

DEWS Studio



DEWS Front Designer

ver 2.0

Copyright (C) DOUZONE BIZON All Rights Reserved.

Manual

Version. 2.6

DOUZONE

목 차

[제1장] DEWS Front Designer 화면 구성요소	8
1. 사용환경	9
2. 설치	9
3. 업데이트	9
4. 로그인	10
5. 화면 구성	13
6. 구성 요소	14
1) 메뉴 영역	14
2) 워크스페이스 탐색기	25
3) 디자이너 영역	25
4) 프로퍼티 영역	27
[제2장] 데이터소스	29
1. 개요	30
2. 일반 정보	31
2. 데이터 정보	32
3. 이벤트	35
4. 컬럼 스키마 (Column Schema)	36
5. 데이터소스 바인딩	36
1) 출력 포맷	40
2) 에디터	44
[제3장] 컨트롤 종류 및 기능	53
1. 상단 메인 버튼	54
2. Canvas	55
3. 컨테이너	56

1) 컨테이너 패널	56
2) 폼 패널	58
3) 컨디션 패널	59
4) 버튼 그룹.....	60
5) 탭 패널	61
6) 참고박스.....	62
7) 아코디언.....	63
4. 폼 컨트롤	66
1) 컨디션 아이템과 폼 아이템.....	66
2) 컨트롤 공통 속성	67
3) 버튼	67
4) 체크박스.....	68
5) 라디오버튼 및 라디오플렉스	69
6) 텍스트박스	72
7) 숫자텍스트박스	73
8) 마스크텍스트박스	74
9) 코드피커	75
10) 멀티코드피커	77
11) 우편번호피커	78
12) 날짜피커	79
13) 시간피커	80
14) 월피커	81
15) 날짜시간피커	81
16) 기간피커	82
17) 주기간피커.....	83

18) 월기간피커.....	83
19) 드롭다운리스트.....	84
20) 멀티드롭다운리스트	85
21) 콤보박스	87
22) 자동완성	88
23) 드롭다운버튼	89
24) 스플리트버튼	90
25) 파일컴포넌트	92
26) 멀티파일 컴포넌트	94
27) 복합 컨트롤	96
5. 데이터	98
1) 그리드	98
2) 트리그리드	100
3) 트리뷰	102
4) 카드리스트	104
5) 피벗그리드	105
[제4장] 스크립트	110
1. 스크립트 사용법.....	111
2. Canvas 의 preReady/Ready 상세.....	114
3. 통합 Script.....	120
4. 조회/추가/삭제/저장/인쇄 스크립트	124
1) 조회 스크립트 예제.....	124
2) 추가 스크립트 예제.....	126
3) 삭제 스크립트 예제.....	127
4) 저장 스크립트 예제.....	128

5) 인쇄 스크립트 예제	131
[제5장] QueryBuilder	135
1. QueryBuilder Setup	136
2. QueryBuilder 실행	139
[제6장] 설계 도구	144
1. 소개 및 실행	145
2. 화면 구성	147
1) 메뉴 툴바	150
2) Diagram 도형과 속성	162
3) 추가 페이지	167
3. 설계 문서 작성	171
1) 문서 개정 이력	171
2) 기능 개요	171
3) 업무 흐름도	172
4) 신규 양식 추가	176
5) 동기화 및 저장	186
6) 개발 도구 그리드 데이터 입력 기능	188
4. 설계 문서 파일 생성	191
1) PDF 파일	191
2) 워드 파일	193
[제7장] Wizard Project	194
1. 소개 및 실행	195
2. 메뉴 개발 예제	197
1) Service	197
2) Local Table	204

3) User Define.....	213
4) Matrix Table.....	226
[제8장] Code Inspection.....	235
1. 소개 및 이점.....	236
2. 기능 설정 및 실행 화면.....	237
1) 기능 설정.....	237
2) Node.js 설치.....	238
3) Code Inspection 화면.....	243
[제9장] 예제 만들어 보기	245
1. 전표삭제현황.....	246
1) 개발 화면.....	246
2) 화면 디자인.....	246
3) 데이터소스 추가 및 바인딩.....	261
4) 이벤트 스크립트 설정	267
2. 자금수지표	274
1) 개발 화면.....	274
2) 화면 디자인.....	275
3) 데이터소스 추가 및 바인딩.....	286
4) 이벤트 스크립트 설정	313
3. 운용상품조회.....	321
1) 개발 화면.....	321
2) 화면 디자인.....	322
3) 데이터소스 추가 및 바인딩.....	338
4) 이벤트 스크립트 설정	356
4. 빠른거래처 등록.....	366

1) 개발 화면.....	366
2) 화면 디자인.....	367
3) 데이터소스 추가 및 바인딩.....	371
4) 이벤트 스크립트 설정	376
[제10장] Git.....	380
1. Git 이란	381
1) Git 개요	381
2. Git 사용 예제.....	381
1) Git Web Server 프로젝트 생성	381
2) DEWS FD 프로젝트 생성	384
3) Git Web Server 병합 요청	403
4) DEWS FD Pull & Merge.....	409
[제11장] JIRA 연동.....	413
1. 소개.....	414
2. JIRA 설정 및 실행 방법.....	415
1) JIRA 설정.....	415
2) JIRA 실행.....	416

[제1장] DEWS Front Designer 화면 구성요소

사용환경

ClickOnce를 통한 설치

업데이트

로그인

화면구성

구성요소

1. 사용환경

DEWS Front Designer을 설치 및 실행하기 위해서는 .NET Framework 4.6(닷넷 프레임워크 4.6 이하 4.6) 이상이 설치되어 있어야 하며, OS 권장사항은 윈도우 7 이상 되어야 한다.

만약 .NET Framework가 설치되어 있지 않는 PC의 경우 설치 웹 사이트에서 .NET Framework가 설치되어 있지 않다는 안내 메시지가 발생하며, 안내 메시지에 따라 .NET Framework를 설치한 후 DEWS Front Designer 설치를 진행 할 수 있다.

2. 설치

DEWS Front Designer 설치는 기존에는 웹 브라우저 URL 접속으로 설치를 진행하였으나, 현재 지금은 설치 파일(setup.exe)을 실행하여 설치한다. 기존 ClickOnce URL 접속으로 설치된 사용자는 제어판에서 삭제 후 setup.exe 파일을 실행한다



<개체 1-1. DEWS Front Designer 설치 압축 파일>

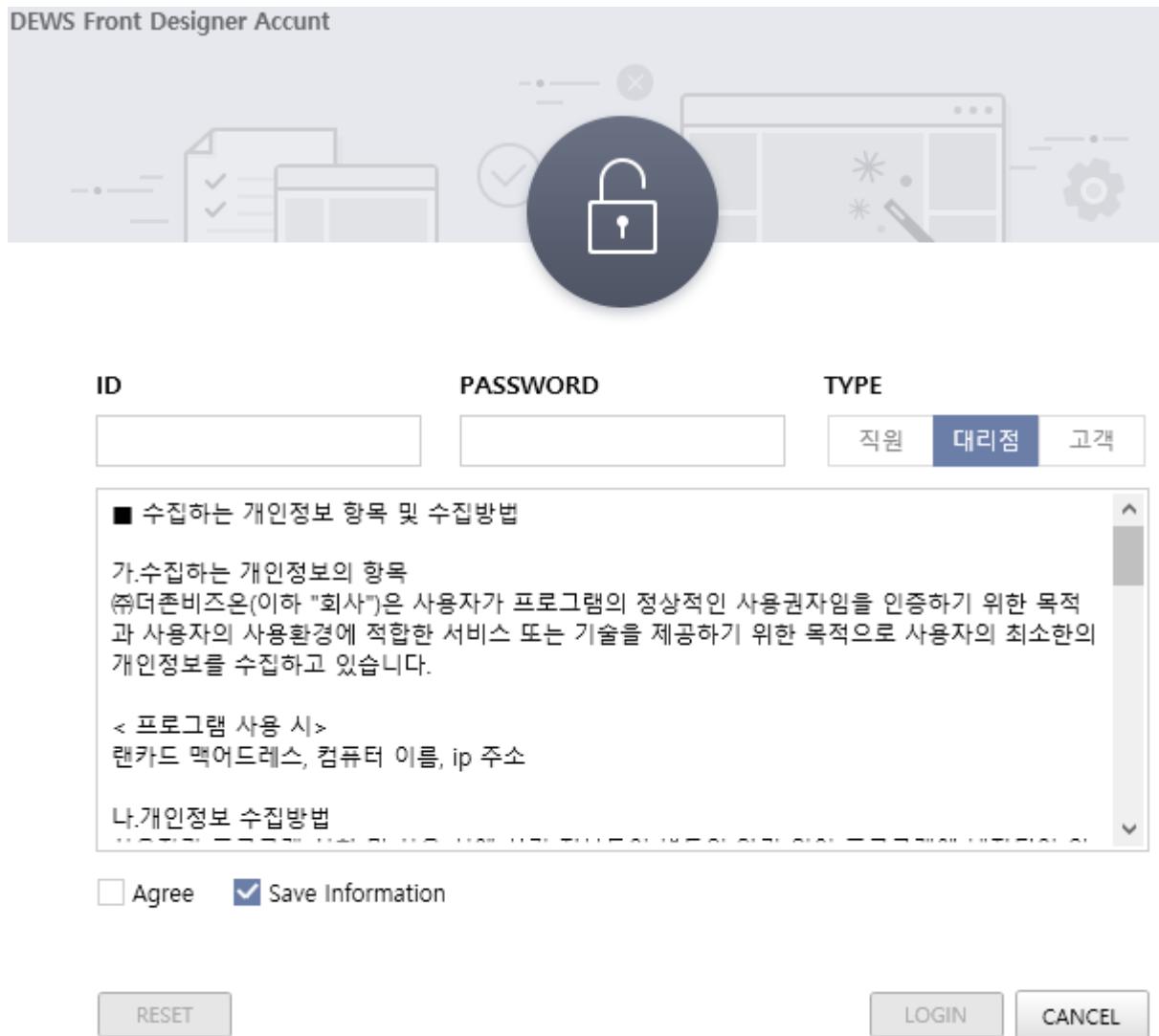
3. 업데이트

DEWS Front Designer는 최초 설치 이후 DEWS Front Designer를 실행 할 때마다 새로운 버전이 존재하는지 확인 후 업데이트가 필요할 때 업데이트 화면과 함께 자동으로 업데이트를 진행한다.



<그림 1-1. DEWS Front Designer 실행 시 자동 업데이트 화면>

4. 로그인



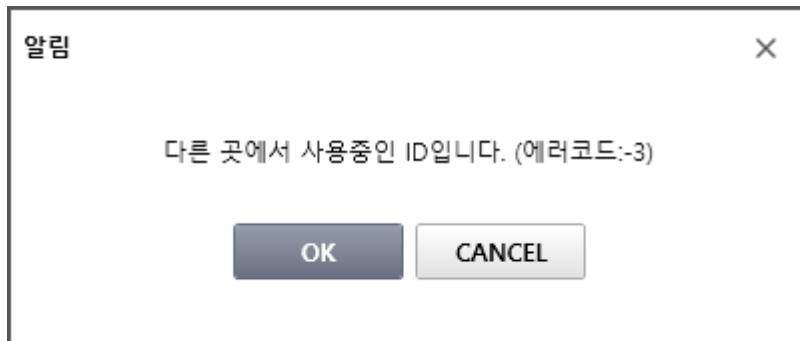
The image shows the login interface for the DEWS Front Designer Account. At the top, there is a header "DEWS Front Designer Account" and a decorative graphic featuring icons related to documents, a lock, and a gear. Below the header is a large central button containing a lock icon. The main form area has three input fields: "ID", "PASSWORD", and "TYPE". Under "TYPE", there are three buttons: "직원" (Employee), "대리점" (Distributor), and "고객" (Customer), with "대리점" being the selected option. To the right of the input fields, there is a section titled "■ 수집하는 개인정보 항목 및 수집방법" (List of collected personal information items and collection methods). This section contains two parts: "가. 수집하는 개인정보의 항목" (List of collected personal information items) and "나. 개인정보 수집방법" (Collection method of personal information). The "나" section is currently collapsed. At the bottom of the form, there are two checkboxes: "Agree" and "Save Information", with "Save Information" being checked. Below the form are three buttons: "RESET", "LOGIN", and "CANCEL".

<그림 1-2. DEWS Front Designer 로그인 창>

사용자가 제품 사용 허가가 되었는지 확인하기 위해 로그인을 해야 한다. 로그인 유형에는 직원, 대리점, 고객 3가지 타입이 존재하며, 내부 직원의 경우 직원을 선택하고 메신저 아이디와 암호를 입력한다. 대리점 직원의 경우 대리점을 선택하며, 고객일 경우 고객을 선택한다.

약관에 동의 후 로그인을 하면 DEWS Front Designer가 실행된다. 이 때 로그인에 3회 실패를 하게 되면 프로그램은 실행이 되지 않고 자동종료 된다.

또한 하나의 아이디를 이용하여 다른 PC에서 로그인을 할 경우 "다른 곳에서 사용중인 ID이다.(에러코드:-3)"이라는 오류가 발생한다.

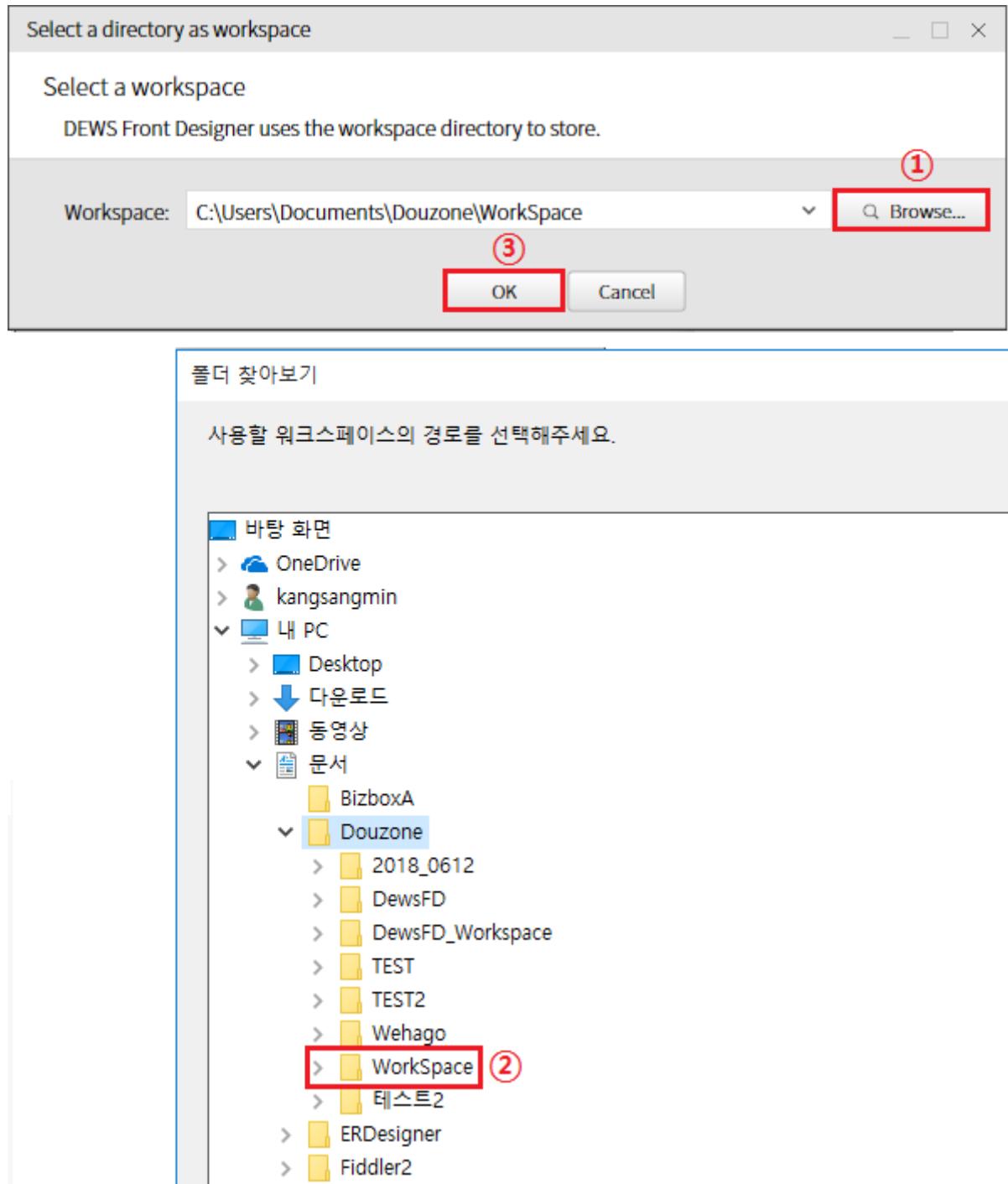


<그림 1-3. 중복 로그인 오류 화면>

이 때는 OK 버튼을 눌러 해당 창을 닫고 <그림 1-4>의 ① Agree 체크하고 ② RESET 버튼을 누르면 로그인이 정상적으로 이루어진다. 단 기존에 로그인 되어있던 PC에서는 DEWS Front Designer가 종료되므로 주의를 해야 한다.

<그림 1-4. 중복 로그인 오류 처리 방법>

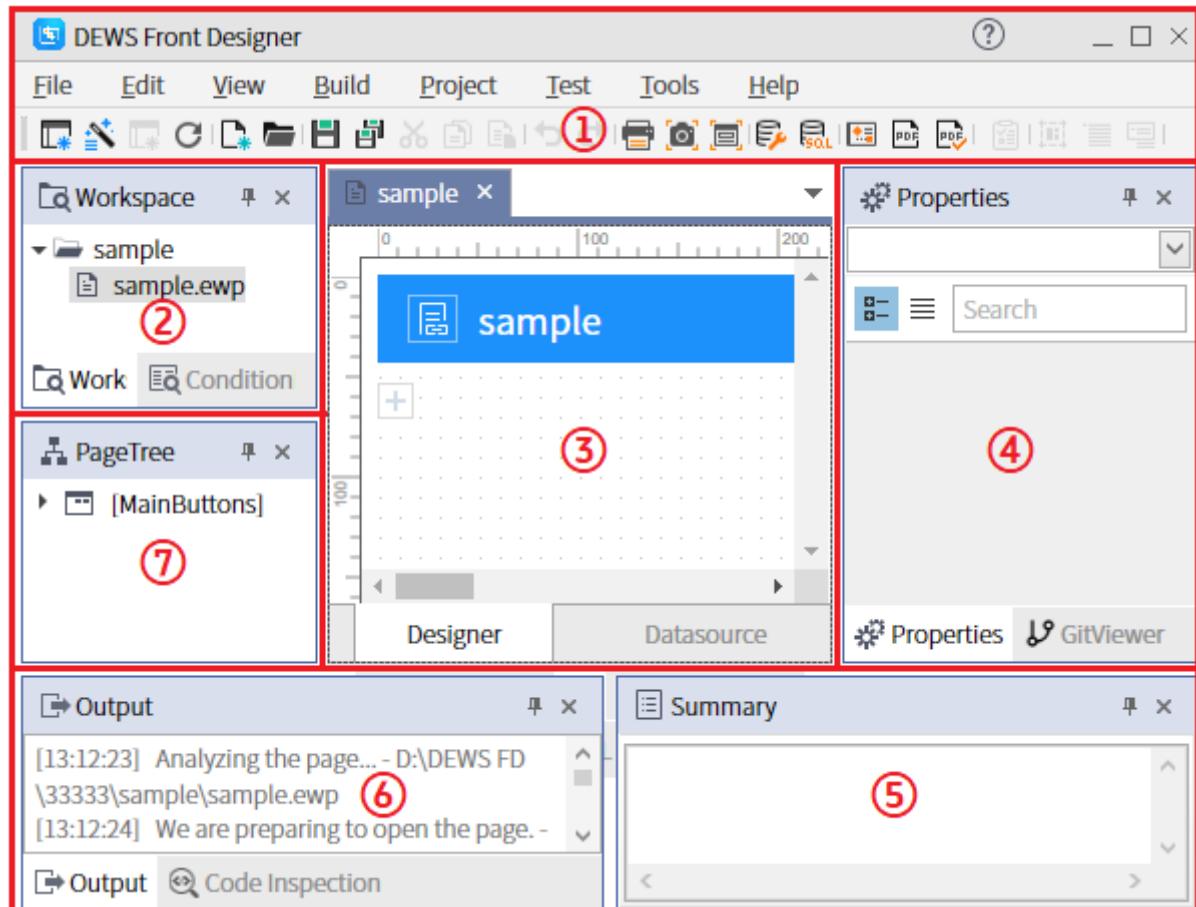
로그인 후에 저장될 작업 영역 디렉토리를 선택하는ダイ얼로그가 나오면 ① Browse 버튼을 누르고 ② 디렉토리를 선택하거나 기본으로 선택되어 있는 폴더(DewsFD_Workspace)를 선택하고 ③ OK 버튼을 누른다.



<그림 1-5. 저장할 작업영역 디렉토리 선택>

5. 화면 구성

<그림 1-6>는 처음 DEWS Front Designer 을 실행하였을 때 나타나는 화면이며, 크게 7개의 화면 구성 요소들로 나눌 수 있다.



<그림 1-5. DEWS Front Designer 처음 실행 시 화면 구성>

- ① 메뉴 영역 : 메뉴와 도구 상자(이미지 아이콘 툴바)
- ② 워크스페이스 및 조회 조건 탐색기 : 프로젝트 파일을 관리하고 모듈별로 사용되는 조회 조건 아이템을 확인
- ③ 작업 영역 : 디자인을 하거나 데이터소스를 생성 또는 통합 스크립트를 작성하거나 CSS를 적용 및 HTML 소스 확인
- ④ 속성 및 GitViewer : 컨트롤의 속성을 확인하거나 변경/Git 메뉴(설정 필요)
- ⑤ Summary : 설계 작성 문서(설정 필요)
- ⑥ 출력창/Code Inspection : 로그를 확인하거나 코드 검사(설정 필요)
- ⑦ 페이지 트리 : 컨트롤 간에 상/하 관계를 확인

6. 구성 요소

1) 메뉴 영역

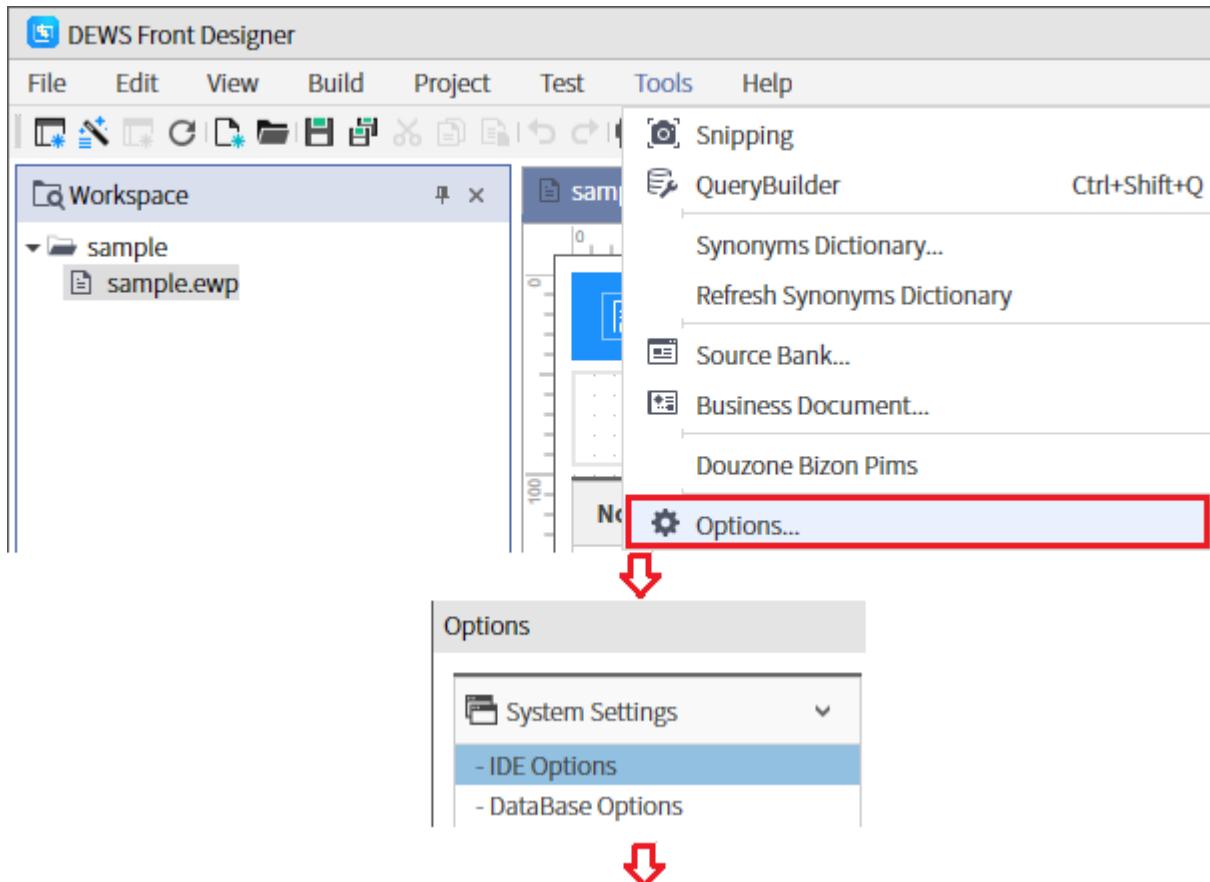
상단에 File, Edit, View, Build, Project, Test, Tools, Help 8개 메뉴가 위치한다.



<그림 1-6. 상단 메뉴 영역>

- ① **File** : 메뉴에 프로젝트를 새로 만들거나 여는 기능
- ② **Edit** : 디자이너 화면의 항목들에 대해 복사/붙여넣기/되돌리기 등을 수행
- ③ **View** : 화면 구성요소를 보이거나 숨김
- ④ **Build** : 빌드 및 미리보기를 진행
- ⑤ **Project** : 메뉴는 프로젝트를 추가하거나 페이지를 추가하는 기능
- ⑥ **Test** : Test Case 실행
- ⑦ **Tools** : 메뉴는 전체 툴의 옵션을 설정하는 메뉴이며, 옵션메뉴를 클릭할 경우 <그림 1-7>과 같은 다이얼로그가 나타남
- ⑧ **Help** : 매뉴얼(pdf)을 보거나 버전을 확인

가장 먼저 Tools에 Options...을 클릭하면 Options 화면이 열리고 왼쪽 System Setting 카테고리 IDE Options을 선택한다.



• 사용할 웹 브라우저

미리보기 용으로 사용할 웹 브라우저 정보를 설정합니다.

경로	브라우저	사용
chrome	Chrome	<input type="checkbox"/>
iexplorer	Internet Explorer	<input type="checkbox"/>
firefox	Firefox	<input type="checkbox"/>
edge	Edge	<input type="checkbox"/>

• Editor Theme White

• Preview Language Korean

• 옵션

마지막 페이지 열기

HTML 창 스크립트 수정기능

Git 가능(다시 실행해야 반영됨)

빌드시 로그보기

빌드 후 같은 실행 경로 지정

Tabbed MDI 사용(다시 실행해야 반영됨)

워크스페이스 탐색기에서 활성화된 항목 추적

Summary 보기(다시 실행해야 반영됨)

통합 스크립트 에디터 사용(열려있는 문서는 새로 열어야 반영됨)

Enable Code Inspection

<그림 1-7. System Settings ▶ IDE Options>

System Settings > IDE Options

사용할 웹 브라우저는 빌드/미리보기 클릭 시 보여줄 브라우저를 설정한다. 지원하는 브라우저는 **Chrome, Internet Explorer, FireFox, Edge**가 있으며, Chrome의 경우 강력한 개발자 도구를 기본으로 제공하고 소스를 수정하거나 최적화하는데 필요한 다양한 기능이 있어 주로 사용한다.

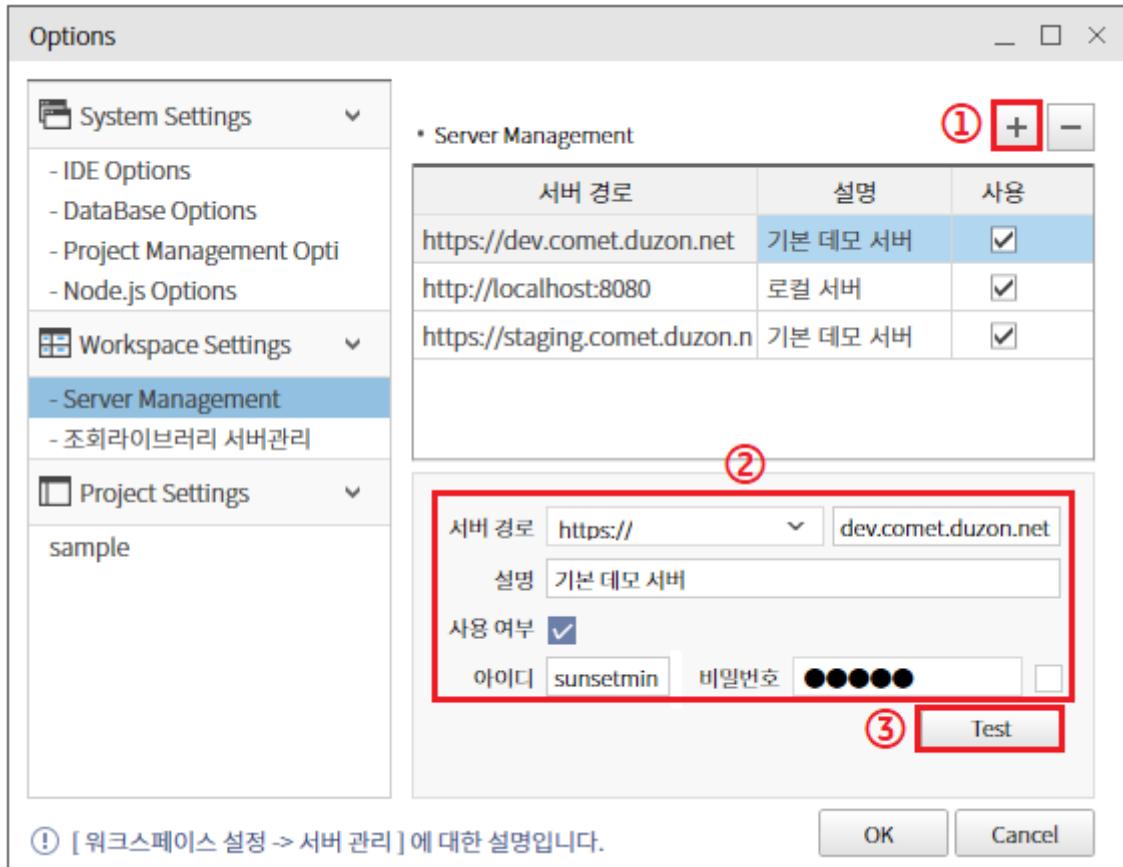
Editor Theme은 Script, CSS, HTML 배경색 설정이며, Preview Language는 미리보기 언어를 설정한다.

아래 옵션을 살펴보면

- ① **마지막 페이지 열기** : DEWS FD를 재실행할 때 가장 마지막에 작업했던 페이지가 자동 열림
- ② **HTML 창 스크립트 수정기능** : 빌드시 기본적으로 HTML은 수정 불가하나 함수 ID가 있는 라인을 클릭하고 F7를 누르면 해당 이벤트 스크립트 수정 가능
- ③ **Git 기능** : Git 기능 사용 및 메뉴 활성화
- ④ **빌드시 로그 보기** : 미리 보기 후 조회할 경우 로그창에서 서버(Back-end) 정보 확인
- ⑤ **빌드 후 같은 실행 경로 지정** : 미리 보기 시 웹 브라우저에서 새 탭을 띄우지 않고 기존 탭에서 미리 보기
- ⑥ **Tabbed MDI 사용** : 탭 다중 문서 인터페이스 즉, 탭 분리 기능 사용 여부
- ⑦ **워크스페이스 탐색기에서 활성화된 항목 추적** : 선택하는 화면에 따라 Workspace 선택된 페이지도 함께 변경
- ⑧ **Summary 보기** : 설계 문서 활성화 여부
- ⑨ **통합 스크립트 에디터** : Script 탭에서 사용자가 작성한 스크립트 전체 보기 및 수정 기능
- ⑩ **Enable Code Inspection** : 코드 검사 활성화 여부

보통 공통으로 Git 기능, 워크스페이스 탐색기에서 활성화된 항목 추적, Summary 보기, Enable Code Inspection을 체크하고 개발자인 경우 HTML 창 스크립트 수정기능, Tabbed MDI 사용을 체크한다.

<그림 1-8>의 **Workspace Settings>Server Management** 메뉴는 모듈정보를 받아오거나, 향후 빌드 결과물을 업로드 하는 등 실제 메뉴가 동작하는 서버를 설정하는 메뉴이다. 기본 서버 경로는 <https://dev.comet.duzon.net>로 설정되어 있으며, ② 아이디, 비밀번호를 반드시 입력해야 미리보기가 가능하다.

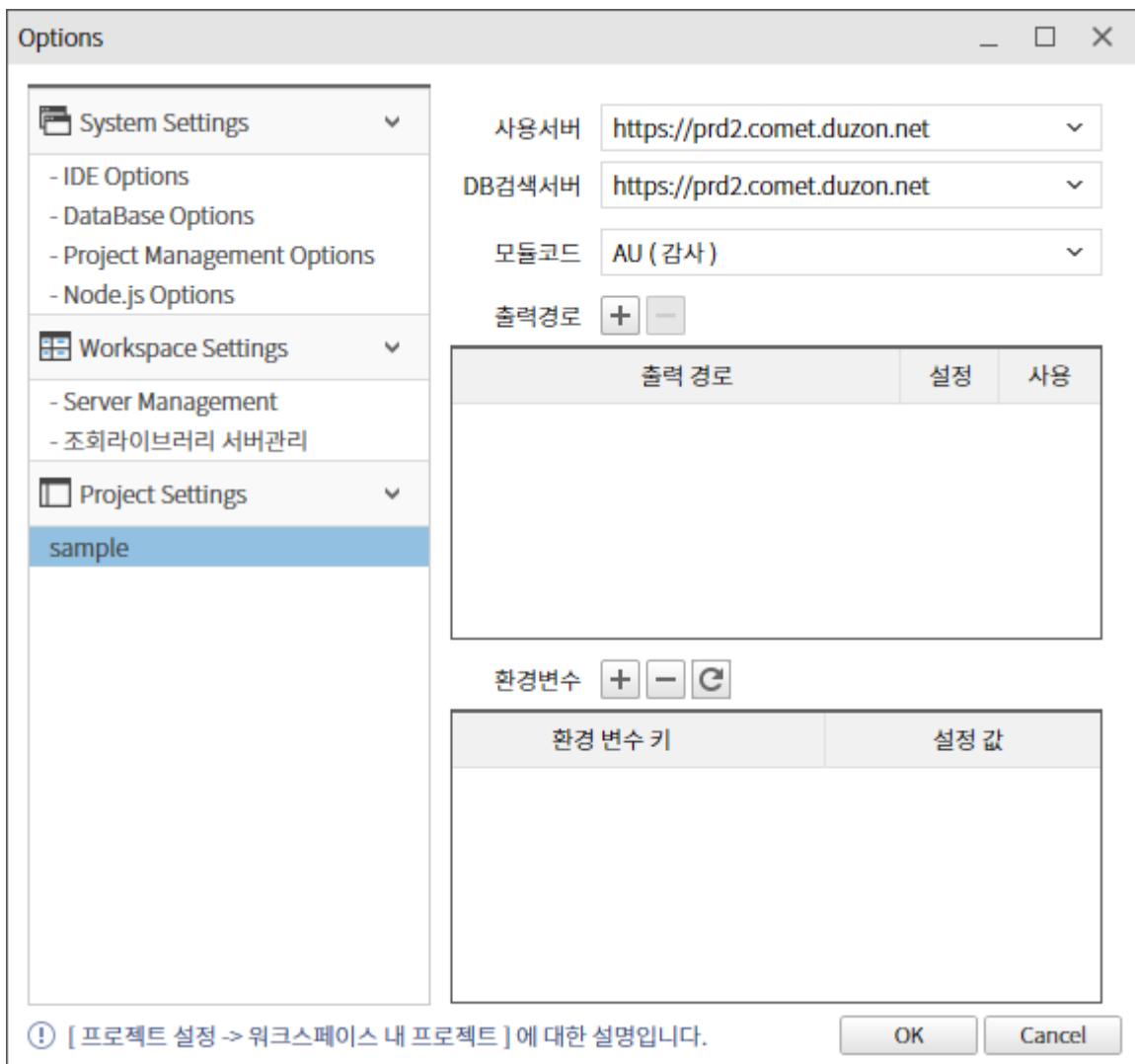


<그림 1-8. 서버관리 서버 경로 및 아이디/패스워드 설정 화면>

또한 ② “사용 여부” 체크에 따라 프로젝트 생성창에서 보여지는 서버리스트에 반영된다.

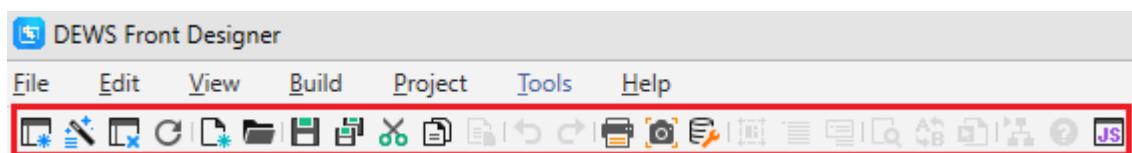
만약 생성한 프로젝트의 서버 경로가 기본값이 아닌 경우, ① 우측 상단 **+** 놀러 서버 경로에 직접 실행 경로를 입력하여 원하는 서버로 바꿀 수 있다.

프로젝트 별로 다른 서버를 사용하기 위해서 이전과 같이 <그림 1-8> ① 우측 상단 **+** 버튼을 눌러 서버를 추가한 다음 ② 서버 정보, 사용 여부 체크, 아이디와 비밀번호를 입력하고 ③ Test , 버튼을 눌러 접속 여부를 확인하여 신규 서버를 만들 수 있다. 만들어진 서버리스트는 **Project Settings>해당 프로젝트명**을 클릭하여 프로젝트별로 “사용 서버”나 “DB검색 서버”를 따로 설정 할 수 있다.



<그림 1-9. 프로젝트 별로 사용서버를 선택하는 화면>

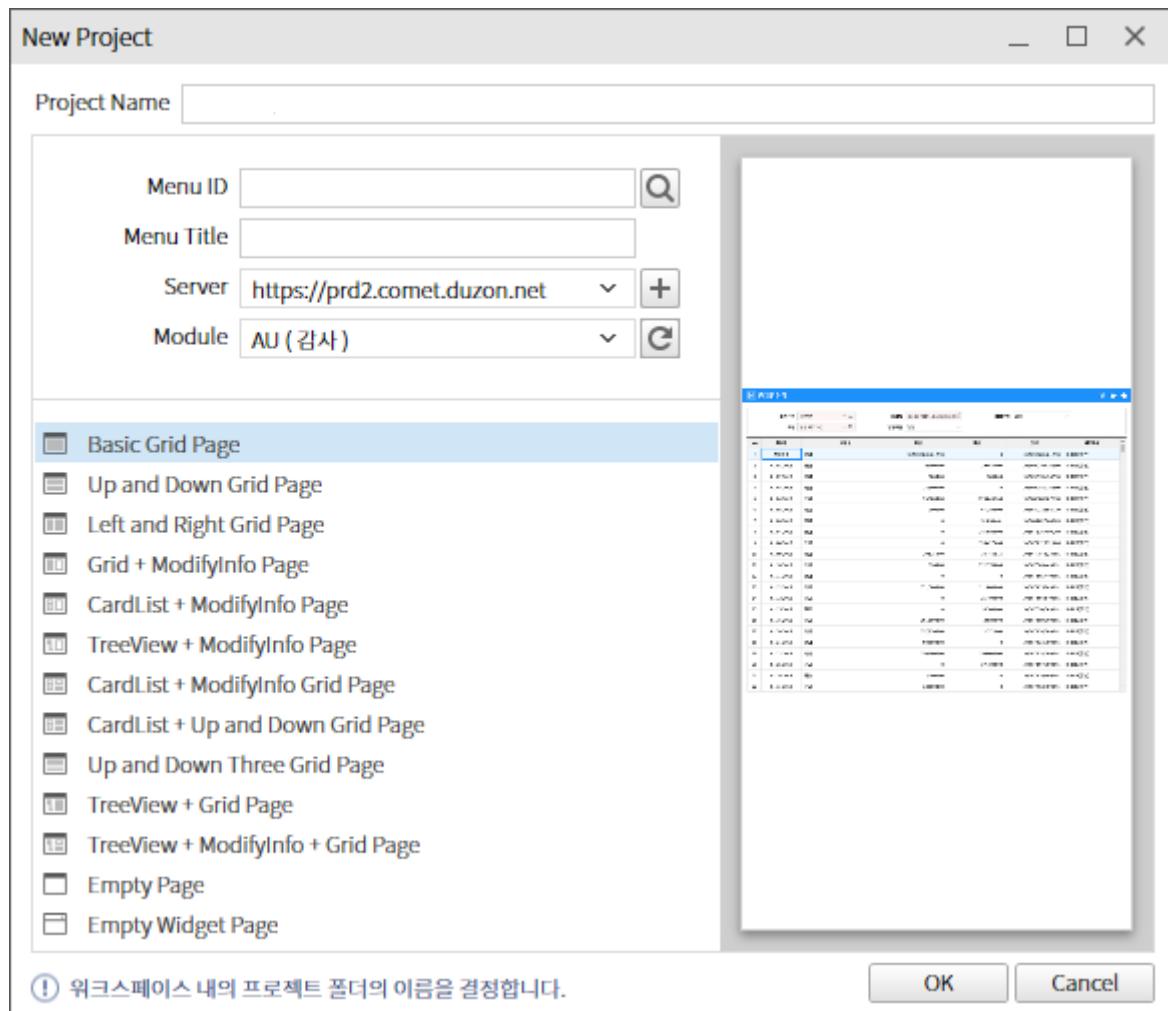
메뉴 밑에는 자주 사용하는 메뉴들을 버튼으로 만든 툴바(도구모음)가 위치한다.



<그림 1-10. 툴바 화면>

이 중에 자주 사용되는 툴바들만 몇 가지 살펴본다.

-  **New Project** : 워크스페이스에 새로운 프로젝트를 추가한다. 버튼을 누르면 <그림 1-11>과 같은 프로젝트 추가 다이얼로그가 나타난다. 기본 템플릿으로 기본 그리드 페이지로 생성할 수 있으며, 오른쪽 미리보기에서 화면을 확인할 수 있다. 그 외에도 상하 좌우 그리드 페이지, 그리드 + 정보수정 페이지 등 다양한 형태의 템플릿을 선택하여 빠르게 메뉴를 개발 할 수 있다.

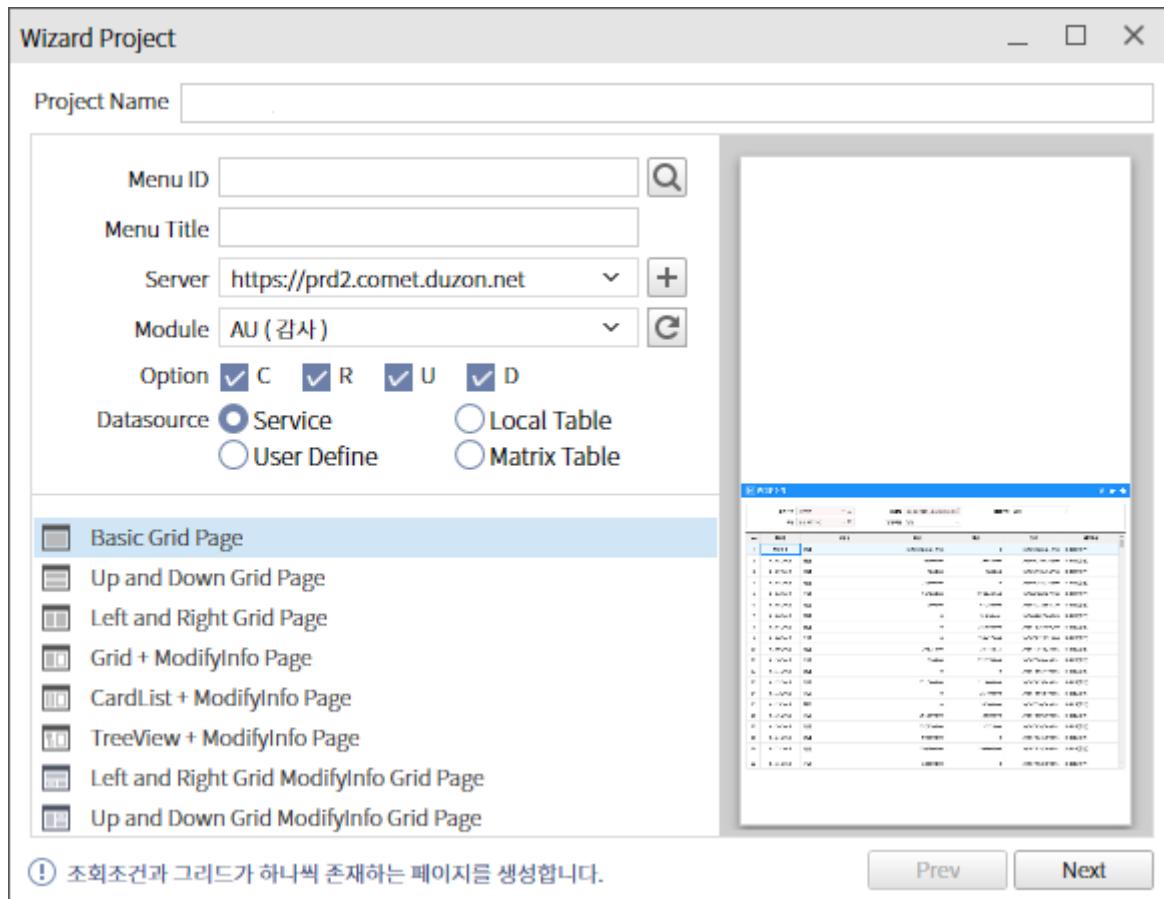


<그림 1-11. 프로젝트 추가 화면>

속성 명	설명
Project Name	이름에 해당하는 폴더가 워크스페이스에 생성. 메뉴 단위
Menu ID	ORGCOA00100과 같은 실제 html 파일로 나타나는 페이지명(ewp)
Menu title	사용자에게 보여지는 문서 제목명
Server	미리보기 서버를 뜻하며, Tools > Options 에서 설정한 Server Management에서 사용 여부에 체크한 서버 리스트
Module	Server에 등록되어 있는 모듈 정보

<표 1-1. 프로젝트 추가 화면 각 속성명 설명>

-  **New Wizard** : 화면 UI를 보다 쉽고 빠르게 자동으로 생성할 수 있는 프로젝트 마법사를 실행한다. 과거 개발자들이 화면을 일일이 디자인했던 기존 UI 개발 툴과 비교했을 때 혁신적인 생산성 향상을 가져다 줄 수 있다.



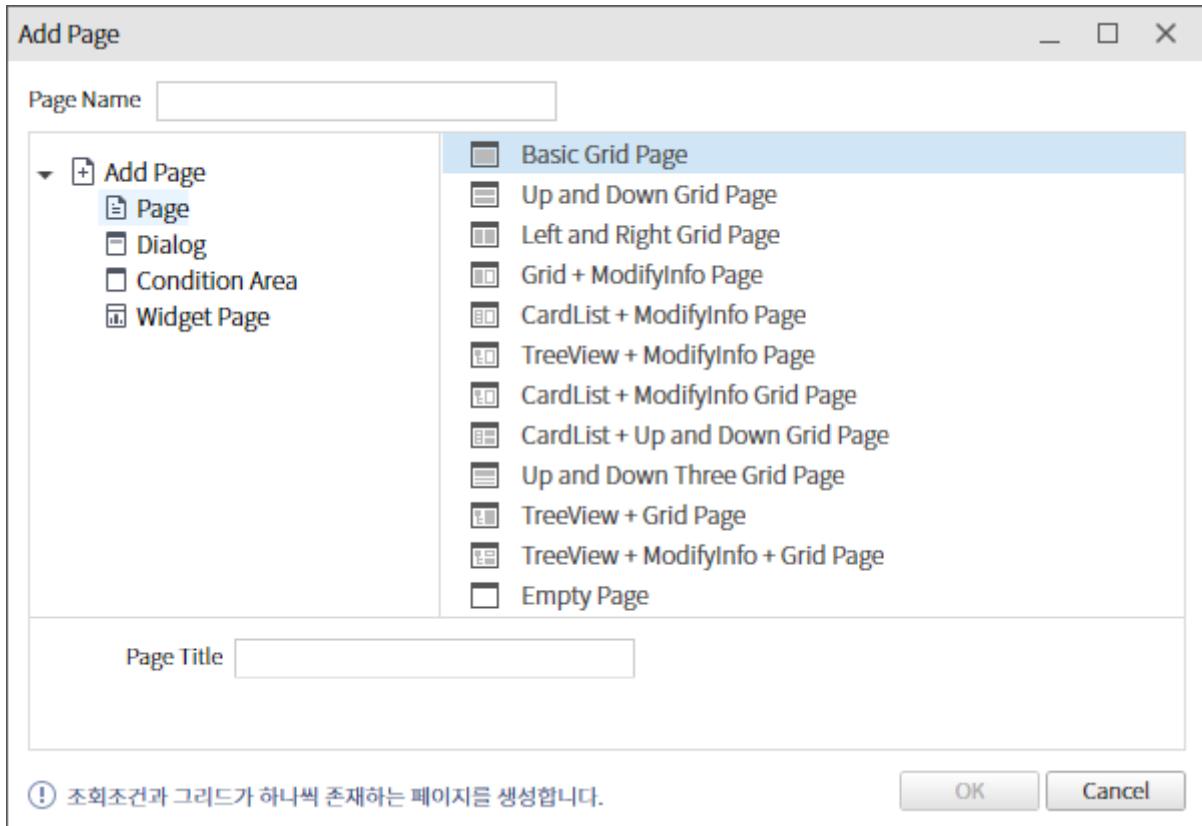
<그림 1-12. New Wizard 화면>

속성 명	설명
Project Name	이름에 해당하는 폴더가 워크스페이스에 생성. 메뉴 단위
Menu ID	ORGCOA00100과 같은 실제 html 파일로 나타나는 페이지명(ewp)
Menu title	상단 사용자에게 보여지는 문서 제목명
Server	미리보기 서버를 뜻하며, Tools > Options에서 설정한 Server Management에서 사용 여부에 체크한 서버 리스트
Module	Server에 등록되어 있는 모듈 정보
Option	만들 메뉴 특성에 따른 체크 (C:Create, R:Read, U:Update, D:Delete) 조회 메뉴인 경우 R 체크
Datasource	연결할 데이터소스 종류

<표 1-2. 프로젝트 마법사 각 속성명 설명>

-  **New Page** : 페이지를 새로 생성하는 버튼이다. 프로젝트가 선택 되어 있는 상태에서 가능하며 프로젝트 아래에 페이지를 생성하는 기능을 한다. 클릭 시 페이지 추가ダイアログ가 나타난다.

서버와 모듈 관련 설정은 페이지가 생성될 때 상위 프로젝트의 설정을 따르게 되기 때문에 프로젝트 관련 설정을 추가로 할 필요가 없다.

