\"");
4247                 }
4248             }
4249         }
4250     });
4251 },
4252 sfUploadFile : function(formId, fileId, fileSn)
4253 {
4254     $('#fileInfo' + fileId).empty();
4255
4256     var formData = new FormData();
4257     var files = null;
4258     if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4259     {
4260         files = $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileupload' + formId)
4261     }
4262     else if($('#' + formId)[0] != undefined)
4263     {
4264         files = $('#fileupload' + formId + ' input[type=file]');
4265     }
4266
4267     formData.append("file0", files[0].files[0]);
4268     formData.append('fileId', fileId);
4269     formData.append('fileSn', fileSn);
4270     $.ajax(
4271     {
4272         url: '/file/upload.do',
4273         type: 'POST',
4274         data: formData,
4275         cache: false,
4276         processData: false, // Don't process the files
4277         contentType: false, // Set content type to false as jQuery will tell the server its a qu
4278         success : function(data)
4279         {
4280             // 파일 업로드 처리가 끝나고 Callback수행 (사용자정의 Callback보다 먼저 실행.)
4281             if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4282             {
4283                 $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId)..
4284                 $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId)..

```

```

4285     }
4286     else if($('#' + formId)[0] != undefined)
4287     {
4288         $('#fileInfo' + formId).append("<span class='upload_file_txt' onClick=\"com.fi
4289         $('#fileInfo' + formId).append("<button class='btn_file_del'    onClick=\"com.fi
4290     }
4291
4292     // 사용자정의 Callback 실행
4293     callmap['callback' + formId](data);
4294 },
4295 error: function(xhr, textStatus, errorThrown)
4296 {
4297     //alert(JSON.parse(xhr.responseText)['error']);
4298     console.log(JSON.parse(xhr.responseText)['error']);
4299 }
4300 });
4301 },
4302 sfDeleteFile : function(formId, fileId, fileSn)
4303 {
4304     $.ajax({
4305         url : '/file/deleteFile.do?fileId=' + fileId + '&fileSn=' + fileSn,
4306         type : 'post',
4307         async : false,
4308         data : '',
4309         success: function(data, textStatus, xhr)
4310         {
4311             if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4312             {
4313                 $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId).
4314             }
4315             else if($('#' + formId)[0] != undefined)
4316             {
4317                 $('#fileInfo' + formId).empty();
4318             }
4319         },
4320         error: function(xhr, textStatus, errorThrown)
4321         {
4322             //alert(JSON.parse(xhr.responseText)['error']);
4323             console.log(xhr.responseText);
4324         }
4325     });
4326 },
4327 staticForm : function(formId, fileInfo, fileId, init, callback)
4328 {
4329     if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4330     {
4331         $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#' + formId).empty();
4332     }
4333     else if($('#' + formId)[0] != undefined)
4334     {
4335         $('#' + formId).empty();
4336     }
4337
4338     var initid = 'init' + formId;
4339     var callid = 'callback' + formId;
4340     callmap[initid] = init;
4341     callmap[callid] = callback;
4342     var readonly = $('#' + formId).data('readonly');
4343
4344     $.ajax(
4345     {
4346         url : '/file/staticForm.do',
4347         type : 'post',
4348         async : false,
4349         contentType : 'application/x-www-form-urlencoded; charset=utf-8',
4350         data : 'formId=' + formId + '&fileId=' + fileId + '&fileInfo=' + JSON.stringify(fileInfo
4351         success: function(data, textStatus, xhr)
4352         {

```

```

4353         if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4354         {
4355             $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#' + formId).append(d
4356         }
4357         else if($('##' + formId)[0] != undefined)
4358         {
4359             $('##' + formId).append(data);
4360         }
4361     },
4362     error: function(xhr, textStatus, errorThrown)
4363     {
4364         //alert(JSON.parse(xhr.responseText)['error']);
4365         console.log(xhr.responseText);
4366     }
4367     });
4368 },
4369 stSelectFileList : function(formId, fileId)
4370 {
4371     $.ajax({
4372         url : '/file/selectFileList.do',
4373         type : 'post',
4374         async : false,
4375         contentType : 'application/x-www-form-urlencoded; charset=utf-8',
4376         data : 'fileId=' + fileId,
4377         success: function(data)
4378         {
4379             $.each(data.fileList, function(idx, row)
4380             {
4381                 if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4382                 {
4383                     $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + form
4384                     $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + form
4385                 }
4386                 else if($('##' + formId)[0] != undefined)
4387                 {
4388                     $('#fileInfo' + formId + idx).append("<span class='upload_file_txt' onClick
4389                     $('#fileInfo' + formId + idx).append("<button class='btn_file_del' onClick
4390                 }
4391             });
4392         }
4393     });
4394 },
4395 stUploadFile : function(formId, fileId, fileSn)
4396 {
4397     var formData = new FormData();
4398     var files = $('#fileupload' + formId + ' input[type=file]');
4399     for(var i = 0; i < files.length; i++)
4400     {
4401         formData.append('file' + i, files[i].files[0]);
4402         formData.append('fileId', fileId);
4403         formData.append('fileDesc' + i, $('#fileupload' + formId + ' input[id=desc' + formId + i
4404     }
4405     $.ajax(
4406     {
4407         url: '/file/staticUpload.do',
4408         type: 'POST',
4409         data: formData,
4410         cache: false,
4411         processData: false, // Don't process the files
4412         contentType: false, // Set content type to false as jQuery will tell the server its a qu
4413         success : function(data)
4414         {
4415             // 파일 업로드 처리가 끝나고 Callback수행 (사용자정의 Callback보다 먼저 실행.)
4416             if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4417             {
4418                 $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId +
4419                 $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId +
4420             }

```