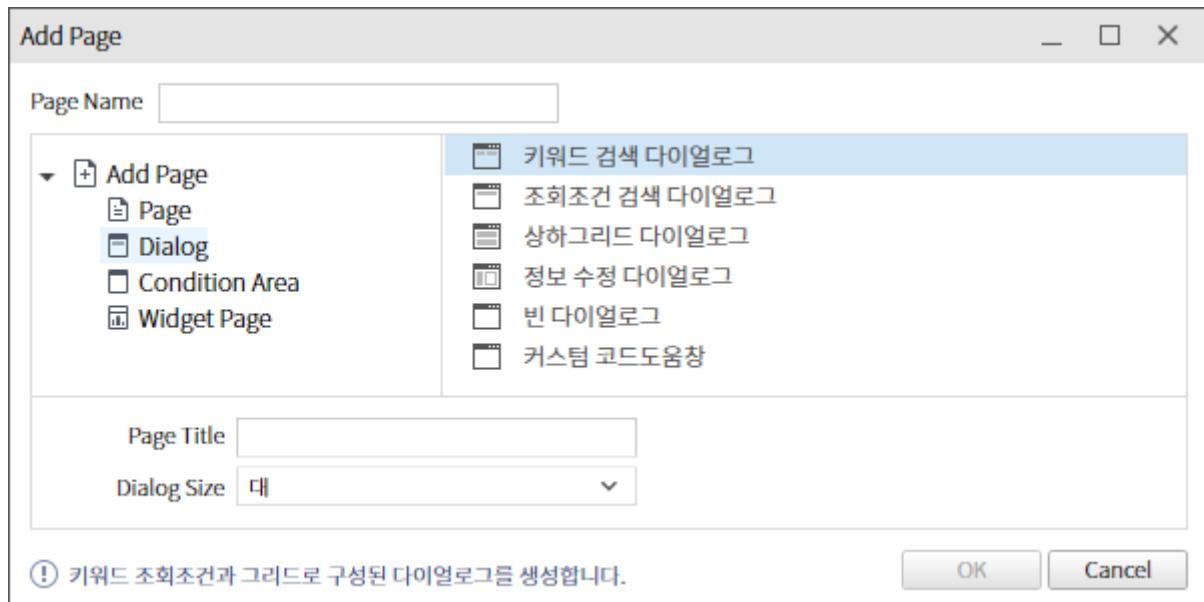


<그림 1-13. 페이지 추가 화면>

속성 명	설명
Page Name	ORGCOA00100과 같은 실제 html 파일로 나타나는 페이지명(ewp)
Page Title	상단 사용자에게 보여지는 문서 제목명

<표 1-3. 페이지 추가 화면 각 속성명 설명>

<그림 1-14>은 다이얼로그 페이지 추가 템플릿 화면 이다.



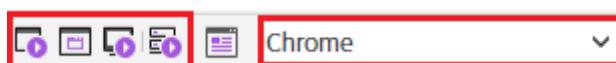
<그림 1-14. 다이얼로그 추가 화면>

속성 명	설명
Page Name	ORGCOA00100과 같은 실제 html 파일로 나타나는 페이지명(ewp)
Page Title	상단 사용자에게 보여지는 문서 제목명
Dialog Size	다이얼로그에 해당하는 설정들이며, 다이얼로그 크기 설정 (소, 중, 대, 사용자정의)

<표 1-4. 다이얼로그 추가 화면 각 속성명 설명>

- Build & Preview

빌드와 미리보기 버튼은 프로젝트내 문서(*.ewp)를 열어야 활성화 된다.



<그림 1-15. 빌드 및 미리보기 화면>

버튼 명	설명
Build	현재 페이지를 빌드하고 저장
Preview	현재 페이지를 브라우저에서 미리보기 (다이얼로그 미리보기에 사용)
Build & Preview	현재 페이지를 빌드하고 브라우저에서 미리보기
Build & Export	현재 페이지를 빌드하고 브라우저 기준 템에서 미리보기
Browser	미리보기가 실행될 브라우저 선택

<표 1-5. 빌드 미리보기 툴바 설명>

- 전체 툴바 기능 정리

전체 이미지 아이콘 툴바 설명은 <표 1-4>에 정리하였다

툴바	설명
New Project (Ctrl+N)	새 프로젝트 만들기
New Wizard	새 마법사(마법사 도구를 이용한 새로운 프로젝트 만들기)
Close Project	현재 열려있는 프로젝트 닫기
Reload Project(Ctrl+Shift+R)	프로젝트 새로고침
New Page(Ctrl+P)	프로젝트의 새로운 페이지 생성
Open(Ctrl+O)	기존 저장된 ewp 파일 불러오기
Save(Ctrl+S)	현재 프로젝트 저장하기
Save All(Ctrl+Shift+S)	전체 프로젝트 저장하기
Cut(Ctrl+X)	잘라내기
Copy(Ctrl+C)	복사하기
Paste(Ctrl+V)	붙여넣기
Undo(Ctrl+Z)	되돌리기(실행취소)
Redo(Ctrl+Y)	되살리기(기존 마지막으로 작업한 내용으로 되돌리는 기능)
Print Designer(Ctrl+Shift+P)	디자이너 영역 인쇄
Snipping	사용자 지정 화면 캡쳐
Designer Screen Capture	디자이너 영역 전체 화면 캡쳐
Query Builder(Ctrl+Shift+Q)	쿼리 빌더 실행 도구
User Define SQL Manager (Ctrl+Shift+U)	Query Service Manager 실행 (사용자 정의 쿼리 리스트 관리)
Business Document(Ctrl+Shift+B)	설계문서 실행 도구
Export Business Document (Ctrl+Shift+D)	설계문서 전체 PDF 파일 생성
Export Selected Business Document (Alt+Shift+D)	선택한 설계문서 PDF 파일 생성
Test Case	테스트 케이스 실행
Auto Align(Ctrl+K+F)	자동 정렬 (소스코드 자동 정렬)
Line Comment	현재 라인을 주석으로 변경
Block Comment	블럭으로 지정된 라인을 주석으로 변경
Find(Ctrl+F)	소스코드 검색 기능
Replace(Ctrl+R)	검색된 소스코드를 다른 코드로 수정 또는 변경하는 기능

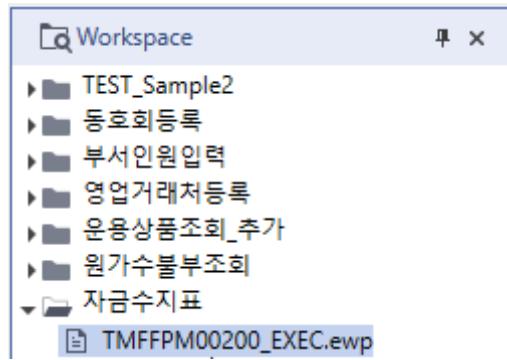
Find All References	모든 참조 찾기(전체 소스 추적 기능)
Editor Minimap	에디터 미니맵(현재 작업영역을 미니맵 형태로 볼 수 있는 기능)
Help Editor	스크립트 에디터 도움창
Create Jira Issue	JIRA 연동을 통한 이슈 등록
DOZONE BIZON PIMS	더존비즈온 PIMS 연동을 통한 요청사항 등록 (더존 내부직원만 사용 가능)
Code Inspection	코드 검사 보이기/숨기기
Code Viewer	사용자가 입력한 전체 스크립트 보기
Event Editor	HTML 탭에서 작성된 이벤트 수정 기능
Build	현재 페이지를 빌드(HTML 파일 생성)하고 저장
Preview(F5)	현재 페이지를 브라우저에서 미리보기 (다이얼로그 미리보기에 사용)
Build & Preview(Ctrl+B)	현재 페이지를 빌드(HTML 파일 생성) 하고 웹 브라우저에서 새 탭으로 미리보기
Build & Export	현재 페이지를 빌드(HTML 파일 생성) 하고 웹 브라우저 기존 탭에서 미리보기 (웹 브라우저 탭에서 새로고침 필수)
Source Bank	소스 뱅크 실행(기존 만들어진 개발 소스 공유)

<표 1-6. 전체 툴바 설명>

2) 워크스페이스 탐색기

메뉴 영역에서 프로젝트를 생성하게 되면 다음과 같이 워크스페이스 탐색기에 프로젝트와 하위 페이지가 생성된다. <그림 1-16>에서 자금수지표는 프로젝트명이며 메뉴에 해당된다.

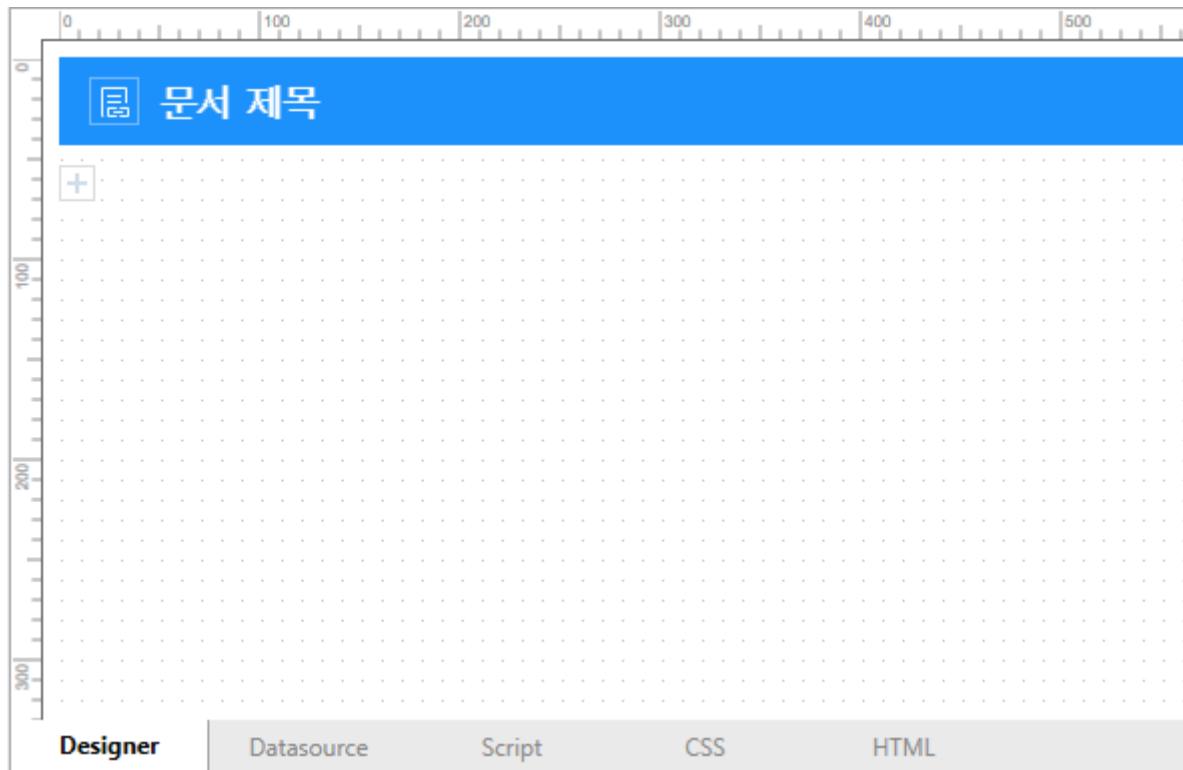
TMFFPM00200_EXEC.ewp은 Menu ID이며 페이지 이름을 의미한다. 빌드 후에는 페이지 이름.html 파일이 생성된다..



<그림 1-16. 워크스페이스 탐색기>

3) 디자이너 영역

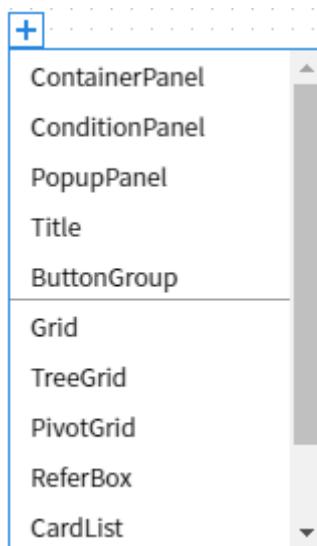
프로젝트를 생성하면 <그림 1-16>와 같은 디자이너 화면을 볼 수 있다. 페이지 생성에서 설정한 대로 만들어진다.



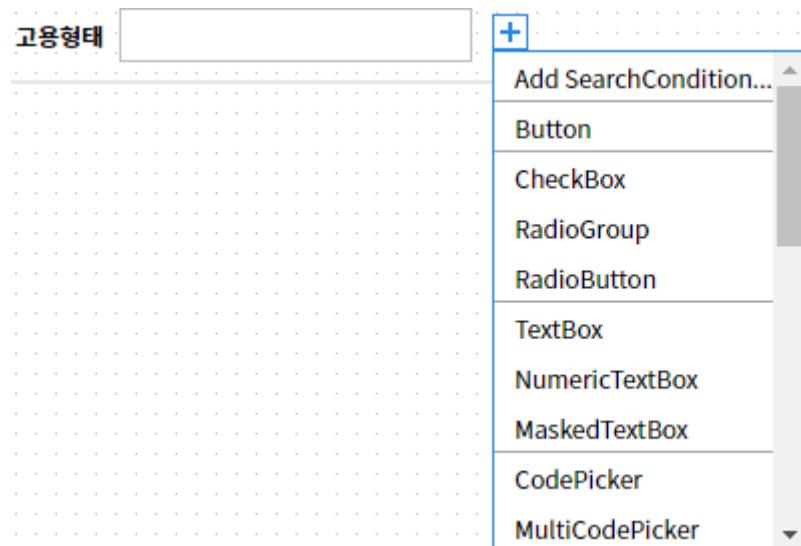
<그림 1-17. 디자이너 영역 화면>

하단에  버튼을 클릭하면 새로운 컨트롤을 추가 할 수 있다.

추가할 수 있는 컨트롤은 패널의 종류에 따라 하위 컨트롤 생성 리스트가 다르다. 예를 들어, 빈 패널인 Canvas에서 컨트롤을 추가할 때는 <그림 1-18>와 같이 12개의 컨트롤을 생성 할 수 있지만, <그림 1-19>처럼 Canvas가 아닌 컨디션 패널일 때는 추가 할 수 있는 컨트롤의 종류도 다르다.



<그림 1-18. 빈 패널(Canvas)에서 컨트롤 추가>



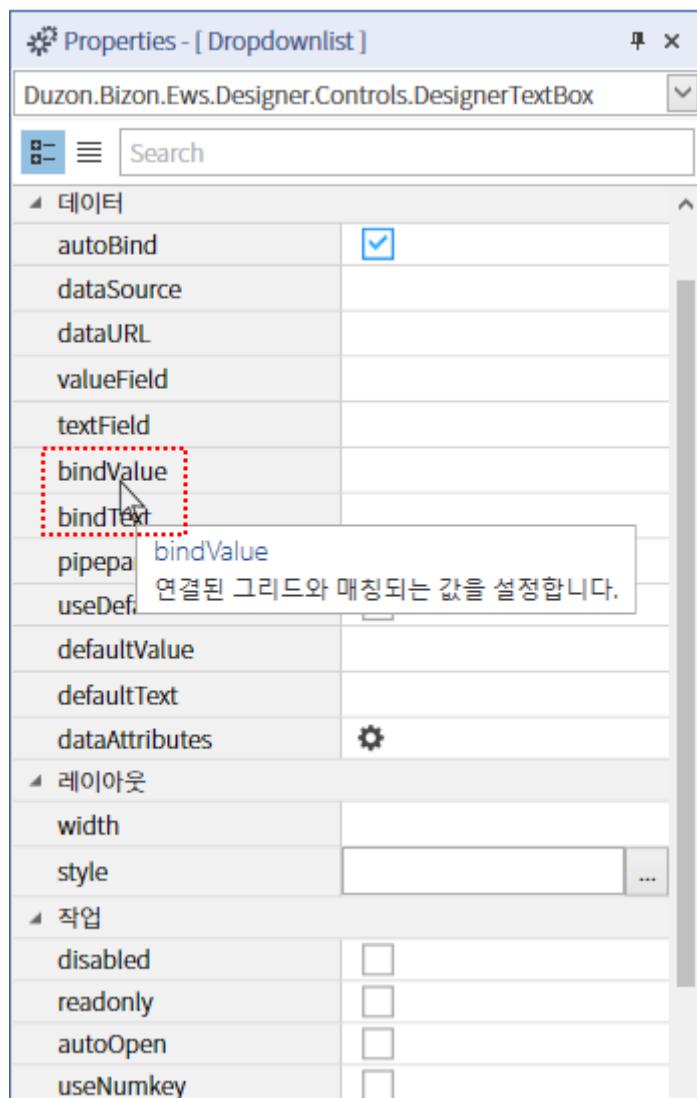
<그림 1-19. 컨디션 패널에서 컨트롤 추가>

4) 프로퍼티 영역

디자이너 영역에서 특정 컨트롤을 클릭하여 선택했을 때 우측 프로퍼티 창에 속성들이 나열된다. 각 컨트롤마다 고유의 속성들을 갖고 있으며, 선택된 컨트롤명이 프로퍼티 창 제목에 표시된다. <그림 1-20>에서 선택한 컨트롤은 드롭다운리스트이며 제목에 프로퍼티-[Dropdownlist]로 표시된다.

캔버스(Canvas)를 제외한 모든 컨트롤들은 ID(아이디) 속성을 가지고 있다. id는 해당 컨트롤을 구분할 수 있는 고유 ID로써 자바스크립트 코드에서 컨트롤들을 식별할 때 사용된다. 또한 텍스트박스, 코드피커와 같이 특정 컨트롤의 경우 생성할 때 아이디가 자동으로 부여된다.

속성에 마우스를 오버하면 속성에 대한 설명이 툴팁으로 표시된다



<그림 1-20. 드롭다운리스트 프로퍼티 창>

<그림 1-20> 하단에 change, open, close 와 같은 속성들이 있는데, 이 속성들은 JQuery 이벤트 또는 DEWS API를 스크립트 상에서 직접 코딩할 수 있도록 하는 속성이다. </> 버튼을 클릭하여 스크립트를 직접 작성 할 수 있으며, 아래와 같은 코드 에디터 화면이 나타난다.

스크립트가 이미 작성 되어 있다면 </> 버튼이 </>파랗게 표시된다. X 버튼은 기존에 작성한 스크립트를 삭제한다. Visual Studio Code와 모나코 에디터를 사용하고, 완성도 높은 인텔리센스를 제공한다.

The screenshot shows a code editor window titled 'mstGrid [Grid]'. A tab labeled 'dblClicked' is selected. The code is as follows:

```
121 function(e) {
122     if(e.cell.field == 'DOCU_NO'){
123         var param = {
124             menuID: 'BILCMR00100' ,
125             pc_cd: self.user.profitCenterCode,
126             docu_no: self.mstGrid.getCellValue(self.mstGrid.select(
127             );
128             dews.ui.openMenu('FI', 'GLDDOC00700', param);
129         }
130     }
}
```

<그림 1-21. 코드 에디터 스크립트 편집 창>

스크립트는 기본적으로 함수명이 없는 익명함수로 function (e) {} 안에 작성될 예정이며, 자바 스크립트 언어나 JQuery, DEWS API를 사용한다. 매개변수로 e가 넘어가며, e는 이벤트가 발생한 객체로 e를 통해 객체에 접근이 가능하다. 작성한 스크립트는 빌드를 하였을 때 그대로 html <script></script> 태그 안에 들어가게 되며, 이외에도 다른 컨트롤들에 대한 속성들은 3장 컨트롤 종류 및 기능에서 다룬다.

[제2장] 데이터소스

기본 정보

데이터 정보

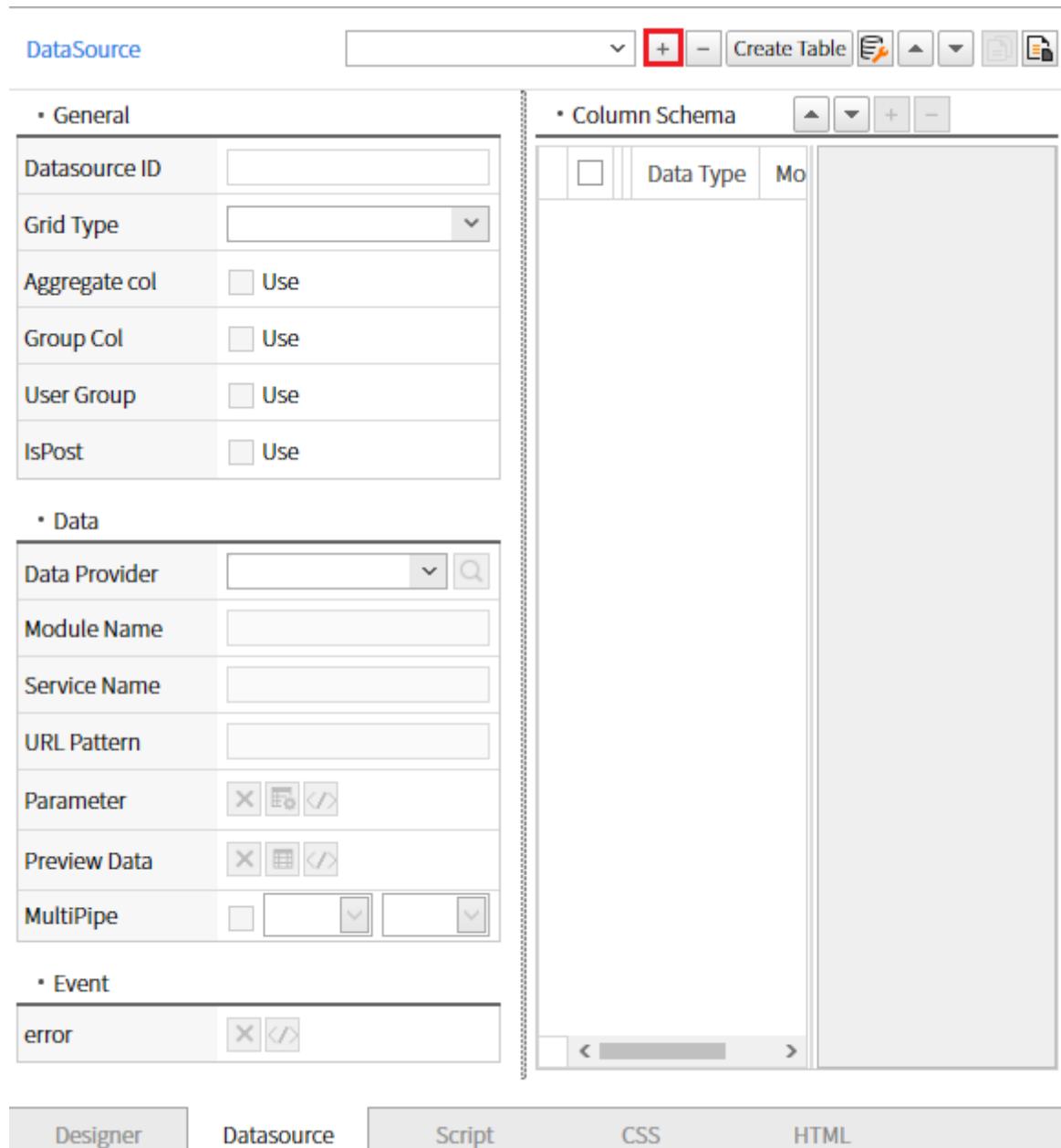
이벤트

컬럼 스키마

그리드에 데이터소스 바인딩

1. 개요

데이터소스 설정이 필요한 컨트롤의 경우 설정을 하지 않으면 비어 있는 컨트롤로 나오게 된다. 따라서 컨트롤에 데이터를 가져오기 위해서 데이터소스를 반드시 설정해주어야 한다. DEWS Front Designer에서는 데이터소스를 쉽게 편집하고 바인딩 할 수 있는 인터페이스를 제공한다.



<그림 2-1. 데이터소스 화면>

디자이너 하단에 데이터소스 탭을 클릭하면 <그림 2-1>와 같은 화면이 나오게 되며, 데이터소스를 추가하기 위해서는 우측 상단에 버튼을 클릭한다.

2. 일반 정보

디폴트로 데이터소스의 ID는 dataSource1, Grid Type은 Grid가 선택되어 있다.

* General

Datasource ID	dataSource1
Grid Type	Grid
Aggregate col	<input type="checkbox"/> Use
Group Col	<input type="checkbox"/> Use
User Group	<input type="checkbox"/> Use
IsPost	<input type="checkbox"/> Use
Paging	false
PagingCount	500

<그림 2-2. DataSource – General 화면>

속성 명	설명
Datasource ID	코드상에 데이터소스 ID로 사용
Grid Type	데이터소스를 바인딩할 컨트롤 타입 설정.(None, Grid, TreeGrid, TreeView, PivotGrid, Widget)
Aggregate col	집계(합계)열 사용 여부 설정
Group Col	그룹 집계(소계)열 사용 여부 설정
User Group	사용자 그룹 사용 여부 설정
IsPost	Back-end 서비스 API 요청 시 Post Method로 호출
Paging	페이지 처리
PagingCount	페이지 처리 시 가져올 데이터 수

<표 2-1. General 각 속성 설명>

2. 데이터 정보

Data Provider : 데이터를 검색한다. Data Service, URL, Local Data, SQL Service 등 4가지 옵션이 있다.

① Data Service를 선택할 경우

• Data

Data Provider	Data Service	<input type="button" value="🔍"/>
Module Name		
Service Name		
URL Pattern		
Parameter	<input type="button" value="X"/> <input type="button" value="☰"/> <input type="button" value="⟨⟩"/>	
Preview Data	<input type="button" value="X"/> <input type="button" value="☰"/> <input type="button" value="⟨⟩"/>	
MultiPipe	<input type="checkbox"/>	

<그림 2-3. Data Service 선택>

이 경우, Back-end 서버에 정의 되어 있는 API를 찾는다. 버튼을 눌러 서비스를 검색한다.

Search for Data Services

URL Description	휴직현황	<input type="button" value="🔍"/>																								
• DataService																										
Module Desc	URL Description	Service Name	Service Descrip	URL Pattern																						
인사관리	휴직현황 조회	PersonalAd...	인사관리-인사...	pamesr00900_list																						
<table border="1"><tr><td>Parameter</td><td>Data Schema</td></tr><tr><th>Parameter Name</th><th>Transfer Type</th><th>Data Type</th><th>Description</th></tr><tr><td>start_dt</td><td>QueryString</td><td>string</td><td>휴직시작일</td></tr><tr><td>end_dt</td><td>QueryString</td><td>string</td><td>휴직종료일</td></tr><tr><td>layoff_tp</td><td>QueryString</td><td>string</td><td>휴직구분</td></tr><tr><td>keyword</td><td>QueryString</td><td>string</td><td>검색어</td></tr></table>					Parameter	Data Schema	Parameter Name	Transfer Type	Data Type	Description	start_dt	QueryString	string	휴직시작일	end_dt	QueryString	string	휴직종료일	layoff_tp	QueryString	string	휴직구분	keyword	QueryString	string	검색어
Parameter	Data Schema																									
Parameter Name	Transfer Type	Data Type	Description																							
start_dt	QueryString	string	휴직시작일																							
end_dt	QueryString	string	휴직종료일																							
layoff_tp	QueryString	string	휴직구분																							
keyword	QueryString	string	검색어																							

<그림 2-4. 데이터서비스 검색>

데이터서비스 검색에서 키워드를 알고 있다면 검색을 할 수 있다. 만약 찾고자 하는 Service의 API 메소드 이름을 알고 있다면 URL Pattern으로 설정 한 후 검색어를 입력하여 검색 할 수 있으며, 찾은 API를 선택한 후 적용(Apply)을 누르게 되면 해당 서비스 API 정보가 자동으로 바인딩 된다. (Back-end service에서 정의한 annotation 중 @DzApi 정보 파싱)

또한 선택했을 때 Parameter와 Data Schema 탭에서 서비스 API에서 필요한 input 파라미터와 output 스키마 정보를 확인할 수 있다.

* Data

Data Provider	Data Service	<input type="button" value="X"/>	<input type="button" value="Mapping..."/>	<input type="button" value="Script..."/>
Module Name	HR			
Service Name	PersonalAdministrationManagementESRService			
URL Pattern	pamesr00900_list			
Parameter	<input type="button" value="X"/>	<input type="button" value="Mapping..."/>	<input type="button" value="Script..."/>	
Preview Data	<input type="button" value="X"/>	<input type="button" value="Mapping..."/>	<input type="button" value="Script..."/>	
MultiPipe	<input type="checkbox"/>			

<그림 2-5. 적용(apply) 후 Parameter 입력>

적용 후 Module Name, Service Name, URL Pattern 항목은 서비스 API 정보를 불러와서 자동으로 입력되며 파라미터는 Parameter의 Mapping 버튼을 눌러 파라미터 매핑 화면에서 선택하거나 Script 버튼을 눌러 코드 에디터 화면에서 직접 스크립트를 작성할 수 있다.

MultiPipe는 드롭다운리스트에 사용되며 하나의 데이터소스를 여러 드롭다운리스트 컨트롤에 설정할 때 사용한다.

② URL로 선택할 경우

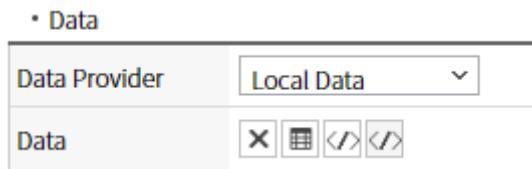
* Data

Data Provider	URL	<input type="button" value="X"/>	<input type="button" value="Mapping..."/>	<input type="button" value="Script..."/>
URL	http://	<input type="button" value="X"/>	<input type="button" value="Mapping..."/>	<input type="button" value="Script..."/>
Parameter	<input type="button" value="X"/>	<input type="button" value="Mapping..."/>	<input type="button" value="Script..."/>	
Preview Data	<input type="button" value="X"/>	<input type="button" value="Mapping..."/>	<input type="button" value="Script..."/>	

<그림 2-6. URL 선택>

<http://URL> 명으로 서비스를 가져 올 수 있다.

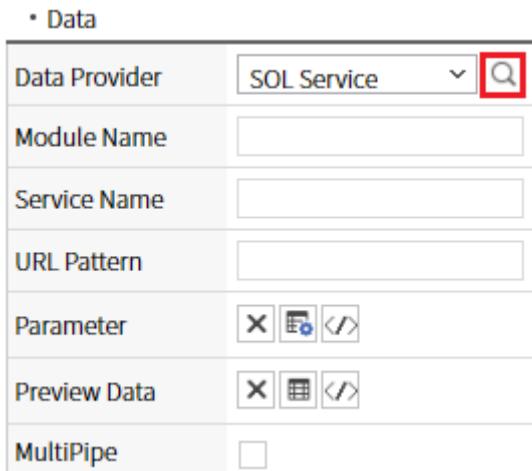
③ Local Data로 선택할 경우



<그림 2-7. Local Data 선택>

직접 스키마(컬럼)를 정의할 수 있다. 이 경우 3번째 스크립창 버튼을 눌러 스크립트에서 컬럼 스키마를 직접 해당 컬럼에 맞게 정의를 해주거나, Column Schema에서 직접 정의 후 4번째 JsonConverter 버튼을 눌러 컬럼에 데이터만 넣거나, 버튼을 눌러 DB에 있는 테이블을 컬럼스키마로 가져올 수 있다.

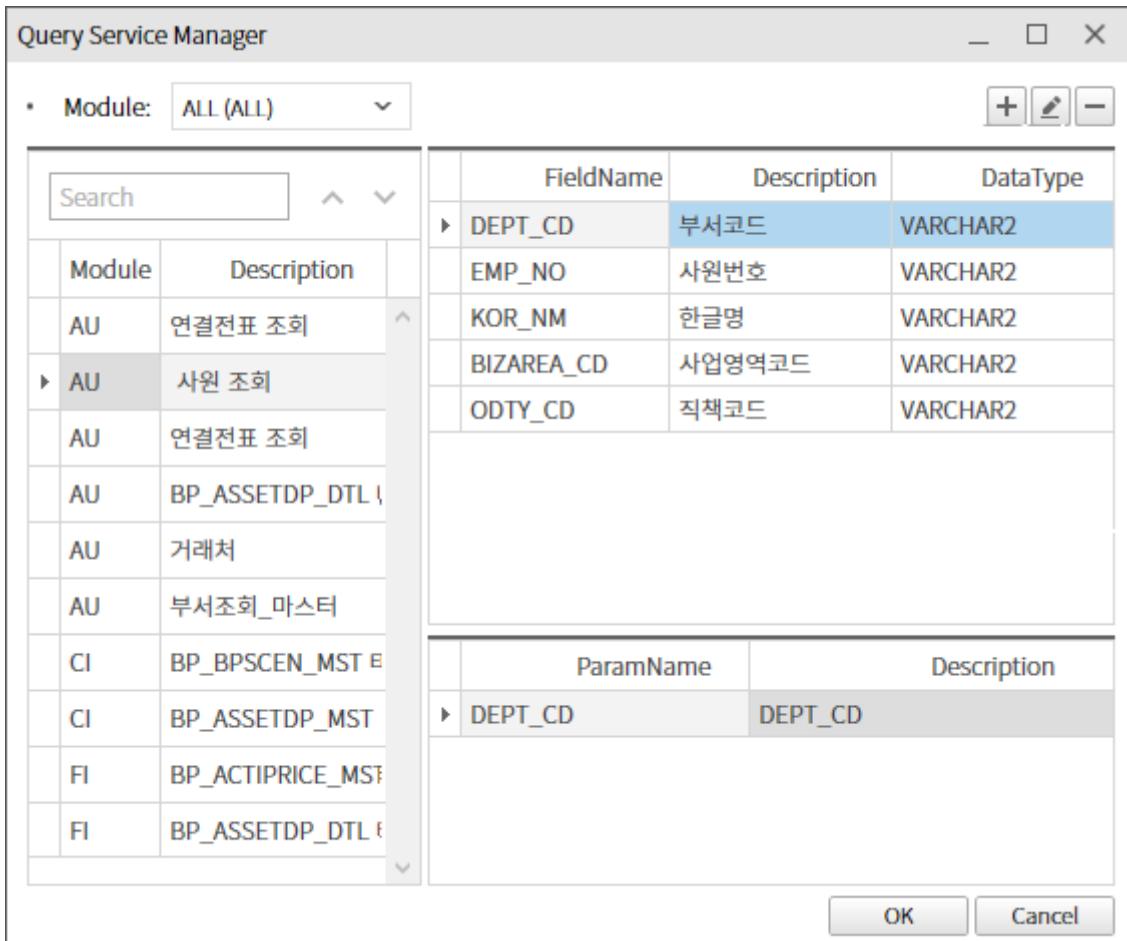
④ SQL Service로 선택할 경우



<그림 2-8. SQL Service 선택>

이 경우, 사용자가 정의한 쿼리를 관리하는 Query Service Manager를 열고 선택한다.

버튼을 눌러 쿼리 리스트를 검색한다.



<그림 2-9. Query Service Manager 화면>

Back-end 서비스 개발 없이 사용자가 QueryBuilder에서 작성한 쿼리문을 Query Service Manager 등록하고 자동 생성된 서비스를 이용하여 메뉴를 개발할 수 있다.

3. 이벤트

서비스명이 잘못 되거나 오류가 발생 했을 시 호출되는 콜백함수를 error 함수로 등록 할 수 있다. 버튼을 눌러 스크립트를 직접 작성한다.



<그림 2-10. 이벤트 화면>

4. 컬럼 스키마 (Column Schema)

Data Provider에서 Data Service를 선택 후 적용할 경우 자동으로 서비스의 Model 정보를 가져와서 생성해주며, 컬럼을 추가할 경우 버튼을 눌러서 직접 추가할 수 있다.

The screenshot shows a 'Column Schema' editor interface. At the top, there are buttons for sorting, a plus sign to add new columns, and a minus sign to remove them. To the right of these are buttons for creating a table, saving, and other operations. Below the buttons is a table with columns: Description, Column Name, Data Type, and Modify. A 'dataSource1' dropdown is also present. On the right side, there is a sidebar titled '기타' (Others) containing various configuration options like '컬럼명' (Column Name), '데이터형식' (Data Type), and '수정가능여부' (Modifiable). The table rows include:

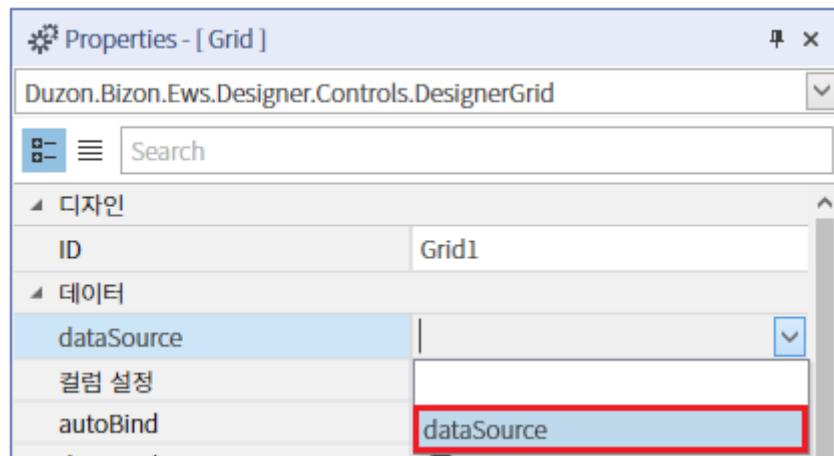
Description	Column Name	Data Type	Modify
계좌코드	FINPRODUCT_CD	string	<input checked="" type="checkbox"/>
순번	NO_SQ	number	<input type="checkbox"/>
계좌번호	FINPRODUCT_NO	string	<input checked="" type="checkbox"/>
계좌명	FINPRODUCT_NM	string	<input checked="" type="checkbox"/>
계좌명2	FINPRODUCT_NM2	string	<input type="checkbox"/>
금융상품구분코드	DPSTSE_FG_CD	string	<input type="checkbox"/>
상품유형코드	ITEM_TP_CD	string	<input type="checkbox"/>
상품유형명	NM_FUNDTYPE	string	<input type="checkbox"/>
금융상품구분명	NM_TP_DEPOSIT	string	<input type="checkbox"/>
거래지점명	PARTNER_NM	string	<input type="checkbox"/>

<그림 2-11. 컬럼 스키마>

또한 우측 상단 Create Table에서 테이블을 생성하거나 기존 사용자가 만든 테이블을 가져올 수 있으며, 데이터소스 순서 변경이나 다른 데이터소스를 복사하거나 붙여넣기 할 수 있다.

5. 데이터소스 바인딩

데이터소스 설정이 완료되면, 컨트롤에 데이터소스를 연결할 수 있다. 디자이너 탭에서 컨트롤을 생성한 후 클릭하면 프로퍼티 창에 <그림 2-12>와 같이 데이터소스를 선택해서 연결한다.



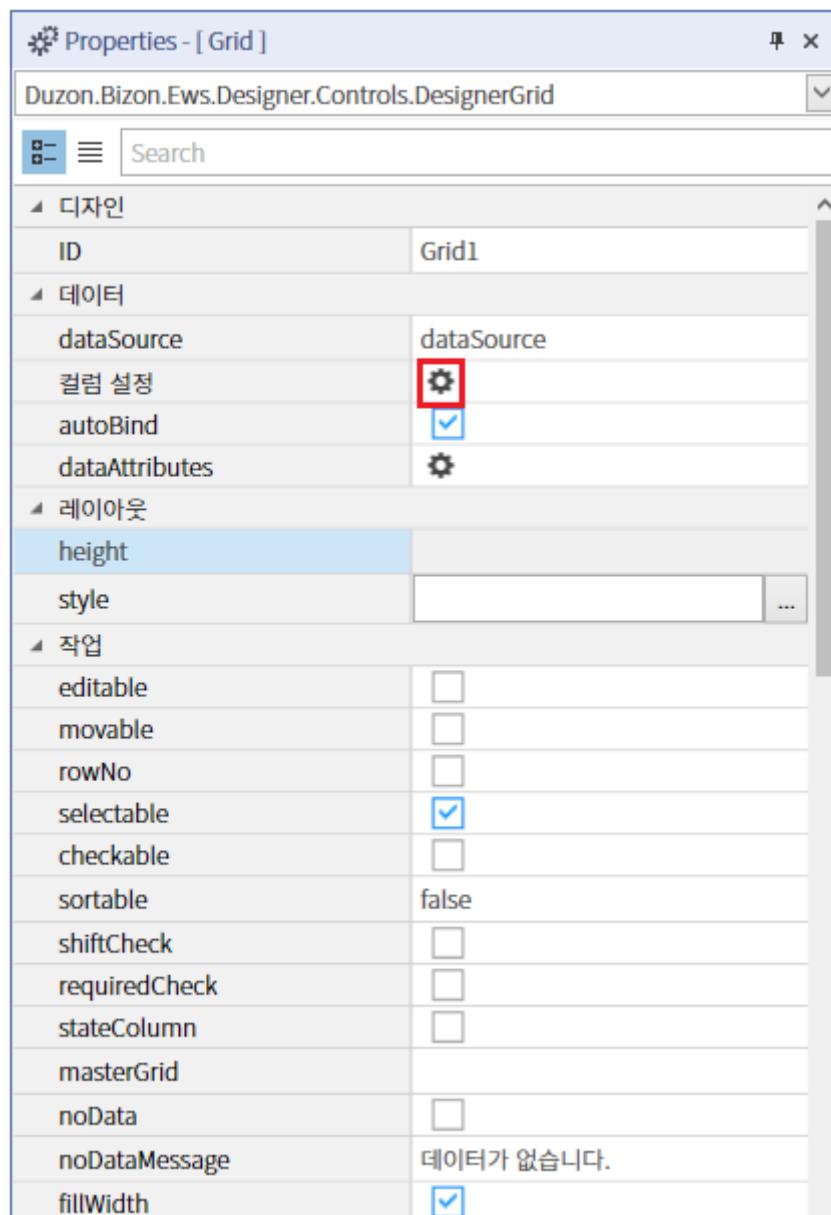
<그림 2-12. 그리드 프로퍼티 창에서 데이터소스를 연결하는 화면>

그리드의 경우 데이터소스를 선택하면 컬럼 스키마에서 정의한 컬럼들이 그리드 컬럼으로 생성되며 상세하게 알아본다.

No	회계단위명	NM_WDEPT	작성사원번호	NM_EMP_W...	회계일
1					

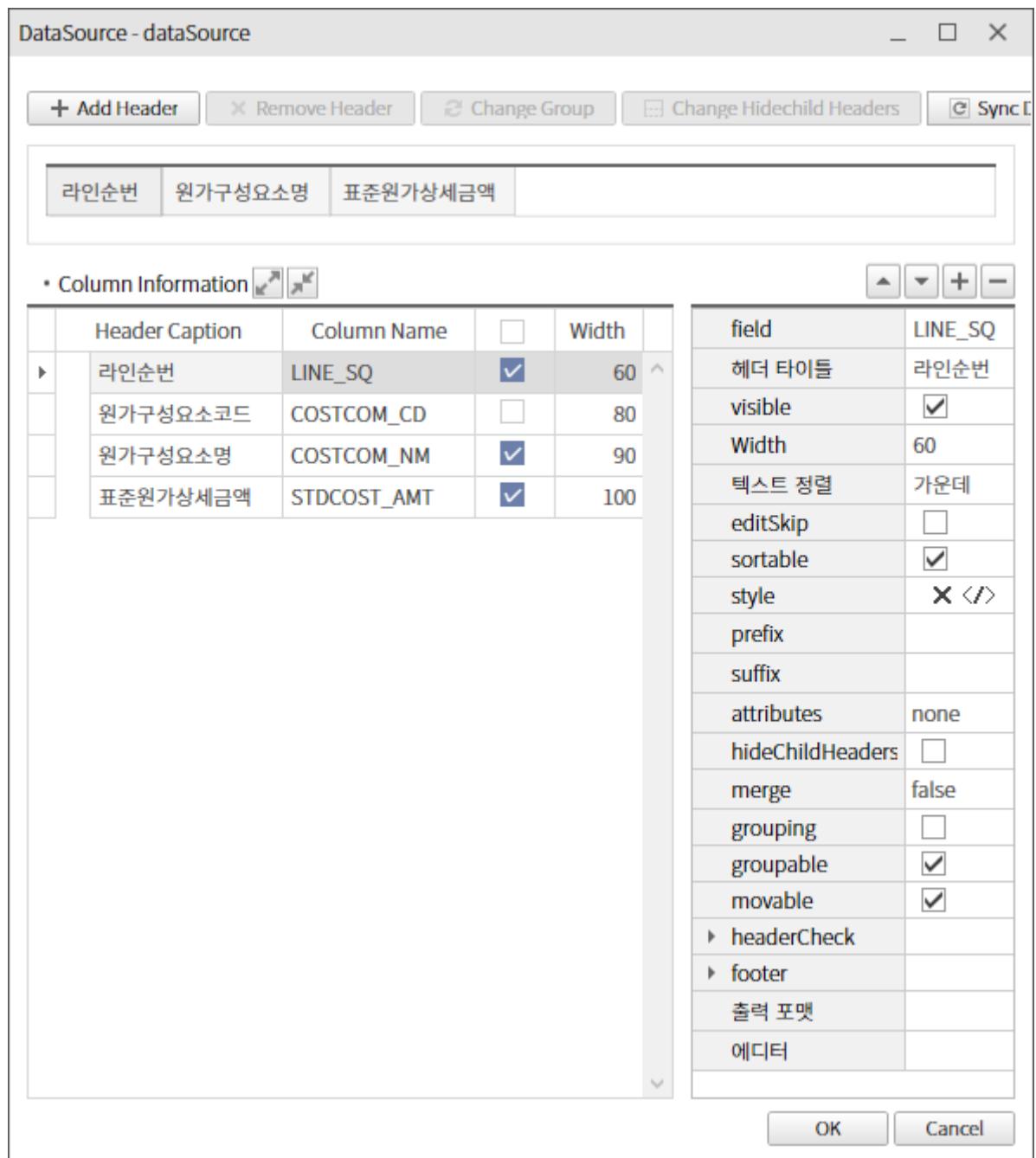
<그림 2-13. 바인딩 된 그리드에 컬럼이 표시>

여기서 컬럼별로 속성을 정해주거나, 에디터를 설정해주려면 프로퍼티 창의 컬럼설정 을 클릭한다.



<그림 2-14. 그리드 프로퍼티 창에서 컬럼 설정>

그리드 컬럼 상세 설정 화면이 열리며 컬럼 순서변경, 에디터, 출력 포맷을 정해줄 수 있다.



<그림 2-15. 그리드 컬럼 설정 다이얼로그>

속성 명	설명	
field	데이터소스 상에 나오는 컬럼명	
헤더 타이틀	그리드에 실제로 보여질 타이틀명	
visible	일반 컬럼에서 자신의 컬럼 보이기 여부	
Width	컬럼의 너비 지정 (px단위)	
텍스트 정렬	정렬 옵션 (왼쪽, 가운데, 오른쪽)	
editSkip	그리드 수정 가능한 행에만 포커스 설정 여부	
sortable	컬럼 정렬 여부	
style	그리드의 스타일 속성 지정	
prefix	그리드 데이터 앞에 설정할 문구 지정	
suffix	그리드 데이터 뒤에 설정할 문구 지정	
attributes	기본, 필수여부, 읽기전용 지정 (none, required, readonly)	
hideChildHeaders	멀티헤더컬럼일 경우 자식 컬럼명 숨김 여부	
merge	그리드의 셀 병합 여부 속성 지정	
grouping	한 데이터 행에 2줄 이상의 데이터를 표시할 경우 설정	
groupable	마우스 드래그를 통한 그룹화 방지 여부	
movable	그리드 컬럼 마우스 드래그를 통한 이동 여부	
headerCheck	used	해당 컬럼 헤더에 체크박스 설정 여부
	location	컬럼 헤더 영역에 체크박스 위치 설정
footer	type	그리드 내 footer 영역에 설정될 연산 (sum(합계), count(개수), average(평균), max(최대값), min(최소값))
	predefined	통제 환경설정 정보에서 가져오는 포맷 사용 여부
	format	날짜/시간 포맷 설정 또는 통제 환경에 포맷값
	currency_type	통화 타입 지정
	currency	거래통화코드 선택
	prefix	데이터 앞에 설정할 문구 지정
	suffix	데이터 뒤에 설정할 문구 지정
	align	footer 영역의 정렬을 설정(left,center,right)
	text	그리드 푸터(footer) 영역에 설정할 텍스트
출력 포맷	group	footer 그룹 소계 영역을 설정 (text, type, format 설정) (General의 Group Col 체크 및 그룹집계열 체크 조건이 우선 선행)
	format	컬럼에 format 옵션을 지정 (숫자 포맷, 코드-텍스트 변환, 링크, 날짜, 시간, 월, 마스크, 체크박스, 이미지, 시간, 비밀번호, function)
에디터	그리드 컬럼에 에디터를 설정 (드롭다운리스트, 텍스트박스, 멀티라인텍스트	

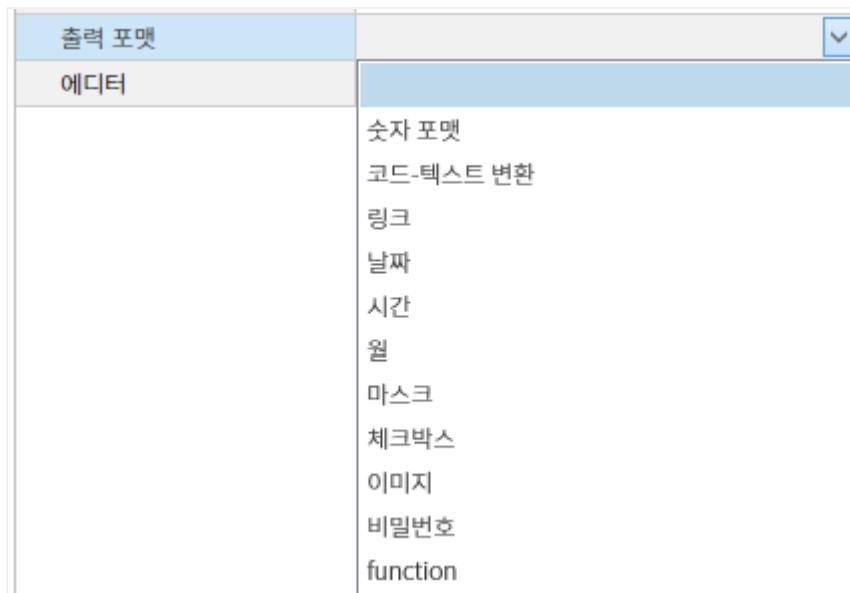
	박스, 숫자텍스트박스, 마스크텍스트박스, 코드피커, 날짜피커, 시간피커, 월피커, 우편번호, 콤보박스, 버튼, 체크박스, 비밀번호, 읽기전용)
--	--

<표 2-2. 그리드 컬럼 설정 - 각 속성에 대한 설명>

출력 포맷과 에디터에 종류에 대해 살펴본다.

1) 출력 포맷

설정할 수 있는 출력 포맷에는 숫자 포맷, 코드-텍스트 변환, 링크, 날짜, 시간, 월, 마스크, 체크박스, 이미지, 비밀번호, function이 있다.



<그림 2-16. 그리드 출력 포맷 종류>

① 숫자 포맷

◀ 출력 포맷	숫자 포맷
predefined	<input checked="" type="checkbox"/>
포맷 문자열	MA00003
currency	KRW <input type="button" value="▼"/> <input type="button" value="X"/> <input type="button" value="</>"/>

<그림 2-17. 숫자포맷>

숫자 포맷을 주게 되면 숫자 이외에 문자는 무시되고 숫자를 입력했을 시 숫자 포맷팅이 된다. predefined는 통제 환경설정 정보에서 가져오는 포맷 사용 여부이며, 포맷 문자열은 숫자 포맷 설정 또는 통제 환경 포맷값을 입력하고, currency는 거래통화코드를 선택한다

② 코드-텍스트 변환

◀ 출력 포맷	코드-텍스트 변환
데이터소스	pipeDataSource1
dataValueField	SYSDEF_CD
dataTextField	SYSDEF_NM
dataPipe	P00840

<그림 2-18. 코드-텍스트 변환>

코드-텍스트 변환은 코드에 해당하는 값을 지정한 텍스트로 변환해준다. 코드-텍스트 변환을 위해서는 데이터소스가 필요하며 데이터소스에서 정의된 컬럼스키마 중 코드, 텍스트 컬럼을 지정해주면 매칭되는 코드 필드가 존재하면 텍스트 필드로 자동으로 변환한다. dataPipe는 pipeparam과 같으며 MultiPipe를 체크한 같은 데이터소스일 때 구분할 수 있는 코드값을 지정한다.

③ 링크

◀ 출력 포맷	링크
URL 필드	BIZAREA_NM
target	
useMenu	<input checked="" type="checkbox"/> true
module	<input type="checkbox"/> X </>
menu	<input type="checkbox"/> X </>
data	<input type="checkbox"/> X </>

<그림 2-19. 링크>

링크 출력 포맷은 URL링크를 타고 다른 페이지로 이동할 수 있는 링크를 그리드 컬럼에 넣어 줄 수 있다. URL필드는 데이터소스에서 URL 데이터에 해당하는 컬럼을 정해줄 수 있다. Target은 _blank 일 경우 새 창에서 링크이동, _parent 일 경우 부모프레임에서 새 문서를 연다. useMenu는 메뉴 페이지 열기 설정을 선택하고 module은 메뉴 페이지 열기 설정 시 모듈명을 설정한다. menu는 메뉴 페이지 열기 설정 시 메뉴명을 설정하며, data는 메뉴 페이지 열기 설정 시 추가로 전달할 데이터를 설정한다. module, menu는 useMenu를 true 설정 시 필수로 설정해야 한다.

④ 날짜

◀ 출력 포맷	날짜
predefined	<input type="checkbox"/>
포맷 문자열	MM-dd-yyyy

<그림 2-20. 날짜>

날짜 포맷을 지정할 수 있다. 날짜 포맷을 지정했다고 해서 날짜 피커가 생성되는 것은 아니며, 날짜 피커는 에디터에서 설정해야 한다. predefined는 통제 환경설정 정보에서 가져오는 포맷 사용 여부이며, 포맷 문자열은 날짜 포맷 설정 또는 통제 환경 포맷값을 입력한다.

⑤ 시간

◀ 출력 포맷	시간
predefined	<input type="checkbox"/>
timeFormat	mm:ss

<그림 2-21. 시간>

시간 포맷을 지정한다. predefined는 통제 환경설정 정보에서 가져오는 포맷 사용 여부이며, timeFormat은 시간 포맷을 설정한다. Ex) HH:mm:ss

⑥ 월

◀ 출력 포맷	월
predefined	<input type="checkbox"/>
monthFormat	yyyy-MM

<그림 2-22. 월>

월 포맷을 지정한다. predefined는 통제 환경설정 정보에서 가져오는 포맷 사용 여부이며, monthFormat은 월 포맷을 설정한다. Ex) yyyy/MM

⑦ 마스크

◀ 출력 포맷	마스크
predefined	<input checked="" type="checkbox"/>
마스크 문자열	###-##-#####
SecureFormat	BIZR_NO

<그림 2-23. 마스크>

마스크 포맷을 주게 되면 해당 컬럼에 마스크가 입력되어 보여지게 된다. predefined는 통제 환경설정 정보에서 가져오는 포맷 사용 여부이며, 마스크 문자열은 마스크 포맷, SecureFormat은 개인정보 마스크 문자열 또는 통제환경 개인정보 마스크 코드 포맷을 설정한다.

⑧ 체크박스

▲ 출력 포맷	체크박스
trueValue	Y
falseValue	N

<그림 2-24. 체크박스>

그리드 해당 컬럼에 체크박스 포맷을 설정한다. trueValue는 체크되었을 때의 데이터 값을, falseValue는 언 체크 되었을 때의 데이터 값을 설정한다.

⑨ 이미지

▲ 출력 포맷	이미지
---------	-----

<그림 2-25. 이미지>

그리드 해당 컬럼에 이미지 포맷을 설정한다. 이미지 경로 데이터는 절대 경로로 지정한다.

⑩ 비밀번호

▲ 출력 포맷	비밀번호
passwordFormat	*****

<그림 2-26. 비밀번호>

패스워드 포맷을 설정하며, 생략시 기본 포맷은 ***** 이다.

⑪ function

▲ 출력 포맷	function
functionFormat	x </> 

[functionFormat]

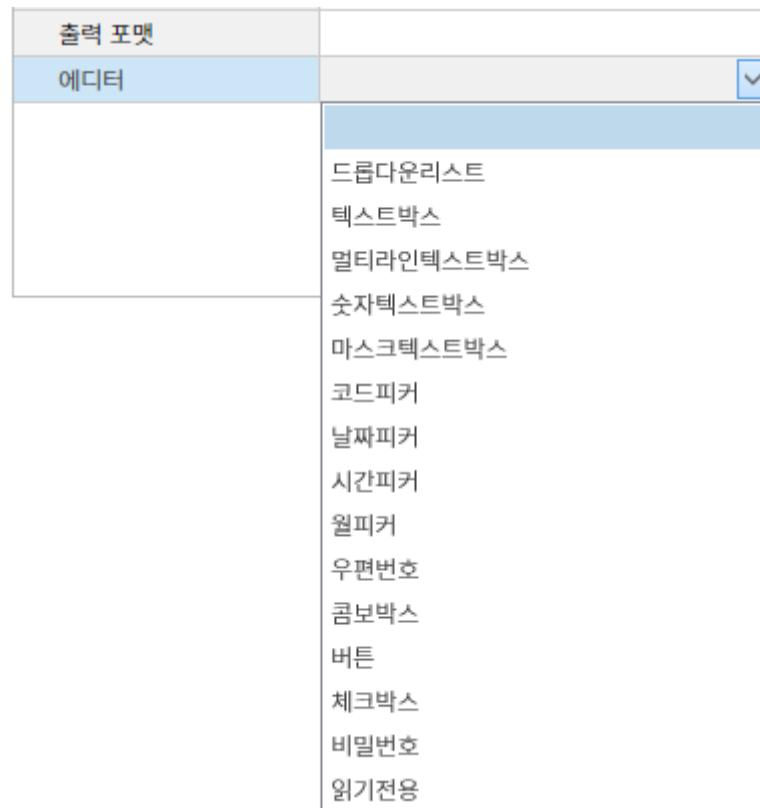
function(value) {
 return value.substring(0,2);
}

<그림 2-27. function>

function을 사용하여 포맷을 직접 설정한다.

2) 에디터

컬럼에 에디터 컨트롤을 설정하여 기본인 텍스트 이외에 다양한 형태의 데이터를 편집할 수 있다. 지정할 수 있는 에디터 컨트롤은 드롭다운리스트, 텍스트박스, 멀티라인 텍스트박스, 숫자텍스트박스, 마스크텍스트박스, 코드 피커, 날짜 피커, 시간피커, 월피커, 우편번호, 콤보박스, 버튼, 체크박스, 비밀번호, 읽기전용이 있다.



<그림 2-28. 그리드 에디터 컨트롤 종류>

① 드롭다운리스트

▲ 에디터	드롭다운리스트
dataSource	pipeDataSource1 <input type="button" value="▼"/> <input type="button" value="X"/> <input type="button" value="</>"/> <input type="button" value="⚙"/>
dataValueField	SYSDEF_CD
dataTextField	SYSDEF_NM
autoOpen	<input checked="" type="checkbox"/>
editable	<input type="button" value="X"/> <input type="button" value="</>"/>
dataPipe	P00880

<그림 2-29. 드롭다운리스트>

드롭다운리스트는 반드시 데이터를 불러올 dataSource를 설정 해주어야 한다. dataValueField는 데이터소스의 컬럼스키마에 있는 값으로 출력할 컬럼을 선택하고, dataTextField는 화면에 표시될 컬럼 즉 텍스트로 출력할 컬럼을 선택한다. dataValueField 값을 텍스트로 출력한다. autoOpen은 해당 컬럼에 포커스 설정시 리스트 자동 오픈 여부를 설정한다. editable은 해당 컨트롤 수정 여부를 설정한다. 기존 조회된 행은 수정이 불가하고 새로 추가한 행만 수정이 되어야 하는 경우 작성한다. dataPipe는 pipeparam과 같으며 MultiPipe를 체크한 데이터소스일 때 구분할 수 있는 코드값을 지정한다.

② 텍스트박스

◀ 에디터	텍스트박스
maxLength	20
editable	<input type="checkbox"/> <input checked="" type="radio"/> X </>

<그림 2-30. 텍스트박스>

기본적으로 그리드에는 텍스트박스가 포함되어 있지만, 최대 길이 설정과 컨트롤 수정 여부를 설정하기 위해 텍스트박스를 사용한다.

③ 멀티라인 텍스트박스

◀ 에디터	멀티라인텍스트박스
maxLength	20
minHeight	100
editable	<input type="checkbox"/> <input checked="" type="radio"/> X </>

<그림 2-31. 멀티라인 텍스트박스>

여러 줄(멀티라인)의 텍스트를 입력할 때 사용한다. maxLength는 최대 길이, minHeight는 최소 높이(px)를 설정한다.. editable은 해당 컨트롤 수정 여부를 설정한다.

④ 숫자텍스트박스

◀ 에디터	숫자텍스트박스
predefined	<input type="checkbox"/>
format	#,##0
currency	KRW <input checked="" type="radio"/> <input type="radio"/> X </>
min	0
max	9999
increaseType	multiple
increaseValue	1
editable	<input type="checkbox"/> <input checked="" type="radio"/> X </>

<그림 2-32. 숫자텍스트박스>

숫자만을 보여주기 위한 텍스트박스와 숫자에 포맷 문자열을 설정하기 위해 숫자텍스트박스 에디터를 사용한다. predefined는 통제 환경설정 정보에서 가져오는 포맷 사용 여부이며, 포맷 문자열은 숫자 포맷 설정 또는 통제 환경 포맷값을 입력하고, currency는 거래통화코드를 선택한다. min은 최소값 max는 최대값을 설정한다. increaseType은 +키를 눌렀을 때 증가할 연산자를 선택 하며 multiple(배수)과 plus(더하기)가 있다. increaseValue는 +키로 증가할 수를 설정한다. editable은 해당 컨트롤 수정 여부를 설정한다. 숫자 포맷 문자열을 보여주기 위해서는 출력 포맷도 숫자 포맷을 설정해야 한다.

⑤ 마스크텍스트박스

◀ 에디터	마스크텍스트박스
predefined	<input type="checkbox"/>
format	000-00-00000
editable	<input type="checkbox"/> <input type="button" value="▼"/> <input type="button" value="X"/> </>

<그림 2-33. 마스크텍스트박스>

텍스트박스에 마스크 포맷 문자열을 설정할 때 사용한다. predefined는 통제 환경설정 정보에서 가져오는 포맷 사용 여부이며, format은 에디터에 마스크 포맷을 설정한다. editable은 해당 컨트롤 수정 여부를 설정한다. 마스크 포맷 문자열을 보여주기 위해서는 출력 포맷도 마스크를 설정해야 한다.

⑥ 코드피커

도움 코드를 지정해줌으로써 해당 컬럼에 코드 피커를 넣을 수 있으며, 옵션은 <표 2-3>에서 정리 하였다.

◀ 에디터	코드피커
helpCode	H_MA_PARTNER_MST_S </>
codeField	PARTNER_CD
textField	PARTNER_NM
helpCustom	<input checked="" type="checkbox"/>
helpViewUrl	~/codehelp/MA/H_MA_PARTNER_MS...
helpApiUrl	~/api/MA/MACustomCodeHelpServic...
helpTitle	거래처도움창
helpSize	대
helpHeight	
helpWidth	
helpParams	<input type="button" value="X"/> <input style="border: 1px solid black; padding: 2px 5px; margin-left: 10px;" type="button" value="却是"/> </>
코드 컬럼	
텍스트 컬럼	
콜백 함수	<input type="button" value="X"/> </>
editable	<input type="button" value="true"/> <input type="button" value="▼"/> <input type="button" value="X"/> </>

<그림 2-34. 코드피커>

속성 명	설명
helpCode	그리드 공통도움창명 설정
codeField	그리드 코드피커의 코드에 매칭시킬 컬럼명 설정
textField	그리드 코드피커의 텍스트에 매칭시킬 컬럼명 설정
helpCustom	그리드 공통도움창이 아닌 사용자 정의도움창 사용 여부
helpViewUrl	그리드 사용자 정의 도움창 페이지 HTML 경로 설정
helpApiUrl	그리드 사용자 정의 back-end 도움창 API 경로 설정
helpTitle	코드 도움창의 제목 지정
helpSize	코드 도움창의 크기 (사용자지정, 소, 중, 대, 특대)
helpHeight	도움 창 크기가 사용자 지정일 때 높이 지정
helpWidth	도움 창 크기가 사용자 지정일 때 너비 지정
helpParams	cd_nation=1과 같이 코드 도움이 열릴 때 전달할 파라미터 지정
코드 컬럼	그리드에 도움창에서 가져온 코드값이 설정될 컬럼
텍스트 컬럼	그리드에 도움창에서 가져온 텍스트값이 설정될 컬럼
콜백 함수(callback)	코드 도움이 열리고 사용자가 적용버튼을 눌렀을 때 호출될 콜백 함수 작성. rowData(현재 행의 데이터), pickerData(코드피커에서 전달된 데이터), index(코드도움이 설정된 행 인덱스) 옵션이 있음
editable	해당 컬럼의 수정 여부 설정

<표 2-3. 코드피커 각 속성 설명>

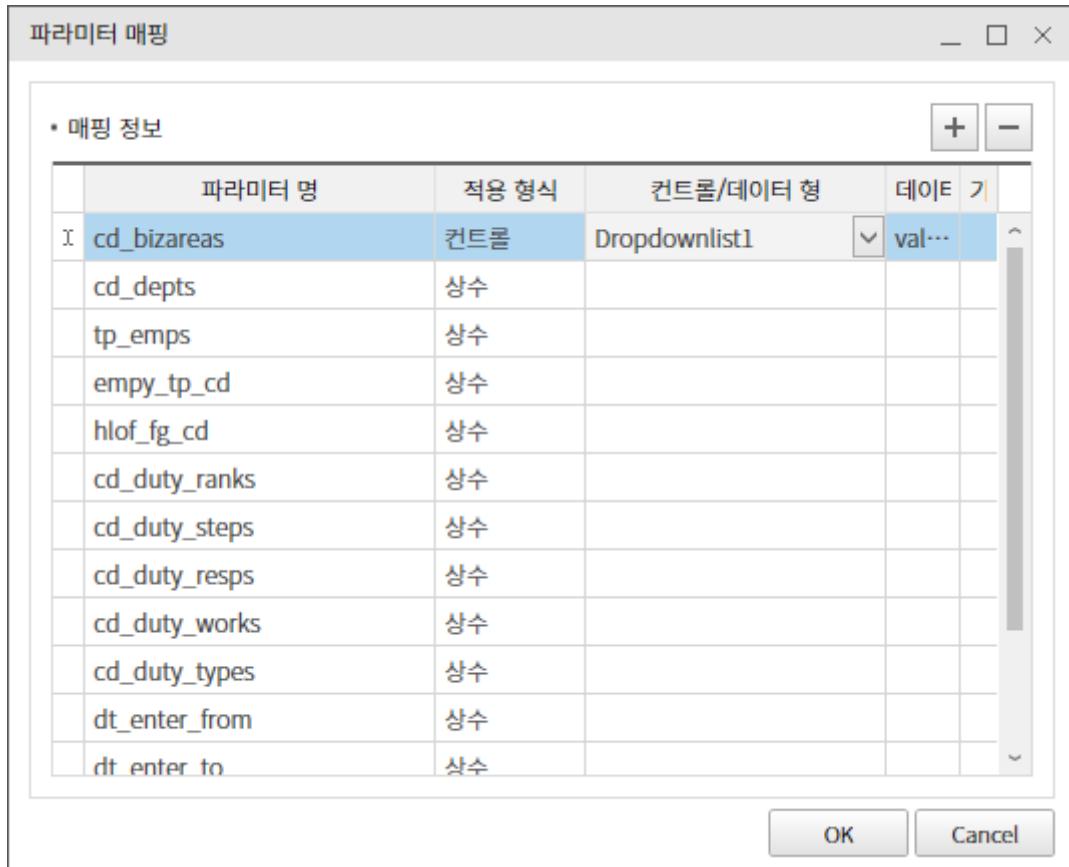
코드피커의 파라미터는 파라미터 매핑  창에서 제공하는 인터페이스를 사용할 수도 있고 직접 스크립트 에디터를 열어 코드를 작성할 수도 있다.

⑦ 파라미터 매핑



<그림 2-35. helpParams 파라미터 매핑 버튼>

가운데 버튼을 눌러 파라미터 매핑 창을 열 수 있으며, 인터페이스는 데이터소스 파라미터 매핑 창과 동일하다.



<그림 2-36. 파라미터 매핑>

위와 같이 파라미터 명을 적어주고 적용 형식으로 컨트롤이나 상수, 환경변수를 정해 줄 수 있다.

⑧ 스크립트 에디터

스크립트 에디터를 통해 직접 코드를 작성 할 수도 있으며, </> 버튼을 눌러 에디터창을 열어 코드를 작성할 수 있다.

⑦ 날짜피커

◀ 에디터	날짜피커
predefined	<input type="checkbox"/>
format	MM-dd-yyyy
editable	<input type="checkbox"/> X </>

<그림 2-37. 날짜피커>

날짜 피커 에디터를 그리드에 표시하며, 포맷을 없을 경우 기본값이 설정된다. predefined는 통제 환경설정 정보에서 가져오는 포맷 사용 여부이며, 포맷 문자열은 날짜 포맷 설정 또는 통제 환경 포맷값을 입력한다. editable은 해당 컨트롤 수정 여부를 설정한다. 날짜 피커를 설정한 경우 출력 포맷을 설정하지 않아도 날짜 피커 포맷으로 표시된다.

⑧ 시간피커

◀ 에디터	시간피커
format	mm:ss
editable	<input type="checkbox"/> <input type="button"/> X </>

<그림 2-38. 시간피커>

그리드 컬럼에 시간 에디터를 설정하며, format은 시간 포맷을 설정한다. editable은 해당 컨트롤 수정 여부를 설정한다. 출력 포맷도 시간을 설정해야 한다.

⑨ 월피커

◀ 에디터	월피커
predefined	<input type="checkbox"/>
format	yyyy-MM
editable	<input type="checkbox"/> <input type="button"/> X </>

<그림 2-39. 월피커>

그리드 컬럼에 시간 에디터를 설정한다. predefined는 통제 환경설정 정보에서 가져오는 포맷 사용 여부이며, format은 월 포맷 설정 또는 통제 환경 포맷값을 설정한다. editable은 해당 컨트롤 수정 여부를 설정한다. 날짜 피커와 마찬가지로 월 피커를 설정한 경우 출력 포맷을 설정하지 않아도 월 피커 포맷으로 표시된다.

⑩ 우편번호

▲ 에디터	우편번호
addressType	zipcode
callback	X </>
editable	<input type="checkbox"/> X </>

<그림 2-40. 우편번호>

```
1 function (rowData, data, index) {  
2     self.grid.setCellValue(index, 'zipcode', data.zipcode);  
3     self.setCellValue(index, 'jibun', data.jibun);  
4     self.setCellValue(index, 'street', data.street);  
5 }
```

<그림 2-41. 우편번호 callback 스크립트 작성>

그리드 컬럼에 우편번호 에디터를 설정한다 addressType은 우편번호 에디터에 설정할 타입명 설정하며 zipcode(우편번호), jibun(지번), street(도로명)가 있다. callback 데이터가 변경이 일어났을 때 처리할 콜백 함수를 작성하며, editable은 해당 컨트롤 수정 여부를 설정한다.

⑪ 콤보박스

▲ 에디터	콤보박스
dataSource	pipeDataSource1
dataValueField	SYSDEF_CD
dataTextField	SYSDEF_NM
editable	<input type="checkbox"/> X </>

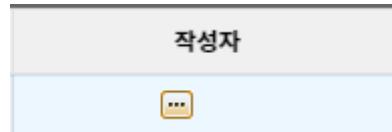
<그림 2-42. 콤보박스>

그리드 컬럼에 콤보박스를 설정한다. 드롭다운리스트는 에디터에 입력이 불가하지만, 콤보박스는 사용자가 직접 값을 입력할 수 있다. 드롭다운리스트와 마찬가지로 반드시 데이터를 불러올 dataSource를 설정 해주어야 한다. dataValueField는 데이터소스에서 값으로 출력할 컬럼을 선택하고, dataTextField는 화면에 표시될 컬럼 즉 dataValueField의 값을 텍스트로 출력할 컬럼을 선택한다. editable은 해당 컨트롤 수정 여부를 설정한다. 콤보 박스보다 드롭다운리스트를 많이 사용한다

⑫ 버튼

◀ 에디터	버튼
buttonWidth	50
buttonHeight	25
basicImg	https://dews.comet.duzon.net/dews/...
hoverImg	https://dews.comet.duzon.net/dews/...
clickImg	https://dews.comet.duzon.net/dews/...
click	X </>

<그림 2-43. 버튼>



<그림 2-44. 버튼 미리보기 화면>

그리드 컬럼에 버튼 에디터를 설정하며, buttonWidth는 버튼 너비, buttonHeight는 높이, basicImg는 기본 버튼 이미지, hoverImg는 hover 시 이미지, clickImg는 click 버튼 이미지 등을 설정한다. Click은 버튼을 클릭 했을 경우 발생하는 이벤트를 설정한다.

⑬ 체크박스

◀ 에디터	체크박스
trueValue	TRUE
falseValue	FALSE
editable	X </>

<그림 2-45. 체크박스>

trueValue는 체크했을 때 데이터 설정하며, falseValue는 체크해제했을 때 데이터를 설정한다. editable은 해당 컨트롤 수정 여부를 설정한다

⑭ 비밀번호

에디터	비밀번호

<그림 2-46. 비밀번호>

그리드 컬럼에 패스워드 에디터를 설정한다. 출력 포맷에도 비밀번호를 설정해야 한다

⑯ 읽기전용



<그림 2-47. 읽기전용>

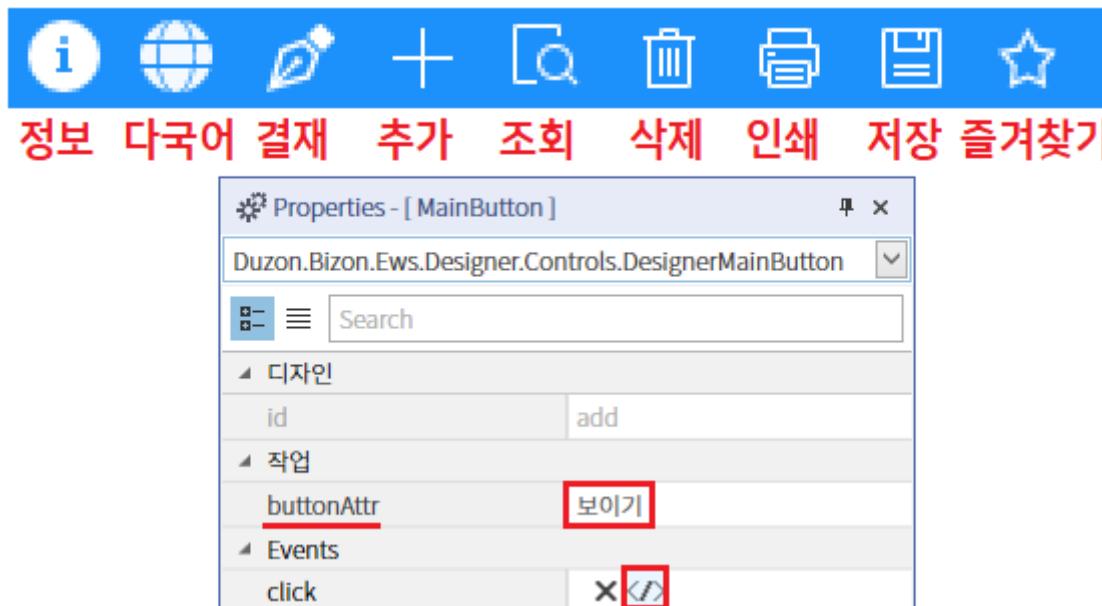
그리드 해당 컬럼을 읽기 전용으로 만든다.

[제3장] 컨트롤 종류 및 기능

상단 메인 버튼
컨테이너
폼 컨트롤
데이터

1. 상단 메인 버튼

상단 우측에 9가지 버튼이 있으며, 버튼을 선택했을 때 id값은 기본적으로 설정되어 있으며 이 값은 DEWS에서 사용하는 고유한 값이므로 변경했을 경우 이벤트가 일어나지 않으며, 삭제는 불가하다.



<그림 3-1. 상단 메인 버튼 및 조회 프로퍼티>

버튼	설명(id)
정보	페이지 정보 확인 이벤트 (information)
다국어	다국어 이벤트 (localize)
결재	결재할 때 발생하는 이벤트 (approval)
추가	행을 추가할 때 발생하는 이벤트 (add)
조회	데이터 조회 할 때 발생하는 이벤트 (search)
삭제	선택 또는 선택된 행들을 삭제할 때 발생하는 이벤트 (delete)
인쇄	조회된 화면을 인쇄할 때 발생하는 이벤트 (print)
저장	변경한 내용을 저장할 때 발생하는 이벤트 저장 (save)
즐겨찾기	해당 메뉴를 즐겨 찾기로 추가할 때 발생하는 이벤트 (configure)

<표 3-1. 버튼 설명 및 id >

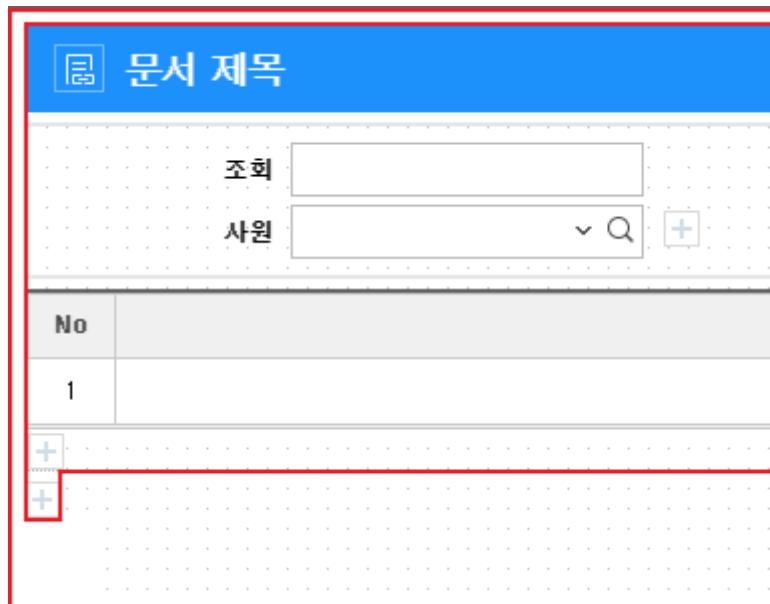
① 메인 버튼 공통 속성

프로퍼티	설명
buttonAttr	버튼 상태 속성 옵션 (비활성화, 보이기, 숨기기)
click	클릭 했을 때 불러질 이벤트

<표 3-2. 메인 버튼 공통 속성>

2. Canvas

폼(Form) 처럼 페이지의 가장 상위 컨트롤인 메인 화면이며, 최상단 공백 영역 혹은 페이지 하단의 공백영역을 클릭해서 선택할 수 있다.



<그림 3-2. Canvas 선택>

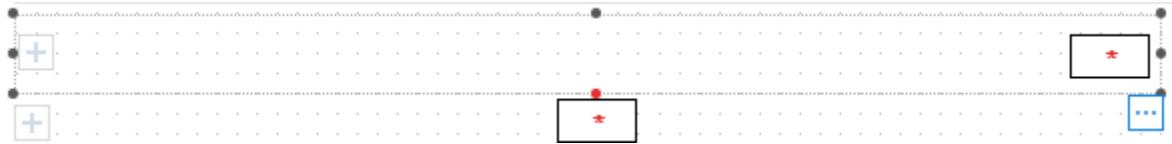
① Canvas 속성

프로퍼티	설명
title	페이지 제목 설정
preReady	페이지 앞부분에 실행되는 스크립트 작성. 주로 페이지 로드 될 때 초기 설정(전역 변수나 함수, 컨트롤 설정)
Ready	페이지 뒷부분에 실행되는 스크립트 작성. 컨트롤 간의 관계 설정이나 초기값 선택에 사용
preDesignSummary	설계 문서의 앞부분 내용 작성
postDesignSummary	설계 문서의 뒷부분 내용 작성
function	설계 문서 function 에 들어가는 부분
DesingerComment	컨트롤에 대한 간단한 설명
DocumentID	설계문서에 사용되는 페이지 이름
jira	Atlassian Jira 정보 조회

<표 3-3. Canvas 속성>

3. 컨테이너

1) 컨테이너 패널



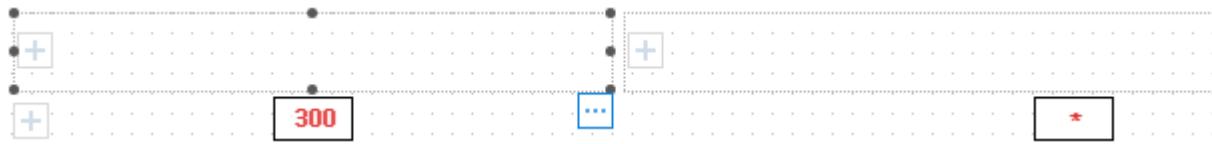
<그림 3-3. 컨테이너 패널>

컨테이너 패널은 데이터 영역을 만들고 화면을 좌우 분할하고 특정 높이 및 전체 화면에 높이를 설정할 수 있는 컴포넌트이다. 패널을 분할하기 위해서는 우측하단에 나오는 더보기 버튼을 클릭하여 Split을 클릭한다.



<그림 3-4. 컨테이너 패널 분할>

분할을 하게 되면 컨테이너 패널이 2개로 분할되어 각각의 영역에 다른 컨트롤들을 배치할 수 있다.



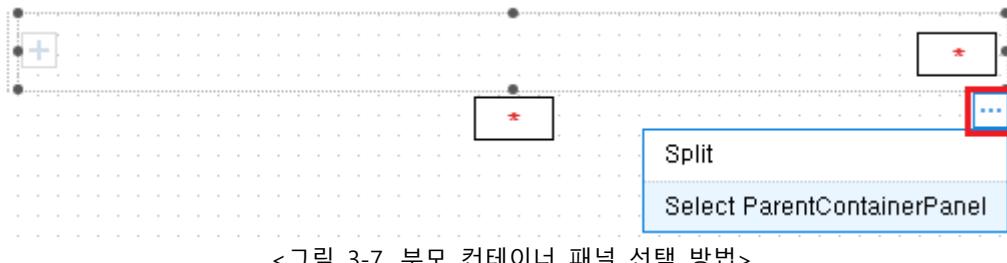
<그림 3-5. 분할된 컨테이너 패널>

분할은 하나의 페이지에 여러 개의 그리드가 있을 때 유용하며, 왼쪽 컨테이너 패널에 마스터 그리드, 오른쪽 컨테이너 패널에 디테일 그리드를 추가하여 메뉴를 작성할 수 있다.



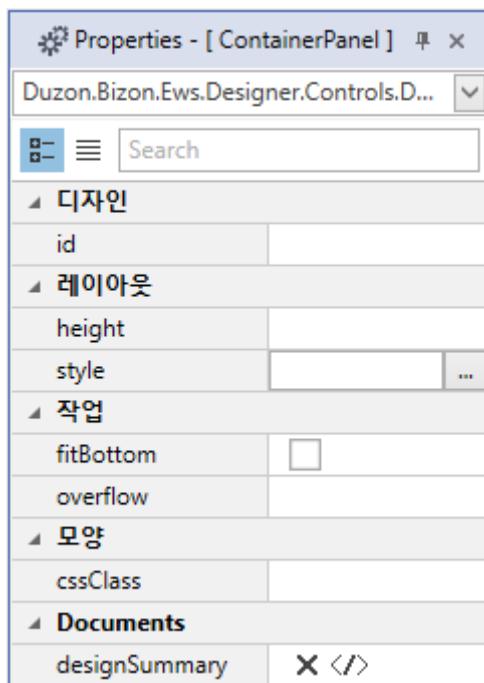
<그림 3-6. 분할된 컨테이너 그리드 예제>

컨테이너 패널이 분할되었을 때 부모 컨테이너 패널에 속성을 설정하기 위해서는 버튼을 눌러 Select ParentContainerPanel을 활성화한 후 프로퍼티 창에서 설정 값을 변경해야 한다.



<그림 3-7. 부모 컨테이너 패널 선택 방법>

① 컨테이너 패널 속성



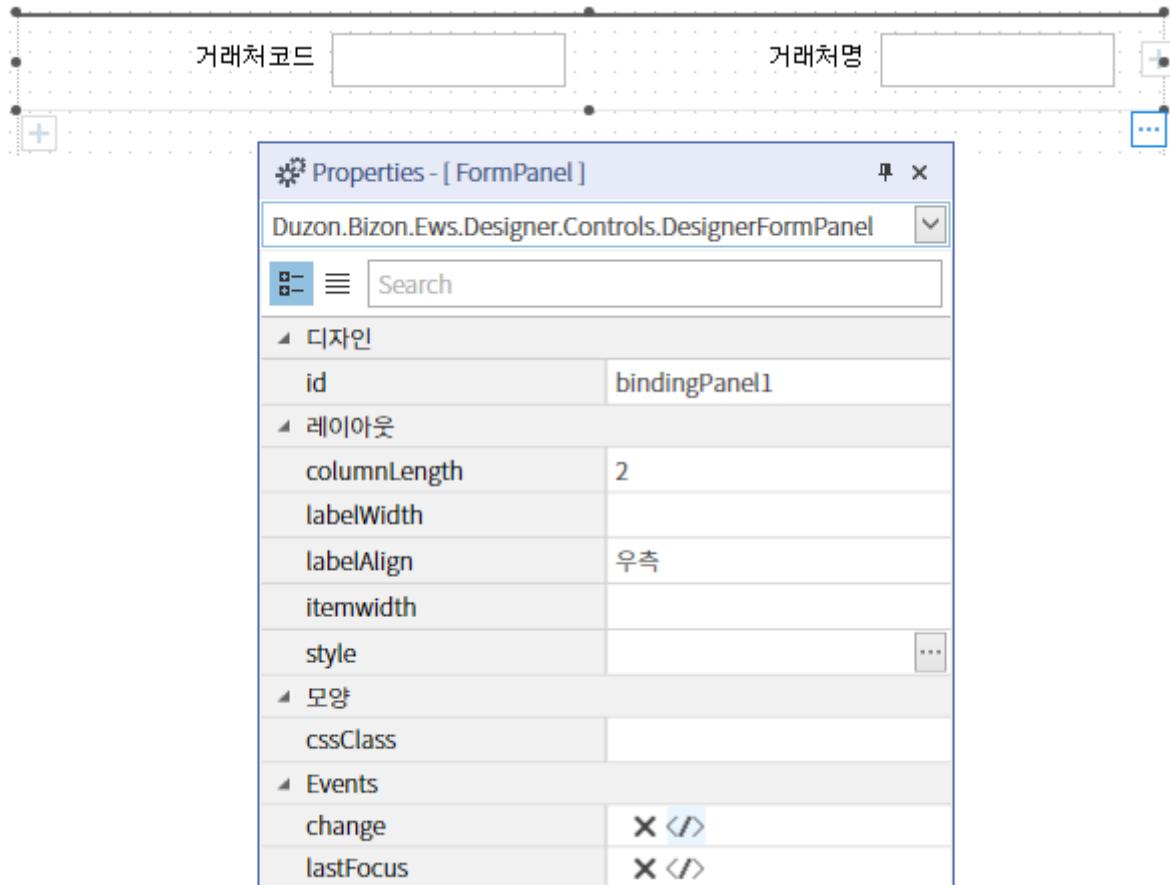
<그림 3-8. 컨테이너 패널 속성 화면>

프로퍼티	설명
id	해당 컨트롤을 구분할 수 있는 고유 ID
height	높이 지정
style	스타일 설정
fitBottom	자동높이 설정(fitBottom) 옵션을 줄 경우 컨테이너패널이 자동으로 화면의 하단높이에 맞춤
overflow	오버플로 되었을 때 화면 출력옵션 (보이기, 숨기기, 스크롤)
cssClass	컨테이너 패널에 CSS Class 명 정의
dataAttributes	사용자 지정 속성 추가(data-*로 시작)
designSummary	컨테이너 패널을 정의하기 위한 비즈니스 문서 작성

<표 3-4. 컨테이너 패널 주요 속성 및 설명>

2) 폼 패널

컨테이너 패널 생성 후 추가할 수 있으며 데이터의 입력/수정/조회를 수행하는 영역을 위한 컨테이너 컴포넌트이다. 폼 패널에는 다양한 종류의 폼 컨트롤들이 들어갈 수 있으며 컨디션 패널과 달리 한 라인에 2개의 컨트롤이 들어간다. ColumnLength 속성 설정을 통해 한 줄에 몇 개의 컨트롤이 들어갈지 지정할 수 있다.



<그림 3-9. 폼 패널 화면>

① 폼 패널 주요 속성

프로퍼티	설명
columnLength	한 라인에 추가할 수 있는 컨트롤 개수 지정 (기본값은 2개)
labelWidth	라벨 너비 지정
labelAlign	라벨의 위치 옵션 (좌측, 중앙, 우측)
itemwidth	폼아이템(FormItem)의 전체 너비 지정
change	폼 패널 내에 있는 컨트롤 중 하나라도 변경이 되었을 경우 발생하는 이벤트

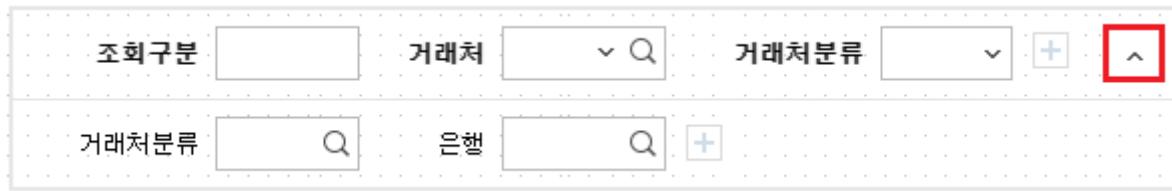
<표 3-5. 폼 패널 속성>

3) 컨디션 패널

컨디션 패널은 조회 조건 영역을 만들고 조회조건의 컨트롤들을 올려 두는데 사용되며, 컨디션 패널에는 필수영역, 옵션영역이 존재한다. 옵션영역 생성은 더보기 버튼을 누르고 OptionArea 를 체크하거나 프로퍼티 창에서 optional 에 체크한다



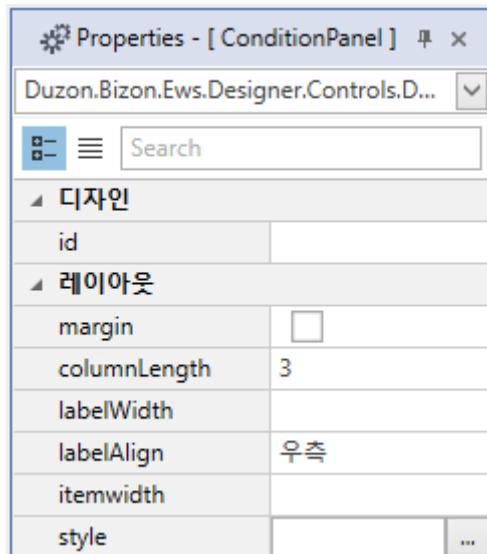
<그림 3-10. 옵션 영역 생성>



<그림 3-11. 옵션 영역 보기>

<그림 3-10>과 같이 ▾ 버튼이 컨디션 패널의 우측에 생성이 되고, 이 버튼을 눌렀을 때 <그림 3-11>과 같이 펼쳐져서 새로운 컨트롤들을 추가할 수 있다.

① 컨디션 패널 속성

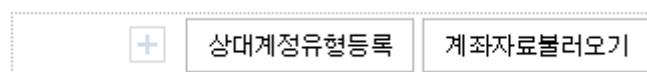


<그림 3-12. 컨디션 패널 속성 화면>

프로퍼티	설명
margin	컨디션 패널 상단에 여백을 주는 옵션
columnLength	한 라인에 추가할 수 있는 컨트롤 개수 지정 (기본값 3개)
labelWidth	라벨 너비 지정
labelAlign	라벨의 위치 옵션 (좌측, 중앙, 우측)
itemwidth	폼아이템(FormItem)의 전체 너비 지정
optional	옵션영역 사용여부
optionalcollapsed	페이지 실행 시 옵션 영역 감추기/펼치기 여부

<표 3-6. 컨디션 패널 주요 속성 및 설명>

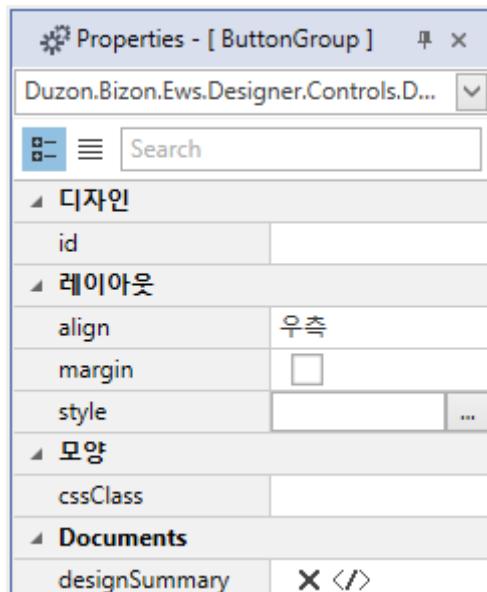
4) 버튼 그룹



<그림 3-13. 버튼그룹 화면>

버튼이나 피커류 등의 단일 컨트롤들을 올릴 수 있는 컨테이너이다. 버튼 그룹의 속성창에서 정렬 옵션을 우측, 좌측으로 할지 선택 할 수 있다. 보통 버튼들을 묶어 한번에 관리하기 위해 사용된다.

① 버튼그룹 속성

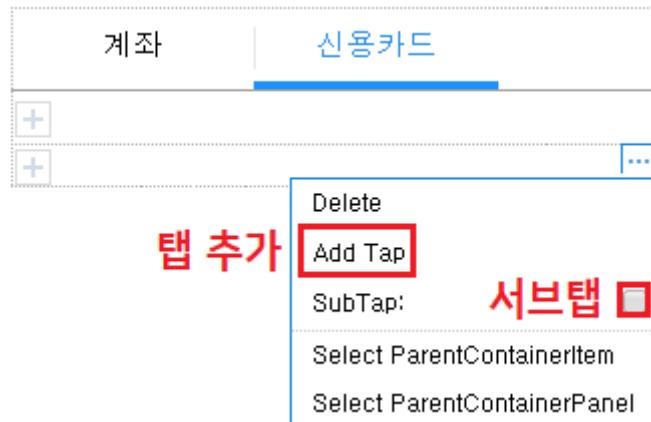


<그림 3-14. 버튼그룹 속성>

프로퍼티	설명
align	버튼의 정렬 속성 (우측, 좌측)
margin	버튼 그룹 상단에 여백을 주는 옵션

<표 3-7. 버튼그룹 주요 속성 및 설명>

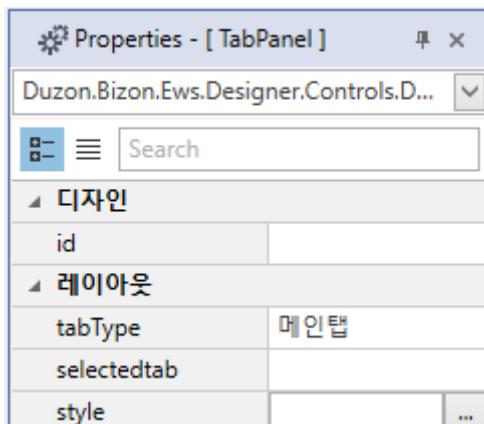
5) 탭 패널



<그림 3-15. 탭 패널 화면>

컨테이너 패널 생성 후 추가할 수 있으며 탭 패널 내부에는 다양한 컨트롤들을 넣을 수 있다. 탭 타이틀을 변경하고 싶다면 탭 타이틀을 클릭하여 프로퍼티 창에 탭 이름을 변경 할 수 있다. 탭을 추가하고 싶다면 버튼을 눌러 Add Tap을 클릭한다. 또한 탭패널을 서브 탭으로 설정 하고 싶다면 SubTab 설정 여부를 체크하거나 프로퍼티 창에서 tabType을 서브 탭으로 선택하면 서브 탭 패널로 바뀐다.

① 탭 패널 속성

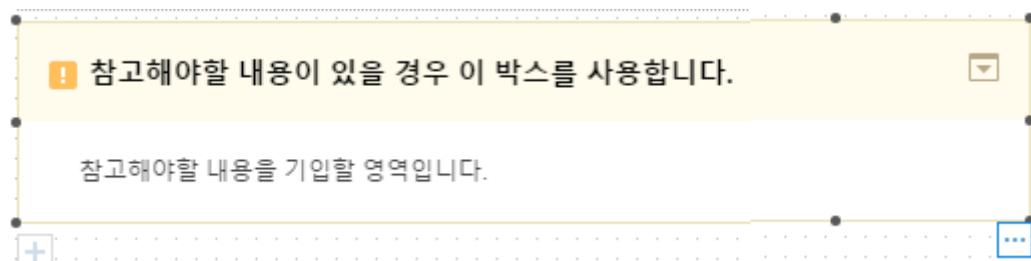


<그림 3-16. 탭 패널 속성 화면>

프로퍼티	설명
tabType	메인 탭, 서브 탭 중에 선택하는 옵션
selectedtab	초기화 시 선택되어 있는 탭 아이템 설정 (0부터 시작하는 인덱스나 탭 아이템 아이디 사용)
cssClass	탭 패널에 CSS Class 명 정의
designSummary	탭 패널을 정의하기 위한 비즈니스 문서 작성

<표 3-8. 탭 패널 주요 속성 및 설명>

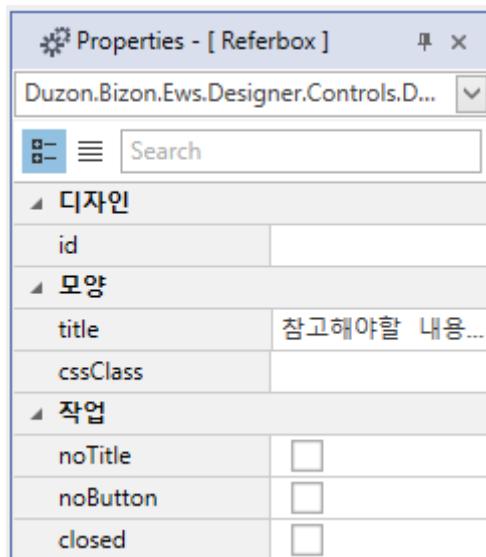
6) 참고박스



<그림 3-17. 참고박스 화면>

참고박스(ReferBox)를 사용하여 특정 내용에 대해 참고사항으로 정의할 수 있다. 현재 제목만 입력이 가능하고, 내용을 입력하기 위해서는 HTMLcontrol를 사용하여 직접 HTML 코드로 입력해야 한다.

① 참고박스 속성

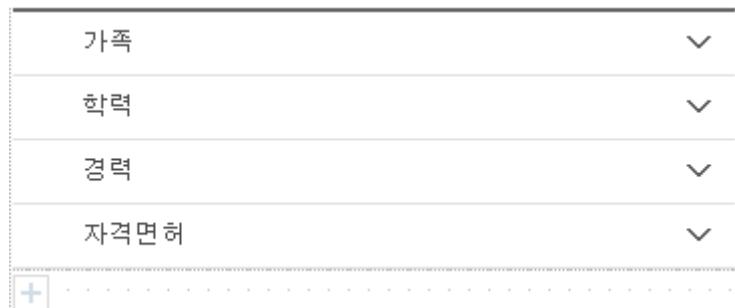


<그림 3-18. 참고박스 속성 화면>

프로퍼티	설명
title	참고박스 제목 설정
notitle	참고박스 제목의 사용여부 설정 (기본값 false)
noButton	참고박스 접힘 기능 버튼 사용여부 설정 (기본값 false)
closed	참고박스 내용 접힘 여부 설정 (기본값 false)

<표 3-9. 참고박스 주요 속성 및 설명>

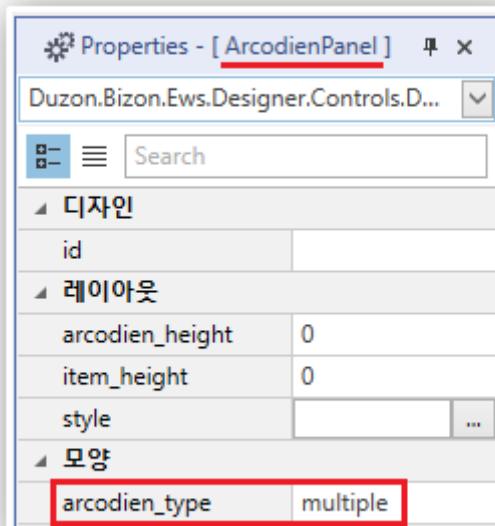
7) 아코디언



<그림 3-19. 아코디언 화면>

아코디언도 폼 패널과 탭 패널처럼 컨테이너 패널 생성 후 추가할 수 있으며 내용이 접히거나 펼쳐질 수 있도록 구성되어 있는 컨테이너이다. 아코디언 아이템은 제목과 내용으로 구성되어 있고 제목 영역 클릭시 내용이 접히거나 펼쳐지는 기능을 한다.

① 아코디언 패널 속성(ArcodienPanel)

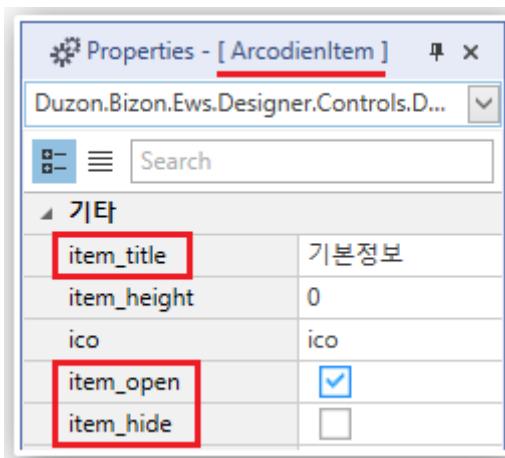


<그림 3-20. 아코디언 패널 속성 화면>

프로퍼티	설명
arcodien_height	아코디언의 높이 설정.
item_height	아코디언 아이템의 기본 높이 설정
arcodien_type	아코디언의 아이템 확장 방법 설정(single, multiple)

<표 3-10. 아코디언 주요 속성 및 설명>

② 아코디언 아이템 속성(ArcodienItem)

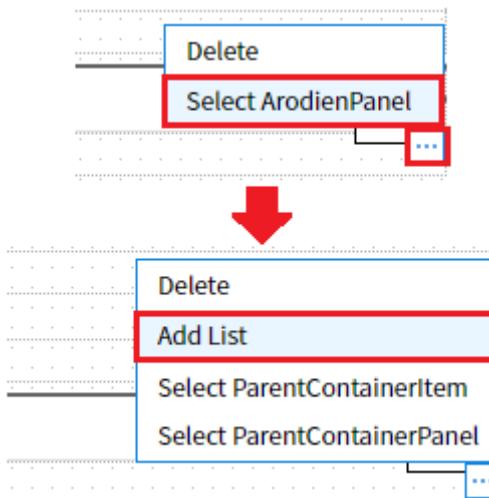


<그림 3-21. 아코디언 아이템 속성 화면>

프로퍼티	설명
item_title	아코디언 아이템의 문구 설정.
item_height	아코디언 아이템의 기본 높이 설정
ico	아코디언 아이템의 이미지를 설정합니다.
item_open	아코디언 아이템의 초기 열림 상태 설정
item_hide	아코디언 아이템의 숨김 상태 설정

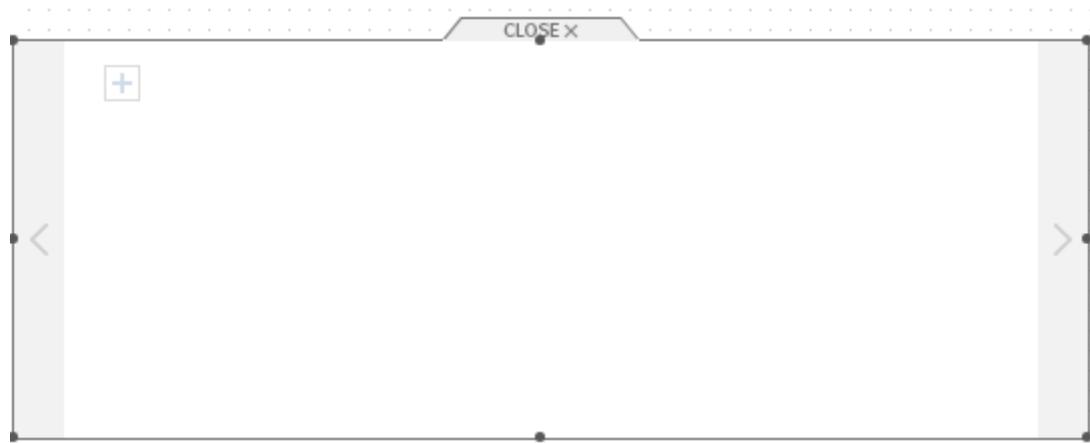
<표 3-11. 아코디언 주요 속성 및 설명>

아코디언 리스트 추가는 새로운 아코디언 패널을 계속 추가하지 않고 아코디언 아이템에서 더보기 버튼을 눌러 Select ArcodienPanel을 선택하고 Add List를 눌러 추가한다.



<그림 3-22. Acrodien List 추가>

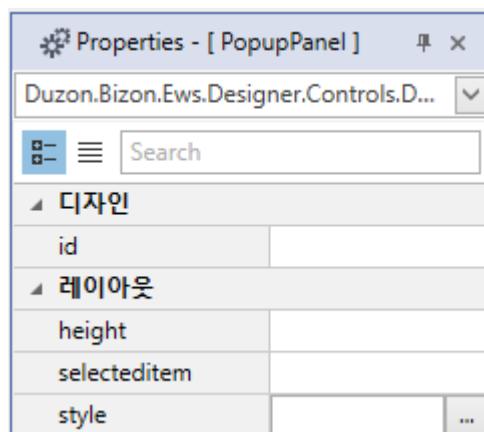
8) 팝업 패널



<그림 3-23. 팝업패널 화면>

팝업 패널은 지정한 영역에 슬라이드 형태로 팝업 패널을 실행시키고 메인 탭, 서브 탭을 지정할 수 있다. 하위 팝업 패널에는 메인 탭 3 개가 존재하고 좌측과 우측에 있는 방향키를 클릭하면 서브 탭으로의 이동이 가능하다.