```

4421     else if($('#' + formId)[0] != undefined)
4422     {
4423         $('#fileInfo' + formId + fileSn).append("<span class='upload_file_txt' onClick="
4424         $('#fileInfo' + formId + fileSn).append("<button class='btn_file_del' onClick="
4425     }
4426
4427     // 사용자정의 Callback 실행
4428     callmap['callback' + formId](data);
4429 },
4430 error: function(xhr, textStatus, errorThrown)
4431 {
4432     //alert(JSON.parse(xhr.responseText)['error']);
4433     console.log(JSON.parse(xhr.responseText)['error']);
4434 }
4435 });
4436 },
4437 stDeleteFile : function(formId, fileId, fileSn)
4438 {
4439     $.ajax({
4440         url : '/file/deleteFile.do?fileId=' + fileId + '&fileSn=' + fileSn,
4441         type : 'post',
4442         async : false,
4443         data : '',
4444         success: function(data, textStatus, xhr)
4445         {
4446             if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4447             {
4448                 $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId).
4449             }
4450             else if($('#' + formId)[0] != undefined)
4451             {
4452                 $('#fileInfo' + formId).empty();
4453             }
4454         },
4455         error: function(xhr, textStatus, errorThrown)
4456         {
4457             //alert(JSON.parse(xhr.responseText)['error']);
4458             console.log(xhr.responseText);
4459         }
4460     });
4461 },
4462 multiForm : function(formId, fileId, init, callback)
4463 {
4464     if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4465     {
4466         $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#' + formId).empty();
4467     }
4468     else if($('#' + formId)[0] != undefined)
4469     {
4470         $('#' + formId).empty();
4471     }
4472
4473     var initid = 'init' + formId;
4474     var callid = 'callback' + formId;
4475     callmap[initid] = init;
4476     callmap[callid] = callback;
4477     var readonly = $('#' + formId).data('readonly');
4478
4479     $.ajax(
4480     {
4481         url : '/file/multiForm.do?formId=' + formId + '&fileId=' + fileId + '&readonly=' + reado
4482         type : 'get',
4483         async : false,
4484         data : '',
4485         success: function(data, textStatus, xhr)
4486         {
4487             if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4488             {

```

```

4489         $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#' + formId).append(d
4490     }
4491     else if($('##' + formId)[0] != undefined)
4492     {
4493         $('##' + formId).append(data);
4494     }
4495 },
4496 error: function(xhr, textStatus, errorThrown)
4497 {
4498     //alert(JSON.parse(xhr.responseText)['error']);
4499     console.log(xhr.responseText);
4500 }
4501 });
4502 },
4503 mfSelectFileList : function(formId, fileId)
4504 {
4505     $('#fileInfo' + formId).empty('');
4506     $.ajax({
4507         url : '/file/selectFileList.do?fileId=' + fileId,
4508         type : 'post',
4509         async : false,
4510         data : '',
4511         success: function(data, textStatus, xhr)
4512         {
4513             $.each(data.fileList, function(idx, row)
4514             {
4515                 if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4516                 {
4517                     $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + form
4518                         "<li>"
4519                         + " <ul>"
4520                         + "         <li id='dwn' + formId + idx + "'><a>" + row.streFileNm + "</
4521                         + "         <li id='fileMg' + formId + idx + "'>" + row.fileMg + "</li>"
4522                         + "         <li id='rgstrId' + formId + idx + "'>" + row.rgstrId + "</li>"
4523                         + "         <li id='rgstrDt' + formId + idx + "'>" + row.rgstrDt + "</li>"
4524                         + "         <li id='del' + formId + idx + "'><a class='g_btn_del'>삭제</a>"
4525                         + " </ul>"
4526                         + "</li>");
4527                 }
4528                 else if($('##' + formId)[0] != undefined)
4529                 {
4530                     $('#fileInfo' + formId).append(
4531                         "<li>"
4532                         + " <ul>"
4533                         + "         <li id='dwn' + formId + idx + "'><a>" + row.streFileNm + "</
4534                         + "         <li id='fileMg' + formId + idx + "'>" + row.fileMg + "</li>"
4535                         + "         <li id='rgstrId' + formId + idx + "'>" + row.rgstrId + "</li>"
4536                         + "         <li id='rgstrDt' + formId + idx + "'>" + row.rgstrDt + "</li>"
4537                         + "         <li id='del' + formId + idx + "'><a class='g_btn_del'>삭제</a>"
4538                         + " </ul>"
4539                         + "</li>");
4540                 }
4541             });
4542         },
4543         error: function(xhr, textStatus, errorThrown)
4544         {
4545             console.log(xhr.responseText);
4546         }
4547     });
4548 },
4549
4550
4551
4552
4553
4554
4555
4556

```

```

4557
4558
4559
4560
4561
4562
4563
4564 downloadFile : function(fileId, fileSn)
4565 {
4566     //location.href = '/file/fileDownload.do';
4567     if(fileId == null || fileId == "" || fileId == undefined || fileId == "undefined" ){
4568         alert("파일아이디가 없습니다.");
4569         return;
4570     }
4571
4572     if(fileSn == null || fileSn == "" || fileSn == undefined || fileSn == "undefined" ){
4573         alert("파일순번이 없습니다.");
4574         return;
4575     }
4576
4577     var downUrl = '/file/fileDownload.do';
4578
4579     if($(".body").length > 0){//임시 form submit 그리고 임시form 삭제
4580         //form submit 할 임시 form 및 attribute 생성
4581         $(".body").append("<form action='"+ downUrl +"' method='post' name='file_"+fileId+"_"+fileSn+"_'>");
4582         $(".form[name=file_"+fileId+"_"+fileSn+"_]_Frm").append("<input type='text' name='fileId' value='"+fileId+"'>");
4583         $(".form[name=file_"+fileId+"_"+fileSn+"_]_Frm").append("<input type='text' name='fileSn' value='"+fileSn+"'>");
4584         $(".form[name=file_"+fileId+"_"+fileSn+"_]_Frm").submit().remove();
4585     }else{
4586         window.open(downUrl+"?fileId="+fileId+"&fileSn="+fileSn, "file_dwon_form", "width=300, height=100");
4587     }
4588 },
4589
4590
4591 selectFile : function(formid, fileId, fileSn)
4592 {
4593     var oldtr = $("#"+ formid + " tr:eq('" + fileSn + "')");
4594     var newtr = oldtr.clone();
4595
4596     $.ajax({
4597         url : '/file/selectFile.do?fileId=' + fileId + '&fileSn=' + fileSn,
4598         type : 'post',
4599         async : false,
4600         data : '',
4601         success: function(data, textStatus, xhr)
4602         {
4603             newtr.find('.rgstrDt').text(data.file != null ? data.file.rgstrDt : '');
4604             newtr.find('.strFilNm').text(data.file != null ? data.file.strFilNm : '');
4605             newtr.find('.mgr').empty();
4606             if(data.file != null)
4607             {
4608                 newtr.find('.mgr').append("<button/>", {'text':'보기', 'class':'g_btn_doc', 'click':function(){com.file.selectFile(formid, fileId, fileSn);}});
4609                 newtr.find('.mgr').append("<button/>", {'text':'삭제', 'class':'g_btn_del', 'click':function(){com.file.deleteFile(formid, fileId, fileSn);}});
4610                 {
4611                     $.ajax(
4612                     {
4613                         url : '/file/deleteFile.do?fileId=' + fileId + '&fileSn=' + fileSn,
4614                         type : 'post',
4615                         async : false,
4616                         data : '',
4617                         success: function(data, textStatus, xhr)
4618                         {
4619                             com.file.selectFile(formid, fileId, fileSn);
4620                         },
4621                         error: function(xhr, textStatus, errorThrown)
4622                         {
4623                             //alert(JSON.parse(xhr.responseText)['error']);
4624                             console.log(xhr.responseText);

```