① 팝업 패널 속성



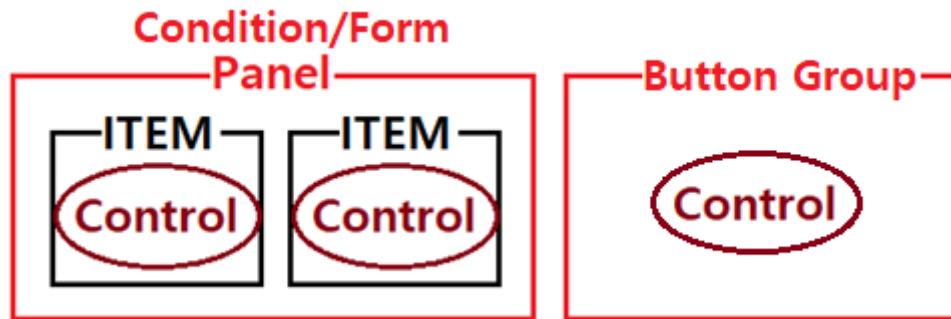
<그림 3-24. 팝업 패널 속성 화면>

프로퍼티	설명
height	팝업 패널의 높이 설정
visible	팝업 패널의 표시 유무 설정
selecteditem	팝업 패널 내 보여줄 아이템의 id 또는 index를 설정 (생략가능. 기본값은 첫번째 아이템)

<표 3-12. 팝업 패널 속성 및 설명>

4. 폼 컨트롤

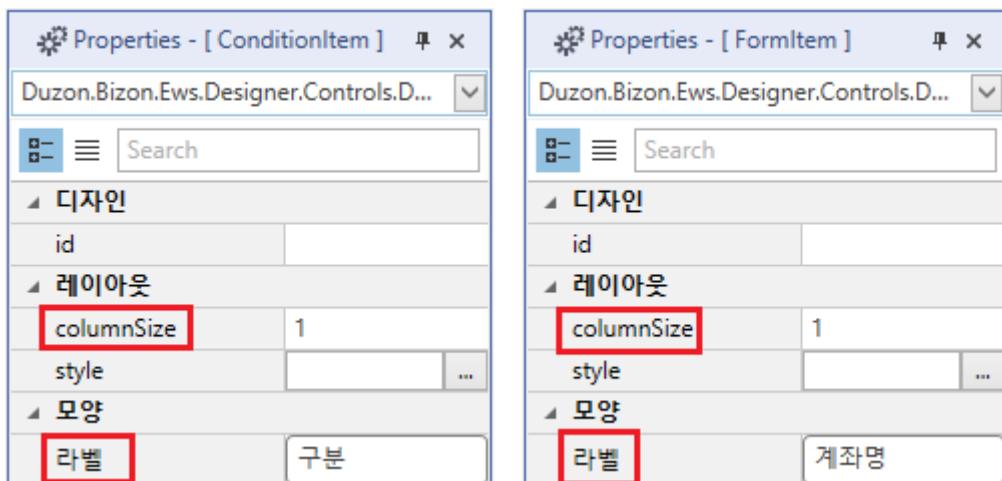
보통 컨디션 패널과 폼 패널, 버튼 그룹에서 추가할 수 있는 컨트롤 종류들이다. 컨디션 패널과 폼 패널에서 추가할 경우 각각 컨디션 아이템과 폼 아이템이 생성되고 하위에 컨트롤이 생성되며 버튼 그룹의 경우 아이템 없이 단일 컨트롤이 생성된다.



<그림 3-25. 컨디션 패널과 폼 패널에서 폼 컨트롤 추가 시 관계도>

1) 컨디션 아이템과 폼 아이템

컨디션 및 폼 아이템에서 폼 컨트롤을 추가했을 때 자동으로 생성되는 컨디션 및 폼 아이템 주요 속성에는 ColumnSize과 라벨이 있다. 우편번호 같이 내용이 긴 컨트롤인 경우 컬럼사이즈를 2로 조정한다. 컨트롤들을 식별할 때 사용되는 id는 아이템이 아닌 하위 컨트롤에 반드시 입력한다.



<그림 3-26. 컨디션 및 폼 아이템>

프로퍼티	설명
columnSize	표시될 컬럼 개수 지정 (기본값 1)
라벨	폼 컨트롤의 라벨 설정

<표 3-13. 컨디션 및 폼 아이템 속성>

폼 컨트롤에 공통으로 사용되는 속성에 대해서 먼저 알아본다.

2) 컨트롤 공통 속성

모든 컨트롤에 해당하는 공통속성은 아래 표와 같다.

프로퍼티	설명
id	해당 컨트롤을 식별 할 수 있는 고유의 ID값
disabled	컨트롤 비활성화
readonly	컨트롤 읽기만 가능
state	기본, 필수 두가지 속성 중 선택 가능. 필수로 선택하게 되면 붉은색 배경이 나타나며 required 상태가 됨
cssClass	컨트롤에 CSS Class 명을 정의
style	스타일 속성 지정
dataAttributes	사용자 지정 속성 추가(data-*로 시작)
DesignSummary	설계 문서 작성 실행
Document	설계 문서 페이지에 사용되는 페이지 이름

<표 3-14. 컨트롤 공통 속성 속성 및 설명>

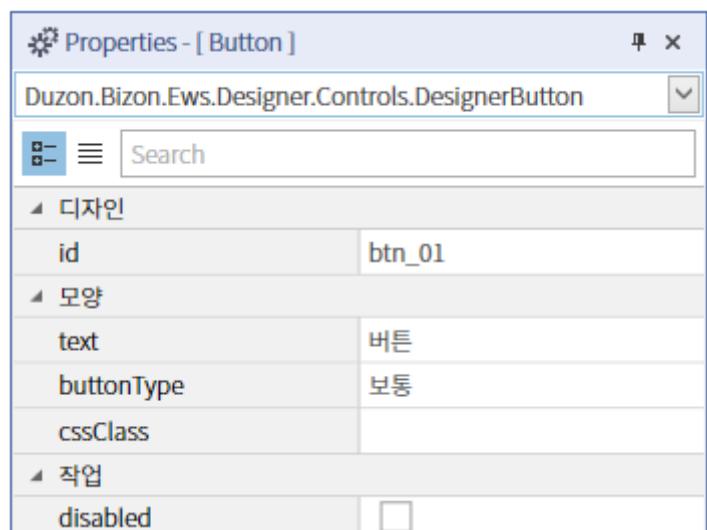
3) 버튼



<그림 3-27. 버튼 화면>

사이즈 별 버튼을 만들 수 있는 컨트롤이다.

① 버튼 속성



<그림 3-28. 버튼 속성 화면>

프로퍼티	설명
text	Default 텍스트 출력
buttonType	해당 컨트롤의 버튼형식(큰, 보통, 작은) 속성을 설정
click	클릭될 때 불러질 이벤트

<표 3-15. 버튼 속성 및 설명>

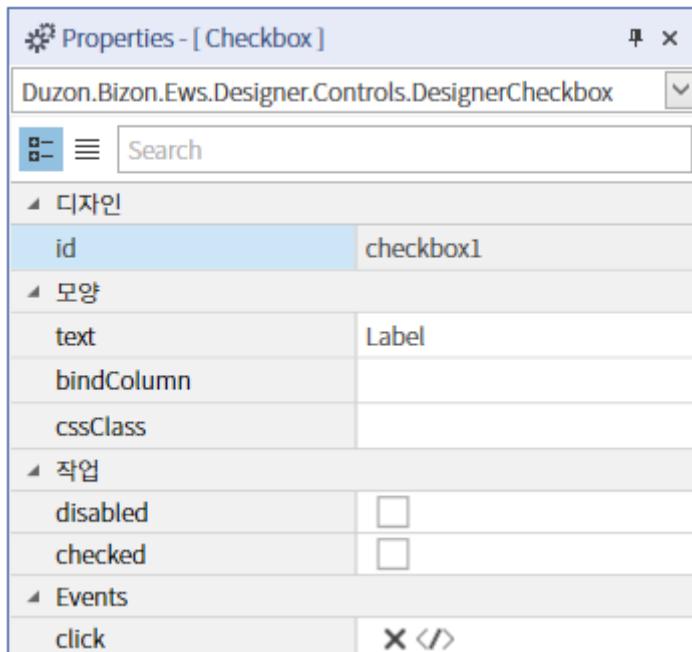
4) 체크박스

체크박스 전체

<그림 3-29. 체크박스 화면>

컨트롤을 생성한 모습이다. 여러 가지의 선택이 필요할 때 사용되며 라디오 버튼과 달리 다중 선택이 가능하다. 다중 선택 생성은 복합 컨트롤에서 생성한다.

① 체크박스 속성



<그림 3-30. 체크박스 속성 화면>

프로퍼티	설명
text	디폴트로 체크 상태 지정
bindColumn	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼 설정
checked	디폴트로 체크 상태 지정
click	클릭 되었을 때 불러질 이벤트
change	체크가 변경 되었을 때 불러질 이벤트

<표 3-16. 체크박스 속성 및 설명>

5) 라디오버튼 및 라디오클립

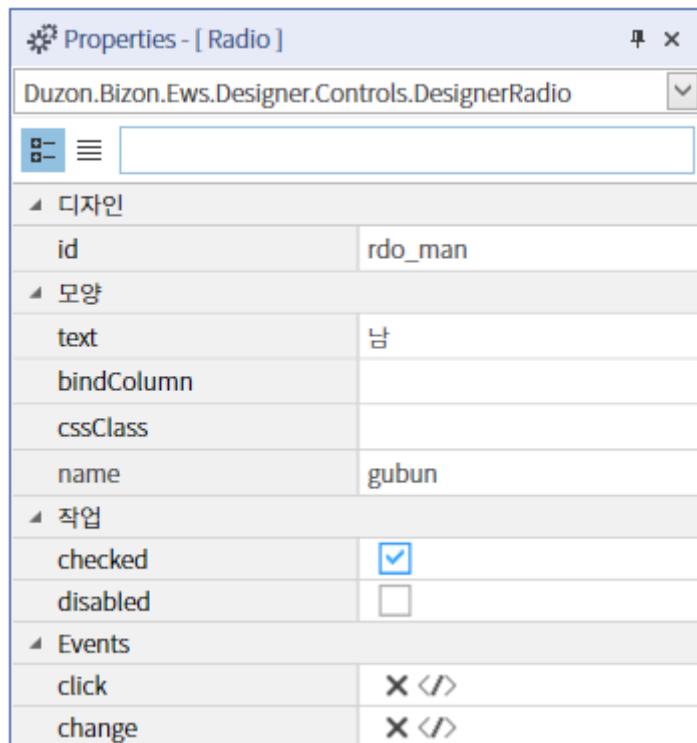
라디오버튼 남

<그림 3-31. 라디오버튼 화면>

컨트롤을 생성한 모습이다. 라디오버튼을 그룹화한 것이 라디오클립이며 보통은 라디오클립을 주로 사용한다.

① 라디오버튼

id는 다르더라도 name이 같으면 같은 그룹이 지정된다.

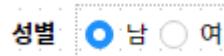


<그림 3-32. 라디오버튼 속성 화면>

프로퍼티	설명
bindColumn	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼 설정
name	라디오 버튼의 그룹명 설정
checked	라디오 버튼의 체크 상태 지정 (디폴트 값은 표시하지 않음)
click	클릭 되었을 때 불러질 이벤트
change	변경 되었을 때 불러질 이벤트

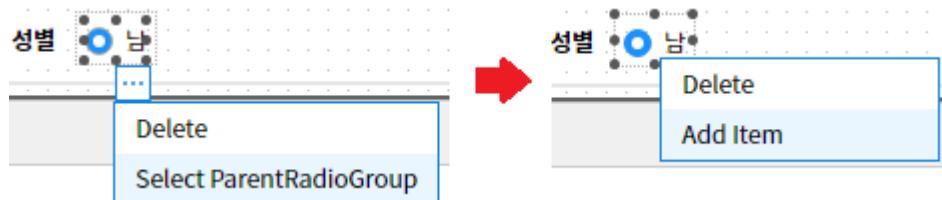
<표 3-17. 라디오버튼 주요 속성 및 설명>

② 라디오그룹

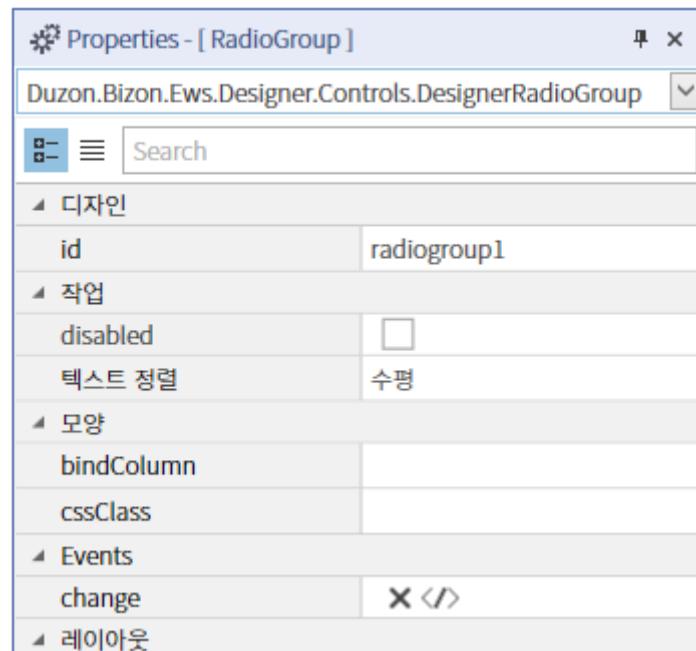


<그림 3-33. 라디어 그룹 화면>

라디오아이템 추가는 라디오그룹을 선택하고 더보기 버튼을 눌러 Add Item을 선택해서 추가한다.



<그림 3-34. 라디오아이템 추가>

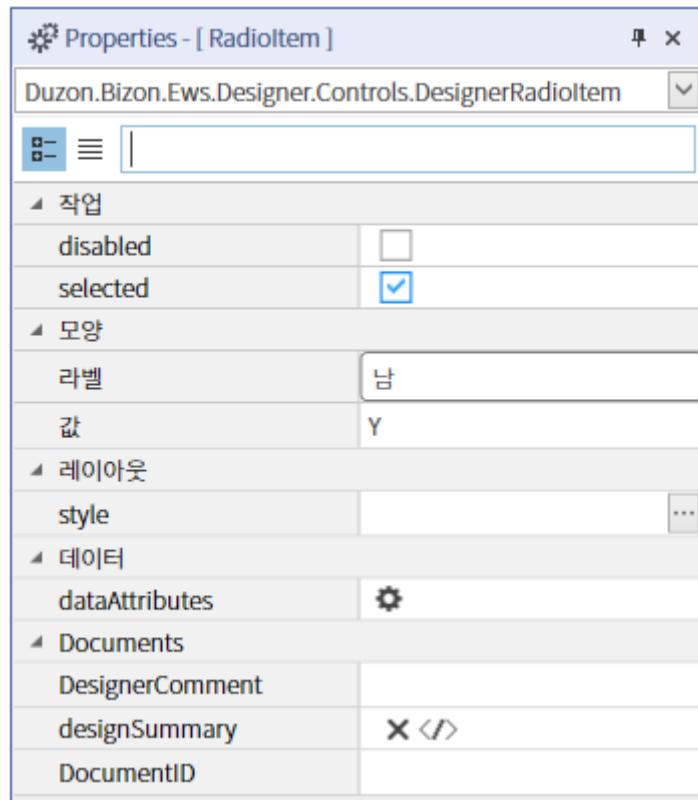


<그림 3-35. 라디오그룹 속성 화면>

프로퍼티	설명
텍스트 정렬	라벨 정렬 설정 (수평, 수직)
bindColumn	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼 설정
change	변경 되었을 때 불러질 이벤트

<표 3-18. 라디오그룹 주요 속성 및 설명>

③ 라디오아이템 속성

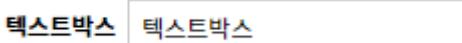


<그림 3-36. 라디오아이템 속성 화면>

프로퍼티	설명
selected	라디오버튼 체크 여부
라벨	라벨 설정
값	값 설정

<표 3-19. 라디오그룹 주요 속성 및 설명>

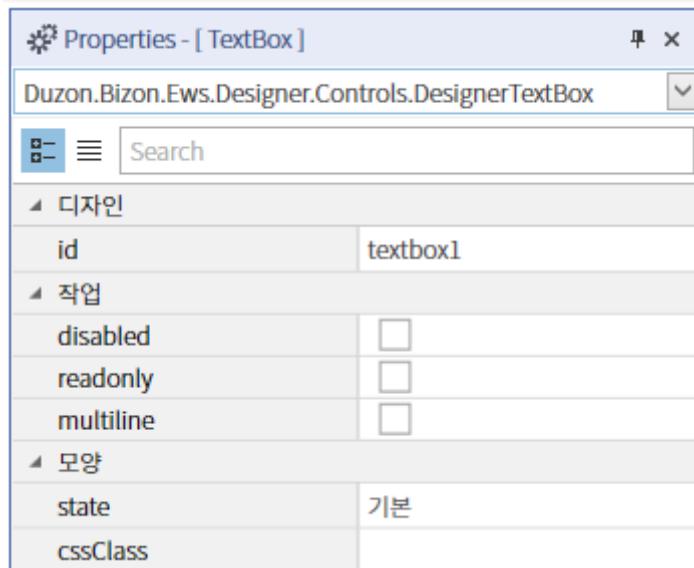
6) 텍스트박스



<그림 3-37. 텍스트박스 화면>

컨트롤을 생성한 모습이며, 문자열을 입력 받는 텍스트박스 컨트롤이다.

① 텍스트박스 속성



<그림 3-38. 텍스트박스 속성 화면>

프로퍼티	설명
multiline	여러 줄 입력 가능하게 하는 옵션
placeholder	Placeholder를 주는 옵션이며, 유저가 입력했을 시 사라지는 텍스트
predefined	통제 환경의 개인정보 마스크 코드 사용 여부
useSecureMask	개인정보 마스크 사용 여부
secureMask	개인정보 마스크 문자열 또는 통제환경 개인정보 마스크 코드 설정
maxLength	최대 입력 가능 글자수
text	Default 텍스트 출력
bindColumn	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼 설정
password	패스워드 텍스트박스로 설정
rows	몇 라인을 차지하게 하는지 설정하는 옵션
change	변경 되었을 때 불러지는 이벤트
focus	포커스인 되었을 시 불러지는 이벤트
blur	포커스아웃 되었을 시 불러지는 이벤트
keyDown	포커스인 되 있는 상태에서 키가 눌러질 때마다 불러지는 이벤트
keyUp	포커스인 되 있는 상태에서 키가 눌렀다 떼어 졌을 때 불러지는 이벤트

	스크립트 작성
nextFocus	포커스를 잃어버릴 때 불러지는 이벤트

<표 3-20. 텍스트박스 주요 속성 및 설명>

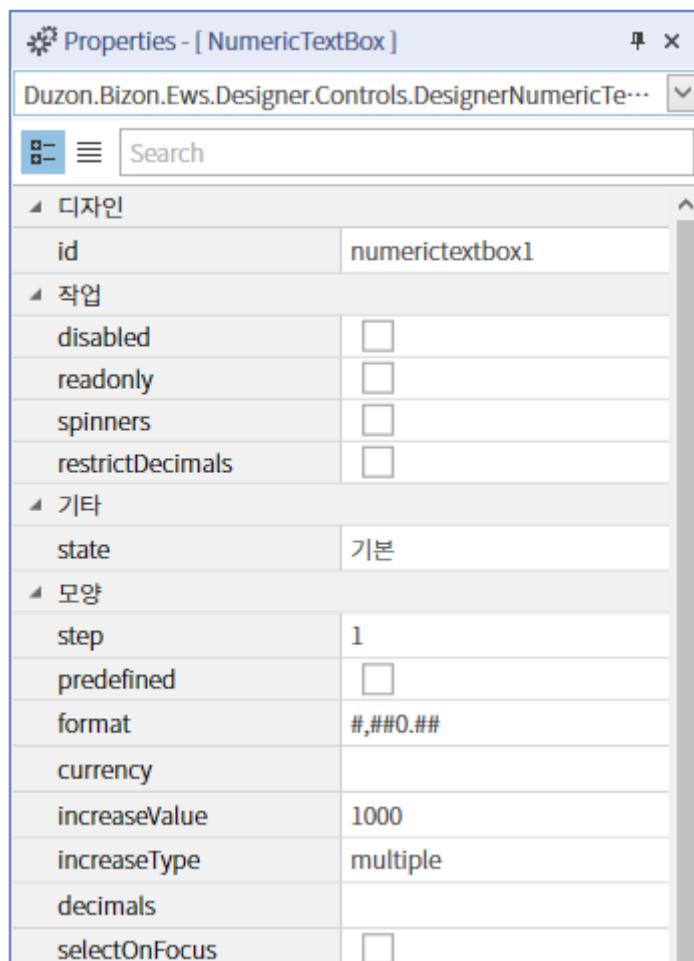
7) 숫자텍스트박스



<그림 3-39 숫자텍스트박스 화면>

컨트롤을 생성한 모습이며, 숫자텍스트박스는 숫자만을 보여주기 위한 텍스트 상자이다.

① 숫자텍스트박스 속성



<그림 3-40. 숫자텍스트박스 속성 화면>

프로퍼티	설명
spinners	증가/감소 스크롤버튼 출력
restrictDecimals	decimals 옵션 설정 시 해당 소수점 자리까지 입력 제한 설정 여부
step	증가/감소 스크롤버튼을 누를 때 증가 또는 감소할 양을 설정

predefined	공통 포맷 코드 사용 여부 설정
format	포맷 설정 또는 통제 환경 포맷값
currency	적용할 환종별 포맷 설정
increaseValue	+키를 입력 시 설정한 숫자 만큼 값이 증가
increaseType	+키를 입력 시 설정한 숫자 만큼 타입에 따라 값이 증가
decimals	소수점 자리 수 통제 설정
selectOnFocus	포커스 시 전체 선택 여부 설정
round	소수점의 반올림(round), 올림(ceil), 내림(floor) 설정

<표 3-21. 숫자텍스트박스 속성 및 설명>

8) 마스크텍스트박스

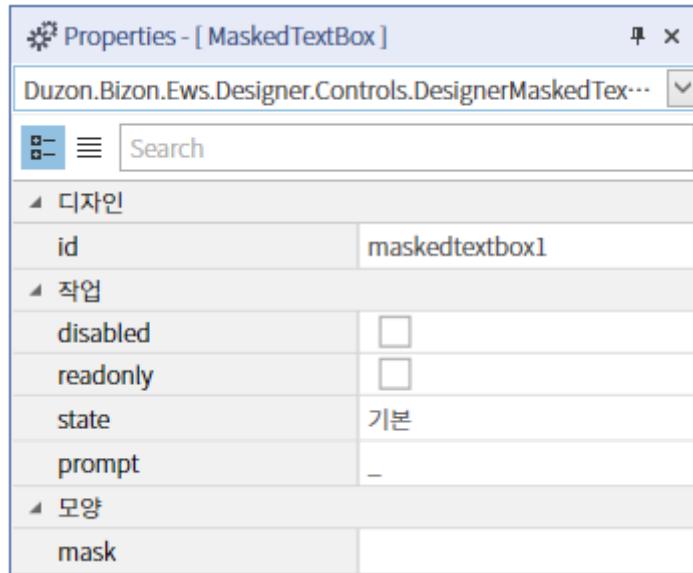


마스크텍스트박스 010-7777-7777

<그림 3-41. 마스크텍스트박스 화면>

컨트롤을 생성한 모습이며, 마스크텍스트박스는 마스크 기능이 추가된 텍스트박스이다.

① 마스크텍스트박스 속성



<그림 3-42. 마스크텍스트박스 속성 화면>

프로퍼티	설명
prompt	프롬프트 문자열 지정 (기본값: _)
mask	마스크 문자열 지정
bindColumn	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼 설정
unmaskOnPost	POST 전송 시 마스크문자열 제거 여부
predefined	통제환경 코드 사용 여부

secureMask	개인정보 마스크 문자열 또는 통제환경 개인정보 마스크 코드를 설정
change	변경 되었을 때 불러지는 이벤트
focus	포커스인 되었을 시 불러지는 이벤트
blur	포커스아웃 되었을 시 불러지는 이벤트
keyDown	포커스인 되어 있는 상태에서 키가 눌러질 때마다 불러지는 이벤트 스크립트 작성
keyUp	포커스인 되어 있는 상태에서 키가 눌렀다 떼어 졌을 때 불러지는 이 벤트
nextFocus	포커스를 잃어버릴 때 불러지는 이벤트

<표 3-22. 마스크텍스트박스 속성 및 설명>

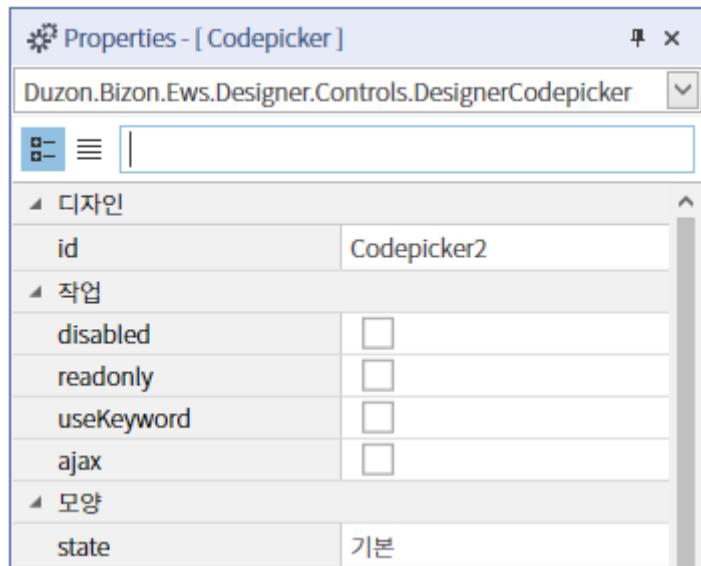
9) 코드피커



<그림 3-43. 코드피커 화면>

컨트롤을 생성한 모습이며, 코드 피커는 도움 창ダイ얼로그를 띄워 데이터를 검색하여 원하는 하나의 데이터를 선택하는 컨트롤이다.

① 코드피커 속성

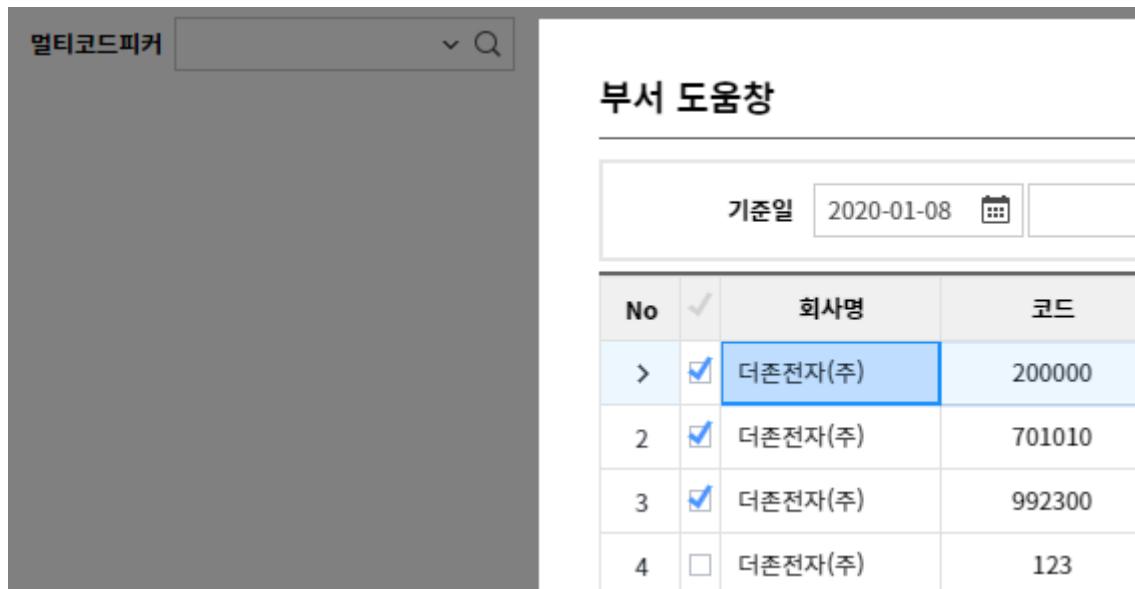


< 그림 3-44. 코드피커 속성 화면 >

프로퍼티	설명
useKeyword	검색어 입력 활성화 여부 지정
ajax	키워드를 AJAX를 이용하여 백그라운드로 검색하여 데이터가 하나만 존재할 경우 다이얼로그의 실행 없이 설정
code	디폴트로 보여질 코드 값
baseText	디폴트로 보여질 텍스트 값
helpCode	코드도움 코드를 지정
codeField	도움 코드에서 반환할 코드 컬럼
textField	도움 코드에서 반환할 텍스트 컬럼
bindCode	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 코드 컬럼 설정
bindText	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 텍스트 컬럼 설정
helpCustom	그리드 공통도움창이 아닌 사용자 정의도움창 사용 여부
helpURLCustom	그리드 사용자 정의 도움창 페이지 HTML 경로 설정
helpApiURL	그리드 사용자 정의 back-end 도움창 API 경로 설정
helpParams	코드 도움에 추가적으로 전달할 데이터
helpTitle	코드도움창 제목을 설정
helpSize	코드 도움창의 크기 선택 (사용자지정, 소, 중, 대, 특대)
helpHeight	도움 창 크기가 사용자 지정일 때 높이
helpWidth	도움 창 크기가 사용자 지정일 때 너비
setData	코드 피커에 코드/텍스트 데이터가 설정될 때 이벤트
codedialog	코드 도움 다이얼로그가 실행될 때 이벤트

<표 3-23. 코드피커 속성 및 설명>

10) 멀티코드피커

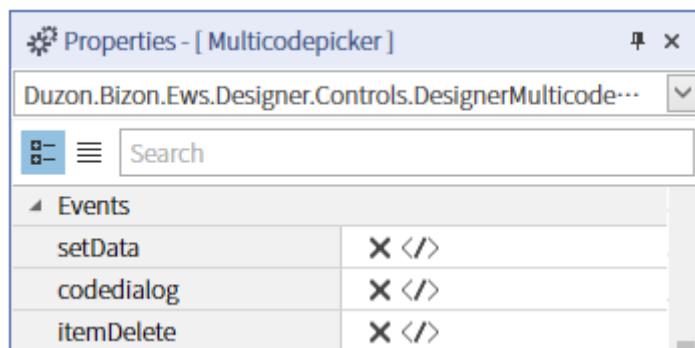


<그림 3-45. 멀티코드피커 화면>

멀티코드피커는 도움창 디자인로그를 띠워 여러 개의 항목을 선택할 수 있는 코드피커이다.

① 멀티코드피커 속성

멀티코드피커는 코드 피커의 속성과 같으며, 추가적으로 이벤트를 더 작성할 수 있다.



<그림 3-46. 멀티코드피커 속성 화면>

프로퍼티	설명
itemDelete	멀티코드피커에 UI 상에서 삭제할 때 이벤트

<표 3-24. 멀티코드피커 속성 및 설명>

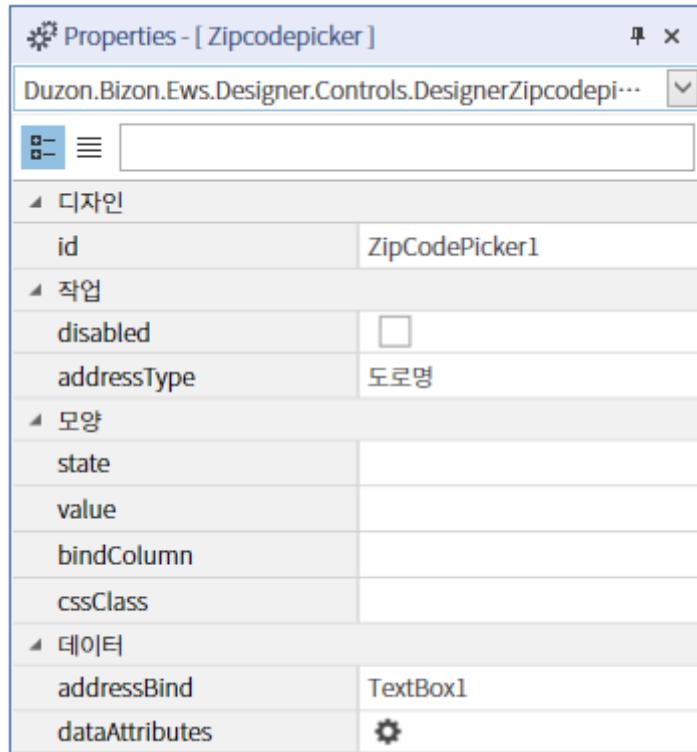
11) 우편번호피커



<그림 3-47. 우편번호피커 화면>

우편번호피커를 이용하여 주소를 검색 우편번호를 조회하여 사용할 수 있다. 일반적으로 복합컨트롤을 사용하여 주소의 상세정보가 입력되는 텍스트박스도 같이 생성한다.

① 우편번호피커 속성

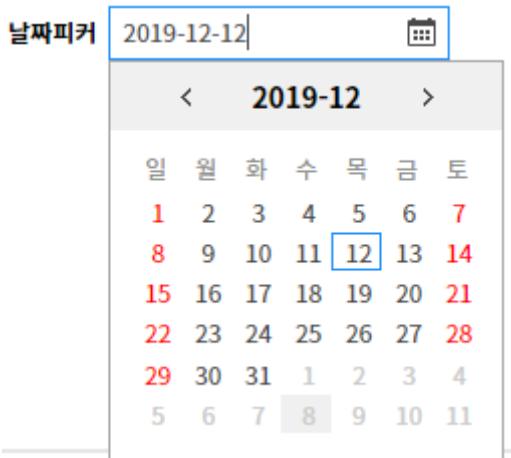


<그림 3-48. 우편번호피커 속성 화면>

프로퍼티	설명
우편번호	우편번호 초기값 지정
bindColumn	그리드와 바인딩되는 컬럼 설정
addressBind	다이얼로그에서 선택된 우편번호 데이터에서 주소 데이터를 바인드 할 컨트롤 아이디 지정 (생략가능)
addressType	주소로 바인딩 될 데이터가 도로명 주소 또는 지번 주소 인지 선택
open	열렸을 때 불러질 이벤트
close	닫혔을 때 불러질 이벤트
change	항목이 바뀌었을 경우 불러질 이벤트
nextFocus	다음 포커스인 되었을 때 불러질 이벤트

<표 3-25. 우편번호피커 속성 및 설명>

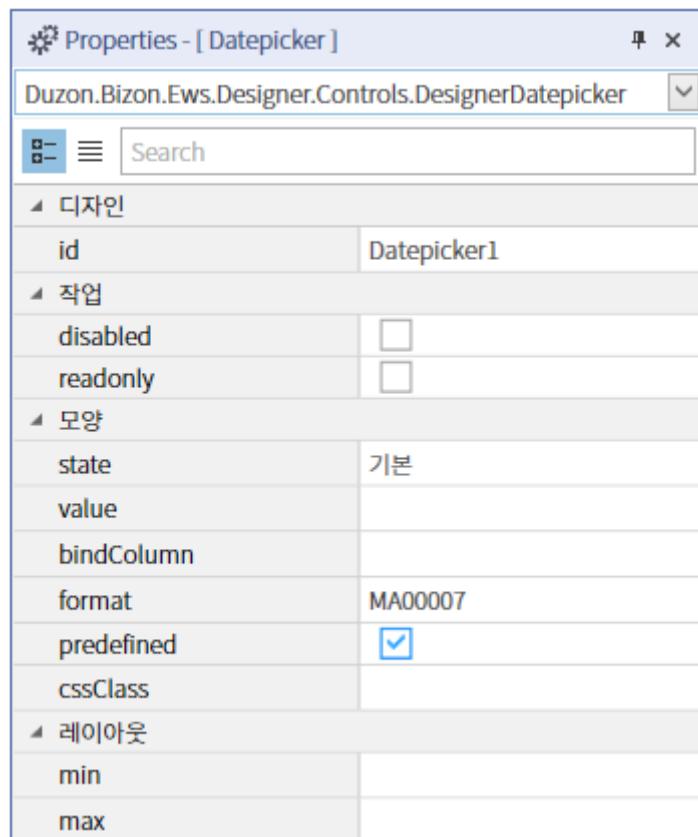
12) 날짜피커



<그림 3-49. 날짜피커 화면>

날짜를 선택하는 컨트롤이다.

① 날짜피커 속성

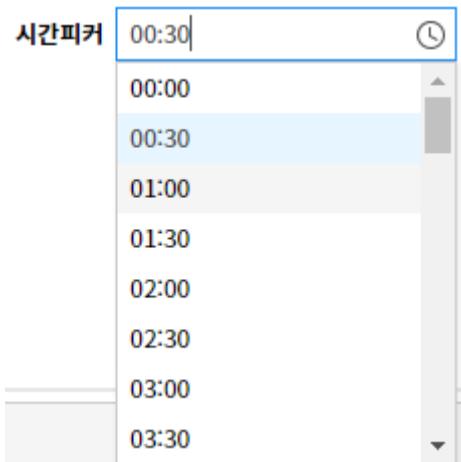


<그림 3-50. 날짜피커 속성 화면>

프로퍼티	설명
value	디폴트값 지정
bindColumn	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼 설정
min	선택할 수 있는 날짜 최소값 지정
max	선택할 수 있는 날짜 최대값 지정
predefined	통제환경 코드 사용 여부
format	포맷 설정 또는 통제 환경 포맷값
width	너비 속성 지정
open	열렸을 때 불러질 이벤트
close	닫혔을 때 불러질 이벤트

<표 3-26. 날짜피커 속성 및 설명>

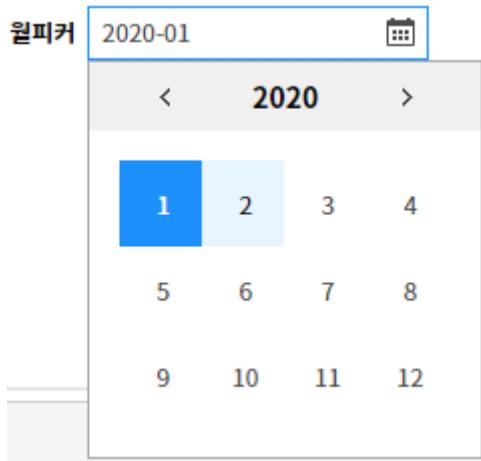
13) 시간피커



<그림 3-51. 시간피커 화면>

시간을 선택하는 컨트롤이며, 속성은 날짜 피커와 동일하다. 시간을 직접 입력하거나 30분 단위로 선택이 가능하다.

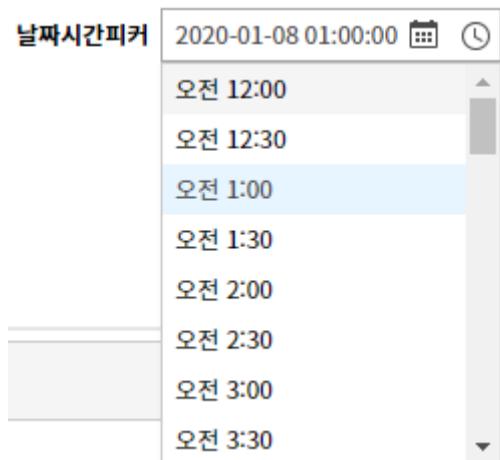
14) 월피커



<그림 3-52. 월피커 화면>

월 날짜를 선택하는 컨트롤이며, 속성은 날짜 피커와 동일하다.

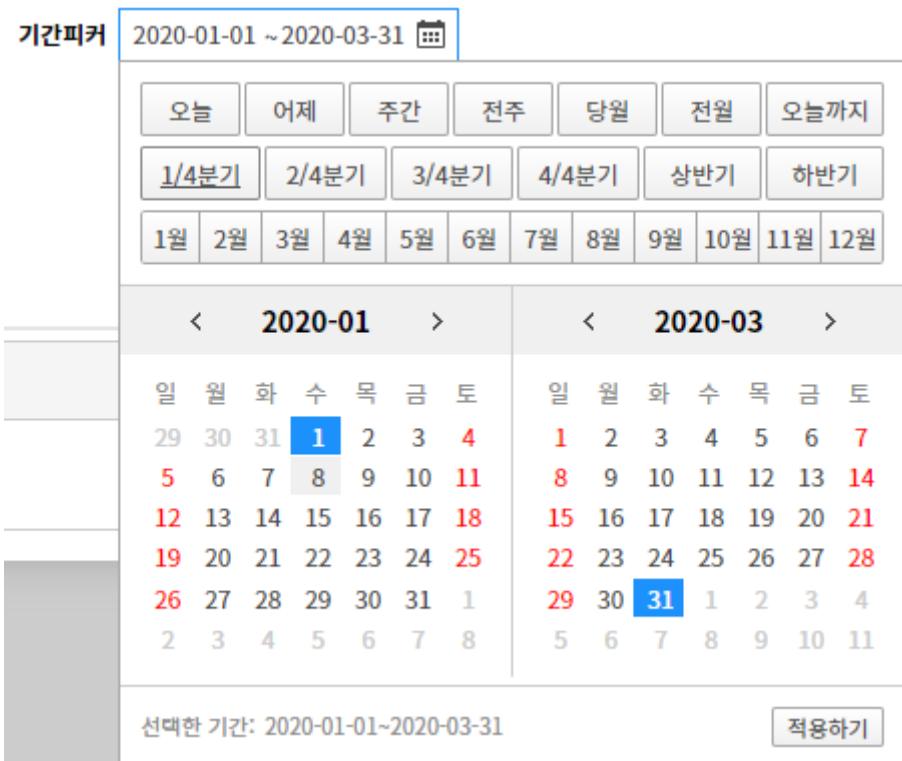
15) 날짜시간피커



<그림 3-53. 날짜시간피커 화면>

날짜와 시간을 선택하는 컨트롤이며, 속성은 날짜 피커와 동일하다.

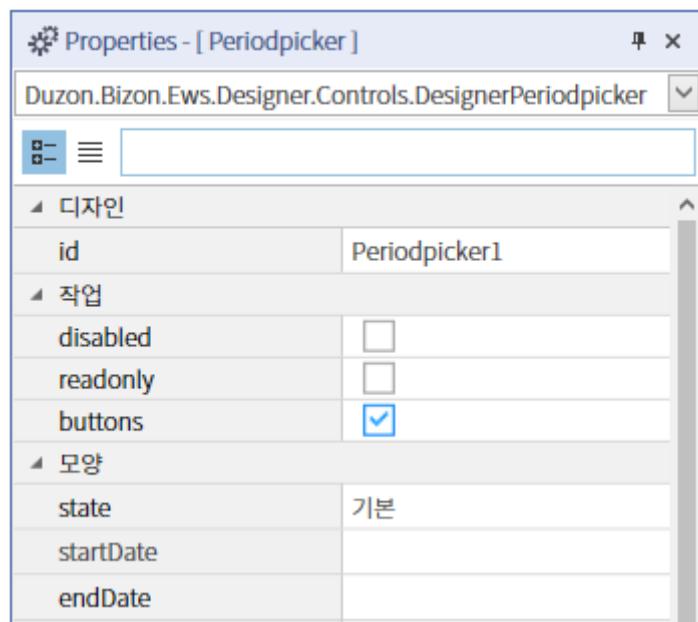
16) 기간피커



<그림 3-54. 기간피커 화면>

기간을 선택하는 컨트롤이다. 상단에 쉽게 기간을 선택할 수 있도록 버튼을 제공한다.

① 기간피커 속성



<그림 3-55. 기간피커 속성 화면>

프로퍼티	설명
buttons	날짜 선택 상단에 기간 선택 버튼 표시 여부
startDate	디폴트 시작날짜
endDate	디폴트 종료날짜
bindStartDate	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼(시작날짜)
bindEndDate	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼(종료날짜)
format	포맷 설정 또는 통제 환경 포맷값
predefined	통제환경 코드 사용 여부
change	항목이 바뀌었을 경우 불러질 이벤트

<표 3-27. 기간피커 속성 및 설명>

17) 주기간피커

<그림 3-56. 월기간피커 화면>

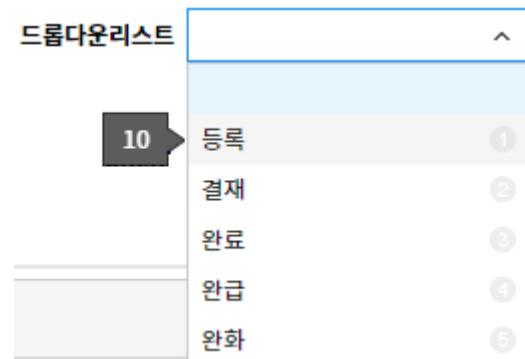
컨트롤을 생성한 모습이며, 주 기간을 선택하는 컨트롤이다. 속성은 기간피커와 동일하다.

18) 월기간피커

<그림 3-57. 월기간피커 화면>

컨트롤을 생성한 모습이며, 월 기간을 선택하는 컨트롤이다. 속성은 기간피커와 동일하다.

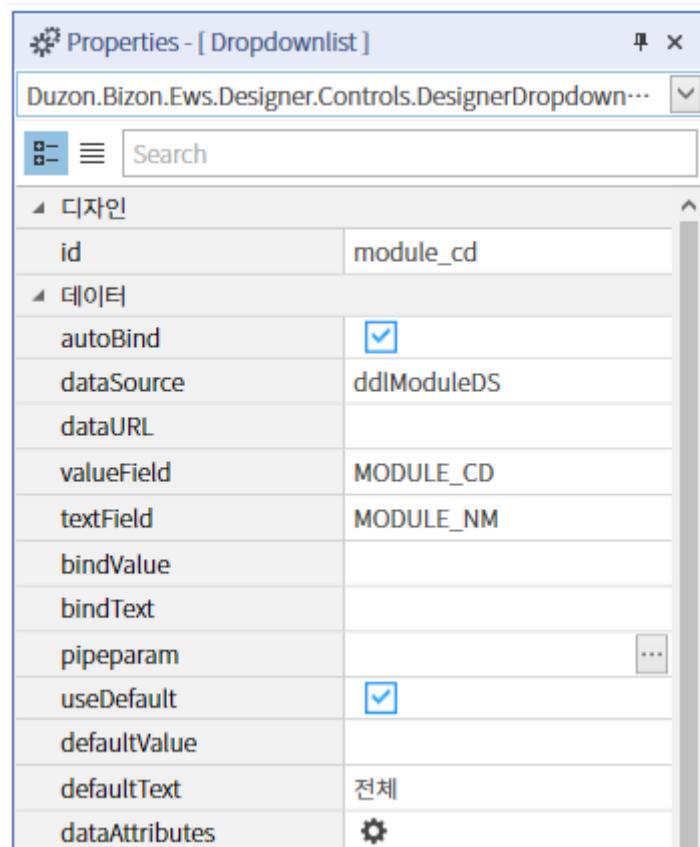
19) 드롭다운리스트



<그림 3-58. 드롭다운리스트 화면>

컨트롤을 생성한 모습이며, 드롭다운리스트 또한 콤보박스와 동일하게 프로퍼티 창에서 데이터소스를 생성하고 바인딩 해주어야 한다. 콤보박스 보다 드롭다운리스트를 많이 사용한다.

① 드롭다운리스트 속성

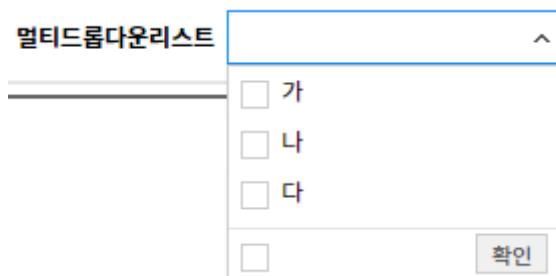


<그림 3-59. 드롭다운리스트 속성 화면>

프로퍼티	설명
autoBind	데이터소스 자동 바인딩 여부
dataSource	바인딩할 데이터소스 지정
dataURL	데이터를 가져올 URL 경로 설정
valueField	바인딩된 데이터소스의 컬럼 중 값으로 출력할 컬럼
textField	바인딩된 데이터소스의 컬럼 중 화면에 표시될 컬럼
bindValue	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼 중 값으로 출력할 컬럼
bindText	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼 중 화면에 표시될 컬럼
pipeparam	MultiPipe를 체크한 데이터소스일 때 구분할 수 있는 코드값
useDefault	디폴트값 사용 여부
defaultValue	디폴트 값
defaultText	디폴트 텍스트
autoOpen	컨트롤에 포커스가 들어갔을 때 자동으로 항목 리스트 열림 여부
useNumkey	숫자키 누를 시 항목 선택 기능 사용 여부(
showToolTip	특정항목의 텍스트가 길어 잘리는 경우 툴팁으로 텍스트 표시 여부
showValueToolTip	아이템에 마우스 오버 시 valueField 값을 좌측에 표시 여부

<표 3-28. 드롭다운리스트 속성 및 설명>

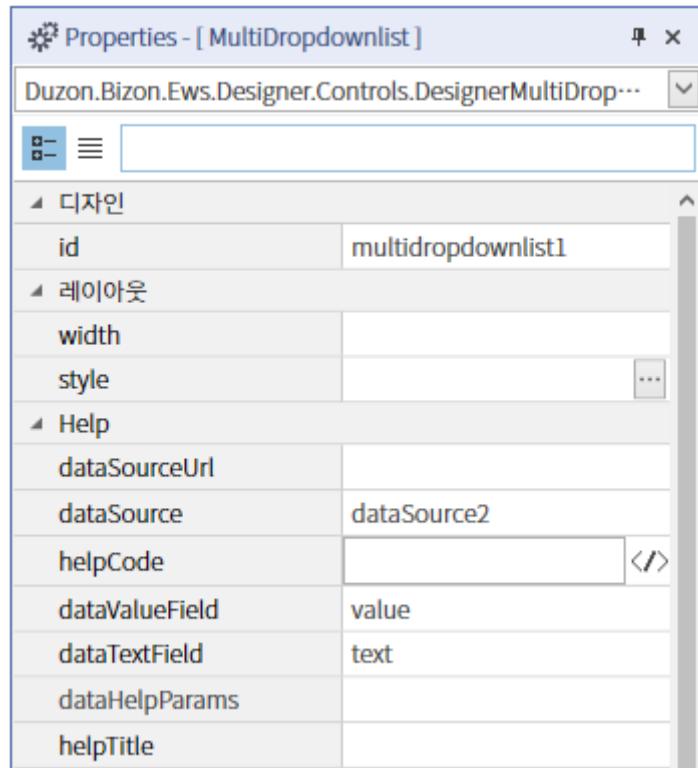
20) 멀티드롭다운리스트



<그림 3-60. 멀티드롭다운리스트 화면>

멀티드롭다운리스트는 드롭다운리스트와 유사한 동작을 하는 컨트롤이며 드롭다운리스트는 하나의 값만 선택하지만 멀티드롭다운리스트는 체크박스를 이용하여 여러 개의 값을 선택할 수 있다.