```

4625         }
4626     });
4627
4628     }));
4629 }
4630 oldtr.replaceWith(newtr);
4631 }
4632 });
4633 },
4634 selectFile1 : function(formid, fileId, fileSn)
4635 {
4636     $.ajax({
4637         url : '/file/selectFile.do?fileId=' + fileId + '&fileSn=' + fileSn,
4638         type : 'post',
4639         async : false,
4640         data : '',
4641         success: function(data, textStatus, xhr)
4642         {
4643             $('#filCn' + formid).val(data.file.strFilNm);
4644             ///file/fileDownload.do?fileId="" + data.file.fileId + "fileSn=" + data.file.fileSn
4645         }
4646     });
4647 }
4648 });
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665 var callmap = {};
4666 com.file = {
4667
4668     // formid : 어느곳에 위치, fileId : 파일ID, formId : 다른파일업로드폼과 구분하기 위한 숫자
4669     simpleForm : function(formId, fileId, fileSn, init, callback)
4670     {
4671         if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4672         {
4673             $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#' + formId).empty();
4674         }
4675         else if($('##' + formId)[0] != undefined)
4676         {
4677             $('##' + formId).empty();
4678         }
4679
4680         var initid = 'init' + formId;
4681         var callid = 'callback' + formId;
4682         callmap[initid] = init;
4683         callmap[callid] = callback;
4684
4685         $.ajax(
4686         {
4687             url : '/file/simpleForm.do',
4688             type : 'get',
4689             async : false,
4690             data : 'formId=' + formId + '&fileId=' + fileId + '&fileSn=' + fileSn,
4691             success: function(data, textStatus, xhr)
4692             {

```

```

4693         if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4694         {
4695             $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#' + formId).append(d
4696         }
4697         else if($('##' + formId)[0] != undefined)
4698         {
4699             $('##' + formId).append(data);
4700         }
4701     },
4702     error: function(xhr, textStatus, errorThrown)
4703     {
4704         console.log(xhr.responseText);
4705     }
4706 });
4707 },
4708 sfSelectFile : function(formId, fileId, fileSn)
4709 {
4710     $.ajax({
4711         url : '/file/selectFile.do',
4712         type : 'post',
4713         async : false,
4714         contentType : 'application/x-www-form-urlencoded; charset=utf-8',
4715         data : 'fileId=' + fileId + '&fileSn=' + fileSn,
4716         success: function(data)
4717         {
4718             if(data.file != null)
4719             {
4720                 if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4721                 {
4722                     $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + form
4723                     $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + form
4724                 }
4725                 else if($('##' + formId)[0] != undefined)
4726                 {
4727                     $('#fileInfo' + formId).append("<span class='upload_file_txt' onClick=\"\
4728                     $('#fileInfo' + formId).append("<button class='btn_file_del' onClick=\"\
4729                 }
4730             }
4731         }
4732     });
4733 },
4734 sfUploadFile : function(formId, fileId, fileSn)
4735 {
4736     $('#fileInfo' + fileId).empty();
4737
4738     var formData = new FormData();
4739     var files = null;
4740     if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4741     {
4742         files = $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileupload' + formId
4743     }
4744     else if($('##' + formId)[0] != undefined)
4745     {
4746         files = $('#fileupload' + formId + ' input[type=file]');
4747     }
4748
4749     formData.append("file0", files[0].files[0]);
4750     formData.append('fileId', fileId);
4751     formData.append('fileSn', fileSn);
4752     $.ajax(
4753     {
4754         url: '/file/upload.do',
4755         type: 'POST',
4756         data: formData,
4757         cache: false,
4758         processData: false, // Don't process the files
4759         contentType: false, // Set content type to false as jQuery will tell the server its a qu
4760         success : function(data)

```

```

4761     {
4762         // 파일 업로드 처리가 끝나고 Callback수행 (사용자정의 Callback보다 먼저 실행.)
4763         if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4764         {
4765             $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId)..
4766             $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId)..
4767         }
4768         else if($('#' + formId)[0] != undefined)
4769         {
4770             $('#fileInfo' + formId).append("<span class='upload_file_txt' onClick=\"com.fi
4771             $('#fileInfo' + formId).append("<button class='btn_file_del'      onClick=\"com.fi
4772         }
4773
4774         // 사용자정의 Callback 실행
4775         callmap['callback' + formId](data);
4776     },
4777     error: function(xhr, textStatus, errorThrown)
4778     {
4779         //alert(JSON.parse(xhr.responseText)['error']);
4780         console.log(JSON.parse(xhr.responseText)['error']);
4781     }
4782 });
4783 },
4784 sfDeleteFile : function(formId, fileId, fileSn)
4785 {
4786     $.ajax({
4787         url : '/file/deleteFile.do?fileId=' + fileId + '&fileSn=' + fileSn,
4788         type : 'post',
4789         async : false,
4790         data : '',
4791         success: function(data, textStatus, xhr)
4792         {
4793             if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4794             {
4795                 $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId)..
4796             }
4797             else if($('#' + formId)[0] != undefined)
4798             {
4799                 $('#fileInfo' + formId).empty();
4800             }
4801         },
4802         error: function(xhr, textStatus, errorThrown)
4803         {
4804             //alert(JSON.parse(xhr.responseText)['error']);
4805             console.log(xhr.responseText);
4806         }
4807     });
4808 },
4809 staticForm : function(formId, fileInfo, fileId, init, callback)
4810 {
4811     if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4812     {
4813         $('#' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#' + formId).empty();
4814     }
4815     else if($('#' + formId)[0] != undefined)
4816     {
4817         $('#' + formId).empty();
4818     }
4819
4820     var initid = 'init' + formId;
4821     var callid = 'callback' + formId;
4822     callmap[initid] = init;
4823     callmap[callid] = callback;
4824     var readonly = $('#' + formId).data('readonly');
4825
4826     $.ajax(
4827     {
4828         url : '/file/staticForm.do',

```