① 멀티드롭다운리스트 속성

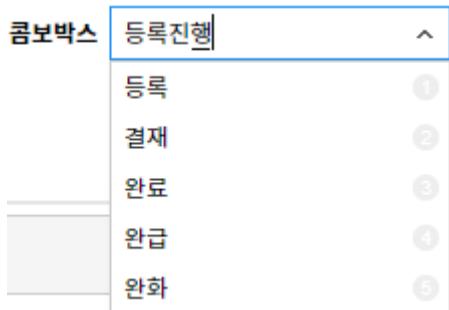


<그림 3-61. 드롭다운리스트 속성 화면>

프로퍼티	설명
dataSourceUrl	특정 url 데이터를 바인딩 하는 경우 사용할 url
dataSource	바인딩할 데이터소스 지정
helpCode	코드도움 데이터를 바인딩하는 경우 사용할 코드도움 코드 지정
dataValueField	바인딩된 데이터소스의 컬럼 중 값으로 출력할 컬럼
dataTextField	바인딩된 데이터소스의 컬럼 중 화면에 표시될 컬럼
dataHelpParams	코드도움 데이터를 바인딩하는 경우 코드도움에 전달할 파라미터 지정
select	멀티드롭다운리스트의 아이템을 선택할 때 발생하는 이벤트
open	열렸을 때 불러질 이벤트
close	닫혔을 때 불러질 이벤트
dataBound	멀티드롭다운리스트에 데이터소스가 바인딩될 때 이벤트

<표 3-29. 멀티드롭다운리스트 속성 및 설명>

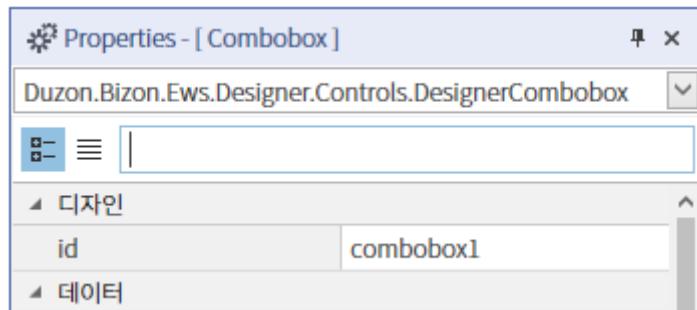
21) 콤보박스



<그림 3-62. 콤보박스 화면>

컨트롤을 생성한 모습이며, 콤보 박스는 드롭다운리스트와 비슷한 컨트롤이다. 다른점은 콤보박스에 데이터를 입력이 가능하다. 프로퍼티 창에서 생성한 데이터소스를 바인딩 해야한다.

① 콤보박스 속성

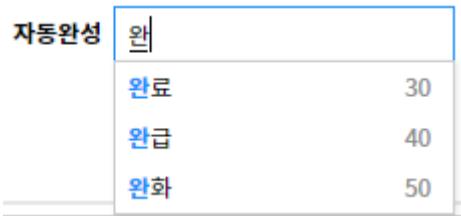


<그림 3-63. 콤보박스 속성 화면>

프로퍼티	설명
dataSource	바인딩할 데이터소스 지정
dataURL	데이터를 가져올 URL 경로 설정
valueField	바인딩된 데이터소스의 컬럼 중 값으로 출력할 컬럼
textField	바인딩된 데이터소스의 컬럼 중 화면에 표시될 컬럼
bindValue	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼 중 값으로 출력할 컬럼
bindText	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 컬럼 중 화면에 표시될 컬럼
pipeparam	MultiPipe를 체크한 데이터소스일 때 구분할 수 있는 코드값
change	항목이 바뀌었을 경우 불러질 이벤트
open	열렸을 때 불러질 이벤트
close	닫혔을 때 불러질 이벤트
select	콤보박스 리스트 아이템을 선택할 때 발생하는 이벤트

<표 3-30. 콤보박스 속성 및 설명>

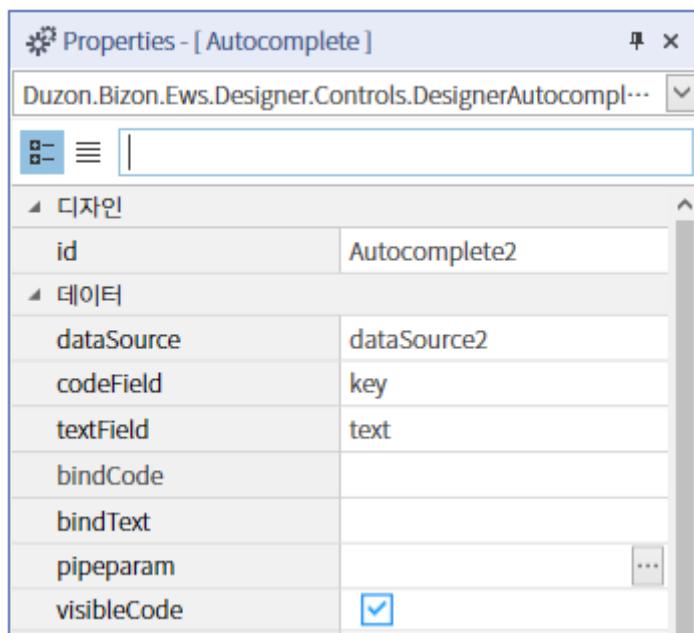
22) 자동완성



<그림 3-64. 자동완성 화면>

컨트롤을 생성한 모습이며, 자동완성 컨트롤은 데이터소스가 필요하다. 데이터 소스 탭에서 데이터 소스를 만든 후 코드, 텍스트 컬럼을 설정하면 유저가 텍스트를 입력하면 프로젝트의 지정된 서버의 정보를 토대로 자동완성 기능이 작동한다.

① 자동완성 속성



<그림 3-65. 자동완성 속성 화면>

프로퍼티	설명
dataSource	바인딩할 데이터소스 지정
codeField	도움 코드에서 반환할 코드 컬럼
textField	도움 코드에서 반환할 텍스트 컬럼
bindCode	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 코드 컬럼 설정
bindText	그리드나 카드리스트 등 데이터 컨트롤과 바인딩(연결)되는 텍스트 컬럼 설정
pipeparam	MultiPipe를 체크한 데이터소스일 때 구분할 수 있는 코드값
visibleCode	코드 값이 우측에 표시 될지 여부
width	너비를 지정 (px단위)

listMaxHeight	최대 리스트 높이 지정
listWidth	리스트 길이 지정 (최소 길이는 검색어 컬럼의 길이).
minLength	사용자가 입력 해야 할 최소 글자수 설정 (기본값 1).
filter	첫글자부터 일치(startwith), 글자내 포함(contains) 옵션 지정 (기본값: startwith).
dataBound	데이터소스가 바인딩 될 때 불러질 이벤트
open	열릴 때 불러질 이벤트
close	닫힐 때 불러질 이벤트

<표 3-31. 자동완성 속성 및 설명>

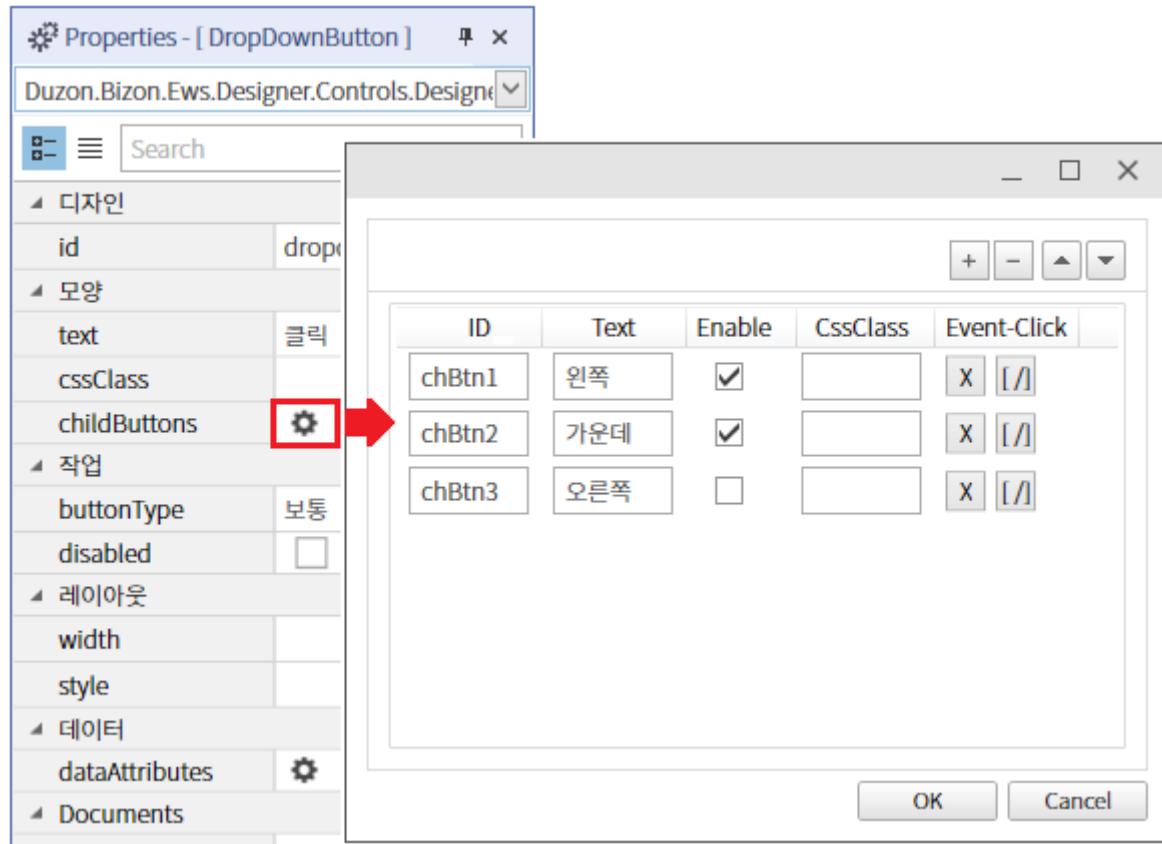
23) 드롭다운버튼



<그림 3-66. 드롭다운버튼 화면>

드롭다운버튼은 내부 버튼이 존재하는 구조로 버튼들의 개별 컨트롤이 가능하다. 프로퍼티창 childButtons 속성을 눌러서 하위 버튼을 관리할 수 있다.

① 드롭다운버튼 속성



<그림 3-67. 드롭다운버튼 속성 화면>

프로퍼티	설명
text	Default 텍스트 출력
childButtons	하위 버튼 설정
buttonType	해당 컨트롤의 버튼형식(큰, 보통, 작은) 속성 설정
width	너비 속성 지정 (px단위)

<표 3-32. 드롭다운버튼 속성 및 설명>

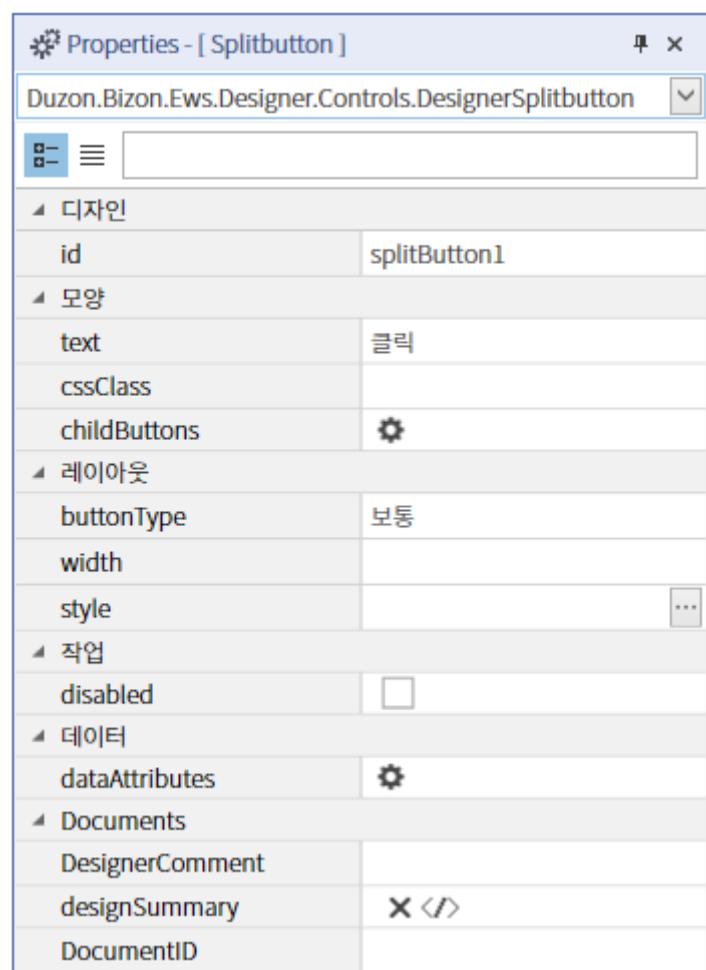
24) 스플리트버튼



<그림 3-68. 스플리트버튼 화면>

스플리트버튼의 내부 버튼이 존재하는 구조로 버튼들의 개별 컨트롤이 가능하며, 드롭다운버튼과 달리 최상위의 버튼은 목록을 열지 않아도 바로 실행할 수 있다.

① 스플리트버튼 속성



<그림 3-69. 스플리트버튼 속성 화면>

프로퍼티	설명
text	Default 텍스트 출력
childButtons	내부 버튼 관리
buttonType	해당 컨트롤의 버튼형식(큰, 보통, 작은) 속성 설정
width	너비 속성 지정 (px단위)
click	클릭될 때 불러질 이벤트

<표 3-33. 스플리트버튼 속성 및 설명>

25) 파일컴포넌트

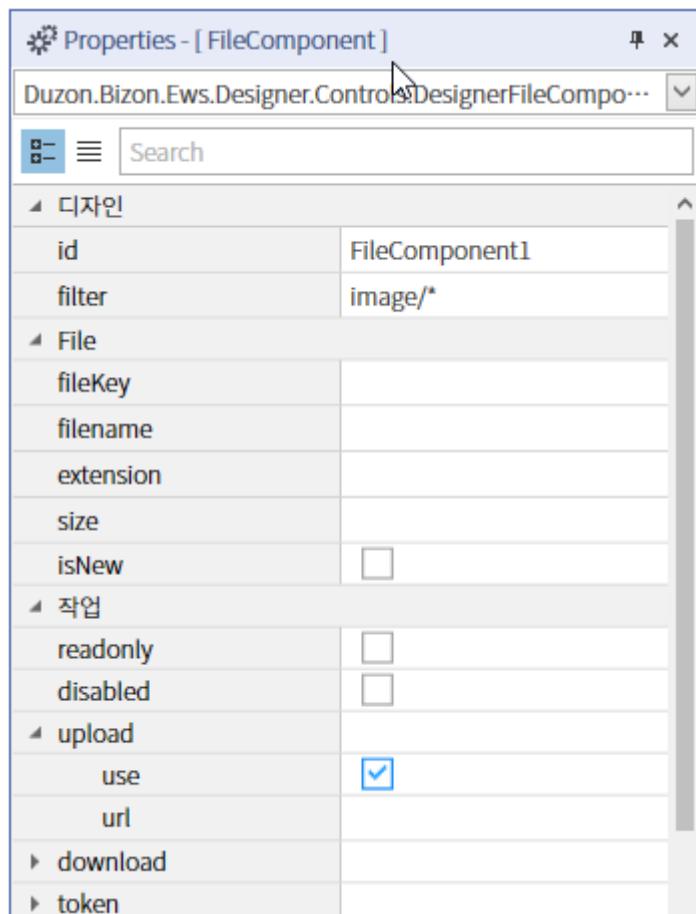


<그림 3-70. 파일컴포넌트 화면>

파일 컴포넌트는 단일 파일에 대한 업로드 및 다운로드를 수행할 수 있는 컴포넌트이며, 파일 업로드 이후 저장 시점까지 임시로 업로드된(서버 temp 폴더) 상태를 관리하기 위해 파일 정보 객체를 이용하여 첨부파일을 관리한다. 컨디션패널과 품패널에서 추가할 수 있다.

업로드 및 다운로드 기능을 자체적으로 가지고 있으므로 첨부 파일을 위한 추가적인 코드가 필요없다. ERP 10에서 사용되는 모든 파일들은 파일 관리 테이블(CM_FILE_INFO)에서 관리한다.

① 파일컴포넌트 속성



<그림 3-71. 파일컴포넌트 속성 화면>

프로퍼티	설명	
filter	파일 컴포넌트에 확장자 필터 설정	
File	fileKey	파일 컴포넌트에 설정할 첨부파일의 키 설정
	filename	파일 컴포넌트에 설정할 첨부파일의 파일명 설정
	extension	파일 컴포넌트에 설정할 첨부파일의 확장자 설정
	size	파일 컴포넌트에 설정할 첨부파일의 파일크기 설정
	isNew	임시파일 여부
upload	use	파일 컴포넌트가 업로드 기능을 수행할지 여부
	url	파일 업로드 API 의 URL 설정
download	use	파일 컴포넌트가 다운로드 기능을 수행할지 여부
	url	파일 컴포넌트의 파일을 다운로드 할 수 있는 API 의 URL을 설정
	tempUrl	파일 컴포넌트의 파일이 임시파일일 때 다운로드 할 수 있는 API의 URL을 설정
	param	파일 다운로드 시 파일 키를 전달할 파라미터 명 설정
token	use	파일 다운로드 시 임시 인증토큰을 발급받아서 사용할지 여부
	url	임시인증 토큰을 받아올 수 있는 API 의 URL 설정
	param	다운로드 시 임시인증 토큰을 전달할 파라미터 명 설정
binding	filekey	바인딩 사용시 첨부파일 키에 해당하는 필드명 설정
	filename	바인딩 사용시 첨부파일명에 해당하는 필드명 설정
	extension	바인딩 사용시 첨부파일의 확장자에 해당하는 필드명 설정
	size	바인딩 사용시 첨부파일의 크기에 해당하는 필드명 설정
	isNew	바인딩 사용시 첨부파일이 임시파일인지 여부
setFile(fileData)		파일 컴포넌트에 파일 정보를 설정하는 함수
deleteFile()		파일 컴포넌트에서 설정된 파일 정보를 삭제하는 함수
getFile()		파일 컴포넌트에서 설정된 파일 정보를 가져오는 함수

<표 3-34. 파일컴포넌트 속성 및 설명>

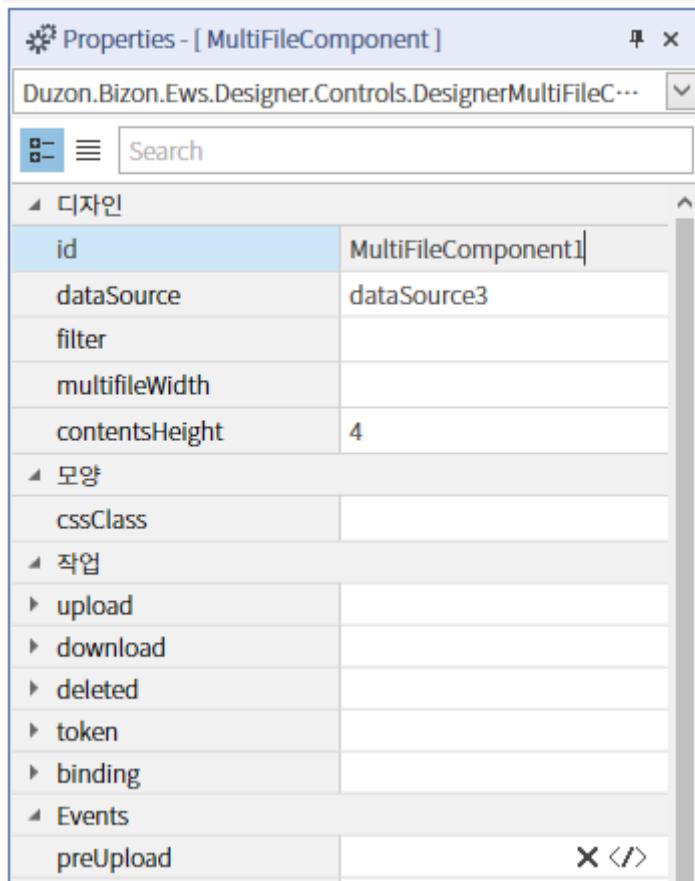
26) 멀티파일 컴포넌트



<그림 3-72. 멀티파일 컴포넌트 화면>

멀티 파일 컴포넌트는 다중 파일에 대한 업로드 및 다운로드를 수행할 수 있는 컴포넌트이며, 파일 업로드 이후 저장 시점까지 임시로 업로드된(서버 temp 폴더) 상태를 관리하기 위해 파일 정보 객체를 이용하여 첨부파일을 관리한다. 파일 컴포넌트와 다르게 컨테이너패널에서 추가할 수 있으며, 데이터소스가 필요하다.

업로드 및 다운로드 기능을 자체적으로 가지고 있으므로 첨부 파일을 위한 추가적인 코드가 필요없다. ERP 10에서 사용되는 모든 파일들은 파일 관리 테이블(CM_FILE_INFO)에서 관리한다.



<그림 3-73. 멀티파일 컴포넌트 속성 화면>

프로퍼티		설명
filter		파일 컴포넌트에 확장자 필터 설정
multifileWidth		멀티파일 컴포넌트의 너비를 지정
contentsHeight		티파일 컴포넌트의 높이를 지정
upload	use	업로드 기능 사용 여부
	url	업로드 API 의 URL (생략가능 / 기본값 : '/upload/file')
	disabled	업로드 버튼 활성화 여부
	visible	업로드 버튼이 보여지는지 여부
download	use	다운로드 기능 사용 여부
	url	다운로드 API 의 URL
	tempUrl	임시파일 다운로드 API 의 URL (생략가능 / 기본값 : '/download/tempfile')
	param	다운로드에서 파일 키를 전달할 파라미터 명
	zipname	다중 다운로드 할 경우 압축파일 이름을 설정
	disabled	다운로드 버튼 활성화 여부 (
delete	disabled	삭제 버튼 활성화 여부
	visible	삭제 버튼이 보여지는지 여부
token	use	임시 인증 토큰 기능 사용 여부
	url	임시 인증 토큰 발급 API 의 URL (생략가능 / 기본값 : '/auth/temporary/token')
	param	다운로드 시 임시 인증 토큰을 전달할 파라미터 명
binding	filekey	파일키에 바인딩할 필드명
	filename	파일명에 바인딩할 필드명
	extension	확장자에 바인딩할 필드명
	size	파일크기에 바인딩할 필드명
	isNew	새로운 임시 파일 여부에 바인딩할 필드명
	isDelete	삭제된 임시파일 여부에 바인딩할 필드명
preUpload		업로드 전에 발생하는 이벤트
preDownload		다운로드 전에 발생하는 이벤트
preDelete		삭제 전에 발생하는 이벤트
uploaded		업로드 완료 시점에 발생하는 이벤트
downloadStarted		서버로 요청 후 다운로드 시작 시점에 발생 이벤트

<표 3-35. 멀티파일 컴포넌트 속성 및 설명>

27) 복합 컨트롤



<그림 3-74. 복합컨트롤 화면>

복합컨트롤은 컨트롤들을 하나의 그룹으로 묶을 수 있다. 체크박스와 같은 컨트롤을 여러 개 생성해야하거나 우편번호피커에 주소의 상세정보가 입력되는 텍스트박스가 필요한 경우 복합컨트롤로 생성해야 한다. .



<그림 3-75. 체크 박스 복합컨트롤 생성 화면>

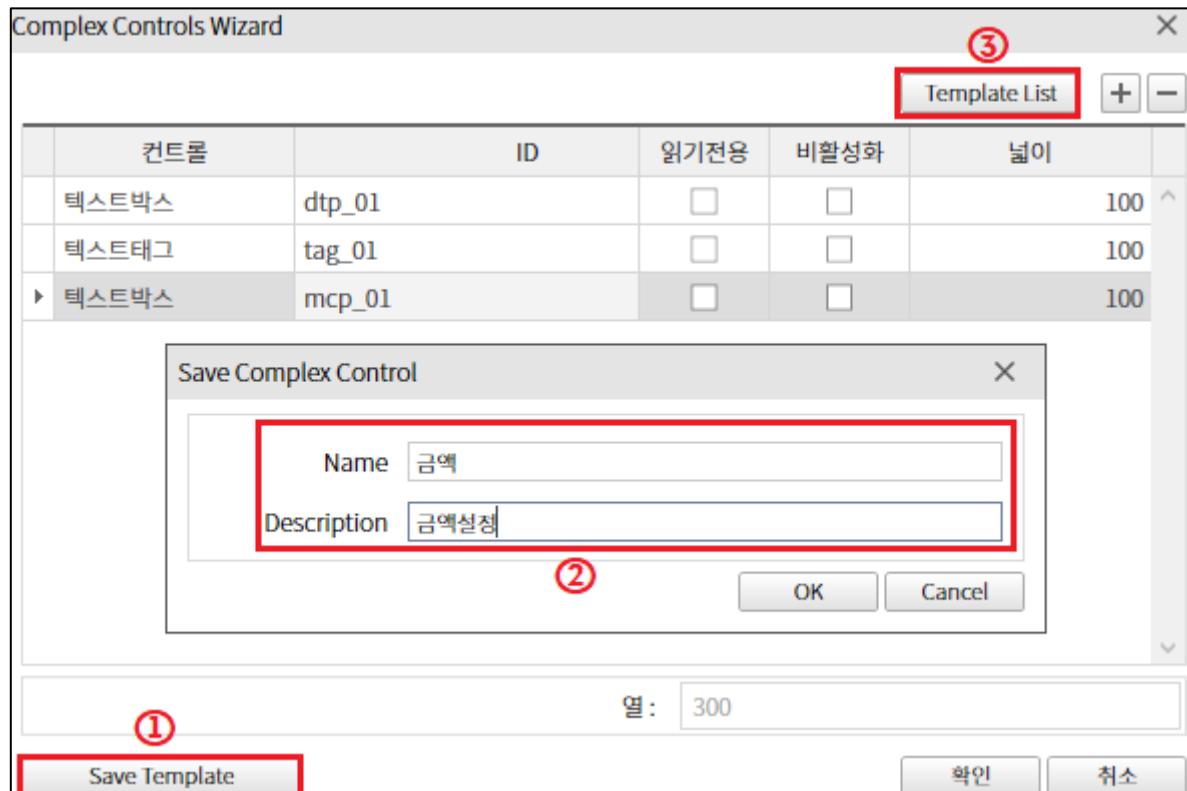
그 외에도 위에서 언급한 모든 컨트롤들을 복합컨트롤로 묶을 수 있다.



<그림 3-76. 날짜피커와 멀티코드피커를 복합컨트롤로 생성하는 화면>

복합 컨트롤을 사용할 경우 여러 개의 컨트롤에 하나의 라벨만 생성이 된다.

자주 사용하는 복합 컨트롤의 경우 복합 컨트롤을 저장하고 불러와서 재사용할 수 있다.



<그림 3-77. 자주 사용되는 복합 컨트롤 저장 화면>

복합 컨트롤 생성 후 ① Save Template 버튼을 클릭하고 Save Complex Control 화면에서 Name 과 Description 을 입력하고 저장버튼을 눌러 저장한다. 저장된 복합 컨트롤은 ③ Template List 버튼을 클릭하고 불러와서 사용할 수 있다.

5. 데이터

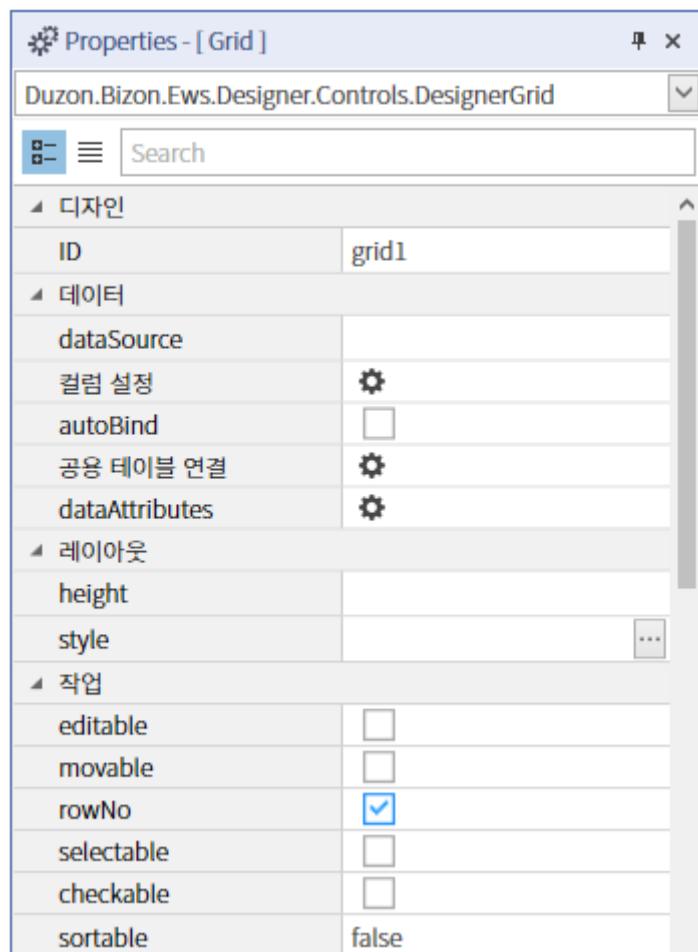
1) 그리드

No	회계단위명	작성부서	작성자	회계일	순번	삭제자명	삭제일시
1							

<그림 3-78. 그리드 화면>

그리드는 각 필드에 행 단위의 데이터가 저장되어 표 형태로 보여지는 가장 기본적인 데이터 컨트롤이다. 각 행에는 텍스트뿐만 아니라 다양한 컨트롤들이 들어갈 수 있다. <그림 3-75>는 그리드의 프로퍼티 화면이다. 데이터소스와 연결되지 않은 그리드의 경우 발사믹 목업 프로그램 (balsamiq mockups)처럼 더블 클릭하여 컬럼명과 데이터를 입력할 수 있다.

① 그리드 속성



<그림 3-79. 그리드 속성 화면>

프로퍼티	설명
ID	그리드 ID는 grid1, grid2 방식으로 번호가 매겨지며, 편집 가능
dataSource	바인딩 될 데이터소스 설정(데이터소스 Grid Type이 Grid)
컬럼 설정	2장 그리드 및 데이터소스 부분 참고
autoBind	데이터소스와 자동 바인딩 여부(체크할 경우 실행 시 자동 조회)
공용테이블 연결	공용테이블 데이터소스를 생성하여 연결
editable	그리드 전체의 수정 여부이며 언체크시 그리드 데이터 수정 불가
movable	헤더를 드래그하여 순서 변경 가능 여부
rowNo	맨 왼쪽 행 번호 표시 여부
selectable	유저가 그리드의 행을 선택할 수 있는지 여부
checkable	행 번호 다음 열에 체크박스를 표시할지 유무
sortable	헤더를 클릭했을 시 정렬 기능이 작동할지 여부
shiftCheck	체크바 다중 체크, 언체크 가능
requiredCheck	그리드의 필수 입력 열에서 엔터키 입력 시 포커스 이동 여부
stateColumn	그리드의 행 상태(추가/수정)를 표시해주는 상태 컬럼 설정 여부
masterGrid	디테일 마스터 그리드 관계에서 마스터 그리드 역할을 할지 여부이며 디테일 그리드 경우 마스터 그리드를 선택하면 자동으로 연결
noData	데이터가 없을 때 데이터 없음 메시지가 출력될지 여부
noDataMessage	데이터가 없을 때 출력될 메시지 설정
fillWidth	그리드 컬럼 전체 너비보다 그리드 너비가 클 경우 컬럼 너비를 지정하지 않은 행에서 자동 크기 설정 여부
resizable	마우스 드래그를 통한 컬럼 너비 변경 여부
context	그리드 컨텍스트 메뉴 기본값을 설정
skipEdit	그리드 내 좌우방향키, 엔터키로 이동시 수정 불가능한 셀일 경우 건너뛰는 기능 적용 여부
copyMode	그리드 복사 시 셀(cell) 또는 행(row) 복사 설정
fixed	컬럼, row 고정 설정(colCount, rightColCount)
useUserGroup	사용자 그룹 사용 여부
group	그룹을 설정할 컬럼 설정
groupHeaderTemplate	그리드 그룹핑 사용시 그룹 헤더 문구 설정
styles	그리드의 다양한 스타일 지정 카테고리(rowNo, cellBorder, font, color, background, textAlign, padding, heightType, height)
그리드 이벤트	change, changing, dataBound, save, validationCheck, rowAdd, rowRemove, cellEditing, selected, dblClicked, clicked, codedialog, keyDown, keyUp, headerClicked, columnHeaderCheck, scrollToBottom...

<표 3-36. 그리드 속성 및 설명>

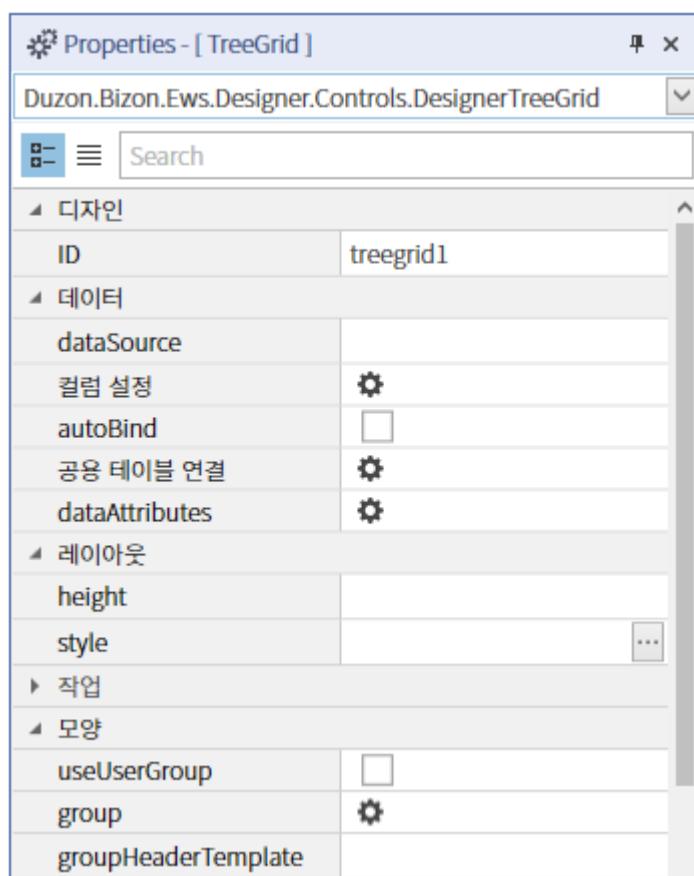
2) 트리그리드

No	✓	아이디	이름	나이	주소
>	<input type="checkbox"/>	1	고길동	40	서울시 은평구
2	<input type="checkbox"/>	2	고길동아들1	20	서울시 은평구
3	<input type="checkbox"/>	3	고길동손주1	1	서울시 은평구
4	<input type="checkbox"/>	5	고길동손주2	1	서울시 은평구
5	<input type="checkbox"/>	5	고길동아들2	18	서울시 은평구

<그림 3-80. 트리그리드 화면>

트리그리드는 각 필드에 행 단위의 데이터가 저장되어 계층 구조 형태로 보여지는 컨트롤이다.
각 행에는 텍스트뿐만 아니라 다양한 컨트롤들이 들어갈 수 있다.(트리그리드의 index 번호는
0번부터가 아닌 1번부터 시작)

① 트리그리드 속성

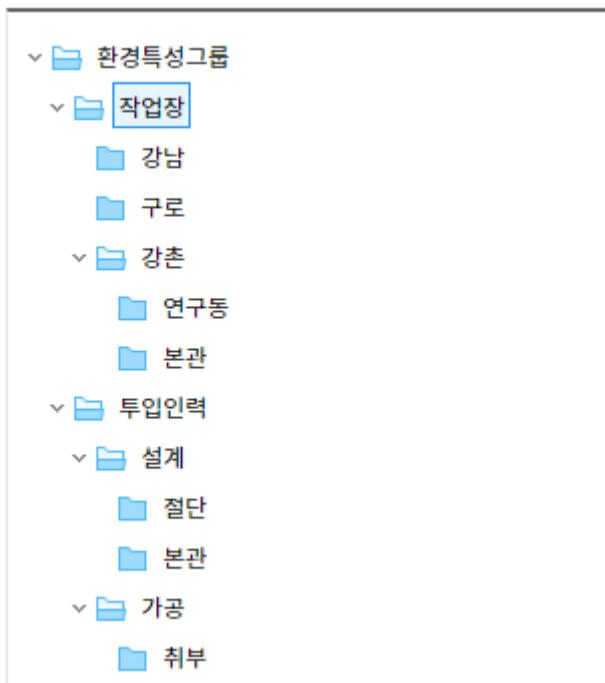


<그림 3-81. 트리그리드 속성 화면>

프로퍼티	설명
ID	트리그리드 ID는 treegrid1 방식으로 번호가 매겨지며, 편집 가능
dataSource	바인딩 될 데이터소스 설정(데이터소스 Grid Type이 TreeGrid)
컬럼 설정	2장 그리드 및 데이터소스 부분 참고
autoBind	데이터소스와 자동 바인딩 여부(체크할 경우 실행 시 자동 조회)
공용테이블 연결	공용테이블 데이터소스를 생성하여 연결
editable	트리그리드 전체의 수정 여부
movable	헤더를 드래그하여 순서 변경 가능 여부
rowNo	맨 왼쪽 행 번호 표시 여부
selectable	유저가 그리드의 행을 선택할 수 있는지 여부
checkable	행 번호 다음 열에 체크박스를 표시할지 유무
sortable	헤더를 클릭했을 시 정렬 기능이 작동할지 여부
shiftCheck	체크바 다중 체크, 언체크 가능
requiredCheck	필수 입력 열에서 엔터키 입력 시 포커스 이동 여부
stateColumn	행 상태(추가/수정)를 표시해주는 상태 컬럼 설정 여부
masterGrid	디테일 마스터 그리드 관계에서 마스터 그리드 역할을 할지 여부이며 디테일 그리드 경우 마스터 그리드를 선택하면 자동으로 연결
noData	데이터가 없을 때 데이터 없음 메시지가 출력될지 여부
noDataMessage	데이터가 없을 때 출력될 메시지 설정
fillWidth	트리그리드 컬럼 전체 너비보다 그리드 너비가 클 경우 컬럼 너비를 지정하지 않은 행에서 자동 크기 설정 여부
resizable	마우스 드래그를 통한 컬럼 너비 변경 여부
context	트리그리드 컨텍스트 메뉴 기본값을 설정
skipEdit	트리그리드 내 좌우방향키, 엔터키로 이동시 수정 불가능한 셀일 경우 건너뛰는 기능 적용 여부
copyMode	트리그리드 복사 시 셀(cell) 또는 행(row) 복사 설정
fixed	컬럼, row 고정 설정(colCount, rightColCount)
useUserGroup	사용자 그룹 사용 여부
group	그룹을 설정할 컬럼 설정
groupHeaderTemplate	트리그리드 그룹핑 사용시 그룹 헤더 문구 설정
styles	트리그리드의 다양한 스타일 지정 카테고리(rowNo, cellBorder, font, color, background, textAlign, padding, heightType, height)
트리그리드 이벤트	change, changing, dataBound, save, validationCheck, rowAdd, rowRemove, cellEditing, selected, dblClicked, clicked, codedialog, keyDown, keyUp, headerClicked, columnHeaderCheck, scrollToBottom...

<표 3-37. 트리그리드 속성 및 설명>

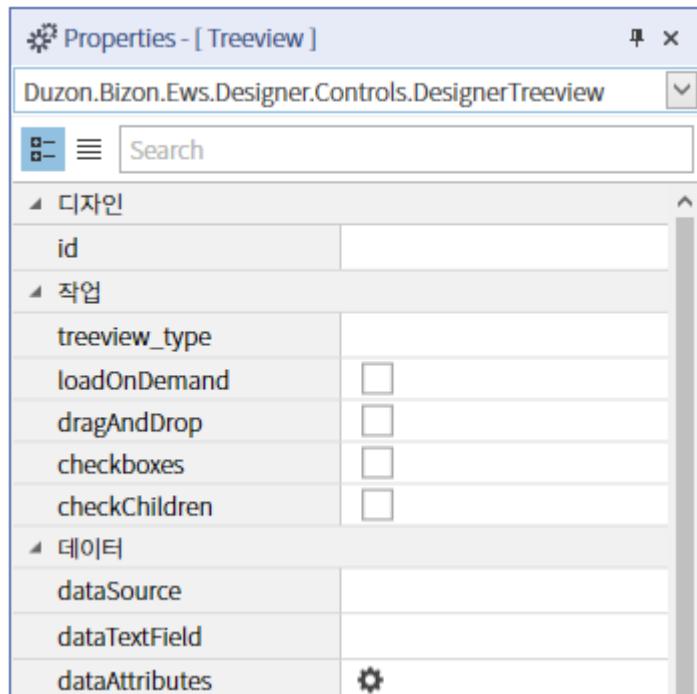
3) 트리뷰



<그림 3-82. 트리그리드 화면>

트리뷰는 폴더 구조로써 자료들의 계층적인 구조를 나타낼 때 사용한다.)

① 트리그리드 속성

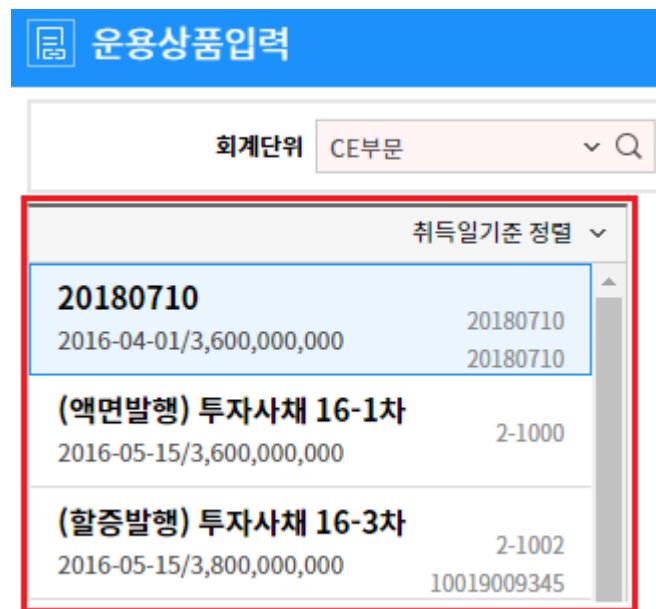


<그림 3-83. 트리그리드 속성 화면>

프로퍼티	설명
ID	트리뷰 ID는 자동으로 생성되지 않으며 사용자가 직접 입력
treeview_type	트리뷰 타입 설정(조직도, 탐색기)
loadOnDemand	부모 노드를 확장 시에만 자식 노드 생성 여부
dragAndDrop	트리뷰 노드의 드래그&드롭 기능 가능 여부
checkboxes	트리뷰의 체크박스 설정 여부
checkChildren	트리뷰의 부모노드 체크 시 자식노드의 체크 설정 여부
dataSource	바인딩 될 데이터소스 설정(트리뷰는 선택이 아닌 수동으로 선택)
dataTextField	트리에 표시될 text 필드 설정
dataSpriteCssClassField	조직도 상태를 나타내는 값 필드 설정
treePadding	트리의 여백 설정(기본 10)
template	노드를 표현할 템플릿 설정(dataTextField 보다 우선)
collapse	트리뷰의 하위노드가 접힐 때 발생하는 이벤트
click	트리뷰 아이템 노드를 선택할 때 발생하는 이벤트
change	트리뷰 아이템의 요소를 클릭할 때 발생하는 이벤트
check	체크박스를 체크할 때 발생하는 이벤트
dataBound	데이터소스의 change 이벤트가 발생한 뒤 발생하는 이벤트
expand	하위 노드가 펼쳐지기 전에 발생하는 이벤트
navigate	사용자가 키보드를 이용하여 포커스를 이동시킬 때 발생하는 이벤트
drag	노드가 드래그 될 때 발생하는 이벤트
drop	노드가 드랍될 때 발생하는 이벤트
dragstart	노드 드래그가 시작되기 전에 발생하는 이벤트
dragend	노드가 드랍된 이후에 발생하는 이벤트

<표 3-38. 트리뷰 속성 및 설명>

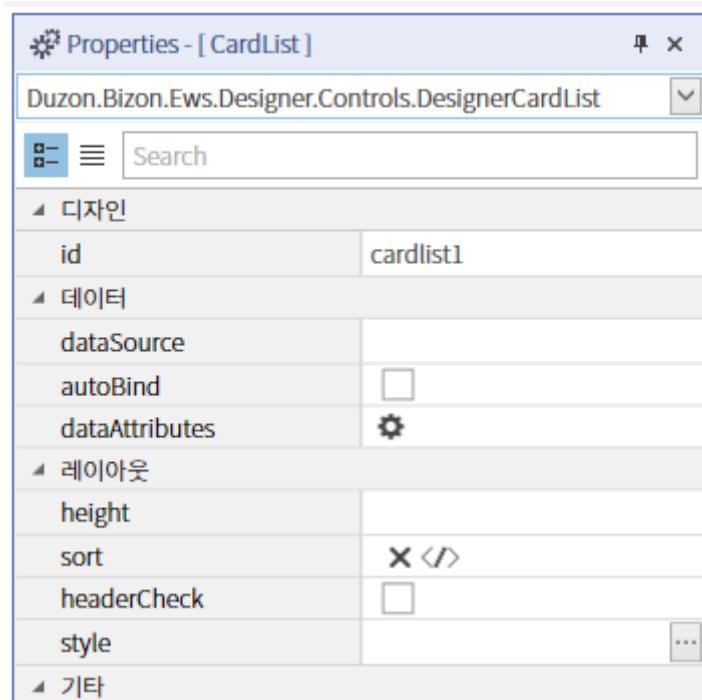
4) 카드리스트



<그림 3-84. 카드리스트 화면>

카드 형태의 정보를 담고 있는 리스트를 생성하여 사용하는 데이터 컨트롤이다.

① 카드리스트 속성



<그림 3-85. 카드리스트 속성 화면>

프로퍼티	설명
ID	카드그리드 ID는 cardlist1, cardlist2 방식으로 자동 번호가 생성되며, 편집 가능
dataSource	바인딩 될 데이터소스
autoBind	처음 로드될 때 자동으로 바인딩 유무
sort	카드리스트의 정렬 설정
headerCheck	카드리스트의 헤더에 체크박스 유무 설정
template	카드리스트의 템플릿 설정
change	행 변경시 발생 하는 이벤트
dataBound	데이터 바인딩 후에 이벤트
rowAdd	행 추가시 발생하는 이벤트
rowRemove	행 삭제시 발생하는 이벤트

<표 3-39. 카드리스트 속성 및 설명>

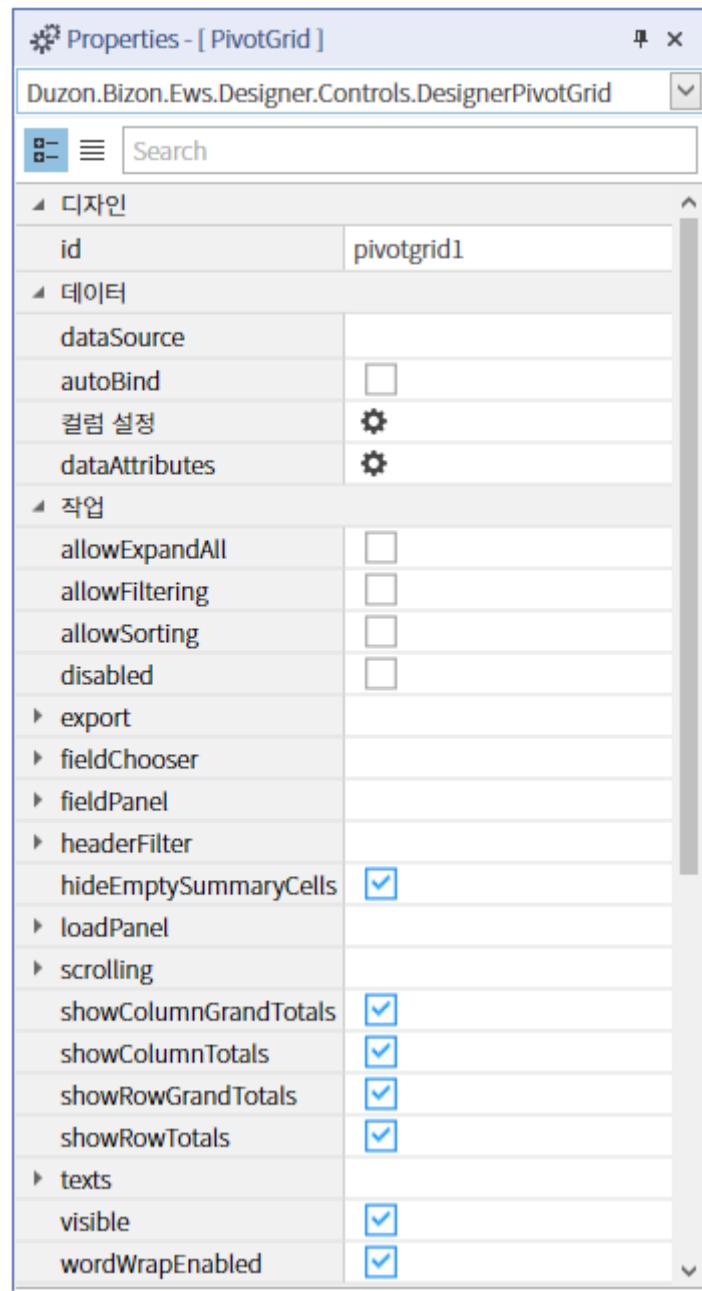
5) 피벗그리드

+	+ 2014		+ 2015		전체 합계	
	합계	누적 합계	합계	누적 합계	합계	누적 합계
+ 남 아메리카	100,480	100,480	50,315	150,795	150,795	150,795
+ 북 아메리카	204,205	204,205	72,925	277,130	277,130	277,130
+ 아시아	200,865	200,865	111,540	312,405	312,405	312,405
+ 아프리카	45,200	45,200	22,020	67,220	67,220	67,220
+ 오스트렐리아	76,130	76,130	42,270	118,400	118,400	118,400
+ 유럽	216,425	216,425	83,635	300,060	300,060	300,060
전체 합계	843,305	843,305	382,705	1,226,010	1,226,010	1,226,010

<그림 3-86. 피벗그리드 화면>

피벗그리드는 각 필드에 컬럼, 행에 데이터가 저장되어 표 형태로 보여지는 컨트롤이다.

① 피벗그리드 속성



<그림 3-87. 피벗그리드 속성 화면>

프로퍼티	설명
ID	피벗그리드 ID는 pivotgrid1, pivotgrid2 방식으로 번호가 매겨 지며, 사용자 지정도 가능
dataSource	피벗 그리드의 데이터 소스 설정
autoBind	피벗 그리드의 데이터를 자동 바인딩 유무
컬럼 설정	<p>피벗 그리드의 컬럼을 설정</p> <p>※ 상세 속성</p> <p>dataField : 필드에 데이터를 제공하는 데이터 소스 필드 지정</p> <p>datatype : 필드에 데이터타입 지정</p> <p>caption : 필드 세팅의 필드명 설정</p> <p>Use : 보이기 여부</p> <p>area : 필드의 영역 지정</p> <p>AreaIndex : 같은 영역의 다른 필드 중 필드의 순서 지정</p> <p>AllowCrossGroupCalculation : 누적 합계 및 절대 또는 백분율 변화를 계산 할 때 이웃 그룹의 요약 값을 고려할지 여부 지정</p> <p>AllowSorting : 사용자가 필드의 정렬을 변경할 수 있는지 여부 지정</p> <p>AllowExpandAll : 컨텍스트 메뉴를 사용하여 동일한 열 또는 행 머리글 수준의 모든 항목을 사용자가 펼치기/접기를 할 수 있는지 여부</p> <p>AllowFiltering : 사용자가 필드 값을 필터링 할 수 있는지 여부</p> <p>Expanded : 필드의 확장여부 설정</p> <p>Format : 필드 값이 표시되는 형식 지정</p> <p>Predefined : 공통포맷 사용여부</p> <p>GroupIndex : 해당 그룹 내의 필드 인덱스 지정</p> <p>GroupInterval : 필드의 값이 그룹으로 결합되는 방법 지정</p> <p>GroupName : 필드의 그룹 이름 지정</p> <p>SummaryType : 필드의 데이터를 집계하는 방법 지정(기본은 Count)</p> <p>RunningTotal : 행 또는 열별로 누적 합계를 계산할지 여부</p>
allowExpandAll	헤더 레벨의 모든 머리글 항목을 확장/축소 사용 여부
allowFiltering	팝업 메뉴에서 값을 선택하거나 선택 취소하여 컬럼을 필터링 할 수 있는지 여부
allowSorting	정렬 옵션 변경 여부
disabled	비활성화 여부
export	<p>엑셀로 내보내기를 구성</p> <p>※ 상세 속성</p> <p>enabled : 엑셀 내보내기 사용 여부</p> <p>ignoreExcelErrors : 내보낸 문서에 오류가 있는 경우 Excel에서</p>

	<p>경고를 숨길 지 여부 지정 filename : 데이터를 내보낼 파일의 기본 이름 지정</p>
fieldChooser	<p>필드 선택의 구성 옵션 ※ 상세 속성</p> <p>allowSearch : 필드 선택창에서 "모든 필드" 섹션에서 검색을 허용하는지 여부 지정</p> <p>enabled : 필드 선택기를 활성화하거나 비활성화 설정</p> <p>title : 필드 선택창 팝업 창의 제목으로 표시 할 텍스트 지정</p> <p>applyChangeMode : 필드 선택창에서 변경 한 내용을 피벗그리드에 적용 할 시기 지정</p> <p>width : 필드 선택창 너비 지정</p> <p>height : 필드 선택창 높이 지정</p> <p>layout : 필드 선택창 레이아웃 지정</p> <p>searchTimeout : 검색이 실행될 때 까지의 지연 지정</p> <p>texts : 피벗그리드의 통합 필드 선택기에서 변경하거나 지역화 할 수 있는 문자열 설정</p>
fieldPanel	<p>필드 패널을 구성 ※ 상세 속성</p> <p>allowFieldDragging : 필드 패널의 컬럼을 드래그 가능 여부</p> <p>showColumnFields : 필드 패널에서 열 필드 표시 여부</p> <p>showDataFields : 필드 패널에서 값 필드 표시 여부</p> <p>showFilterFields : 필드 패널에서 필터 필드 표시 여부</p> <p>showRowFields : 필드 패널에서 행 필드 표시 여부</p> <p>visible : 필드 패널 표시 여부</p> <p>text : 필드 영역의 문자열 지정</p>
headerFilter	<p>헤더 필터 기능을 구성 ※ 상세 속성</p> <p>allowSearch : 헤더 필터에서 검색을 사용할지 여부 지정</p> <p>width : 필터링 값이 포함 된 팝업 메뉴의 너비 지정</p> <p>height : 필터링 값이 포함 된 팝업 메뉴의 높이 지정</p> <p>searchTimeout : 검색이 실행 될 때까지의 지연 지정</p> <p>texts : 팝업 메뉴의 요소의 텍스트 구성</p>
hideEmptySummaryCells	데이터가 없는 행과 열을 숨길지 여부
loadPanel	<p>로드 패널을 구성하는 옵션 지정 ※ 상세 속성</p> <p>enabled : 로드 패널 사용 여부</p>
scrolling	스크롤 옵션 지정

	※ 상세 속성 mode : 스크롤 모드 지정
showColumnGrandTotals	총합계 열을 표시할지 여부 지정
showColumnTotals	소계 열을 표시할지 여부 지정
showRowGrandTotals	총합계 행을 표시할지 여부 지정
showRowTotals	소계 행을 표시할지 여부
visible	피벗그리드 컴포넌트가 표시되는지 여부
wordWrapEnabled	머리글 항목의 긴 텍스트를 줄 바꿈 할지 여부 지정
dataFieldArea	데이터 필드 머리글이 보여야 하는 영역 지정
rowHeaderLayout	행 머리글의 항목 레이아웃 지정
state	피벗그리드의 필드 구성을 저장할지 설정
style	스타일 속성 지정
contentReady	피벗그리드 컴포넌트의 내용이 준비되면 발생하는 이벤트
cellClick	사용자가 피벗그리드 셀을 클릭하면 발생하는 이벤트
cellPrepared	셀이 준비된 이후에 발생하는 이벤트
optionChanged	피벗그리드 옵션이 변경된 후에 발생하는 이벤트

<표 3-40. 피벗그리드 속성 및 설명>

[제4장] 스크립트

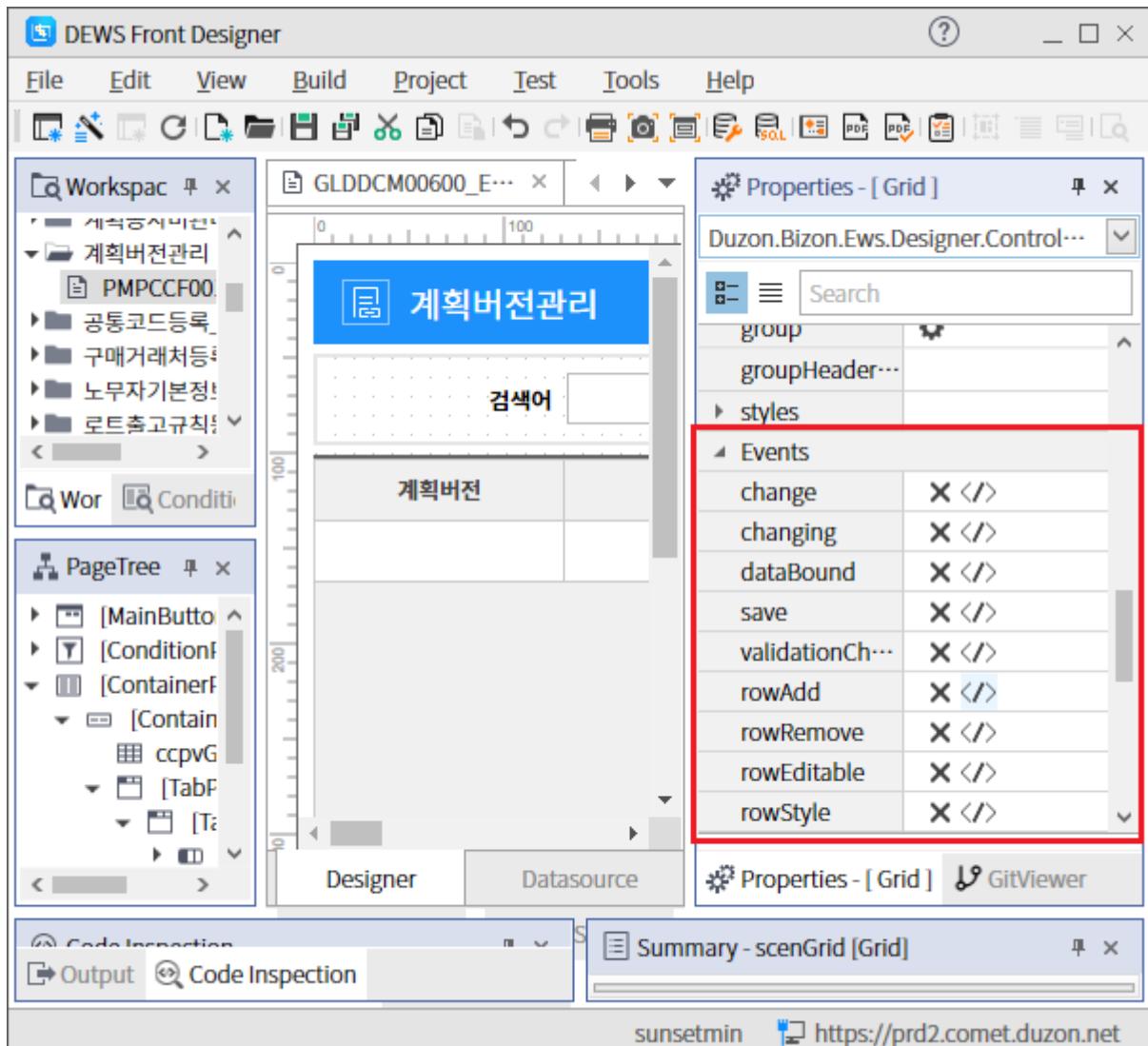
스크립트 사용법

PreReady/Ready 스크립트

통합 Script

조회/추가/삭제/저장/인쇄 스크립트 작성

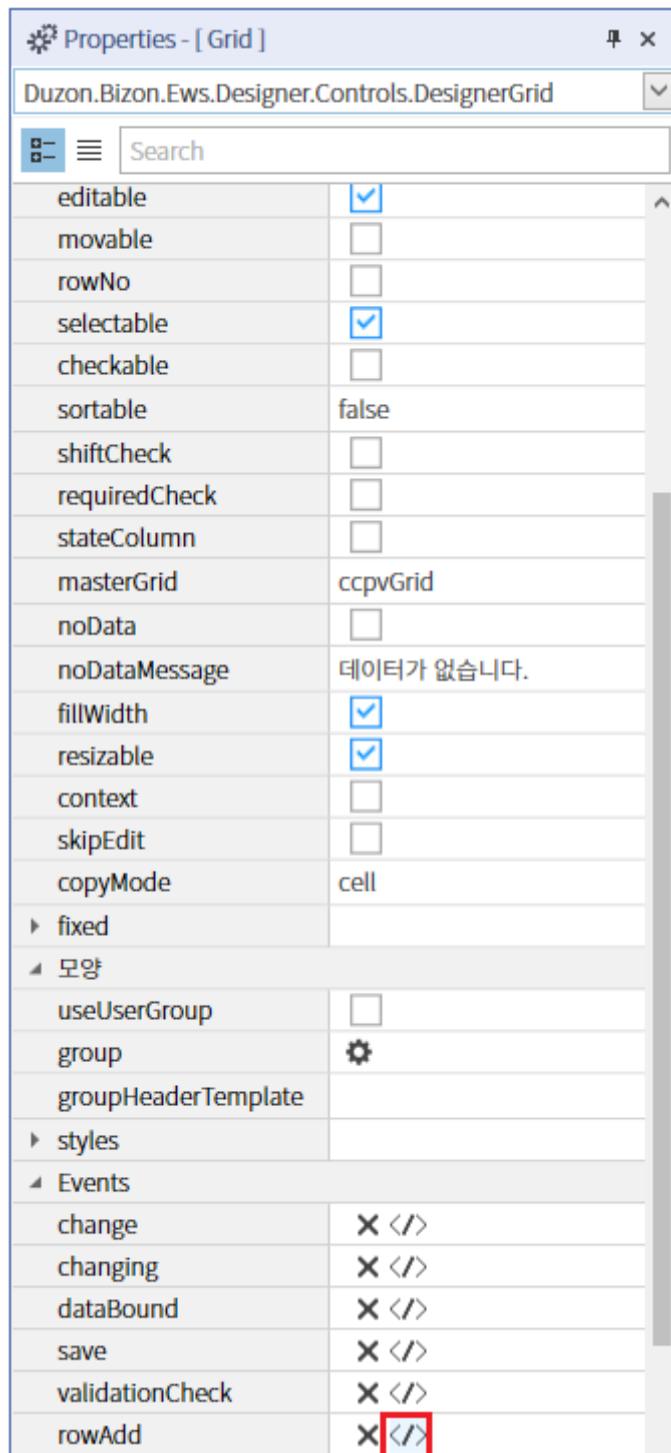
1. 스크립트 사용법



<그림 4-1. 우측 상단의 프로퍼티 창에 이벤트 목록이 표시된 화면>

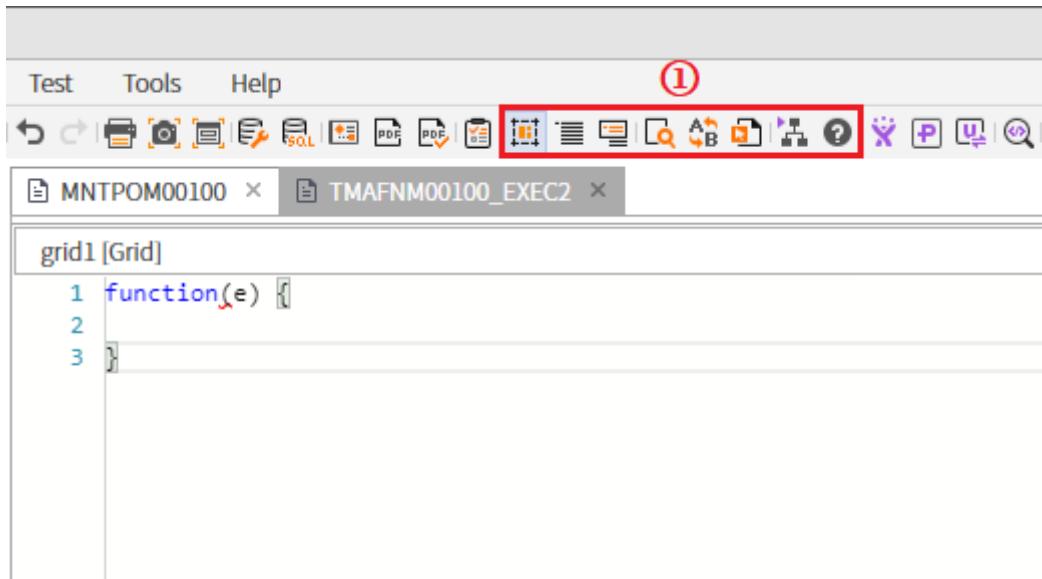
화면 디자인 작업과 데이터소스 생성 및 설정 후 이벤트 스크립트를 작성하여 동작을 제어할 수 있다. 기본적으로 스크립트 작성은 자바스크립트, 제이쿼리, dews 문법을 사용한다.

또한 스크립트를 작성하기 위해서는 이벤트는 오른쪽 프로퍼티창의 Events 카테고리에서 확인할 수 있다. 컨트롤에 따라 작성할 수 있는 이벤트 목록이 다르므로 이벤트 종류를 확인하고 스크립트를 작성한다. 또한 통합 Script 에서 직접 컨트롤을 선택하고 이벤트를 선택하여 스크립트를 작성할 수 있다.



<그림 4-2. 이벤트 목록이 표시된 프로퍼티 창>

이미 작성된 스크립트가 있다면 위 그림처럼 해당 이벤트의 </> 버튼이 </>으로 표시된다.

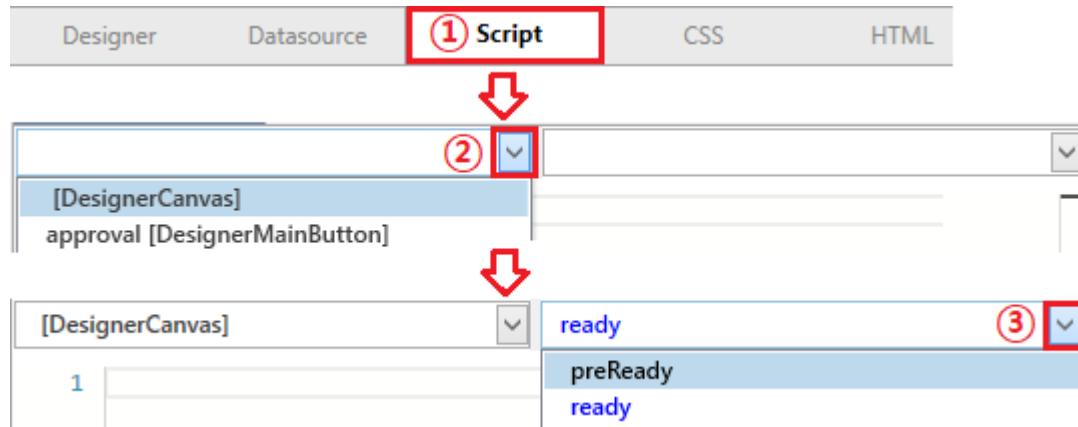


<그림 4-3. 이벤트 스크립트 창 실행된 화면>

스크립트 버튼(<!-->)을 누르면 <그림 4-3>처럼 자바스크립트 편집창이 나타난다.

툴바 아이콘(①)으로는 자동정렬, 라인 주석, 블록 주석, 찾기, 바꾸기, 모든 참조 찾기, 미니맵 표시, 도움창 등의 기능이 있다. 특히 오른쪽의 물음표 아이콘을 누르면 (F1 키) 에디터 사용에 관한 도움말을 확인 할 수 있다.

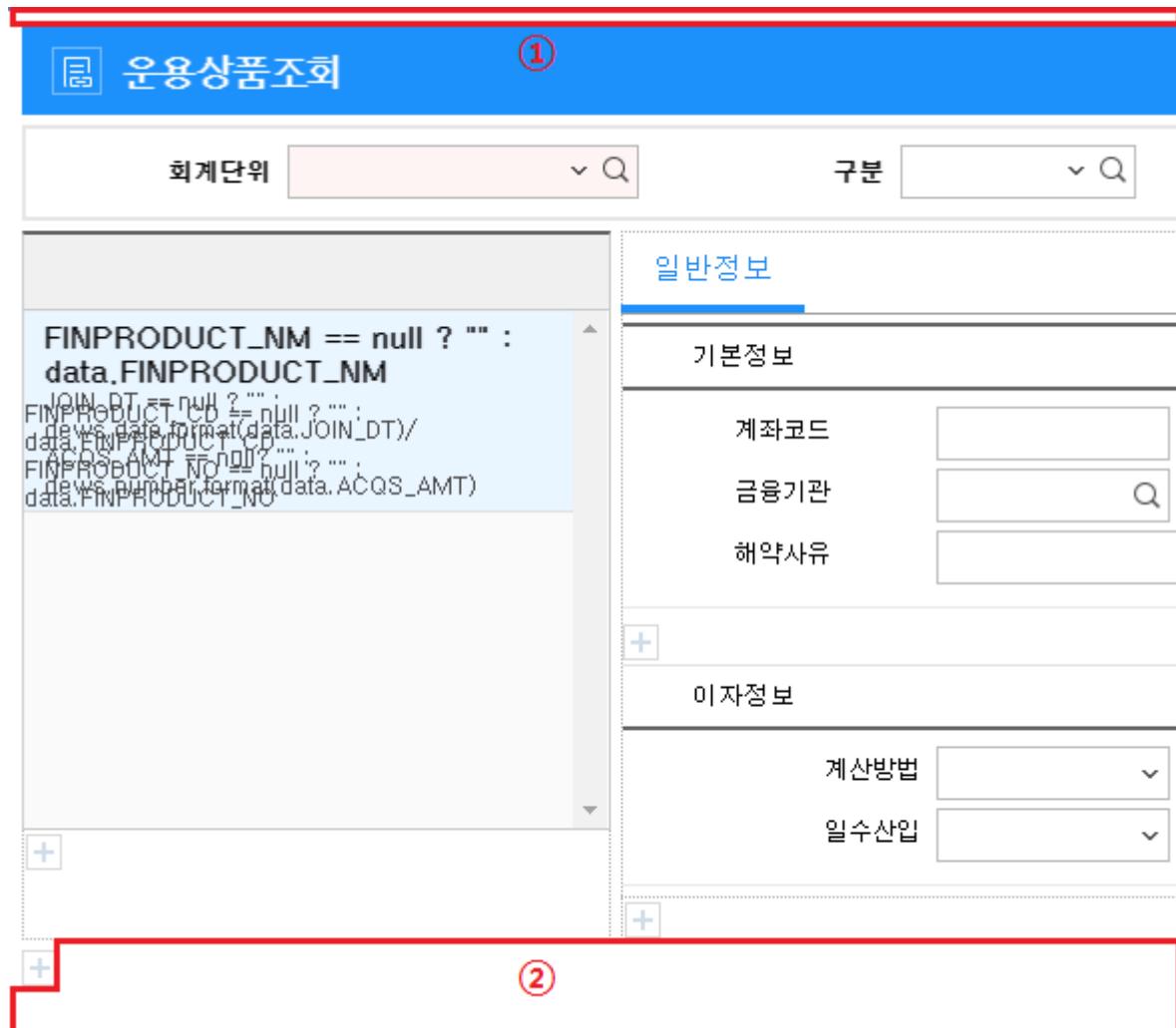
다른 방법으로 <그림 4-4>와 같이 ① 스크립트 탭에서 ② 상단 왼쪽 콤보박스에서 컨트롤을 선택하고 ③ 오른쪽 콤보박스에서 해당 컨트롤에 해당하는 이벤트를 선택하여 작성한다. 자세한 내용은 뒤에 통합 Script 부분에서 다룬다.



<그림 4-4. 스크립트 탭에서 직접 이벤트 스크립트 입력>

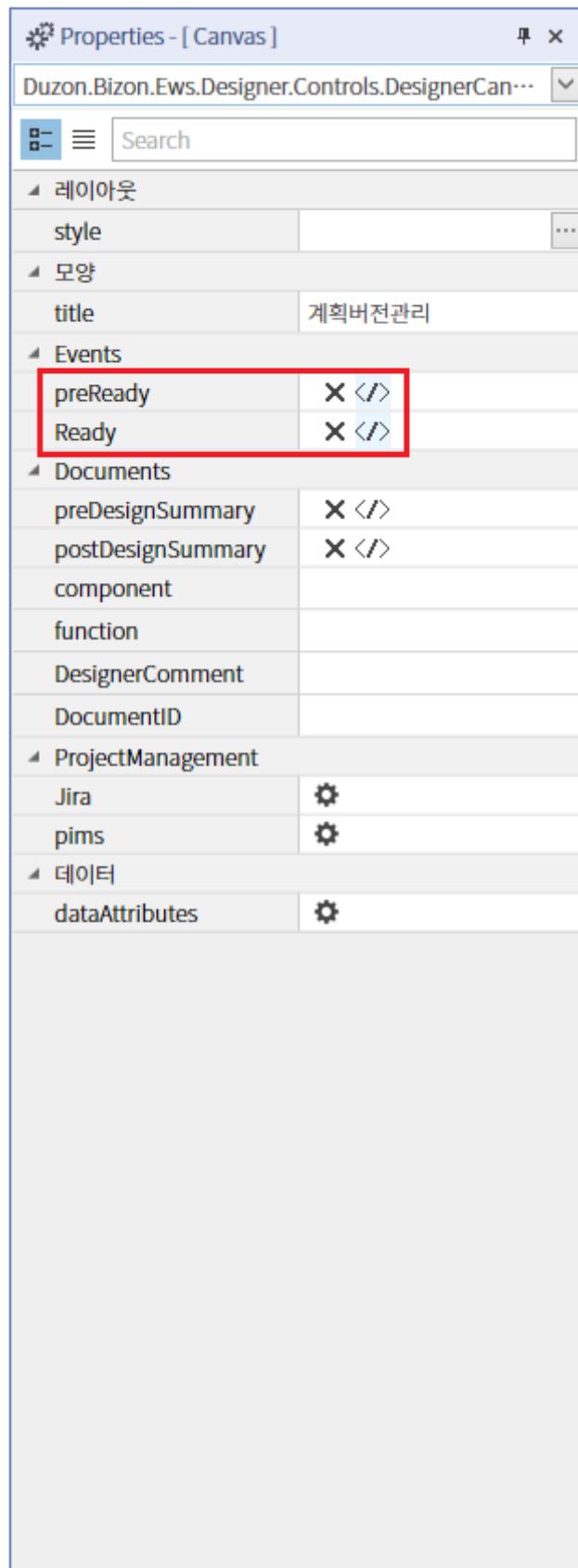
스크립트를 모두 작성한 후 마우스를 다른 탭이나 이벤트를 선택하면 자동 저장된다.

2. Canvas의 preReady/Ready 상세



<그림 4-5. 디자이너에서 Canvas 영역을 선택하는 화면>

① 페이지 타이틀 위 또는 ② 빈 영역을 클릭하면 Canvas가 선택되며, 브라우저에서 해당 페이지가 로드될 때 실행되는 preReady/Ready 스크립트를 작성 할 수 있다.



<그림 4-6. Canvas 영역 선택시 나타나는 속성창>

preReady 이벤트는 페이지가 로드될 때 스크립트 태그에서 가장 앞부분에서 구현되고, Ready 이벤트는 스크립트 태그의 마지막 부분에서 구현된다.

The screenshot shows a software interface for designing web pages. On the left, there's a 'DesignerCanvas' panel where a script editor displays code. On the right, there are several toolbars and panels related to the design environment. The script editor contains the following code:

```
1 var dewself = this;
2 var mstGrid , mstGridDataSource;
3 var dtlGrid , dtlGridDataSource;
4 var subGrid1 , subGridDataSource1;
5 var subGrid2 , subGridDataSource2;
6 var subGrid3 , subGridDataSource3;
7 var objCodeDtl = {};
8 var chkSearch = '';
9 dewself.ui.mainbuttons.delete.useDefaultConfirm = false;
10 dewself.ctrlCdDisch.options.helpParams = {cd_saorg: ""};
11 dewself.ctrlCdDivision.options.helpParams = {cd_saorg: "", cd_
12
13 var CD_SEARCHDataSource = dewself.ui.dataSource("CD_SEARCHDataSource");
14 |   data: [
15 |     { CD_SYSDEF: "1", NM_SYSDEF: "영업거래처 등록" },
16 |     { CD_SYSDEF: "2", NM_SYSDEF: "영업거래처 미등록" }
17 |   ]
18 });
19 /*
20 var ynUseDataSource = dewself.ui.dataSource("ynUseDataSource", {
21 |   data: [
22 |     { CD_SYSDEF: "Y", NM_SYSDEF: "Yes" },
23 |     { CD_SYSDEF: "N", NM_SYSDEF: "No" }
24 |   ]
25 });
26 */
27
28 //코드디테일
29 function getCodeData(cd_module, cd_field_pipe, yn_sycode, yn_
30   if (!objCodeDtl.hasOwnProperty(cd_module)) {
31     objCodeDtl[cd_module] = {};
32   }
33   $.each(cd_field_pipe.split("|"), function (i, v) {
34     if (v != null && v != "") {
35       objCodeDtl[cd_module][v] = [];
36     }
37   });

```

<그림 4-7. preReady 스크립트 예시 화면 – 페이지에서 사용할 전역 변수들을 세팅>

먼저 preReady 스크립트에는 보통 전역으로 사용할 변수와 함수들을 세팅한다. 위의 예시화면에서도 전역으로 사용할 페이지의 변수에 대해 세팅하고 있다.

preReady에서 선언한 변수들은 다른 컨트롤의 이벤트에서 스크립트를 작성할 때 참조 할 수 있다. 예를 들어 <그림4-7>에서 mstGrid 라고 선언한 변수는 전역변수로써 어디서나 접근 가능한 변수로 사용 가능하다.

또한 전역으로 사용할 함수들을 미리 선언하여 다른 컨트롤에 이벤트를 작성할 때 호출하여 사용한다.

<그림 4-8. preReady 스크립트 예시 화면 – 페이지에서 사용할 전역 함수들을 세팅>

위 그림에서도 `getCodeData()`라는 함수는 전역함수로써 어디서나 접근 가능한 함수로 동작한다.

또한 preReady에는 처음 실행 시 초기 설정 하는 부분이 들어간다. <그림 4-9>에서 로그인 사용자 회계단위 정보와 현재 날짜를 초기값으로 설정하는 스크립트이다.

The screenshot shows a code editor window titled "[DesignerCanvas]" with a dropdown menu and a tab labeled "preReady". The code in the editor is:

```
1 //로그인 사용자 회계단위 Default
2 self.PC_CD_C.setData([{
3   PC_CD: self.user.profitCenterCode,
4   PC_NM: self.user.profitCenterName
5 }]);
6
7 //시스템 일자 Default
8 self.STD_DT_C.value(new Date());
```

<그림 4-9. preReady 스크립트 예시 화면 – 실행 시 초기 설정 부분>

Ready에는 데이터소스 생성이나 컨트롤 설정 후에 실행되며, 컨트롤 간의 관계 설정하는 코드들을 작성한다. <그림 4-10>은 카드리스트와 그리드 컨트롤 간에 Master와 Detail을 설정하는 코드이다. 만약 preReady에 이 코드를 작성했을 경우 컨트롤 생성 및 설정 이전이므로 오류가 발생한다.

The screenshot shows a code editor window titled "[DesignerCanvas]" with a dropdown menu and a tab labeled "ready". The code in the editor is:

```
1 // 카드리스트와 그리드 M-D 설정 (Master : Card - Detail : mstGrid)
2 // 참조: 그리드와 그리드 M-D 설정은 디테일 그리드 속성에서 설정
3 self.mstCard.setDetail(self.mstGrid);
4
5 |
```

<그림 4-10. Ready 스크립트 예시 화면 – 컨트롤간의 관계 설정>

스크립트를 작성할 때 컨트롤이나 데이터소스에 접근해야 할 필요가 있다. id는 해당 컨트롤을 식별할 고유 속성이며, 전역 객체인 self를 이용하여 컨트롤 id에 접근 할 수 있다.

DataSource

	Description
<input type="checkbox"/>	_uid
<input type="checkbox"/>	계획버전
<input type="checkbox"/>	계획버전명
<input type="checkbox"/>	계획대상기간(from)
<input type="checkbox"/>	계획대상기간(to)
<input type="checkbox"/>	확정여부
<input type="checkbox"/>	비고
<input type="checkbox"/>	시나리오
<input type="checkbox"/>	시나리오명

<그림 4-11. 데이터소스 ID 확인>

예를 들어, 데이터소스의 ID가 ccpvDs인 경우 스크립트에서 self.ccpvDs로 접근이 가능하다.

```

search [MainButton] click
334 function() {
335   if (self.ccpvDs.getDirtyDataCount() != 0 || self.scenDs.getDirtyDataCount() != 0) {
336     dews.confirm('저장하지 않은 데이터가 있습니다.\n조회를 계속하시겠습니까?').no(function () {
337       dews.ui.snackbar.warning('취소되었습니다.');
338       return false;
339     });
340   } else {
341     searchData();
342   }
343 }
344
345
346

```

<그림 4-12. 전역 객체 self를 이용한 스크립트 세팅 화면>

<그림 4-12>에서 self.ccpvDs를 통해 데이터소스의 getDirtyDataCount() 함수를 호출한다.

3. 통합 Script

The screenshot shows the 'Designer' interface with the 'Script' tab highlighted by a red box. The code editor contains the following JavaScript code:

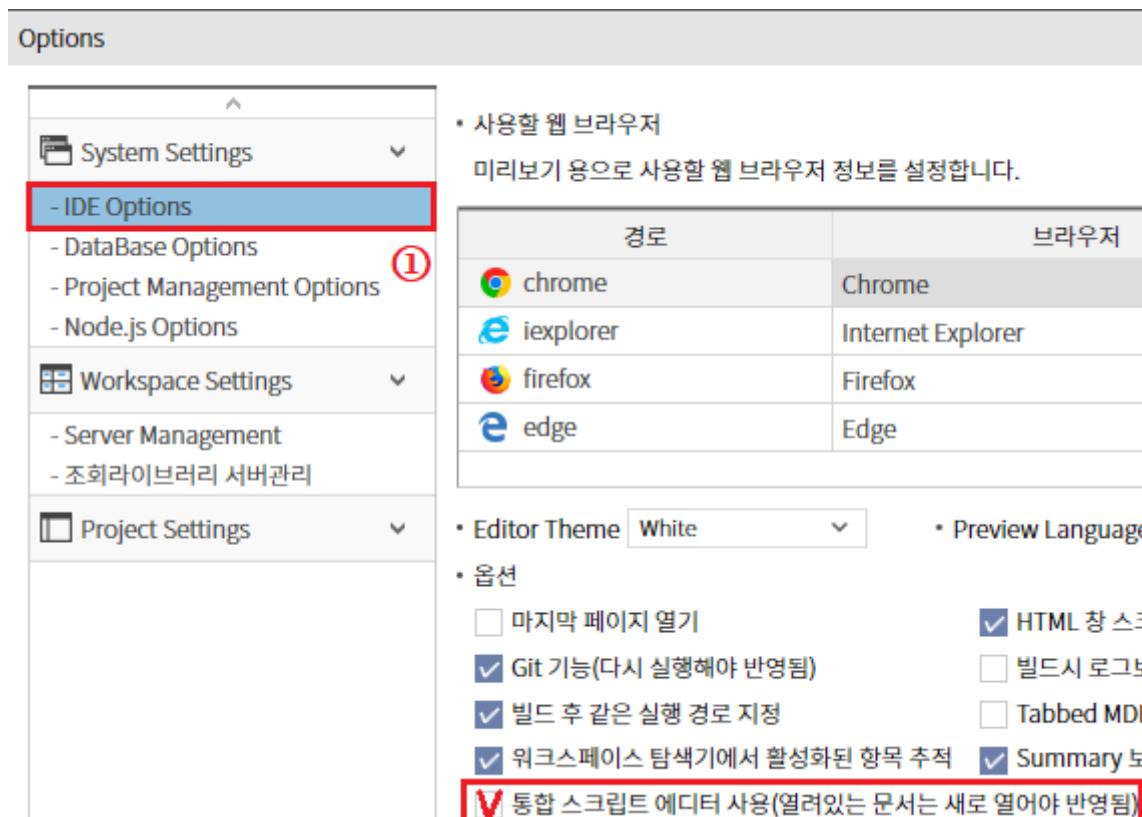
```

334 function(e) {
335     var ccpvData = self.ccpvGrid.dataItems(self.ccpvGrid.select());
336     this.setCellValue(e.row.index, "VER_CD", ccpvData.VER_CD);
337     this.setCellValue(e.row.index, "CNFM_YN", "N");
338 }

```

<그림 4-13. 통합 Script 화면>

디자이너 화면 세번째에 Script 탭이 있으며. 이 영역에서는 디자이너 화면에 그려진 모든 컨트롤들의 이벤트를 추가 및 수정할 수 있다. Tools의 Options의 IDE Options에서 통합 스크립트 에디터 사용을 체크한 경우(탭을 닫고 다시 열어야 반영) 작성한 모든 스크립트를 볼 수 있으며 편집도 가능하다.



<그림 4-14. Options의 통합스크립트 에디터 사용 체크>

단, 통합 스크립트 에디터 사용에 체크했을 때 주의할 점은 주석 부분과 자동 생성된 코드를 수정할 경우 오류가 발생하므로 반드시 콜백 함수 영역만 수정해야 한다

```
scenGrid [Grid] rowAdd
334 function(e) {
335     var ccpvData = self ccpvGrid.dataItems(self ccpvGrid.select());
336     this.setCellValue(e.row.index, "VER_CD", ccpvData.VER_CD);
337     this.setCellValue(e.row.index, "CNFM_YN", "N");
338 }
```

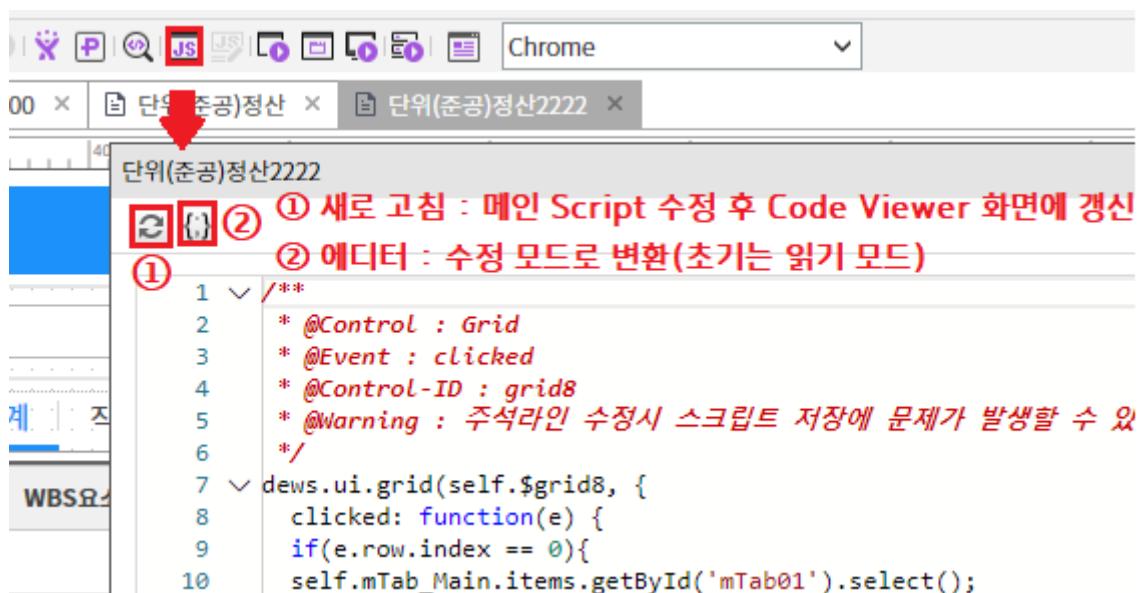
〈통합 스크립트 에디터 사용 미체크〉

```
341 /**
342 * @Control : Grid
343 * @Event : rowAdd
344 * @Control-ID : scenGrid
345 * @Warning : 주석라인 수정시 스크립트 저장에 문제가 발생할 수 있습니다.
346 */
347 dews.ui.grid(self.$scenGrid, {
348     rowAdd: function (e) {
349         var ccpvData = self ccpvGrid.dataItems(self ccpvGrid.select());
350         this.setCellValue(e.row.index, "VER_CD", ccpvData.VER_CD);
351         this.setCellValue(e.row.index, "CNFM_YN", "N");
352     }
353 });
★ 콜백 함수 영역만 반드시 수정
```

〈통합 스크립트 에디터 사용 체크〉

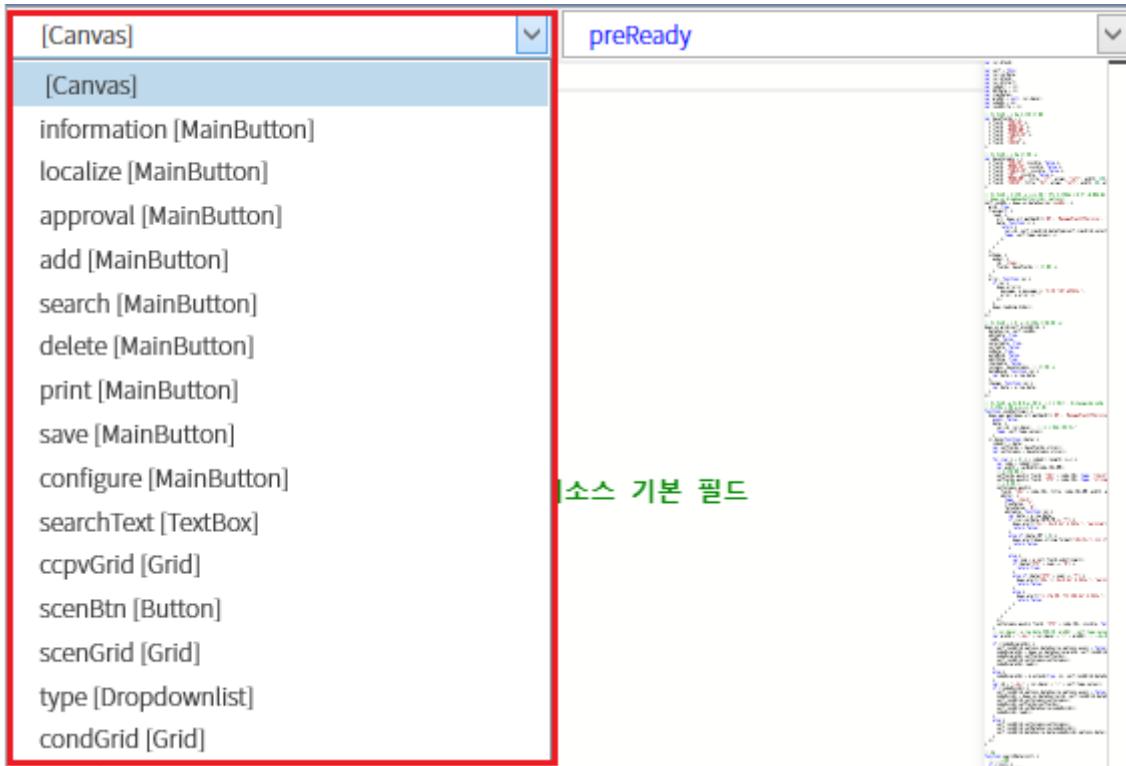
<그림 4-15. 통합 스크립트 에디터 사용 미체크와 체크 화면>

또한 사용자가 작성한 모든 스크립트는 툴바 Code Viewer를 클릭하여 확인할 수 있다.



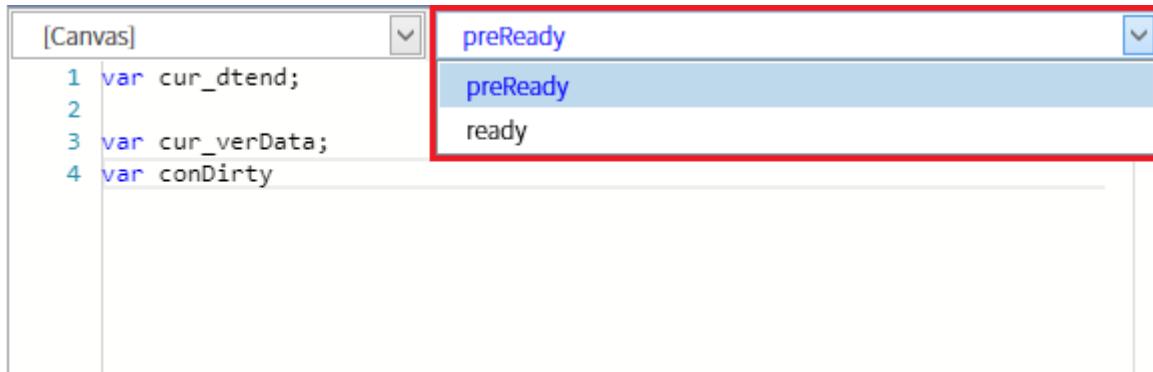
<그림 4-16. 작성한 모든 스크립트를 볼 수 있는 Code Viewer 화면>

Script 화면은 <그림 4-17>처럼 왼쪽 콤보박스에는 디자이너에 그려진 컨트롤 리스트를 확인할 수 있고, <그림 4-18>와 같이 오른쪽 콤보박스에는 선택한 컨트롤에 해당하는 이벤트가 표시된다.



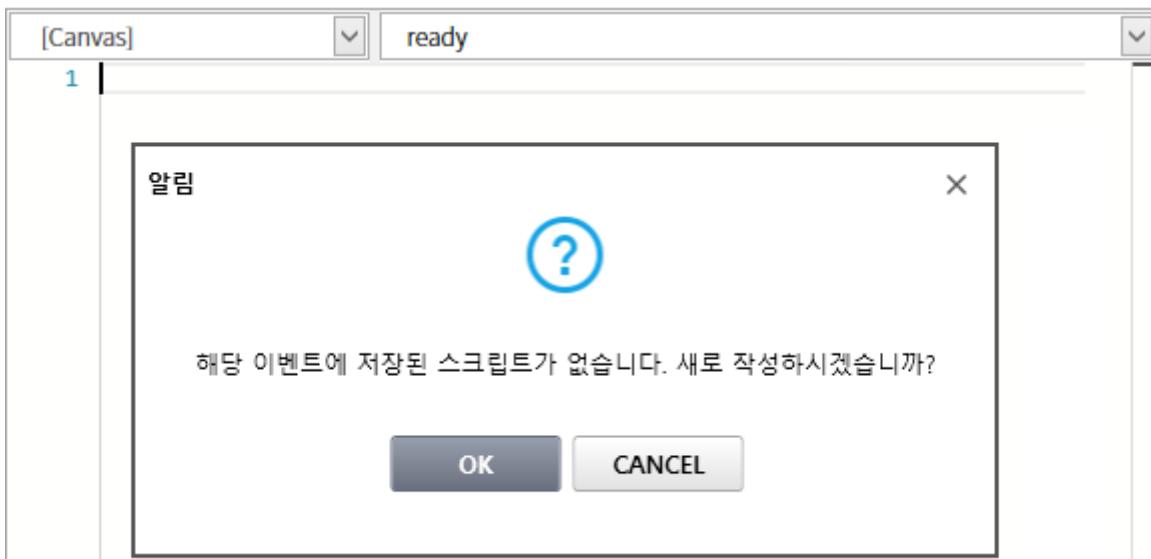
<그림 4-17. 통합 Script 디자이너 컨트롤 목록 된 화면>

이벤트 중 이미 작성된 이벤트는 파란색으로 표시되고 미작성된 이벤트는 검은색으로 표시된다.



<그림 4-18. 통합 Script에 컨트롤을 이벤트 목록 표시된 화면>

왼쪽에서 컨트롤을 선택하고 오른쪽에서 이벤트를 선택했을 때 이미 작성한 스크립트가 있을 경우 저장된 스크립트가 표시되며 처음 작성한 경우 <그림 4-19>처럼 “해당 이벤트에 저장된 스크립트가 없습니다. 새로 작성하시겠습니까?”라는 메세지가 나타난다. OK를 선택하면 해당 이벤트의 스크립트를 작성할 수 있고, CANCEL을 누르면 해당 이벤트 스크립트 작성을 취소한다.



<그림 4-19. 이벤트 작성 선택 메세지박스 화면>

```
//검색수행
function searchData() {
    dews.ui.loading.show({ text: '조회중입니다.' });

    try {
        self.mstCardDataSource.read();
        setTimeout(function () { dews.ui.loading.hide(); }, 100)
    } catch (exception) {
        dews.ui.loading.hide();
        dews.error(exception || '작업이 실패하였습니다.');
    }
}

// 컨트롤들 일괄로 Readonly 처리하는 함수!
function readonlyControl(flag) {
    $(".PSAPRF00300_MST").each(function (index, item) {
        if (self[item.id] != undefined) {
            self[item.id].readonly(false);
        }
    });
}
```

A screenshot of a code editor window titled '[Canvas] ready'. The code is written in a script-like language with line numbers on the left. The code defines two functions: 'searchData()' and 'readonlyControl()'. The 'searchData()' function shows a loading dialog and then reads data from a dataSource. The 'readonlyControl()' function iterates over elements with class 'PSAPRF00300_MST' and sets their 'readonly' property to false if they have an id.

<그림 4-20. 이벤트에 스크립트가 작성된 화면>

4. 조회/추가/삭제/저장/인쇄 스크립트



<그림 4-21. 디자이너 우측 상단 메인 버튼>

디자이너 우측 상단에 9개의 메인 버튼이 있다. 왼쪽부터 정보, 다국어, 결재, 추가, 조회, 삭제, 인쇄, 저장, 즐겨찾기 기능을 담당하는 메인 버튼의 기본적인 스크립트 작성법에 대해 알아본다.

추가, 조회, 삭제, 저장 버튼의 경우 click 이벤트에 저장된 스크립트가 없을 때 쉽게 작성할 수 있도록 가이드 스크립트가 주석으로 되어 있다. 필요한 경우 주석 부분을 지우고 사용할 수 있다.

1) 조회 스크립트 예제

```
search [MainButton]
click
1 function() {
2
3 // TODO: 필수항목 입력을 체크한 후 데이터소스를 조회합니다.
4 // if (self.searchCondition.validate({tooltip:true,message:'필수 항목을 입
5 // {
6 self.dataSource1.read();
7 // }
8
9 // TODO: 저장여부를 확인한 후 데이터소스를 조회합니다.
10 // if (self.dataSource1.getDirtyDataCount() != 0) {
11 // dews.confirm('저장하지 않은 데이터가 있습니다.\n조회를 계속하시겠습니까?
12 // self.dataSource1.read();
13 // }).no(function () {
14 // dews.ui.snackbar.warning('취소되었습니다.');
15 // return false;
16 // });
17 // } else {
18 // self.dataSource1.read();
19 // }
20 }
```

<그림 4-20. 조회 스크립트 예제 화면>

<그림 4-20>는 데이터소스에서 데이터를 불러오는 화면이다. script 가장 앞부분에 자동으로 선언되어 있는 self라는 전역변수를 통해 사용자가 추가한 dataSource1 컨트롤에 접근해 read()라는 메소드 호출을 통해 데이터를 조회하고 있다.

조회 가이드 스크립트는 2개 부분으로 되어 있다. 첫번째는 조회조건에 필수 입력 항목을 체크 하며 이 경우 반드시 컨디션패널 id에 searchCondition을 입력해야 한다. 두번째는 편집 중에 저장하지 않고 재조회를 할 경우 저장하지 않고 조회를 계속할 것인지 물어보는 메시지박스를 출력한다. 필요한 경우 각각 주석을 해제하고 활용할 수 있다. 또한 두 부분을 같이 사용할 경우

저장 여부 메세지를 띄울 때 필수 입력 항목 체크에 있는 데이터소스 read 함수가 실행되어 데이터를 조회하므로 <그림 4-21>과 같이 저장 여부 확인 메세지 부분을 필수 항목 입력 체크 부분으로 이동하고 사용한다.

The screenshot shows a code editor with a tab labeled "search [MainButton]" and a button labeled "click". The code is as follows:

```
1 function() {
2
3     // TODO: 필수항목 입력을 체크한 후 데이터소스를 조회합니다.
4     if (self.searchCondition.validate({ tooltip: true, message: '필수 항목을 \n// TODO: 저장여부를 확인한 후 데이터소스를 조회합니다.' },
5         // TODO: 저장여부를 확인한 후 데이터소스를 조회합니다.
6         if (self.dataSource1.getDirtyDataCount() != 0) {
7             dews.confirm('저장하지 않은 데이터가 있습니다.\n조회를 계속하시겠습니까?'),
8             self.dataSource1.read();
9         }).no(function () {
10             dews.ui.snackbar.warning('취소되었습니다.');
11             return false;
12         });
13     } else {
14         self.dataSource1.read();
15     }
16 }
17 }
```

<그림 4-21. 필수항목체크와 저장여부 메세지를 같이 사용하는 경우 처리 화면>

만약 조회되는 데이터가 많아 조회 시간이 오래 걸리는 경우 조회 중이라는 로딩 화면을 보여줄 경우 dews.ui.loading.show 함수를 사용하여 구현한다. 이후 <그림 4-22>처럼 그리드나 카드리스트 등 데이터 컨트롤의 데이터가 조회 된 이후 또는 데이터가 컨트롤에 뿌려진 이후에 실행되는 이벤트인 dataBound에서 dews.ui.loading.hide() 함수를 사용하여 로딩 화면을 숨긴다.

The screenshot shows a code editor with a tab labeled "grid1 [Grid]" and an event name "dataBound". The code is as follows:

```
1 function(e) {
2     //실행되고 있는 로딩을 닫음
3     dews.ui.loading.hide();
4 }
```

<그림 4-22. 데이터 컨트롤의 dataBound 이벤트>

2) 추가 스크립트 예제

```
add [MainButton] click
1 function() {
2     // TODO: 그리드에 데이터를 추가합니다.
3     self.grid1.addRow();
4 }
```

<그림 4-23. 추가 스크립트 예제 화면>

<그림 4-23>는 추가 버튼을 눌렀을 때, 그리드 행이 추가되는 스크립트이다. script 가장 앞부분에 자동으로 선언되어 있는 self라는 전역변수를 통해 grid1 컨트롤의 addRow() 함수를 호출한다. 행 추가 시에 기본적으로 맨 아래에 삽입된다.

추가한 후에 초기값이 필요한 경우 그리드의 경우 setCellValue() 함수를 사용하여 초기값을 설정하거나 그리드 행이 추가된 이후에 발생하는 이벤트인 rowAdd에 구현할 수 있다.

```
add [MainButton] click
334 function() {
335     self ccpvGrid.addRow();
336     var ccpvData = self ccpvGrid.dataItems(self ccpvGrid.select());
337     self ccpvGrid.setCellValue(self ccpvGrid.select(), "VER_CD", ccpvData.VER_CD);
338     self ccpvGrid.grid.setCellValue(self ccpvGrid.select(), "CNFM_YN", "N");
339 }
<메인버튼 add>

scnGrid [Grid] rowAdd
334 function(e) {
335     var ccpvData = self ccpvGrid.dataItems(self ccpvGrid.select());
336     e.grid.setCellValue(e.row.index, "VER_CD", ccpvData.VER_CD);
337     e.grid.setCellValue(e.row.index, "CNFM_YN", "N");
338 }
<그리드 rowAdd>
```

<그림 4-24. 초기값 설정 방법 화면>

그리드 이벤트의 경우 매개변수 e를 통해서 이벤트를 발생시킨 객체에 접근이 가능하므로 e.grid나 e.row를 사용하여 표준화된 코드로 작성할 수 있는 이점이 있다.

3) 삭제 스크립트 예제

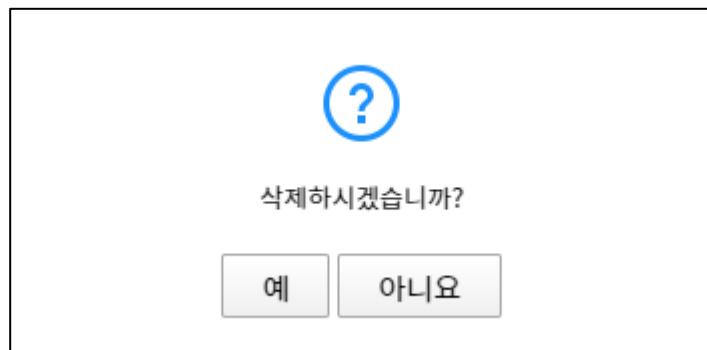
The screenshot shows a script editor window with the title bar 'delete [MainButton]' and a dropdown menu 'click'. The code is as follows:

```
1 function() {
2     // TODO: 삭제시 강제 팝업에 대한 알림창을 끕니다.
3     // self.mainButtons.delete.useDefaultConfirm = false;
4
5     // TODO: 그리드에서 체크된 항목을 삭제합니다.
6     // if (self.grid1.getCheckedIndex().length == 0) {
7     //     dews.alert('선택된 데이터가 없습니다.', 'warning');
8     //     return false;
9     // }
10    // self.grid1.removeRow(self.grid1.getCheckedIndex());
11
12    // TODO: 그리드에 선택된 항목을 삭제합니다.
13    self.grid1.removeRow(self.grid1.select());
14 }
```

<그림 4-25. 삭제 스크립트 예제 화면>

<그림 4-25>은 삭제 버튼을 눌렀을 때, 그리드의 선택된 행이 삭제되는 스크립트이다. 삭제 가이드 스크립트는 3개 부분으로 되어 있다..