```

4829     type : 'post',
4830     async : false,
4831     contentType : 'application/x-www-form-urlencoded; charset=utf-8',
4832     data : 'formId=' + formId + '&fileId=' + fileId + '&fileInfo=' + JSON.stringify(fileInfo)
4833     success: function(data, textStatus, xhr)
4834     {
4835         if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4836         {
4837             $('#'+ $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#' + formId).append(d
4838         }
4839     else if($('#'+ formId)[0] != undefined)
4840     {
4841         $('#'+ formId).append(data);
4842     }
4843     },
4844     error: function(xhr, textStatus, errorThrown)
4845     {
4846         //alert(JSON.parse(xhr.responseText)['error']);
4847         console.log(xhr.responseText);
4848     }
4849     });
4850 },
4851 stSelectFileList : function(formId, fileId)
4852 {
4853     $.ajax({
4854         url : '/file/selectFileList.do',
4855         type : 'post',
4856         async : false,
4857         contentType : 'application/x-www-form-urlencoded; charset=utf-8',
4858         data : 'fileId=' + fileId,
4859         success: function(data)
4860         {
4861             $.each(data.fileList, function(idx, row)
4862             {
4863                 if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4864                 {
4865                     $('#'+ $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + form
4866                     $('#'+ $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + form
4867                 }
4868             else if($('#'+ formId)[0] != undefined)
4869             {
4870                 $('#fileInfo' + formId + idx).append("<span class='upload_file_txt' onClic
4871                 $('#fileInfo' + formId + idx).append("<button class='btn_file_del' onClic
4872             }
4873             });
4874         }
4875     });
4876 },
4877 stUploadFile : function(formId, fileId, fileSn)
4878 {
4879     var formData = new FormData();
4880     var files = $('#fileupload' + formId + ' input[type=file]');
4881     for(var i = 0; i < files.length; i++)
4882     {
4883         formData.append('file' + i, files[i].files[0]);
4884         formData.append('fileId', fileId);
4885         formData.append('fileDesc' + i, $('#fileupload' + formId + ' input[id=desc' + formId + i
4886     }
4887     $.ajax(
4888     {
4889         url: '/file/staticUpload.do',
4890         type: 'POST',
4891         data: formData,
4892         cache: false,
4893         processData: false, // Don't process the files
4894         contentType: false, // Set content type to false as jQuery will tell the server its a qu
4895         success : function(data)
4896         {

```

```

4897 // 파일 업로드 처리가 끝나고 Callback수행 (사용자정의 Callback보다 먼저 실행.)
4898 if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4899 {
4900     $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId +
4901     $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId +
4902     }
4903 else if($('##' + formId)[0] != undefined)
4904 {
4905     $('#fileInfo' + formId + fileSn).append("<span class='upload_file_txt' onClick="
4906     $('#fileInfo' + formId + fileSn).append("<button class='btn_file_del' onClick="
4907     }
4908
4909 // 사용자정의 Callback 실행
4910 callmap['callback' + formId](data);
4911 },
4912 error: function(xhr, textStatus, errorThrown)
4913 {
4914     //alert(JSON.parse(xhr.responseText)['error']);
4915     console.log(JSON.parse(xhr.responseText)['error']);
4916 }
4917 });
4918 },
4919 stDeleteFile : function(formId, fileId, fileSn)
4920 {
4921     $.ajax({
4922         url : '/file/deleteFile.do?fileId=' + fileId + '&fileSn=' + fileSn,
4923         type : 'post',
4924         async : false,
4925         data : '',
4926         success: function(data, textStatus, xhr)
4927         {
4928             if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4929             {
4930                 $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId).
4931             }
4932             else if($('##' + formId)[0] != undefined)
4933             {
4934                 $('#fileInfo' + formId).empty();
4935             }
4936         },
4937         error: function(xhr, textStatus, errorThrown)
4938         {
4939             //alert(JSON.parse(xhr.responseText)['error']);
4940             console.log(xhr.responseText);
4941         }
4942     });
4943 },
4944 multiForm : function(formId, fileId, init, callback)
4945 {
4946     if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4947     {
4948         $('##' + $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#' + formId).empty();
4949     }
4950     else if($('##' + formId)[0] != undefined)
4951     {
4952         $('##' + formId).empty();
4953     }
4954
4955     var initid = 'init' + formId;
4956     var callid = 'callback' + formId;
4957     callmap[initid] = init;
4958     callmap[callid] = callback;
4959     var readonly = $('##' + formId).data('readonly');
4960
4961     $.ajax(
4962     {
4963         url : '/file/multiForm.do?formId=' + formId + '&fileId=' + fileId + '&readonly=' + reado
4964         type : 'get',

```

```

4965     async : false,
4966     data : '',
4967     success: function(data, textStatus, xhr)
4968     {
4969         if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4970         {
4971             $('#'+ $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#' + formId).append(d);
4972         }
4973         else if($('#'+ formId)[0] != undefined)
4974         {
4975             $('#'+ formId).append(data);
4976         }
4977     },
4978     error: function(xhr, textStatus, errorThrown)
4979     {
4980         //alert(JSON.parse(xhr.responseText)['error']);
4981         console.log(xhr.responseText);
4982     }
4983     });
4984 },
4985 mfSelectFileList : function(formId, fileId)
4986 {
4987     $('#fileInfo' + formId).empty('');
4988     $.ajax({
4989         url : '/file/selectFileList.do?fileId=' + fileId,
4990         type : 'post',
4991         async : false,
4992         data : '',
4993         success: function(data, textStatus, xhr)
4994         {
4995             $.each(data.fileList, function(idx, row)
4996             {
4997                 if($p.top().wdcMain != null && $p.top().wdcMain.getFrame(gcm['menuId']) != null)
4998                 {
4999                     $('#'+ $p.top().wdcMain.getFrame(gcm['menuId']).id).find('#fileInfo' + formId)
5000                     + "<li>"
5001                     + " <ul>"
5002                     + " <li id='dwn' + formId + idx + "'><a>" + row.streFileNm + "</li>"
5003                     + " <li id='fileMg' + formId + idx + "'>" + row.fileMg + "</li>"
5004                     + " <li id='rgstrId' + formId + idx + "'>" + row.rgstrId + "</li>"
5005                     + " <li id='rgstrDt' + formId + idx + "'>" + row.rgstrDt + "</li>"
5006                     + " <li id='del' + formId + idx + "'><a class='g_btn_del'>삭제</a>"
5007                     + " </ul>"
5008                     + "</li>");
5009                 }
5010                 else if($('#'+ formId)[0] != undefined)
5011                 {
5012                     $('#fileInfo' + formId).append(
5013                     + "<li>"
5014                     + " <ul>"
5015                     + " <li id='dwn' + formId + idx + "'><a>" + row.streFileNm + "</li>"
5016                     + " <li id='fileMg' + formId + idx + "'>" + row.fileMg + "</li>"
5017                     + " <li id='rgstrId' + formId + idx + "'>" + row.rgstrId + "</li>"
5018                     + " <li id='rgstrDt' + formId + idx + "'>" + row.rgstrDt + "</li>"
5019                     + " <li id='del' + formId + idx + "'><a class='g_btn_del'>삭제</a>"
5020                     + " </ul>"
5021                     + "</li>");
5022                 }
5023             });
5024         },
5025         error: function(xhr, textStatus, errorThrown)
5026         {
5027             console.log(xhr.responseText);
5028         }
5029     });
5030 },
5031
5032

```