첫번째는 삭제 여부 메세지를 닫는다. 삭제를 할 경우 <그림 4-26> 와 같이 삭제하시겠습니까?라는 메세지가 출력되는데 이 메세지를 닫는다. 삭제 스크립트 안에 있을 경우 삭제 시 한번 출력되고 다시 삭제 버튼을 눌렀을 때는 출력되지 않는다. 만약 삭제 버튼을 처음 눌렀을 때 메세지 출력이 필요없는 경우 Canvas의 preReady 이벤트에 알림창 스크립트를 작성한다.



<그림 4-26. 삭제 메세지 박스>

두번째는 그리드에 체크 박스를 표시할 경우 체크한 항목을 삭제한다. 세번째는 선택한 항목을 삭제한다. 필요한 경우 각각 주석을 해제하고 사용한다.

4) 저장 스크립트 예제

The screenshot shows a script editor window with the title bar "save [MainButton]". Below the title bar are two buttons: a dropdown arrow labeled "click" and a "click" button itself. The main area contains the following JavaScript code:

```
1 function() {
2     // TODO: 데이터소스를 저장합니다.
3     if (self.dataSource1.getDirtyDataCount() == 0) {
4         dews.alert('변경된 데이터가 없습니다.');
5         return false;
6     }
7
8     if (self.grid1.validate().result == false) {
9         dews.alert('필수 값이 입력되지 않은 항목이 있습니다.');
10        return false;
11    }
12
13 // 서비스 파라미터명과 데이터소스명이 같은 경우
14 dews.confirm('저장하시겠습니까?', 'ico2').yes(function () {
15     self.dataSource1.batchSave(dews.url.getApiUrl("BP", "ManagePlanCCFSer
16     .done(function (data) {
17         dews.ui.snackbar.ok('저장되었습니다.');
18         dews.ui.loading.show({
19             text: '조회중입니다.'
20         });
21         self.dataSource1.read();
22     }).fail(function (message) {
23         dews.ui.snackbar.warning(message);
24     }).always(function () {
25         dews.ui.loading.hide();
26     });
27 }).no(function () {
28     dews.ui.snackbar.warning('취소되었습니다.');
29 });
```

```

<그림 4-27 저장 스크립트 유효성 검사 스크립트>

<그림 4-27>은 저장 버튼을 눌렀을 때, 그리드의 변경된 사항들을 저장하는 스크립트이다.

저장 가이드 스크립트는 크게 3부분으로 되어 있다.

첫번째는 데이터 소스의 변경된 데이터가 있는지 체크를 한다. 두번째는 그리드의 필수값이 모두 입력되었는지 유효성 검사 함수(validate)를 호출하여 체크 한다. 세번째는 DEWS에서 제공하는 AJAX함수를 통해 서버에 등록된 저장 서비스 API를 호출하는 부분으로 서비스 파라미터명과 데이터소스명이 같은 경우와 다른 경우로 구분할 수 있다.

먼저 서비스 파라미터명과 데이터소스명이 같은 경우 데이터소스의 변경된 항목들을 저장하는 함수인 batchSave함수를 사용하여 저장한다. <그림 4-28>의 경우 데이터소스명은 ds\_emp이며 <그림 4-29>를 보면 back-end 서버 저장 서비스 파라미터도 ds\_emp이므로 이름이 서로 같은 경우이다.

save [DesignerMainButton] click

```

15 function() {
16 // 서비스 파라미터명과 데이터소스명이 같은 경우
17 dews.confirm('저장하시겠습니까?', 'ico2').yes(function () {
18 self.ds_emp.batchSave(dews.url.getApiUrl('HR', 'CompetencyManagement')
19 .done(function (data) {
20 dews.ui.snackbar.ok('저장되었습니다.');
21 dews.ui.loading.show({
22 text: '조회중입니다.'
23 });
24 self.ds_emp.read();
25 }).fail(function (message) {
26 dews.ui.snackbar.warning(message);
27 }).always(function () {
28 dews.ui.loading.hide();
29 });
30 }).no(function () {
31 dews.ui.snackbar.warning('취소되었습니다.');
32 });
33
34
35
36

```

<그림 4-28. 서비스 파라미터와 데이터소스명이 같은 경우>

```

@DzApi(url="/emp_save1", desc="사원 저장", httpMethod=DzRequestMethod.POST)
public void emp_save1(
 @DzParam(key = "ds_emp", desc = "데이터소스 ", paramType = DzParamType.Body)

 DbTransaction trans = null;

 try{
 trans = this.beginTransaction();

```

<그림 4-29. back-end 서버 서비스 저장 부분(이클립스)>

save [DesignerMainButton] click

```

15 function() {
16 // 서비스 파라미터명과 데이터소스가 다른 경우
17 if (self.dataSource1.getDirtyDataCount() > 0){
18 dews.confirm("수정된 데이터를 저장하시겠습니까?", "question");
19 dews.api.post(dews.url.getApiUrl('HR', 'CompetencyList'), {
20 async: false,
21 data: {
22 saveData: JSON.stringify(self.dataSource1.getDirtyData());
23 }
24 }).done(function (data) {
25 dews.ui.snackbar.ok('자료가 정상적으로 저장되었습니다.');
26 self.dataSource1.read();
27 }).fail(function () {
28 dews.error('저장이 실패하였습니다.', 'icon');
29 });
30 });
31 }
32 else{
33 dews.ui.snackbar.info('저장할 데이터가 없습니다.');
34 }

```

<그림 4-30. 서비스 파라미터와 데이터소스명이 다른 경우>

<그림 4-30>의 경우 데이터소스명은 dataSource1이며 <그림 4-31>에서 서비스 파라미터는 saveData이므로 서로 다른 경우이다. 이런 경우 자바스크립트 객체를 이름과 값을 가지는JSON 형식의 문자열로 변환해주는 함수인 JSON.stringify()를 사용하여 서비스 파라미터(saveData)로 데이터를 전송하여 저장한다.

```

@DzApi(url="/gldsf00200_save", desc="현금흐름표계정속성등록_저장", httpMethod=DzRequestMethod.POST)
public boolean gldsf00200_save(
 @DzParam(key = "saveData", desc = "그리드 데이터", paramType = DzParamType.Body) DzGridData saveData)
 throws Exception{
 DbTransaction trans = null;
 try{

```

<그림 4-31. 서비스의 저장 부분>

## 5) 인쇄 스크립트 예제

```
476 function() {
477
478 /* 파라미터 저장 요청 */
479 dews.api.post(dews.url.getApiUrl("CM", "printService", "setPrintPar
480 async: false,
481 data: {
482 reportCode: reportId,
483 items: JSON.stringify(items)
484 }
485 }).done(function (data){
486 /* 파라미터 키 리턴 */
487 var parameterKey = data;
488 console.log("data = ", data);
489 if (parameterKey != "" && parameterKey != null){
490 /* 인증토큰 */
491 var authToken = JSON.parse(dewself.token).access_token;
492 /* 출력물 정보 */
493 var reportInfoUrl = window.location.protocol + "//" + window.lo
494 console.log("reportInfoUrl = ", reportInfoUrl);
495 /* DRWebViewer 호출 */
496 location.href = 'GERP://" + authToken + "' + reportId +
497 console.log("location.href = ", location.href);
498 }
499 });
500 }
```

<그림 4-32. 인쇄 스크립트 예제 화면>

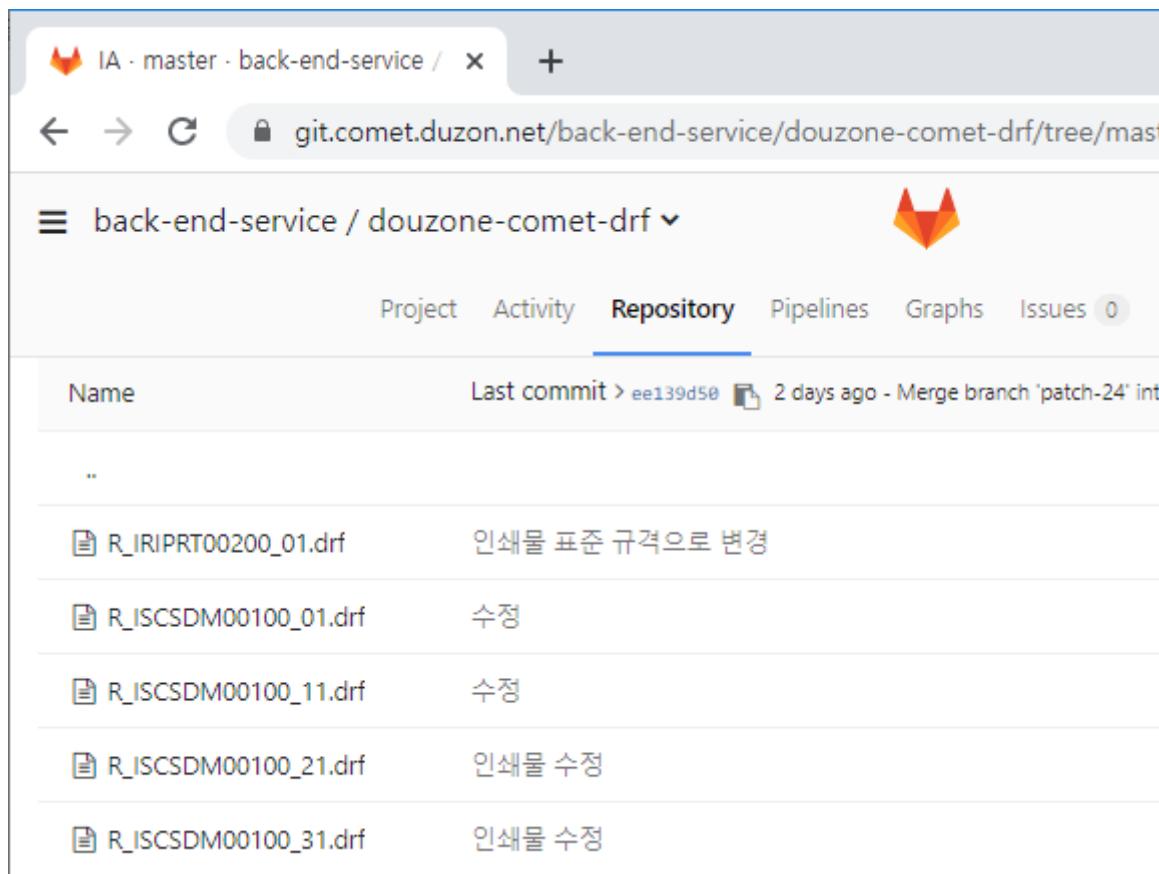
<그림 4-32>은 인쇄 버튼을 눌렀을 때, 인쇄하는 스크립트 예제이다.

인쇄를 하기 위해선 ERP 10 전용 인쇄 모듈의 설치가 기본적으로 되어 있어야 한다. 데이터 조회 후 인쇄를 눌렀을 때 ERP 10 전용 인쇄 모듈인 G\_ERP\_PRINT가 실행되면서 인쇄 양식이 없을 경우 자동으로 기본 형태의 양식으로 출력되고 양식이 있을 경우 지정한 양식으로 출력된다.

인쇄 스크립트를 작성하기 전에 인쇄 툴인 DRDesigner에서 만든 인쇄 양식인 DRF 파일은 <그림 4-33>와 같이 git 서버의 douzone-comet-drf 프로젝트에 등록을 하며, 백엔드 인쇄 서비스 API는 ERP 10 통합리포트관리 메뉴에서 등록을 한다.

통합리포트관리 메뉴를 실행하면 <그림 4-34>와 같이 왼쪽 카드리스트에는 메뉴 제목과 리포트 코드가 있고 오른쪽 그리드에는 출력물 리스트로 출력물ID, 출력물명, 서비스 URL이 있다.

출력물 ID는 리포트코드에 인덱스를 추가해서 작성을 하고, 출력물 ID는 DRF 파일명과 같고, 서비스 URL은 back-end 인쇄 API URL를 입력한다.



<그림 4-33. git 서버의 douzone-comet-drf 프로젝트 화면>

The screenshot shows the '통합리포트관리' (Unified Report Management) interface. The top navigation bar includes '메뉴 검색', '검색', '통합리포트관리', and other icons. The main area has two sections: '모듈' (Module) and '코드' (Code). The '코드' section lists report modules: '고정자산총괄현황' (R\_AACASR00100\_0), '고정자산대장' (R\_AACASR00200\_0), '고정자산명세서' (R\_AACASR00300\_0), '자산변동현황' (R\_AACASR00400\_0), and '감가상각비현황' (R\_AACASR00500\_0). To the right, a '표준 출력물' (Standard Output Document) table is displayed, showing a single row for '고정자산총괄현황' with ID 'R\_AACASR00100\_01'. The table columns are: No, 출력물ID, 출력물명, 출력물명(영문), 서식뱅크ID, and 서비스 URL.

| No | 출력물ID            | 출력물명     | 출력물명(영문) | 서식뱅크ID | 서비스 URL                      |
|----|------------------|----------|----------|--------|------------------------------|
| >  | R_AACASR00100_01 | 고정자산총괄현황 |          |        | AssetAccountingASRService/aa |

<그림 4-34. ERP 10 통합리포트관리 화면>

이제 인쇄 스크립트를 작성한다.