```

5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046 downloadFile : function(fileId, fileSn)
5047 {
5048     //location.href = '/file/fileDownload.do';
5049     if(fileId == null || fileId == "" || fileId == undefined || fileId == "undefined" ){
5050         alert("파일아이디가 없습니다.");
5051         return;
5052     }
5053
5054     if(fileSn == null || fileSn == "" || fileSn == undefined || fileSn == "undefined" ){
5055         alert("파일순번이 없습니다.");
5056         return;
5057     }
5058
5059     var downUrl = '/file/fileDownload.do';
5060
5061     if($(".body").length > 0){//임시 form submit 그리고 임시form 삭제
5062         //form submit 할 임시 form 및 attribute 생성
5063         $(".body").append("<form action='"+ downUrl +"' method='post' name='file_"+fileId+"_"+fileSn+"'>");
5064         $(".form[name=file_"+fileId+"_"+fileSn+"_Frm]").append("<input type='text' name='fileId' value='"+fileId+"'>");
5065         $(".form[name=file_"+fileId+"_"+fileSn+"_Frm]").append("<input type='text' name='fileSn' value='"+fileSn+"'>");
5066         $(".form[name=file_"+fileId+"_"+fileSn+"_Frm]").submit().remove();
5067     }else{
5068         window.open(downUrl+"?fileId="+fileId+"&fileSn="+fileSn, "file_dwon_form", "width=300, height=100");
5069     }
5070 },
5071
5072
5073 selectFile : function(formid, fileId, fileSn)
5074 {
5075     var oldtr = $("#"+ formid + " tr:eq('" + fileSn + "')");
5076     var newtr = oldtr.clone();
5077
5078     $.ajax({
5079         url : '/file/selectFile.do?fileId=' + fileId + '&fileSn=' + fileSn,
5080         type : 'post',
5081         async : false,
5082         data : '',
5083         success: function(data, textStatus, xhr)
5084         {
5085             newtr.find('.rgstrDt').text(data.file != null ? data.file.rgstrDt : '');
5086             newtr.find('.strFilNm').text(data.file != null ? data.file.strFilNm : '');
5087             newtr.find('.mgr').empty();
5088             if(data.file != null)
5089             {
5090                 newtr.find('.mgr').append("<button/>", {'text':'보기', 'class':'g_btn_doc', 'click':function(){selectFile(formid, fileId, fileSn);}});
5091                 newtr.find('.mgr').append("<button/>", {'text':'삭제', 'class':'g_btn_del', 'click':function(){deleteFile(formid, fileId, fileSn);}});
5092             }
5093             $.ajax(
5094             {
5095                 url : '/file/deleteFile.do?fileId=' + fileId + '&fileSn=' + fileSn,
5096                 type : 'post',
5097                 async : false,
5098                 data : '',
5099                 success: function(data, textStatus, xhr)
5100                 {

```

```

5101         com.file.selectFile(formid, fileId, fileSn);
5102     },
5103     error: function(xhr, textStatus, errorThrown)
5104     {
5105         //alert(JSON.parse(xhr.responseText)['error']);
5106         console.log(xhr.responseText);
5107     }
5108     });
5109
5110     }
5111     }
5112     oldtr.replaceWith(newtr);
5113 }
5114 });
5115 },
5116 selectFile1 : function(formid, fileId, fileSn)
5117 {
5118     $.ajax({
5119         url : '/file/selectFile.do?fileId=' + fileId + '&fileSn=' + fileSn,
5120         type : 'post',
5121         async : false,
5122         data : '',
5123         success: function(data, textStatus, xhr)
5124         {
5125             $('#filCn' + formid).val(data.file.strFilNm);
5126             ///file/fileDownload.do?fileId="" + data.file.fileId + "fileSn=" + data.file.fileSn
5127         }
5128     });
5129 }
5130 });
5131
5132 //-----
5133 // Session
5134 //-----
5135 com.session = {
5136     userDetails : function()
5137     {
5138         return top.gcm['userdetails'];
5139     },
5140     hasRedAuth : function()
5141     {
5142         return (JSON.parse(sessionStorage.getItem('menuList')).filter(function(obj){ return (obj.men
5143     },
5144     hasUldAuth : function()
5145     {
5146         return (JSON.parse(sessionStorage.getItem('menuList')).filter(function(obj){ return (obj.men
5147     },
5148     hasDwnAuth : function()
5149     {
5150         return (JSON.parse(sessionStorage.getItem('menuList')).filter(function(obj){ return (obj.men
5151     },
5152     hasPrtAuth : function()
5153     {
5154         return (JSON.parse(sessionStorage.getItem('menuList')).filter(function(obj){ return (obj.men
5155     },
5156     hasSavAuth : function()
5157     {
5158         return (JSON.parse(sessionStorage.getItem('menuList')).filter(function(obj){ return (obj.men
5159     },
5160     hasDelAuth : function()
5161     {
5162         return (JSON.parse(sessionStorage.getItem('menuList')).filter(function(obj){ return (obj.men
5163     },
5164     hasBizAuth : function()
5165     {
5166         return (JSON.parse(sessionStorage.getItem('menuList')).filter(function(obj){ return (obj.men
5167     },
5168     check : function()

```

```

5169 {
5170     $.ajax({
5171         url : '/sctest.do',
5172         type : 'get',
5173         async : false,
5174         dataType: 'text',
5175         success : function(data)
5176         {
5177             if(data != 0)
5178             {
5179                 alert('세션종료');
5180                 sessionStorage.clear();
5181                 location.href='/websquare.html?w2XPath=/ui/com/index.xml';
5182             }
5183         },
5184         error: function(xhr, textStatus, errorThrown)
5185         {
5186             com.error(xhr.responseText);
5187         }
5188     });
5189 },
5190 adjustAuthority : function()
5191 {
5192     console.log('.....read auth : ' + com.session.hasRedAuth());
5193     console.log('.....upload auth : ' + com.session.hasUldAuth());
5194     console.log('.....download auth : ' + com.session.hasDwnAuth());
5195     console.log('.....print auth : ' + com.session.hasPrtAuth());
5196     console.log('.....save auth : ' + com.session.hasSavAuth());
5197     console.log('.....delete auth : ' + com.session.hasDelAuth());
5198     console.log('.....bisuness auth : ' + com.session.hasBizAuth());
5199
5200     if(com.session.hasRedAuth())
5201     {
5202         $('#'+ $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnSearch]').css(
5203             $('#'+ $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnInit]').css('d
5204         )
5205     }
5206     else
5207     {
5208         $('#'+ $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnSearch]').css(
5209             $('#'+ $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnInit]').css('d
5210     }
5211
5212     if(com.session.hasUldAuth())
5213     {
5214         $('#'+ $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnUpload]').css(
5215     }
5216     else
5217     {
5218         $('#'+ $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnUpload]').css(
5219     }
5220
5221     if(com.session.hasDwnAuth())
5222     {
5223         $('#'+ $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnDownload]').cs
5224     }
5225     else
5226     {
5227         $('#'+ $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnDownload]').cs
5228     }
5229
5230     if(com.session.hasPrtAuth())
5231     {
5232         $('#'+ $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnPrint]').css('
5233     }
5234     else
5235     {
5236         $('#'+ $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnPrint]').css('

```

```

5237
5238     if(com.session.hasSavAuth())
5239     {
5240         $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnSave]').css('d
5241     }
5242     else
5243     {
5244         $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnSave]').css('d
5245     }
5246
5247     if(com.session.hasDelAuth())
5248     {
5249         $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnDelete]').css(
5250     }
5251     else
5252     {
5253         $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnDelete]').css(
5254     }
5255
5256     if(com.session.hasBizAuth())
5257     {
5258         let bizBtns = JSON.parse(sessionStorage.getItem('menuList')).filter(function(obj){ return
5259         for(var i = 0; i < bizBtns.length; i++)
5260         {
5261             $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnUserDef0'
5262             $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnUserDef0'
5263         }
5264         for(var i = bizBtns.length; i < 5; i++)
5265         {
5266             $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnUserDef0'
5267         }
5268     }
5269     else
5270     {
5271         $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnUserDef01]').c
5272         $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnUserDef02]').c
5273         $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnUserDef03]').c
5274         $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnUserDef04]').c
5275         $('#' + $p.top().wdcMain.getFrame(top.gcm['menuId']).id).find('div[id$=btnUserDef05]').c
5276     }
5277 }
5278 });
5279
5280 //-----
5281 // Report
5282 //-----
5283 com.report =
5284 {
5285     oditestform : function(divId, opt)
5286     {
5287         // 버튼 처리
5288         var button1 = ($('<button/>', {id:'btn' + divId, name:'btn' + divId, text:'submit', css:{wid
5289         click:function(e)
5290         {
5291             $('#' + divId + 'form')[0].target='ifr' + divId;
5292             $('#' + divId + 'form')[0].submit();
5293         }
5294     });
5295     $('#.w2window_selected_header').siblings('.w2window_body').find('#' + divId).append(button1);
5296
5297     // 버튼 클릭시 Submit할 Form
5298     var iframe1 = ($('<iframe/>', {id:'ifr' + divId, name:'ifr' + divId, css:{width:'500px', hei
5299     var form1 = ($('<form/>', {id:divId + 'form', name:divId + 'form', action:'http://11.11.1
5300     var input1 = ($('<input/>', {id:'filename', name:'filename', value:opt.filename, css:{disp
5301     var input2 = ($('<input/>', {id:'url', name:'url', value:opt.url, css:{display:'
5302     var input3 = ($('<input/>', {id:'param1', name:'param1', value:opt.param1, css:{displa
5303     $(input1).appendTo(form1);
5304     $(input2).appendTo(form1);
5305     $(input3).appendTo(form1);

```

```

5305     $('w2window_selected_header').siblings('w2window_body').find('#' + divId).append(iframe1)..
5306 },
5307 jsonestform : function(divId, opt)
5308 {
5309     // 버튼 처리
5310     var button1 = ($('<button/>', {id:'btn' + divId, name:'btn' + divId, text:'submit', css:{wid
5311         click:function(e)
5312         {
5313             $(''#' + divId + 'form')[0].target='ifr' + divId;
5314             $(''#' + divId + 'form')[0].submit();
5315         }
5316     }));
5317     $('w2window_selected_header').siblings('w2window_body').find('#' + divId).append(button1);
5318     // 버튼 클릭시 Suubmit할 Form
5319     var iframe1 = ($('<iframe/>', {id:'ifr' + divId, name:'ifr' + divId, css:{width:'500px', hei
5320     var form1 = ($('<form/>', {id:divId + 'form', name:divId + 'form', action:'http://11.11.1
5321     var input1 = ($('<input/>', {id:'filename', name:'filename', value:opt.filename, css:{disp
5322     var input2 = ($('<input/>', {id:'url', name:'url', value:opt.url, css:{display:'
5323     var input3 = ($('<input/>', {id:'param1', name:'param1', value:opt.param1, css:{displa
5324     $(input1).appendTo(form1);
5325     $(input2).appendTo(form1);
5326     $(input3).appendTo(form1);
5327     $('w2window_selected_header').siblings('w2window_body').find('#' + divId).append(iframe1)..
5328 },
5329 odilayerform : function(divId, opt)
5330 {
5331     // 버튼 처리
5332     var button1 = ($('<button/>', {id:'btn' + divId, name:'btn' + divId, text:'submit', css:{wid
5333         click:function(e)
5334         {
5335             opt.divId = divId;
5336             com.popup.open('/ui/sys/sol/oz/odiform.xml', opt);
5337         }
5338     }));
5339     $(''#' + divId).append(button1);
5340     // 버튼 클릭시 Suubmit할 Form
5341     var form1 = ($('<form/>', {id:divId + 'form', name:divId + 'form', action:'http://11.11.1
5342     var input1 = ($('<input/>', {id:'filename', name:'filename', value:opt.filename, css:{disp
5343     var input2 = ($('<input/>', {id:'url', name:'url', value:opt.url, css:{display:'
5344     var input3 = ($('<input/>', {id:'param1', name:'param1', value:opt.param1, css:{displa
5345     $(input1).appendTo(form1);
5346     $(input2).appendTo(form1);
5347     $(input3).appendTo(form1);
5348     $(''#' + divId).append(form1);
5349 },
5350 jsonlayerform : function(divId, opt)
5351 {
5352     // 버튼 처리
5353     var button1 = ($('<button/>', {id:'btn' + divId, name:'btn' + divId, text:'submit', css:{wid
5354         click:function(e)
5355         {
5356             opt.divId = divId;
5357             com.popup.open('/ui/sys/sol/oz/jsonform.xml', opt);
5358         }
5359     }));
5360     $(''#' + divId).append(button1);
5361     // 버튼 클릭시 Suubmit할 Form
5362     var form1 = ($('<form/>', {id:divId + 'form', name:divId + 'form', action:'http://11.11.1
5363     var input1 = ($('<input/>', {id:'filename', name:'filename', value:opt.filename, css:{disp
5364     var input2 = ($('<input/>', {id:'url', name:'url', value:opt.url, css:{display:'
5365     var input3 = ($('<input/>', {id:'param1', name:'param1', value:opt.param1, css:{displa
5366     $(input1).appendTo(form1);
5367     $(input2).appendTo(form1);
5368     $(input3).appendTo(form1);
5369     $(''#' + divId).append(form1);
5370 },
5371 odiForm : function(divId, opt)
5372 {

```

```

5373 // 버튼 처리
5374 var button1 = ($('<button/>', {id:'btn' + divId, name:'btn' + divId, text:'submit', css:{wid
5375 click:function(e)
5376 {
5377     var width = 500;
5378     var height = 500;
5379     const post = { x:(screen.width/2) - (width/2), y:(screen
5380     const features = 'width=${width} height=${height} left=$
5381
5382     window.open('about:blank', 'reportpopup', features);
5383     $('# + divId + 'form')[0].target = 'reportpopup';
5384     $('# + divId + 'form')[0].submit();
5385     }));
5386     $('.w2window_selected_header').siblings('.w2window_body').find('# + divId).append(button1);
5387
5388 // 버튼 클릭시 Suubmit할 Form
5389 var form1 = ($("<form/>", {id:divId + 'form', name:divId + 'form', action:'http://11.11.1
5390 var input1 = ($("<input/>", {id:'filename', name:'filename', value:opt.filename, css:{disp
5391 var input2 = ($("<input/>", {id:'url', name:'url', value:opt.url, css:{display:'
5392 var input3 = ($("<input/>", {id:'param1', name:'param1', value:opt.param1, css:{displa
5393 $(input1).appendTo(form1);
5394 $(input2).appendTo(form1);
5395 $(input3).appendTo(form1);
5396 $('.w2window_selected_header').siblings('.w2window_body').find('# + divId).append(form1);
5397 },
5398 jsonForm : function(divId, opt)
5399 {
5400     // 버튼 처리
5401     var button1 = ($('<button/>', {id:'btn' + divId, name:'btn' + divId, text:'submit', css:{wid
5402     click:function(e)
5403     {
5404         var width = 500;
5405         var height = 500;
5406         const post = { x:(screen.width/2) - (width/2), y:(screen
5407         const features = 'width=${width} height=${height} left=$
5408
5409         window.open('about:blank', 'reportpopup', features);
5410         $('# + divId + 'form')[0].target = 'reportpopup';
5411         $('# + divId + 'form')[0].submit();
5412         }));
5413         $('.w2window_selected_header').siblings('.w2window_body').find('# + divId).append(button1);
5414
5415 // 버튼 클릭시 Suubmit할 Form
5416 var form1 = ($("<form/>", {id:divId + 'form', name:divId + 'form', action:'http://11.11.1
5417 var input1 = ($("<input/>", {id:'filename', name:'filename', value:opt.filename, css:{disp
5418 var input2 = ($("<input/>", {id:'url', name:'url', value:opt.url, css:{display:'
5419 var input3 = ($("<input/>", {id:'param1', name:'param1', value:opt.param1, css:{displa
5420 $(input1).appendTo(form1);
5421 $(input2).appendTo(form1);
5422 $(input3).appendTo(form1);
5423 $('.w2window_selected_header').siblings('.w2window_body').find('# + divId).append(form1);
5424     }
5425 };
5426
5427 //-----
5428 // 기타
5429 //-----
5430 /**
5431 * 파라미터를 읽어 온다.
5432 *
5433 * @date 2019.03.11
5434 * @memberOf com
5435 * @param {String} 파라미터 키
5436 * @author InswaveSystems
5437 * @return {Object} 파라미터 값
5438 * @example var code = com.getParameter("code"); // 특정 파라미터 값을 얻어오기 var param = com.getParameter
5439 */
5440 com.getParameter = function(paramKey) {

```

```

5441     try {
5442         if (typeof paramKey !== "undefined") {
5443             var param = $p.getParameter(paramKey);
5444             if ((typeof param !== "undefined") && (param !== "")) {
5445                 return param;
5446             }
5447         };
5448
5449         var param = $p.getParameter("param");
5450         if ((typeof param !== "undefined") && (param !== "")) {
5451             return param[paramKey];
5452         }
5453
5454         return param;
5455     } else {
5456         var param = $p.getParameter("param");
5457         return com.data.getJSON(param);
5458     }
5459 } catch (ex) {
5460     return "";
5461 }
5462 };
5463
5464 /**
5465  * contextRoot가 포함된 path를 반환한다.
5466  *
5467  * @date 2016.11.16
5468  * @memberOf com
5469  * @param {String} path 파일경로(Context가 포함되지 않은)
5470  * @return {String} Context가 포함된 파일경로
5471  * @example
5472  * // context가 /sample 인경우
5473  * com.getFullPath("/ui/SP/Parameter/confirm.xml");
5474  * // return 예시) "/sample/ui/SP/Parameter/confirm.xml"
5475  */
5476 com.getFullPath = function(path) {
5477     var rtn_path = "";
5478     if (com.CONTEXT_PATH == "") {
5479         rtn_path = path;
5480     } else {
5481         rtn_path = com.CONTEXT_PATH + path;
5482     }
5483     return rtn_path;
5484 };
5485
5486 /**
5487  * 다국어 처리함수
5488  *
5489  * @date 2016.08.02
5490  * @private
5491  * @param {String} xmlUrl 전체 URL중 w2xPath이하의 경로
5492  * @memberOf
5493  * @author InswaveSystems
5494  * @example
5495  * com.getI18NUrl( "/ui/DEV/result.xml" );
5496  * //return 예시)"/websquare/I18N?w2xPath=/ui/DEV/result.xml"
5497  * com._getI18NUrl( "/ui/SW/request.xml" );
5498  * //return 예시)"/websquare/I18N?w2xPath=/ui/SW/request.xml"
5499  */
5500
5501 com._getI18NUrl = function(xmlUrl) {
5502     var baseUrl = com.CONTEXT_PATH + "/I18N"; // org"/websquare/engine/servlet/I18N.jsp?w2xPath=";
5503     var rsUrl;
5504     var locale = WebSquare.cookie.getCookie("locale");
5505     var bXml = "/blank.xml";
5506     xmlUrl = xmlUrl.replace(/\?.*/ , '');
5507     xmlUrl = xmlUrl.replace(gcm.CONTEXT_PATH, '');
5508

```

```

5509     if (xmlUrl.search(bXml) > -1 && xmlUrl.search(WebSquare.baseURI) == -1) {
5510         xmlUrl = WebSquare.baseURI + "/blank.xml";
5511     }
5512     rsURL = baseUrl + "?w2xPath=" + xmlUrl;
5513
5514     if (locale != null && locale != '') {
5515         rsURL = rsURL + "&locale=" + unescape(locale);
5516     }
5517
5518     return rsURL;
5519 };
5520
5521 com.startBizLayer = function()
5522 {
5523     $p.top().wdcWelcome.hide();
5524     $p.top().wdcMain.show();
5525
5526     if($('.grpAlertInfo').css('display') != 'none') $('.grpAlertInfo').css('display', 'none');
5527     if($('.grpUserInfoBtn').css('display') != 'none') $('.grpUserInfoBtn').css('display', 'none');
5528     if($('.left_user_id').css('display') != 'none') $('.left_user_id').css('display', 'none');
5529     if($('.left_payment').css('display') != 'none') $('.left_payment').css('display', 'none');
5530     if($('.left_area_btn').css('display') != 'none') $('.left_area_btn').css('display', 'none');
5531     $('.left_area_main').removeClass("main_menu");
5532 };
5533
5534 com.tempPopup = function(row) {
5535
5536     var width = parseInt(row.popupHrztLen) || 995;
5537     var height = 600;
5538     try {
5539         var deviceWidth = parseFloat($(".body").css("width"));
5540         var deviceHeight = parseFloat($(".body").css("height"));
5541         if (!opt.notMinSize) {
5542             if (deviceWidth > 0 && width > deviceWidth) {
5543                 width = deviceWidth - 0; // 팝업 border 고려
5544             }
5545
5546             if (deviceHeight > 0 && height > deviceHeight) {
5547                 height = deviceHeight - 0; // 팝업 border 고려
5548             }
5549         }
5550     } catch (e) {
5551
5552     }
5553     var data = {};
5554     data.url = row.url;
5555     data.id = row.menuId;
5556
5557     var top = ((document.body.offsetHeight / 2) - (parseInt(height) / 2) + $(".body").scrollTop()) +
5558     var left = ((document.body.offsetWidth / 2) - (parseInt(width) / 2) + $(".body").scrollLeft()) +
5559
5560     var _dataObj = {
5561         type : "json",
5562         data : data,
5563         name : "param"
5564     };
5565
5566     var options = {
5567         id : row.menuId,
5568         popupName : row.menuNm,
5569         type : "wframePopup", // iframePopup, wframePopup, browserPopup
5570         width : width + "px",
5571         height : height + "px",
5572         top : top || "140px",
5573         left : left || "500px",
5574         modal : true,
5575         dataObject : _dataObj,
5576         alwaysOnTop : true,

```

```

5645
5646     var popupData = {
5647         "callbackFn" : "com.CB_FUNCTION_MANAGER.cbFuncSave." + idx,
5648         "isReturnValue" : true
5649     };
5650
5651     if(com.isEmpty($('#' + elementId).val())) {
5652         popupData['searchValue'] = $('#' + elementId).val();
5653     }
5654
5655     $.extend(option, opt);
5656     popupData["userData"] = option;*/
5657
5658     com.popup.open(viewString, option, {});
5659 }
5660
5661 com.closePopup = function()
5662 {
5663     console.log('%c.....This API is deprecated. Use com.popup.open, com.popup.close()', 'background:
5664     com.popup.close();
5665 };
5666
5667 com.executeSubmission = function(options, requestData, obj)
5668 {
5669     console.log('%c.....This API is deprecated. Use com.sbm.executeSubmission(options, requestD
5670     com.sbm.executeSubmission(options, requestData, obj);
5671 };
5672
5673 com.getCodeData = function(param)
5674 {
5675     console.log('%c.....This API is deprecated. Use com.code.getCodeData(param)', 'background:#
5676     return com.code.getCodeData(param);
5677 };
5678
5679 com.alert = function(msg)
5680 {
5681     console.log('%c.....This API is deprecated. Use com.popup.alert(msg, callbackname)', 'backgr
5682     return com.popup.alert(msg);
5683 };
5684
5685 com._getParameters = function(func)
5686 {
5687     return new RegExp('(?:'+func.name+'\\s*|^)\\s*\\(((.*?)\\))').exec(func.toString().replace(/\n/g, '
5688 };
5689
5690 com._resize = function()
5691 {
5692     h = 0;
5693     $.each($(".div[id*='contents'][id$='body']").children(), function(idx, row)
5694     {
5695         h += $(row).height();
5696     });
5697     $(".div[id*='contents'][id$='body']").height(h);
5698 };
5699
5700 com._printAllProperties = function(obj)
5701 {
5702     $.each(Object.getOwnPropertyNames(obj), function(idx, value)
5703     {
5704         if(typeof obj[value] == 'function') console.log(value + '(' + com._getParameters(obj[value].t
5705     });
5706 };

```