```
print [MainButton] click
68 function() {
69 var i;
70 var tmpCdCompany = self.user.companyCode;
71 var tmpNmCompany = self.user.companyName;
72 var tmpCdPc = ""; // 회계단위코드(멀티)
73
74 // 조회조건의 컨트롤들의 값 배열로 만들기
75 var searchParams = [
76 { name: "company_cd", value: tmpCdCompany },
77 { name: "company_nm", value: tmpNmCompany },
78 { name: "pc_cd", value: tmpCdPc },
79];
80
81 // 파라미터 ReportParam 모델 형태의 배열로 만들기
82 var items = [];
83 var reportId = "R_GLDFSM00100_0"; //Report ID
84 var objectId = "R_GLDFSM00100_01"; //Object ID
85
86 $(searchParams).each(function (idx, item) {
87 items.push({
88 RPRT_CD: reportId, OBJECT_CD: objectId, PARA_CD: item.name, PARA_T: 1
89 });
90 });
91
92 // 파라미터 저장 요청
93 dews.api.post(dews.url.getApiUrl("CM", "printService", "setPrintParam"),
94 async: false,
95 data: {
96 requestCode: reportId, items: JSON.stringify(items)
97 }
98 }).done(function (data) {
99 // 파라미터 키 리턴 ex) d3e05154-ebbd-4fbd-8f4b-f18aab2875e7
100 var parameterKey = data;
101
102 // 인증토큰 ex) eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb21wYW55Q29i
103 var authToken = JSON.parse(self.token).access_token;
104
105 // 인쇄정보 서비스 URL(출력물 정보) ex) https://staging.comet.duzon.ne
106 var reportInfoUrl = window.location.protocol + "//" + window.location.
107
108 // DRWebViewer 호출 (인증토큰, 리포트코드, 인쇄정보 서비스 URL, 파라미터
109 location.href = 'GERP://' + authToken + '/' + reportId + '/';
110 });
}
①
②
③
④
```

<그림 4-35. 인쇄 이벤트 스크립트 화면>

인쇄 버튼 이벤트 스크립트는 <그림 4-35>를 참고한다.

- ① 먼저 조회조건의 컨트롤들 값을 가져와서 배열 형태로 생성한다.
- ② 다음으로 ReportParam 모델 형태 배열로 만들기 위해서 통합리포트 관리의 왼쪽 카드리스트에 등록한 리포트코드와 오른쪽 그리드 출력물 ID를 저장한다.

조회조건 컨트롤 갯수 만큼 반복하면서 리포트코드와 출력물 ID를 추가해서 ReportParam 형태의 배열로 다시 생성한다.

- ③ 인쇄물 데이터 조회에 필요한 조회조건을 DataBase에 저장하기 위해 파라미터 저장 API를 호출한다.

ReportService의 setReportParam API에 리포트 코드와 ReportParam 형태의 배열을 인자로 하여 저장 API를 호출한다.

파라미터가 저장되면 UID 즉 고유식별자 형태의 키가 data로 반환된다.

- ④ 이제 ERP 10 Web 인쇄 프로그램인 DRWebViewer를 호출한다. 호출 시 필요한 파라미터는 인증토큰, 리포트코드, 인쇄정보 서비스 URL, 파라미터 키, 메뉴코드로 구성되어 있다.

인증토큰은 로그인한 사용자의 인증토큰을 받아오기 위해 self.token을 사용한다. JSON.parse() 메서드를 사용하여 JSON 문자열의 구문을 분석하고 access\_token 속성값을 사용한다.

리포트코드는 통합리포트관리의 리포트 코드이다.

인쇄 정보 서비스 URL은 브라우저 객체의 최상위 객체인 window와 사용자 브라우저와 관련된 속성과 메서드를 제공하는 location 정보를 가져온다.

파라미터 키는 앞에서 저장 API에서 반환받은 발급 받은 파라미터 키이며, 메뉴 코드는 페이지 이름 즉, 메뉴 ID이다.

## [제5장] QueryBuilder

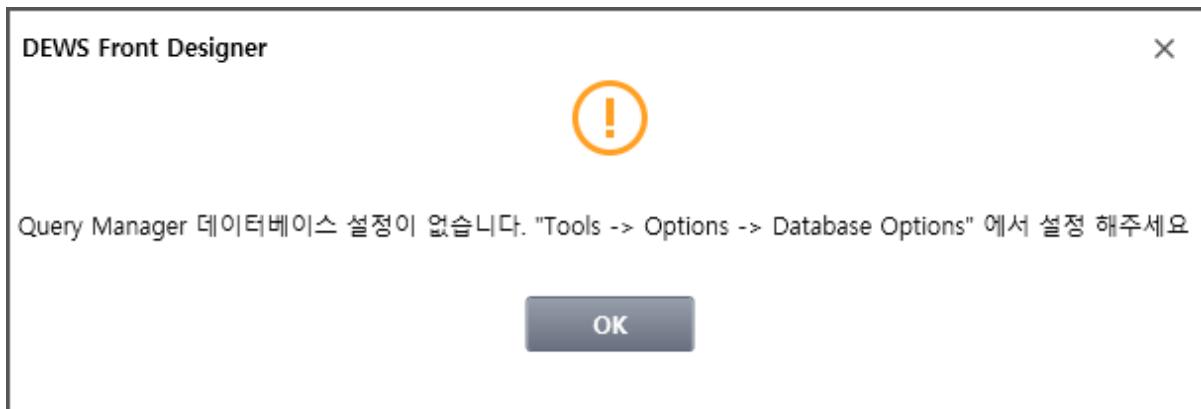
QueryBuilder Setup

QueryBuilder 실행

## 1. QueryBuilder Setup

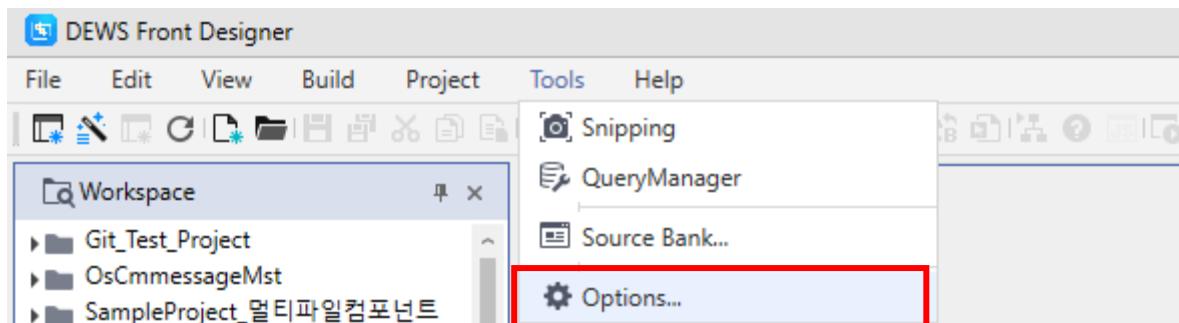
DEWS Front Designer에서 쿼리 빌더를 제공하여 데이터베이스 조회 전용으로 사용할 수 있다. ERD 형태의 다이어그램을 이용하여 쿼리를 직접 만드거나. 쿼리 추천기를 이용하여 기존 만들어진 쿼리를 활용할 수 있다.

쿼리 빌더를 사용하기 위해서는 먼저 툴의 데이터베이스 옵션에서 설정을 해야 한다. 만약 쿼리 빌더를 설정하지 않고 실행하는 경우 설정 요청 메세지가 나타난다.



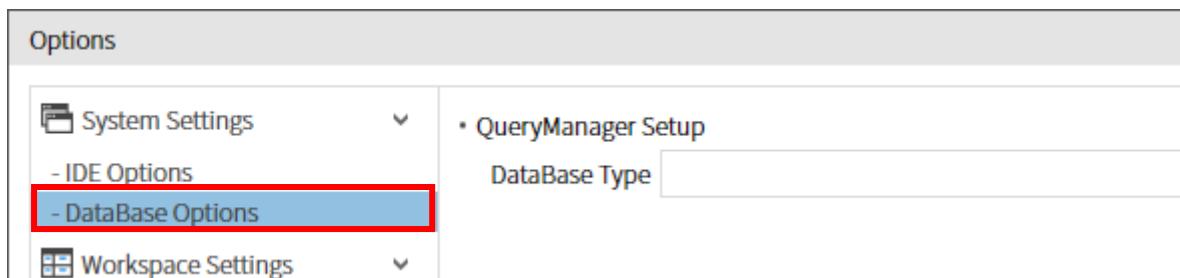
<그림 5-1. 쿼리빌더 설정 요청 화면>

Tools의 Options을 실행한다.



<그림 5-2. Tools의 옵션 실행>

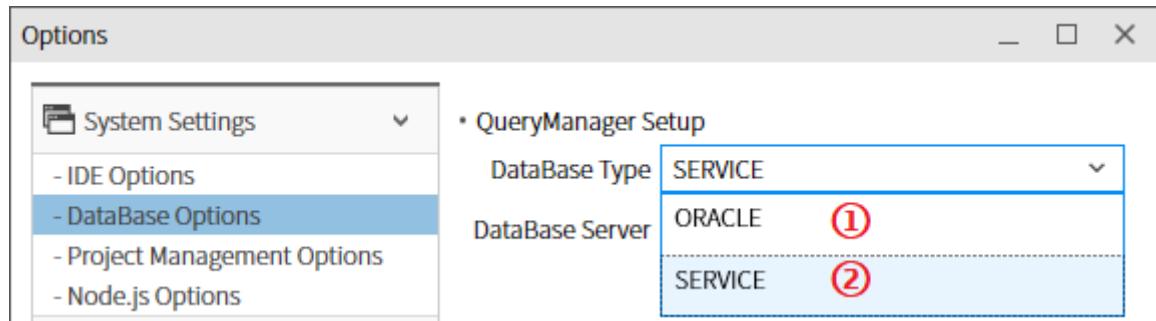
System Settings > DataBase Options을 선택한다.



<그림 5-3. DataBase Options 선택>

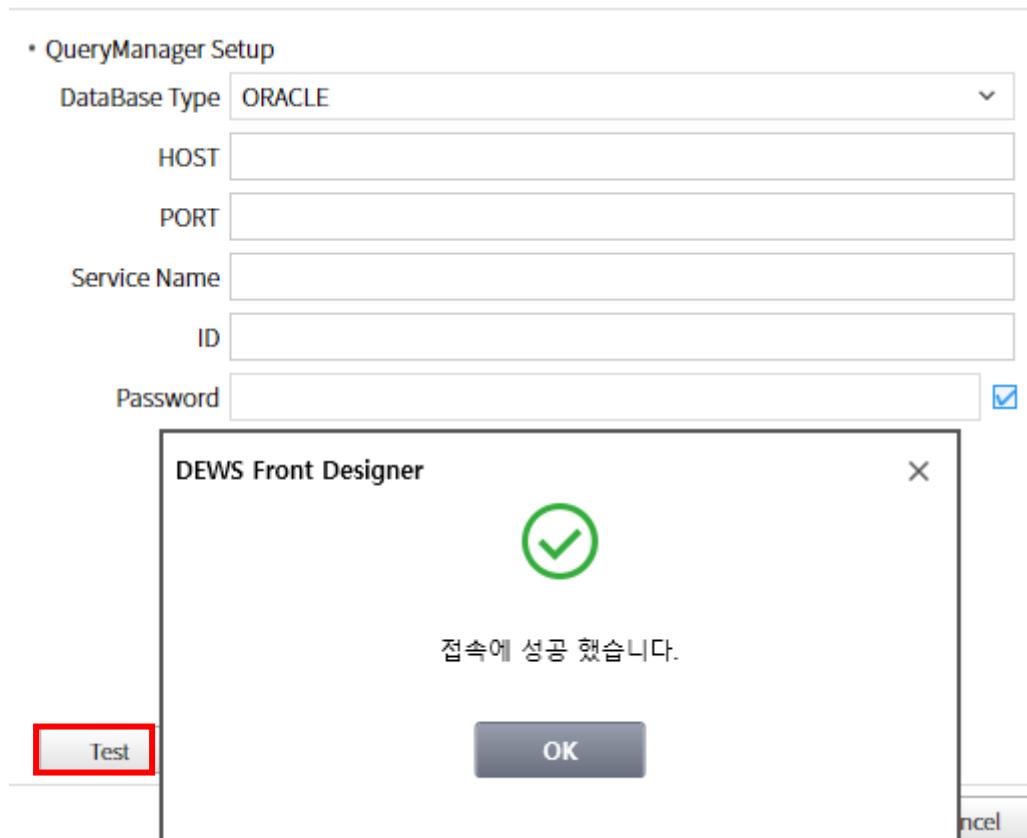
오른쪽 QueryManager Setup 의 DataBase Type 을 선택한다

- ① 직접 DB 에 접속할 경우와 ② Server Management 에서 설정한 서버에 연결된 DB 를 선택할 수 있다.



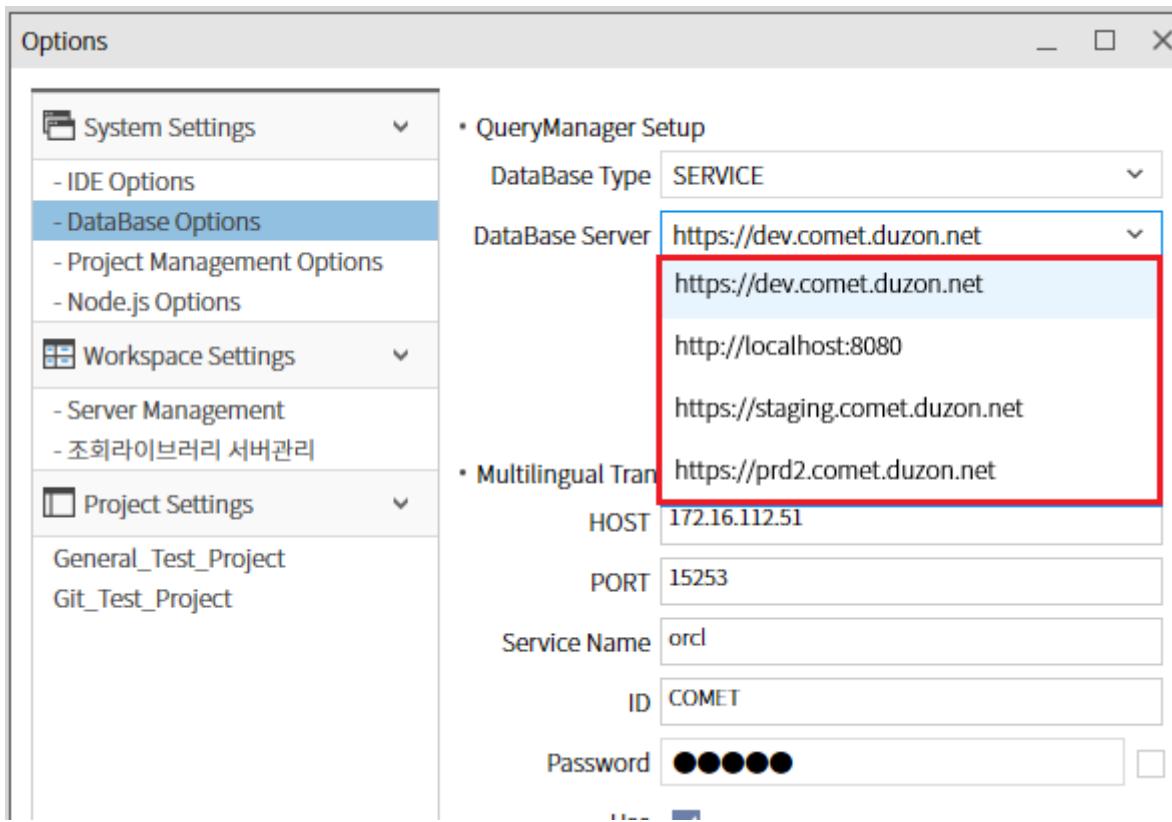
<그림 5-4. DataBase Type 선택>

직접 DB 에 접속할 경우 접속 정보를 모두 입력하고 Test 버튼을 누르면 접속 성공 여부를 확인할 수 있다. 현재는 오라클 DB 만 지원하며 다른 DB 는 추후 지원 예정이다.



<그림 5-5. 직접 DB 선택>

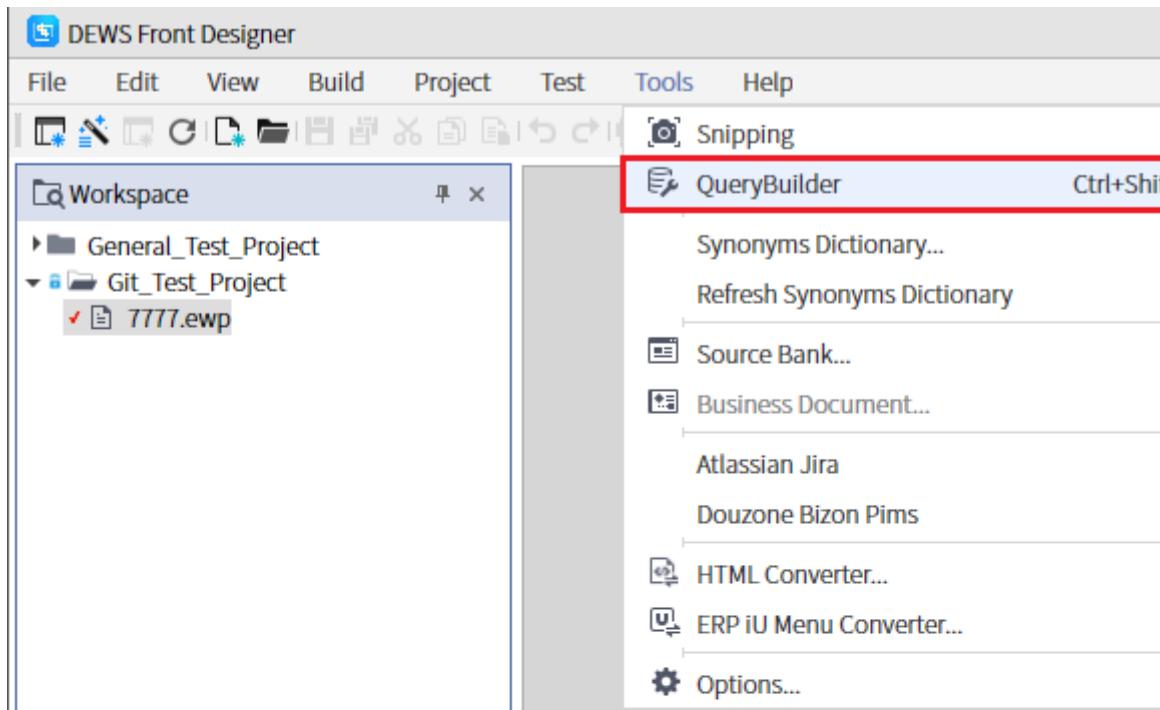
서비스를 선택할 경우 Workspace Settings > Server Management 에 있는 서버 정보들의 리스트를 보여준다.



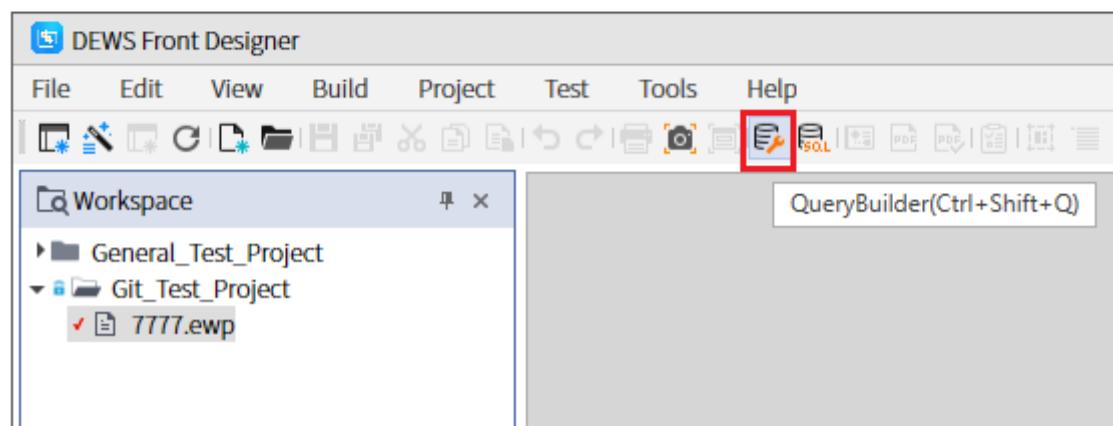
일반적으로 직접 DB 연결하는 방법보다 Server Management 에서 설정한 서버에 연결된 DB 로 사용한다.

## 2. QueryBuilder 실행

메뉴 Tools의 QueryBuilder나 툴바의 을 눌러 쿼리 빌더를 실행한다.

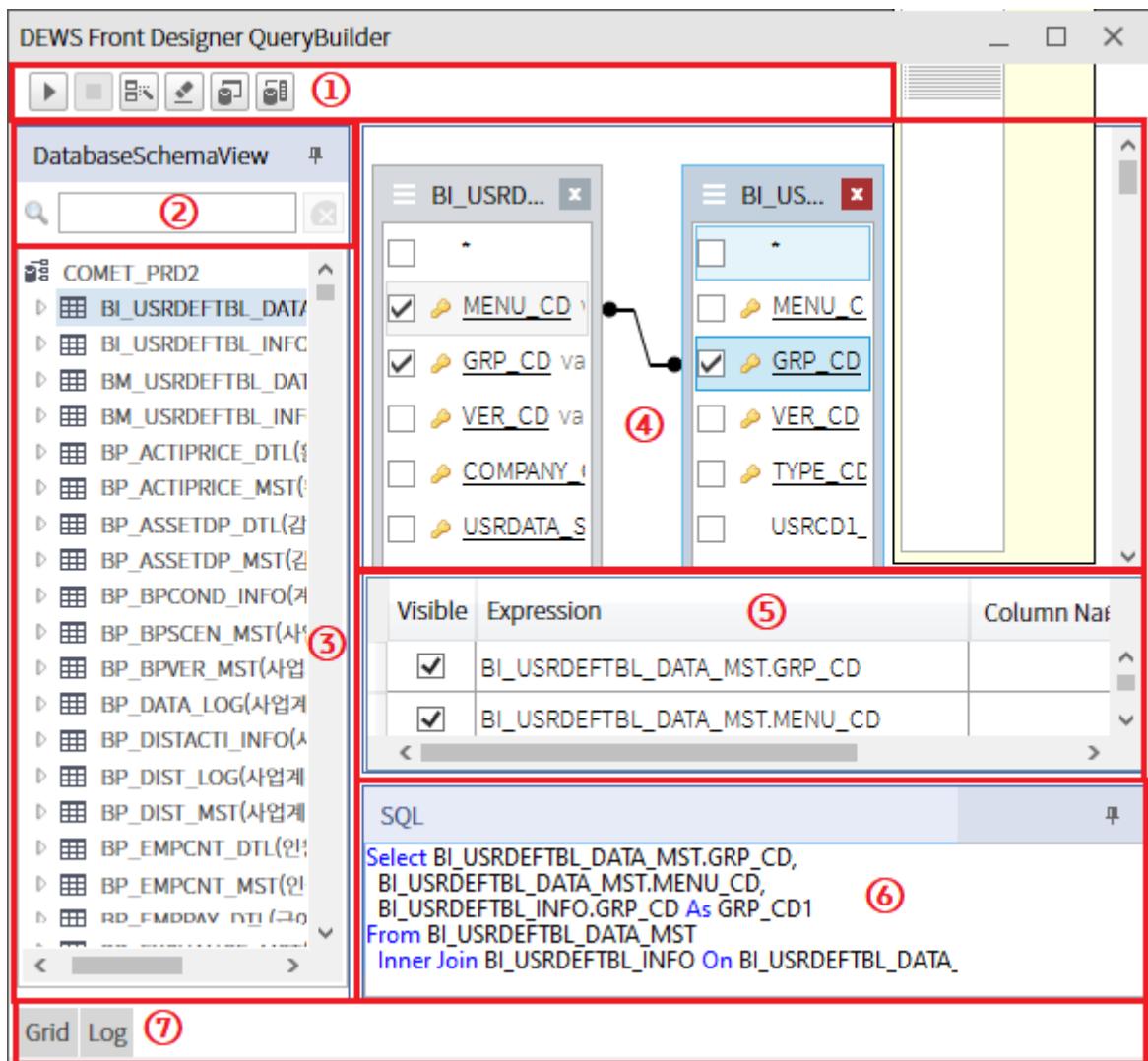


OR



<그림 5-7. 쿼리매니저 실행>

아래 <그림 5-8>은 쿼리 빌더를 실행한 화면이다.



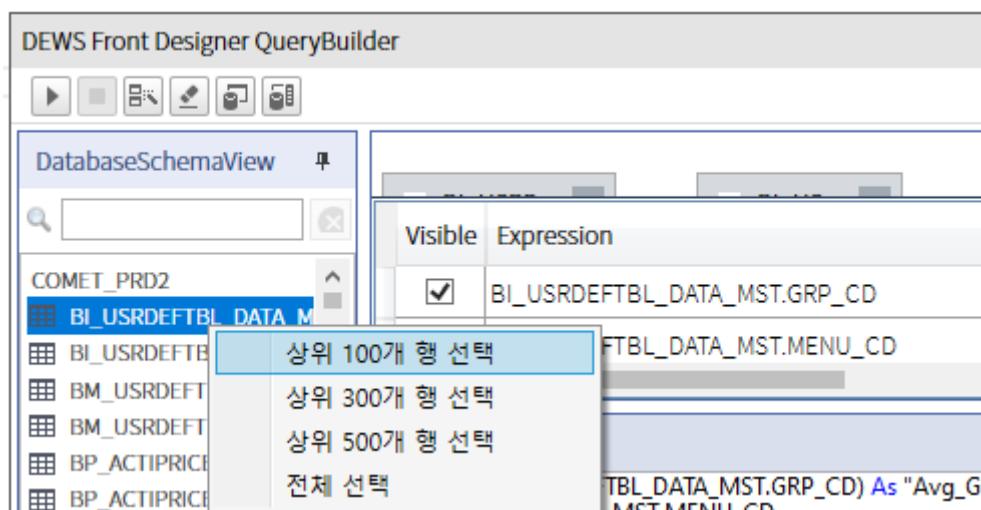
<그림 5-8. 쿼리빌더 화면>

① 메뉴 툴바 : 6개 툴바로 이루어짐

| 툴바                            | 설명                                     |
|-------------------------------|----------------------------------------|
| Run                           | 쿼리를 실행 (F5)                            |
| Stop                          | 쿼리 실행 중지                               |
| Editor Synchronization        | 쿼리창의 쿼리를 디아이그램 화면에 동기화 (Ctrl+Shift+F5) |
| Diagram Clear                 | 다이어그램 초기화 (Ctrl+Delete)                |
| Query Recommender             | 쿼리 추천기 실행 (Ctrl+F6)                    |
| Diagram Parameter Information | 쿼리 스키마 정보 확인 (Ctrl+Shift+i)            |

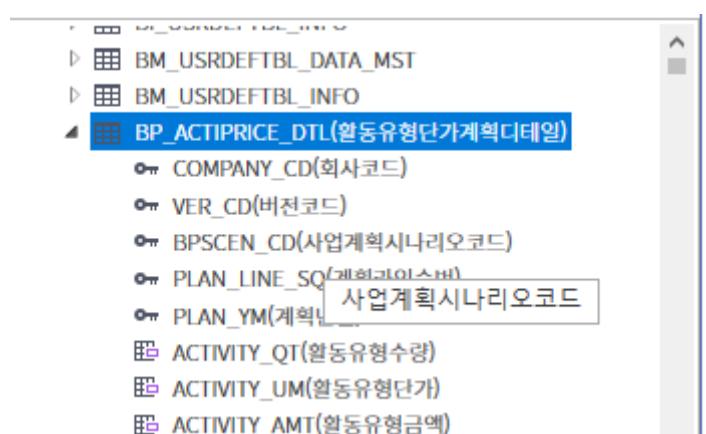
<표 5-1. 메뉴 툴바에 대한 설명>

- ② 테이블 검색 : 테이블 검색 입력
  - ③ 테이블 리스트 : 모든 테이블 리스트 표시
  - ④ Diagram 및 미니맵 표시 : 테이블을 끌어다 놓으면 ERD 형태의 다이어그램으로 표시되고  
테이블 관계선 설정은 컬럼을 클릭 다른 테이블 컬럼에 끌어다  
놓음
  - ⑤ 컬럼 상세 설정 : 표시될 컬럼에 대한 쉬운 설정. 쿼리 에디터에 바로 반영
  - ⑥ 쿼리 에디터 : 다이어그램에서 체크하거나 상세 설정시 쿼리가 자동 작성되며 직접 편집
  - ⑦ 쿼리 결과 : 쿼리문을 실행한 결과(Grid) 및 로그(Log) 확인. 화면이 자동 숨김으로 표시
- ③ 테이블 리스트 화면에서 우클릭 했을 때 자동 조회(Select) 및 결과행 수를 선택할 수 있다



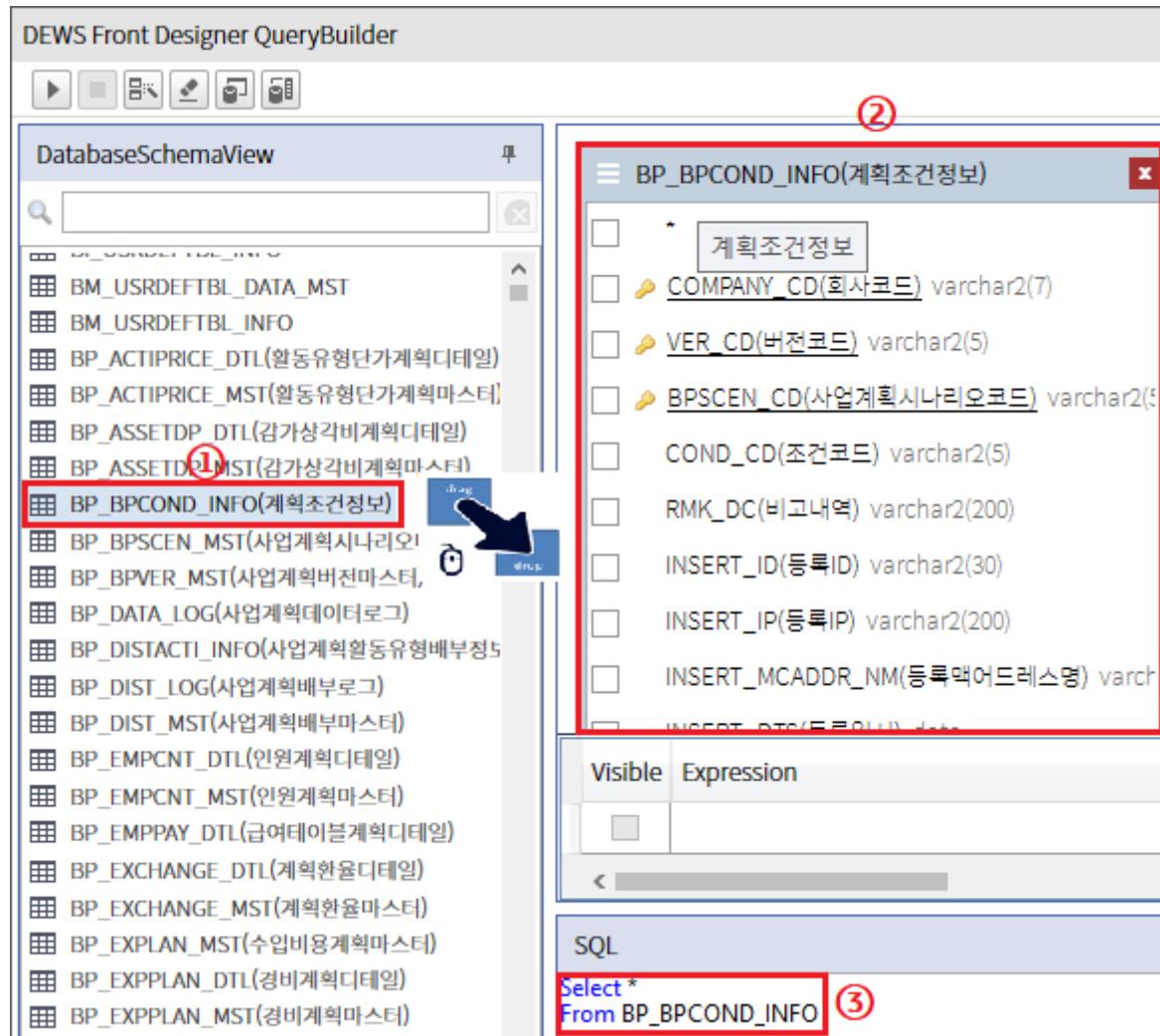
<그림 5-9. 테이블 우클릭 메뉴>

또한 테이블을 더블클릭했을 때는 오른쪽으로 ERD 형태의 다이어그램으로 이동하고 테이블을 펼쳤을 때는 컬럼 상세 정보를 확인 할 수 있다.



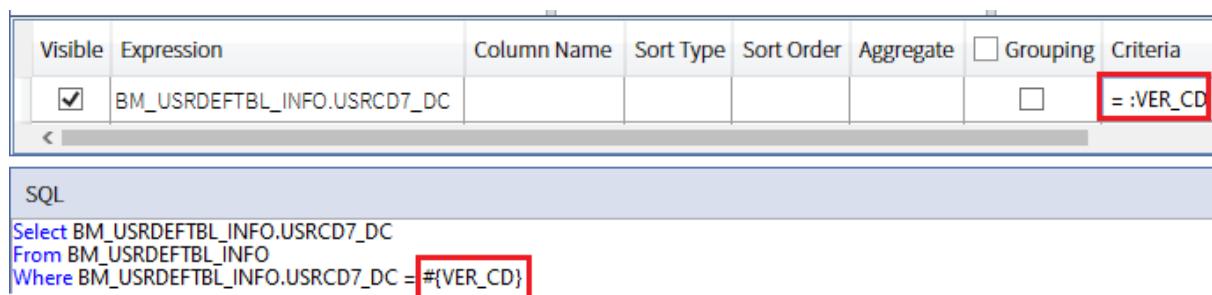
<그림 5-10. 테이블을 펼쳤을 때 화면>

<그림 5-11> 과 같이 ① 테이블 선택 후 ② 다이어그램 화면에 끌어다 놓으면(드래그&드롭)  
 ③ 모든 컬럼을 조회하는 쿼리문이 자동으로 만들어 진다.



<그림 5-11. 테이블 드래그 & 드롭>

Where 조건의 경우 아래 쿼리 에디터에서 #{파라미터명} 처럼 직접 작성할 수도 있고, Diagram에서 조회할 컬럼을 체크하면 컬럼 상세 설정에서 추가되고 컬럼에 대한 조건을 Criteria에서 직접 선택하고 작성할 수도 있다. 쿼리 에디터에서도 자동으로 동기화된다.



<그림 5-12. 조회조건 입력 화면>

만들어진 쿼리를 실행하기 위해서는 상단 ▶ 버튼을 누르거나 F5 키를 눌러 실행한다 쿼리 결과 화면에서는 데이터에서 우클릭하면 엑셀 출력과 데이터 찾기를 할 수 있다.

A screenshot of a query result grid titled "Grid". The grid has columns: COMPANY\_CD, VER\_CD, BPSCEN\_CD, PLAN\_LINE\_SQ, and ASSET\_CD. The first row shows COMPANY\_CD as 5000 and VER\_CD as 001. A context menu is open over this cell, with the "Search Ctrl+F" option highlighted. Other options in the menu include "Excel Export.." and "Search Ctrl+F". The menu is enclosed in a red box.

| COMPANY_CD | VER_CD | BPSCEN_CD | PLAN_LINE_SQ | ASSET_CD      |
|------------|--------|-----------|--------------|---------------|
| 5000       | 001    | 02        | 26           | 201805030003  |
| 5000       | 001    | 02        | 27           | A2018060060   |
| 5000       | 001    | 02        | 28           | A2018060065   |
| 5000       | 001    | 02        | 29           | 201805030006  |
| 5000       | 001    | 02        | 30           | 2018052800001 |
| 5000       | 001    | 02        | 31           | 201805030002  |
| 5000       | 001    | 02        | 32           | C20180101     |
| 5000       | 001    | 02        | 33           | A2018060066   |

<그림 5-13. 쿼리 결과 데이터 우클릭 메뉴>

컬럼명에서 우클릭 했을 때 정렬, 컬럼 그룹, 컬럼 크기 자동 맞춤 등 많은 메뉴를 제공한다.

A screenshot of a query result grid titled "Grid". The grid has columns: ACCT\_CD. The first row shows ACCT\_CD as 1203040. A context menu is open over this cell, with the "Sort Ascending" option highlighted. Other options in the menu include "Sort Descending", "Clear Sorting", "Show Group Panel", "Show Column Chooser", "Best Fit", "Best Fit (all columns)", and "Show Search Panel". The menu is enclosed in a red box.

| ACCT_CD |
|---------|
| 1203040 |

<그림 5-14. 쿼리 결과 컬럼명 선택 후 우클릭 메뉴 화면>

## [제6장] 설계 도구

소개 및 실행

화면 구성

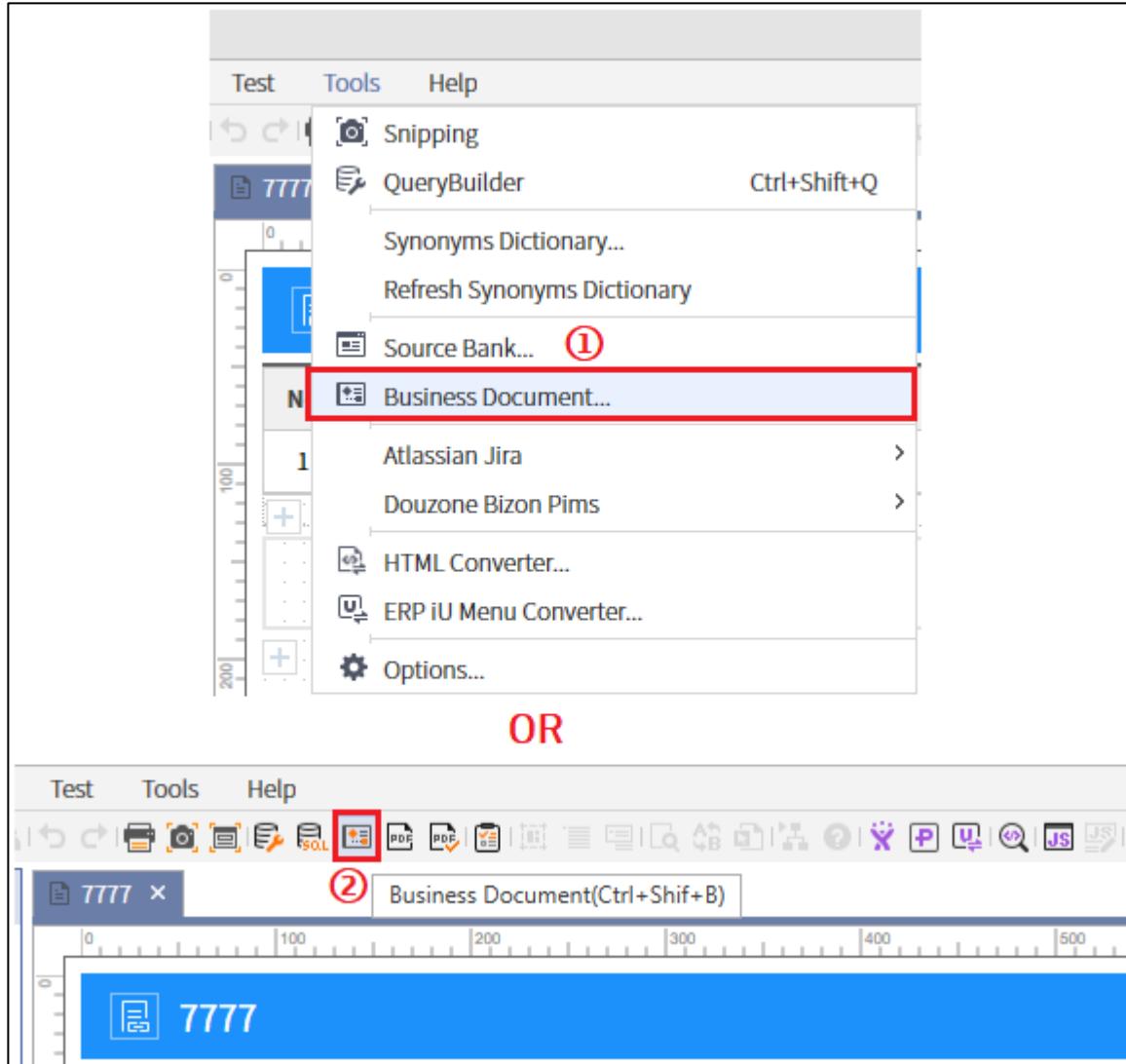
설계 문서 작성

설계 문서 파일 만들기

## 1. 소개 및 실행

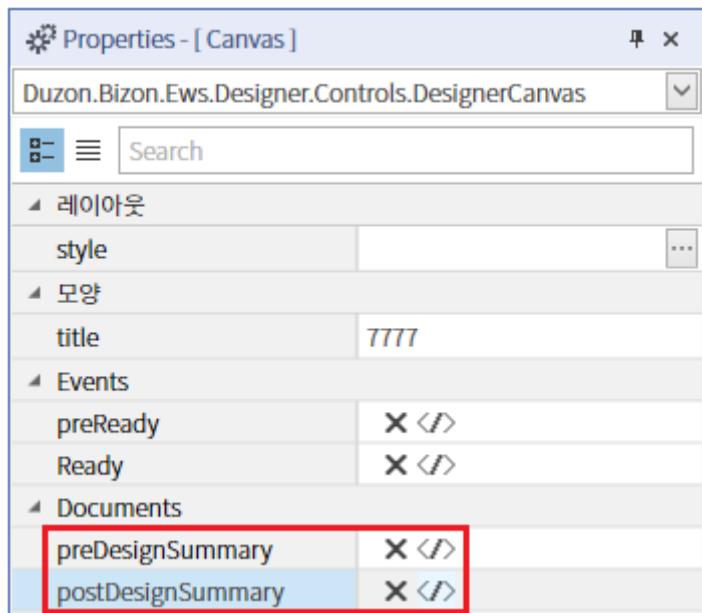
설계 도구(Business Document)는 설계 문서 작성 도구로 간단히 설계 도구라고 하며 개발할 화면을 분석하거나 기능 개요, 업무 흐름도, 테이블 정보 등의 설계 문서를 작성할 수 있는 툴이다.

설계 도구 실행 방법은 <그림 6-1>와 같이 ① 메뉴 Tools > Business Document (Ctrl+Shift+B)을 실행하거나 ② 툴바에서 Business Document를 클릭해서 실행한다.



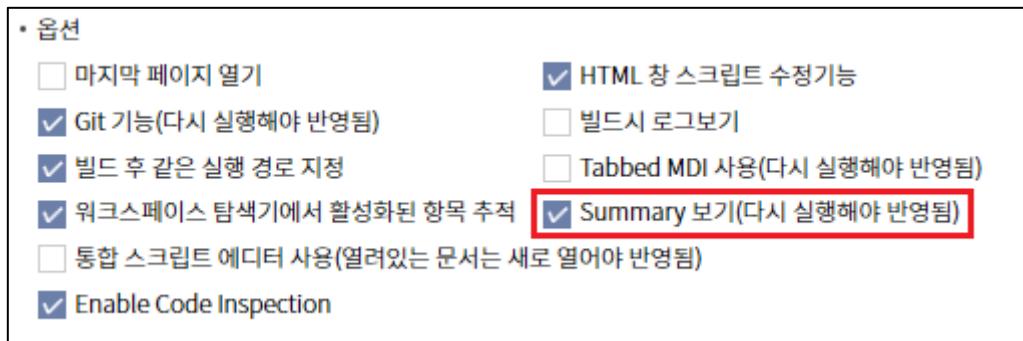
<그림 6-1 쿼리빌더 설정 요청 화면>

다른 방법으로 컨트롤을 선택하고 Properties 창 속성 중에 designSummary 스크립트 버튼을 누른다. 화면 최상위 컨트롤인 Canvas 인 경우 preDesignSummary(개요)와 postDesignSummary (맺음말) 2 개 버튼을 눌러 설계 도구를 실행할 수 있다



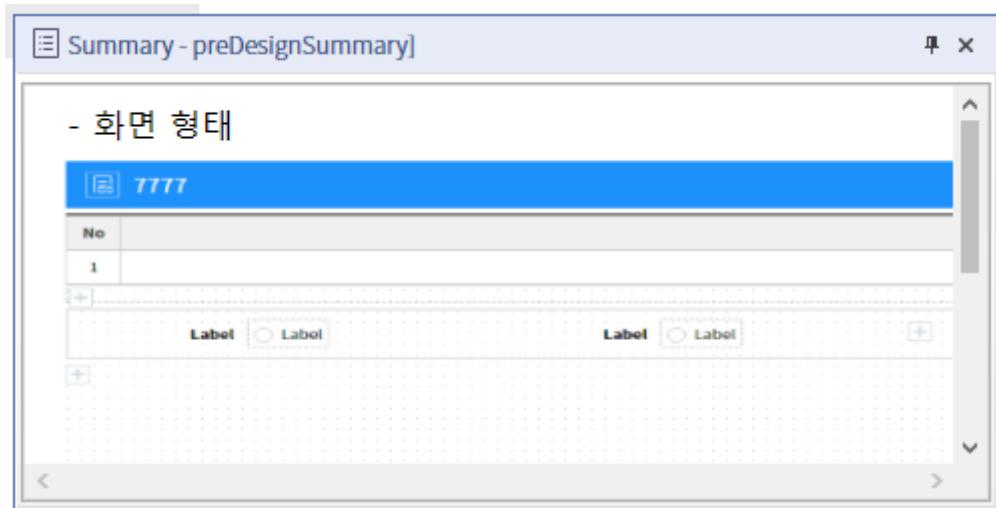
<그림 6-2. Canvas 속성 화면>

또한 설계 도구를 따로 열지 않고 개발 도구에서도 설계 문서 작성이 가능하며, 이 경우 옵션에서 Summary 보기에 체크가 되어 있어야 한다.



<그림 6-3. Tools – Options의 옵션 Summary 보기 체크 화면>

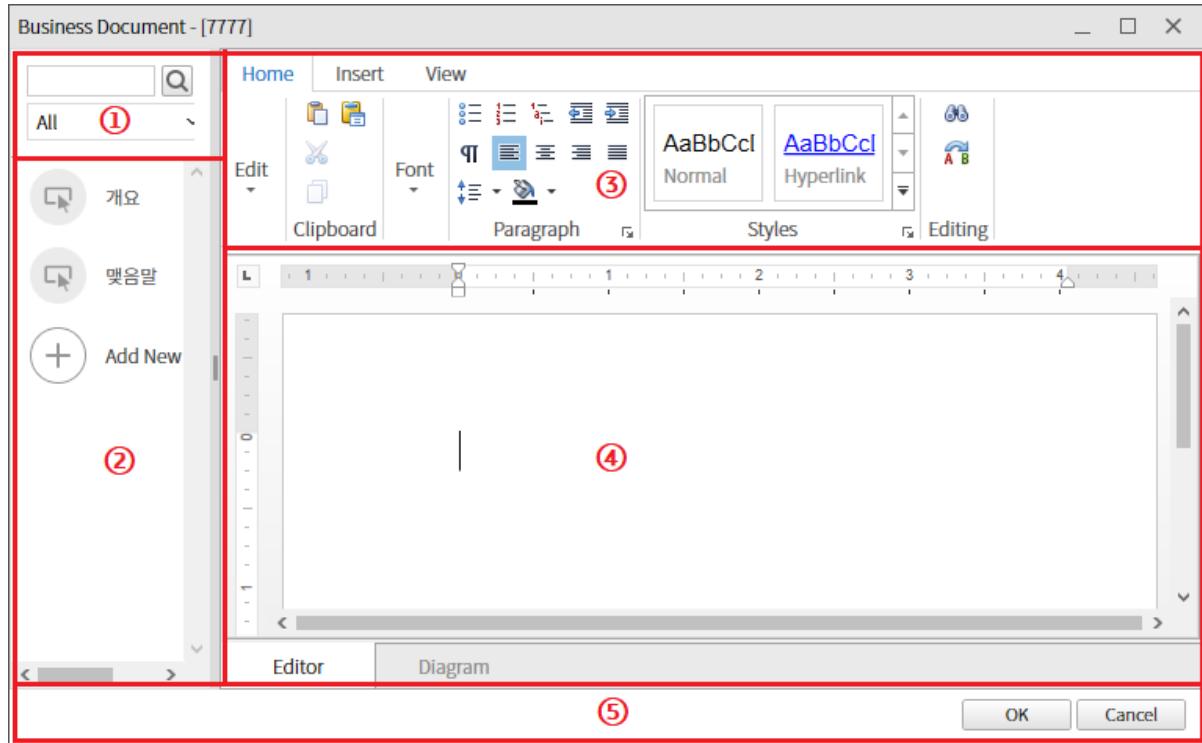
Summary 보기 를 체크한 경우 오른쪽 하단 Summary 에서 설계 문서를 확인할 수 있으며 편집도 가능하다.



<그림 6-4. 개발도구 Summary 화면>

## 2. 화면 구성

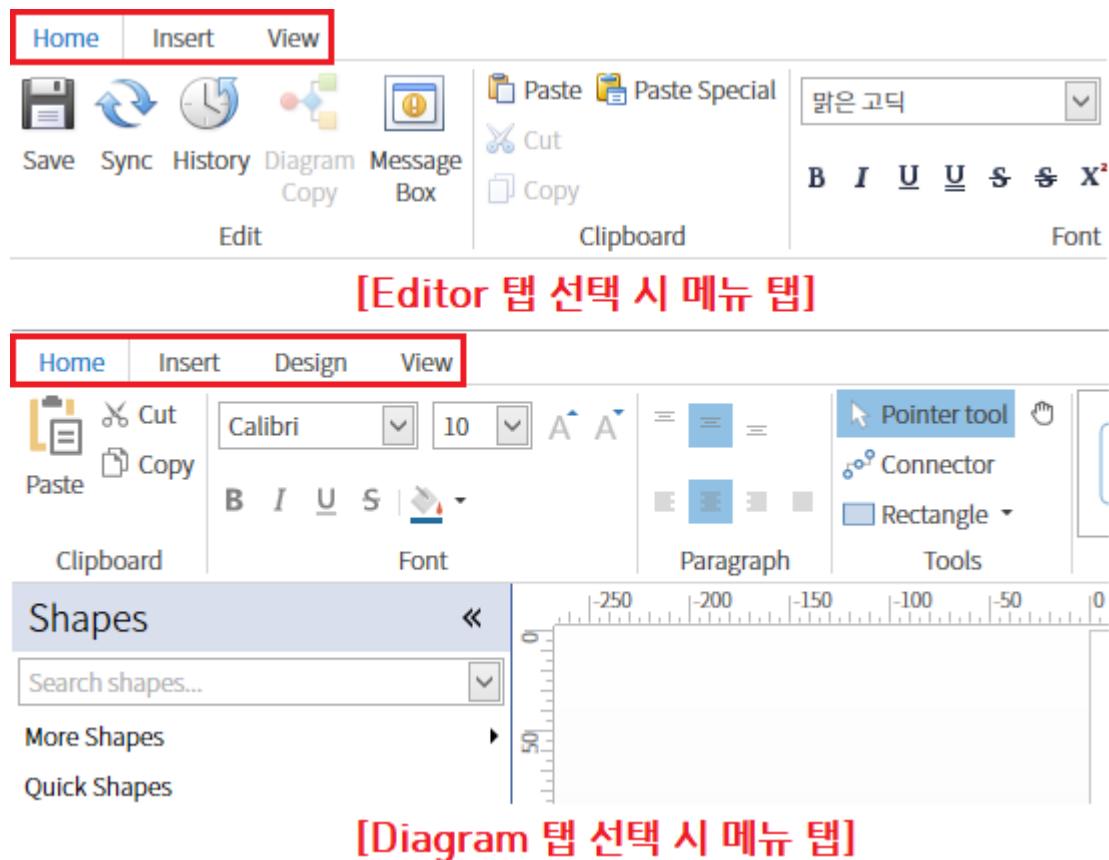
설계 도구를 실행한 화면이다. 크게 5가지 화면으로 구성되어 있다.



<그림 6-5. 설계 도구 실행 화면>

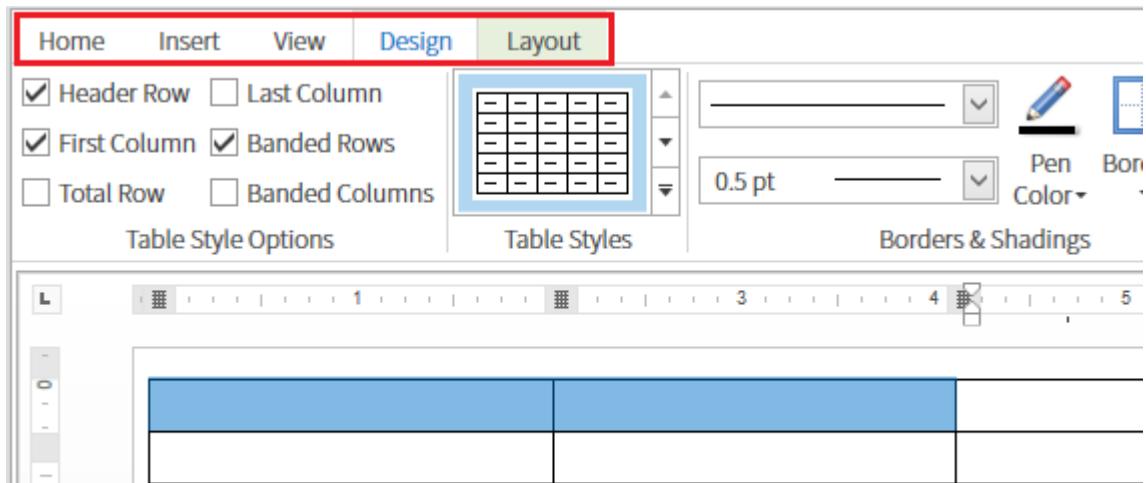
- ① 페이지 검색 : 작성된 페이지를 검색
- ② 페이지 : 설계 문서가 작성된 컴포넌트(컨트롤) 또는 사용자가 추가한 페이지 목록과 새로운 페이지 추가. 기본으로 개요와 맷음말 페이지가 있음
- ③ 메뉴 툴바 : 설계 문서 편집 메뉴 툴바
- ④ 작업 영역 : Editor 탭과 Diagram 탭으로 구분되며, 텍스트 작성 및 편집 또는 다이어그램 생성 등의 실제 문서 작성 영역
- ⑤ 확인 버튼 : 작업 내역 저장 또는 취소하고 화면 닫기

<그림 6-6> 처럼 ③ 메뉴 툴바의 메뉴 탭 구성은 아래 Editor 탭 또는 Diagram 탭을 선택했을 때와 표나 그림, 다이어그램, 컨테이너 등 선택한 개체에 따라서 메뉴 탭 구성이 다르다. Editor 탭 선택 시 3개 탭이며, Diagram 탭은 4개 탭으로 되어 있다.

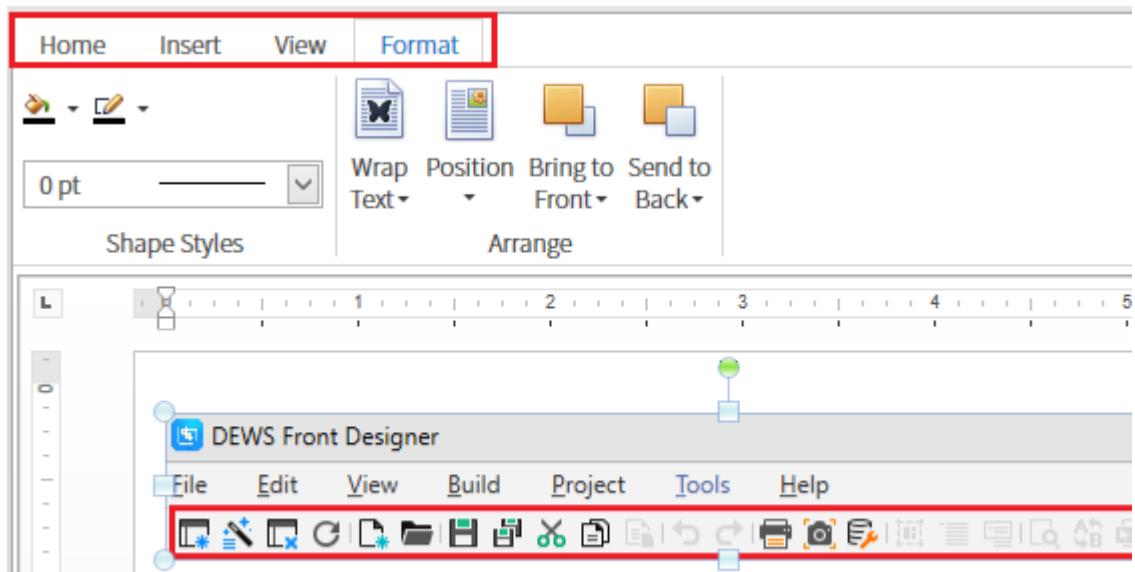


<그림 6-6. 탭 선택에 따른 메뉴 탭 구성 화면>

또한 같은 Editor나 Diagram 탭이라도 선택한 개체(테이블, 그림 선택)에 따라 메뉴 구성 탭도 다르다.



[테이블 선택 시 메뉴 탭]



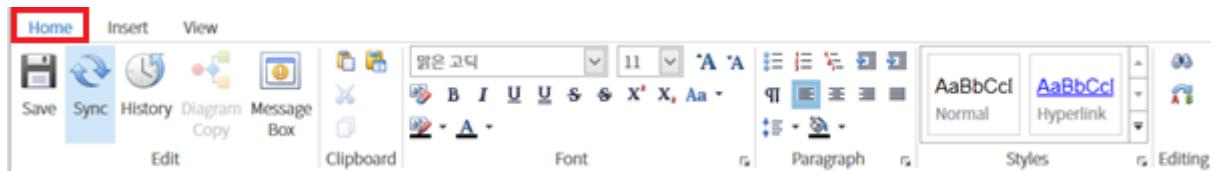
[그림 선택 시 메뉴 탭]

<그림 6-7. 선택한 개체에 따라 메뉴 탭 구성 화면>

## 1) 메뉴 툴바

### ① Editor

#### - Home 탭



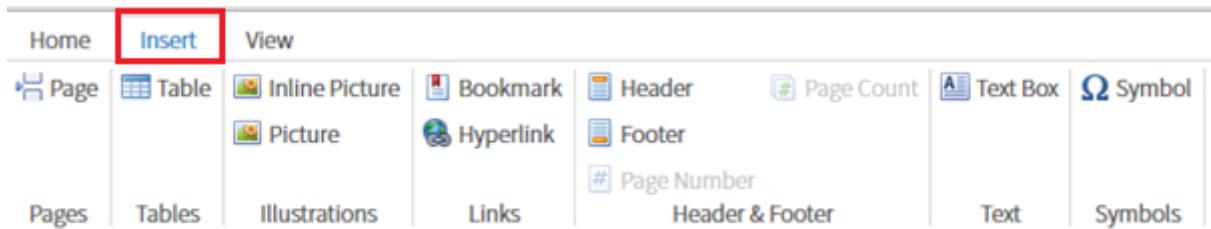
<그림 6-8. Home 화면>

Edit > Save의 버튼의 경우 팝업 화면 처럼 내용만 저장되며, 파일 저장은 반드시 개발 도구 화면에서 저장 버튼을 눌러야 저장되니 주의한다.

| 구분        | 하위 메뉴        | 설명                             |
|-----------|--------------|--------------------------------|
| Edit      | Save         | 저장 (※ 설계 문서 내용만 저장)            |
|           | Sync         | 개발 도구에서 작성한 Summary 를 가져옴      |
|           | History      | 개정 이력 작성                       |
|           | Diagram Copy | 다이어그램 탭에서 작성한 모양을 그림으로 삽입      |
|           | MessageBox   | 메시지 박스 삽입                      |
| Clipboard |              | 붙여넣기, 복사, 잘라내기                 |
| Font      |              | 글꼴 수정하거나 다양한 효과 적용             |
| Paragraph |              | 글머리 목록을 매기거나 단락의 간격 조정, 번호, 정렬 |
| Styles    |              | 텍스트 서식을 적용                     |
| Editing   |              | 텍스트 찾기 및 바꾸기                   |

<표 6-1. Home 메뉴 설명>

#### - Insert 탭



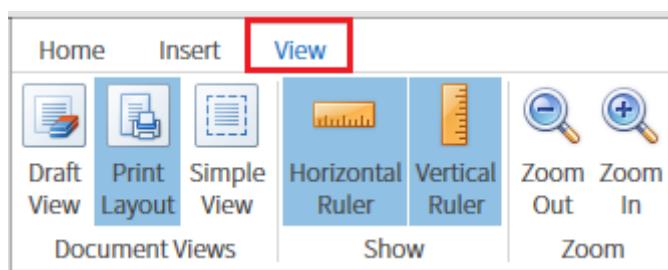
<그림 6-9. Insert 화면>

Text > Text Box는 테두리 강조할 때 사용할 수 있다.

| 구분              | 하위 메뉴 | 설명                                  |
|-----------------|-------|-------------------------------------|
| Pages           |       | 새로운 페이지 삽입                          |
| Tables          |       | 표 삽입                                |
| Illustrations   |       | 그림 삽입                               |
| Links           |       | 책갈피나 하이퍼링크 삽입                       |
| Header & Footer |       | 머리글/바닥글/페이지 번호/전체 페이지 삽입            |
| Text            |       | 텍스트 상자 삽입 ( <b>※ 테두리 강조 시에 사용</b> ) |
| Symbols         |       | 기호 삽입                               |

<표 6-2. Insert 메뉴 설명>

- View 탭

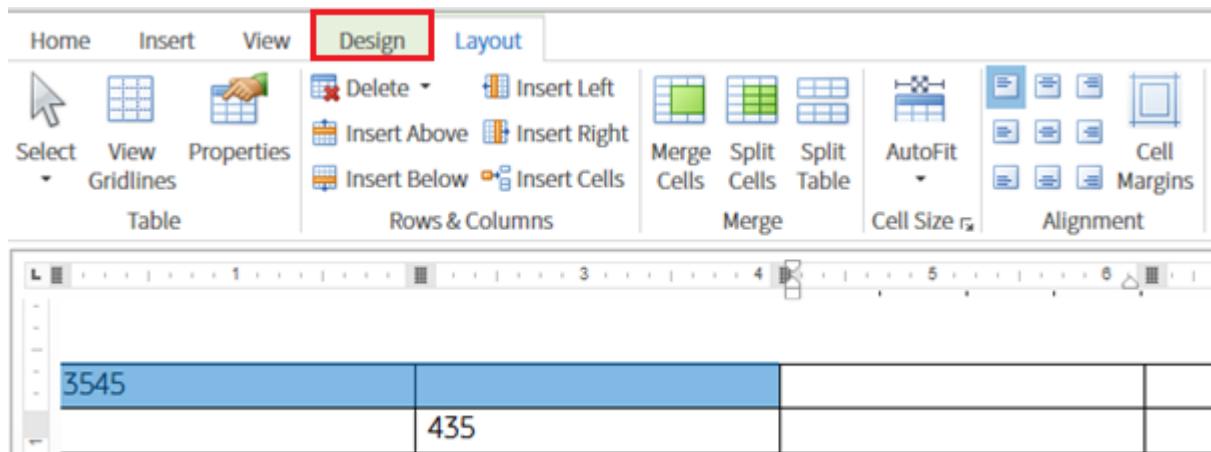


<그림 6-10. View 화면>

| 구분             | 하위 메뉴        | 설명                                        |
|----------------|--------------|-------------------------------------------|
| Document Views | Draft View   | 초안 보기. 텍스트를 빠르게 편집할 수 있으며 머리글 바닥글은 보이지 않음 |
|                | Print Layout | 인쇄 레이아웃. 인쇄될 페이지로 문서를 볼 수 있음(기본)          |
|                | Simple View  | 심플 보기. 페이지 레이아웃을 무시하여 텍스트 편집에 주의          |
| Show           |              | 수평선/수직선 자 보기                              |
| Zoom           |              | 화면 확대/축소                                  |

<표 6-3. View 메뉴 설명>

- 테이블 삽입 후 선택 했을 때 Design 탭

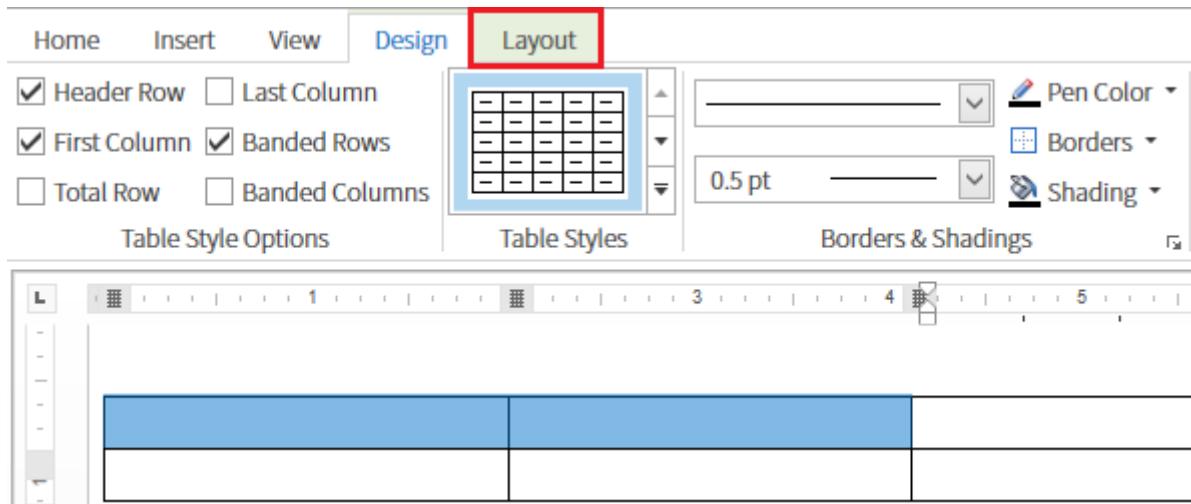


<그림 6-11. 테이블 삽입 후 선택했을 때 Design 화면>

| 구분             | 하위 메뉴          | 설명                                               |
|----------------|----------------|--------------------------------------------------|
| Table Styles   | Select         | 현재 셀, 행, 열 또는 전체 테이블을 선택.                        |
|                | View Gridlines | 테이블 내에서 눈금 선을 표시하거나 숨김                           |
|                | Properties     | 테이블 속성 대화 상자 표시                                  |
| Rows & Columns | Delete         | 행, 열, 셀 또는 전체 테이블 삭제                             |
|                | Insert Above   | 선택한 행 바로 위에 새 행 추가                               |
|                | Insert Below   | 선택한 행 바로 아래에 새 행 추가                              |
|                | Insert Left    | 선택한 열의 왼쪽에 새 열을 직접 추가.                           |
|                | Insert Right   | 선택한 열 오른쪽에 새 열을 직접 추가                            |
|                | Insert Cells   | 셀 삽입                                             |
| Merge          | Merge Cells    | 선택한 셀을 하나의 셀로 병합                                 |
|                | Split Cells    | 선택한 셀을 여러 개의 새 셀로 분할                             |
|                | Split Table    | 테이블을 두 개의 테이블로 분할 되며,<br>선택한 행이 새 테이블의 첫 번째 행이 됨 |
| Cell Size      | AutoFit        | 텍스트에 따라 열 너비의 크기 자동으로 조정                         |
| Alignment      |                | 정렬 및 여백 설정                                       |

<표 6-4. 테이블 Design 메뉴 설명>

- 테이블 삽입 후 선택 했을 때 Layout 탭

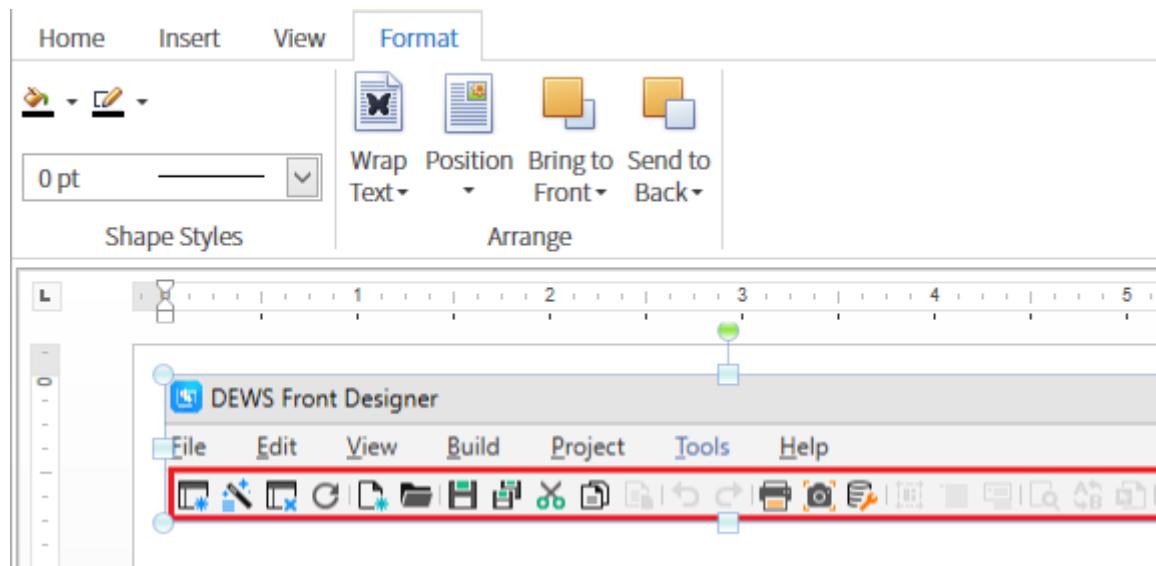


<그림 6-12. 테이블 삽입 후 선택했을 때 Layout 화면>

| 구분                  | 하위 메뉴          | 설명                               |
|---------------------|----------------|----------------------------------|
| Table Style Options | Header Row     | 테이블의 첫 번째 행에 대한 특수 서식 표시         |
|                     | Last Columns   | 테이블의 마지막 열에 대한 특수 서식 표시          |
|                     | First Column   | 테이블의 첫 번째 열에 대한 특수 서식 표시         |
|                     | Banded Rows    | 짝수 행이 홀수 행과 다르게 서식이 지정된 줄무늬 행 표시 |
|                     | Total Rows     | 테이블의 마지막 행에 대한 특수 서식 표시          |
|                     | Banded Columns | 짝수 열이 홀수 열과 다르게 서식이 지정된 줄무늬 열 표시 |
| Table Styles        |                | 표 스타일 지정                         |
| Border & Shadings   |                | 표 테두리 및 음영 설정                    |

<표 6-5. 테이블 Layout 메뉴 설명>

- 그림 삽입 후 선택 했을 때 Format 탭

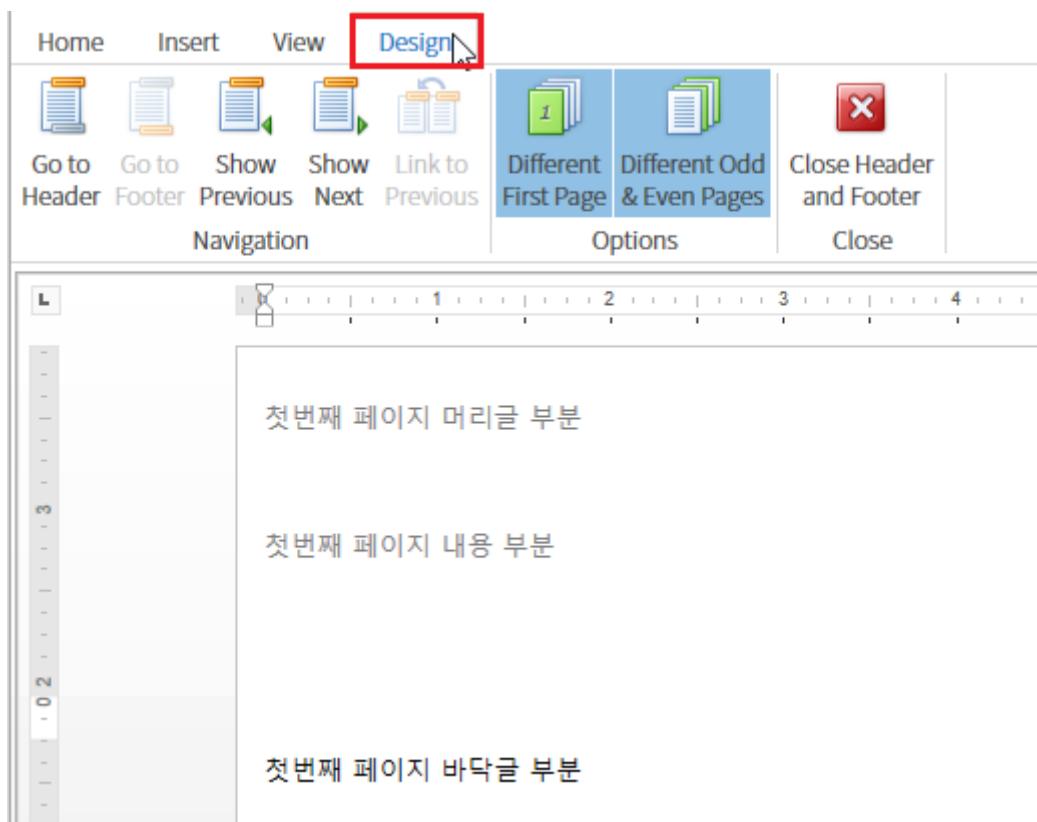


<그림 6-12. 그림 삽입 후 선택했을 때 Format 화면>

| 구분           | 하위 메뉴                | 설명           |
|--------------|----------------------|--------------|
| Shape Styles | Shape Fill           | 그림 채우기 색상 설정 |
|              | Shape Outline        | 그림 테두리 설정    |
|              | Shape Outline Weight | 그림 테두리 두께 설정 |
| Arrange      | Wrap Text            | 그림과 텍스트 배치   |
|              | Position             | 그림 위치 설정     |
|              | Bring to Front       | 앞으로 가져오기     |
|              | Send to Back         | 뒤로 보내기       |

< 표 8-5. Format 메뉴 설명>

- 머리글(Header) 또는 바닥글(Footer) 실행 했을 때 Design 탭



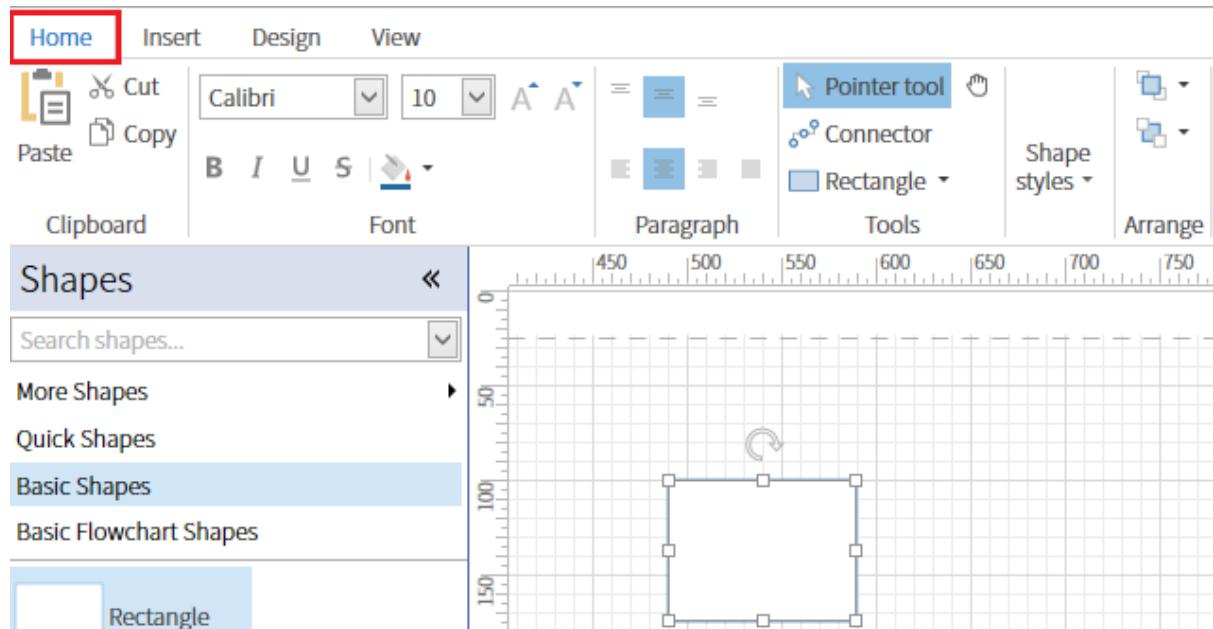
<그림 6-13. 머리글 또는 바닥글 실행했을 때 Layout 화면>

| 구분         | 하위 메뉴                     | 설명                                            |
|------------|---------------------------|-----------------------------------------------|
| Navigation | Go to Header              | 머리글 이동 및 활성화, 편집                              |
|            | Go to Footer              | 바닥글 이동 및 활성화, 편집                              |
|            | Show Previous             | 이전 구역의 머리글 또는 바닥글로 이동                         |
|            | Show Next                 | 다음 구역의 머리글 또는 바닥글로 이동                         |
|            | Link to Previous          | 현재 구역의 머리글과 바닥글이 이전 구역과 동일한 내용을 갖도록 이전 구역과 연결 |
| Options    | Different First Page      | 문서의 첫페이지에 고유한 머리글과 바닥글 지정                     |
|            | Different Odd & Even Page | 홀수 페이지와 짝수 페이지 머리글과 바닥글 다르게 지정                |
| Close      | Close Header and Footer   | 머리글 및 바닥글 편집 닫기                               |

<표 6-6. 머리글 바닥글 Design 메뉴 설명>

## ② Diagram

- Home 탭



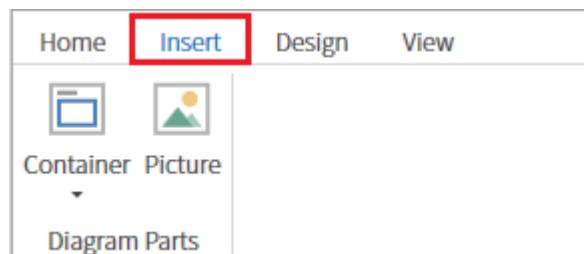
<그림 6-14. Home 화면>

Tools > Connector는 도형 사이 연결하는 선으로 자주 사용된다.

| 구분           | 하위 메뉴          | 설명                               |
|--------------|----------------|----------------------------------|
| Clipboard    |                | 붙여넣기, 복사, 잘라내기 작업                |
| Font         |                | 글꼴 설정, 스타일, 색상 변경 등의 글꼴 편집       |
| Paragraph    |                | 도형 내의 폰트 단락 설정,                  |
| Tools        | Pointer tool   | 도형 선택, 이동 및 크기 조정                |
|              | Connector      | 도형 사이 연결                         |
|              | Rectangle      | 사각형 그리기                          |
| Shape styles |                | 도형 스타일 설정, 도형 채우기 및 도형 윤곽선 색상 설정 |
| Arrange      | Bring to Front | 맨 앞으로 가져오기                       |
|              | Send to Back   | 맨 뒤로 보내기                         |

<표 6-7. Home 메뉴 설명>

- Insert 탭

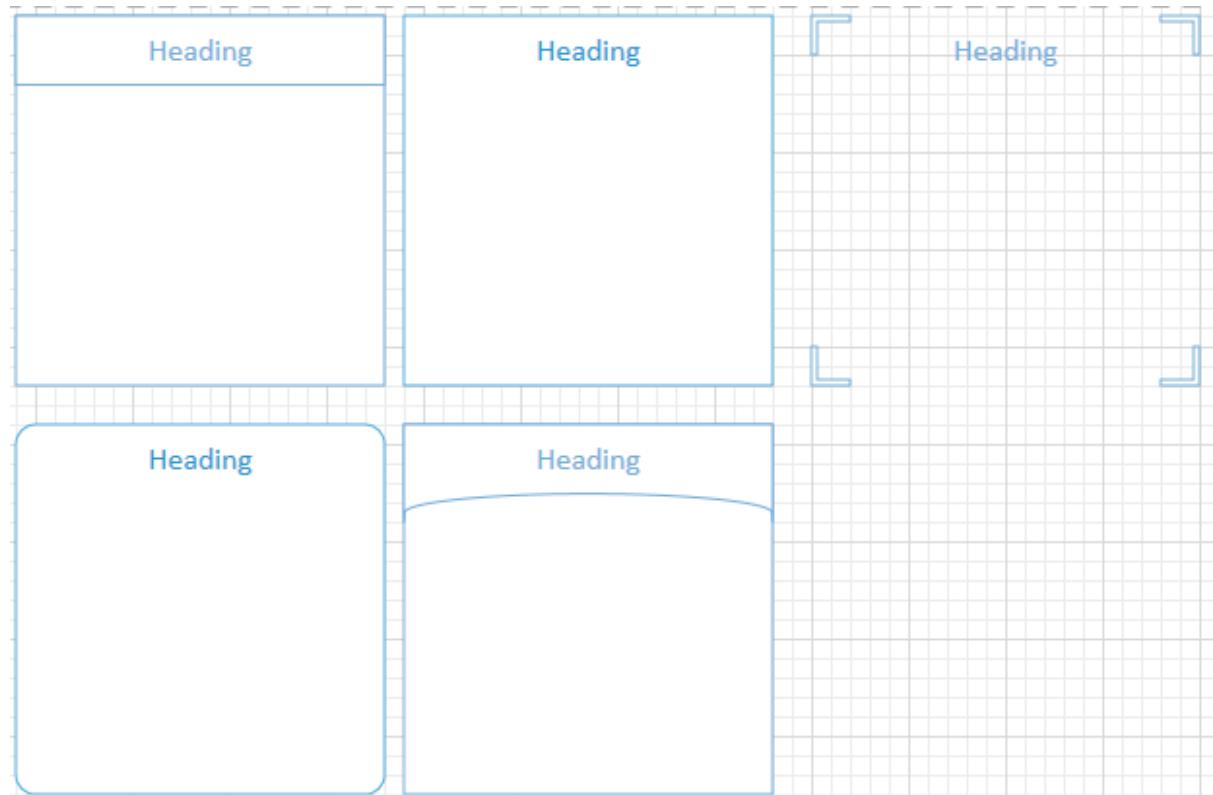


<그림 6-15. Home 화면>

| 구분            | 하위 메뉴     | 설명      |
|---------------|-----------|---------|
| Diagram Parts | Container | 컨테이너 삽입 |
|               | Picture   | 그림 삽입   |

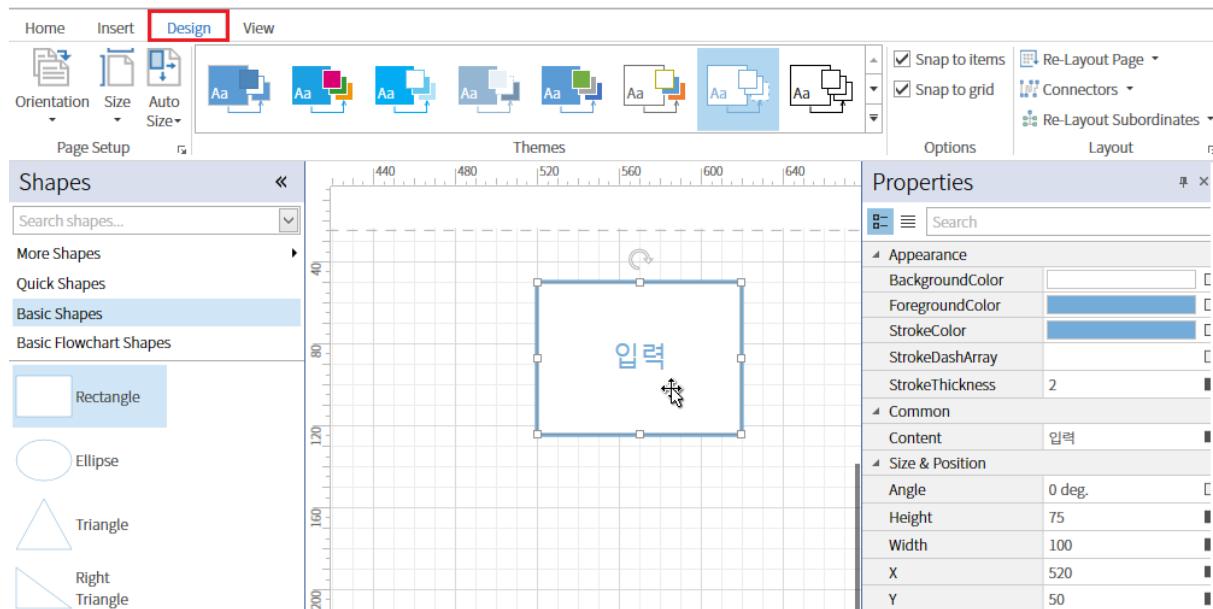
<표 6-7. Home 메뉴 설명>

컨테이너는 <그림 6-16>과 같이 다양한 모양을 선택해서 삽입할 수 있다.



<그림 6-16. Container 화면>

- Design 탭

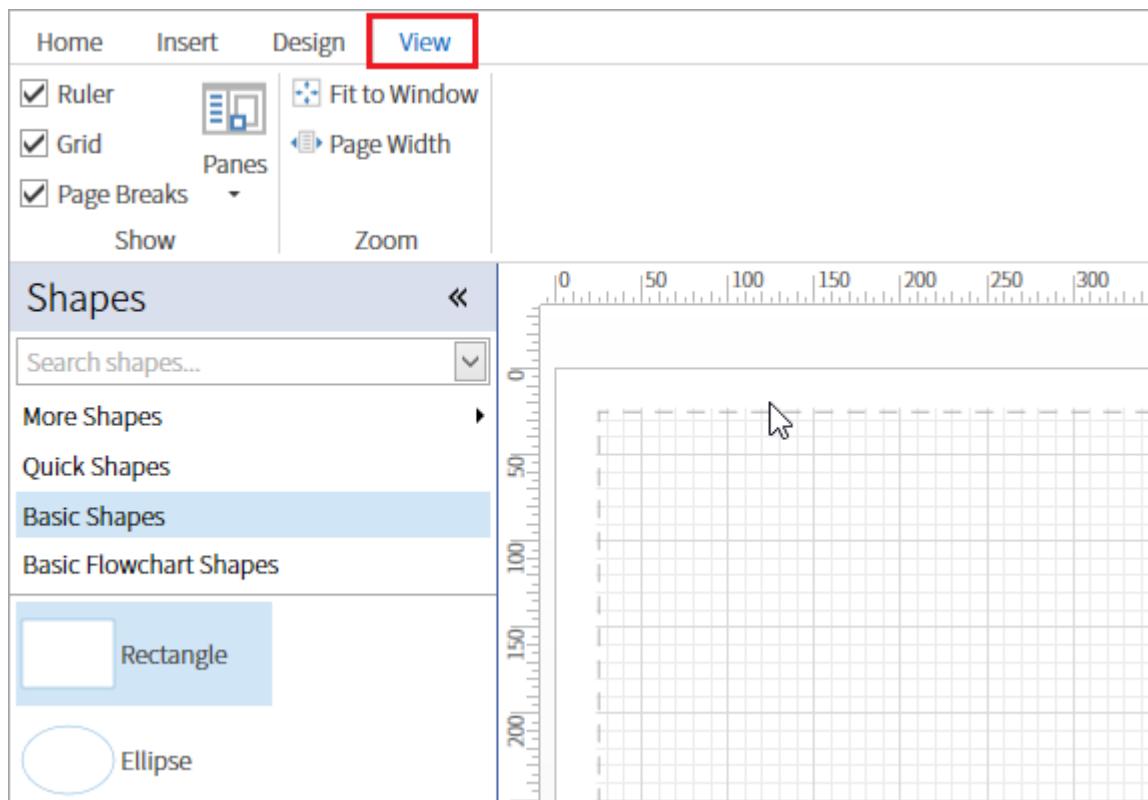


<그림 6-17. Home 화면>

| 구분         | 하위 메뉴                  | 설명                      |
|------------|------------------------|-------------------------|
| Page Setup | Orientation            | 용지 방향                   |
|            | Size                   | 크기                      |
|            | Auto Size              | 자동 크기                   |
| Themes     |                        | 테마 설정                   |
| Options    | Snap to items          | 아이템에 맞추기                |
|            | Snap to grid           | 그리드에 맞추기                |
| Layout     | Re-Layout Page         | 페이지 재배치                 |
|            | Connectors             | 연결 모양 변경                |
|            | Re-Layout Subordinates | 루트 또는 선택된 항목의 하위 항목 재배치 |

<표 6-8. Diagram Design 설명>

- View 탭

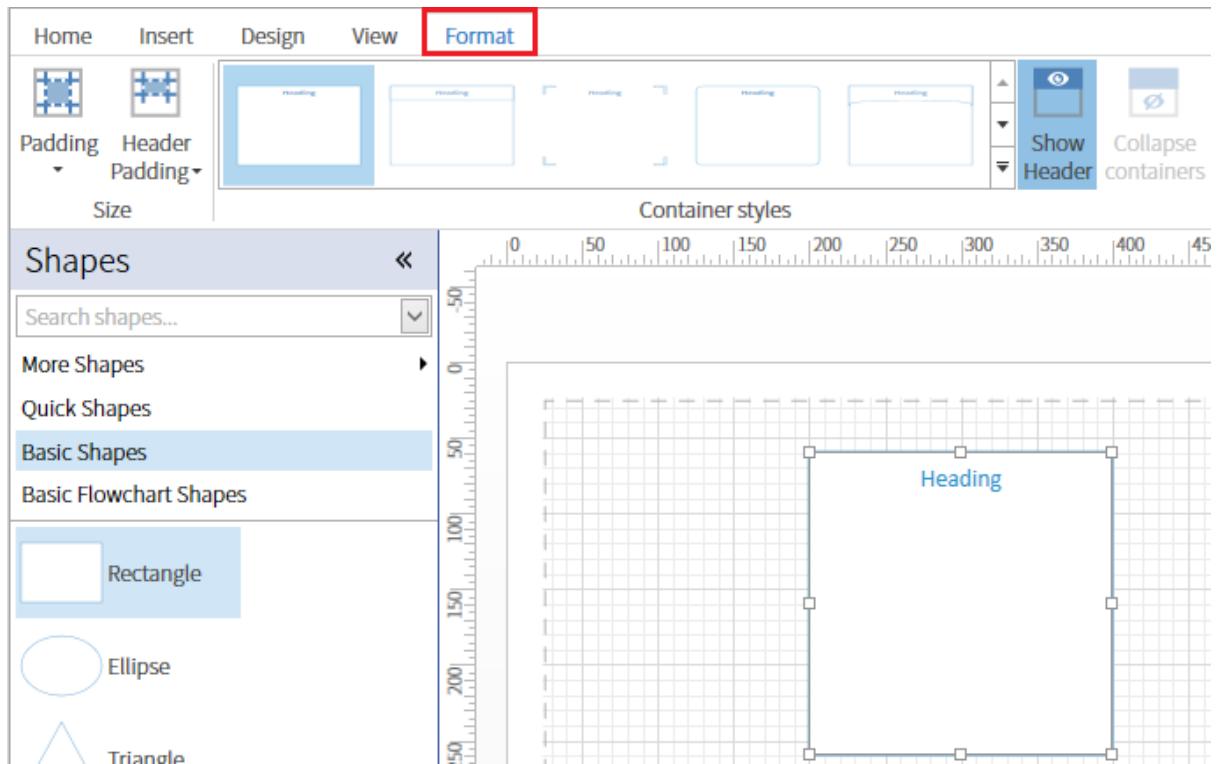


<그림 6-18. View 화면>

| 구분   | 하위 메뉴         | 설명                 |
|------|---------------|--------------------|
| Show | Ruler         | 눈금자 표시/감추기         |
|      | Grid          | 눈금선 표시/감추기         |
|      | Page Breaks   | 인쇄 나누기 표시/감추기      |
|      | Panes         | 도형 선택과 속성 창 표시/감추기 |
| Zoom | Fit to Window | 가로, 세로 화면에 맞게 맞춤   |
|      | Page Width    | 가로 페이지에 맞춤         |

<표 6-9. Diagram View 설명>

- 컨테이너 (Container) 선택 시 Format 탭

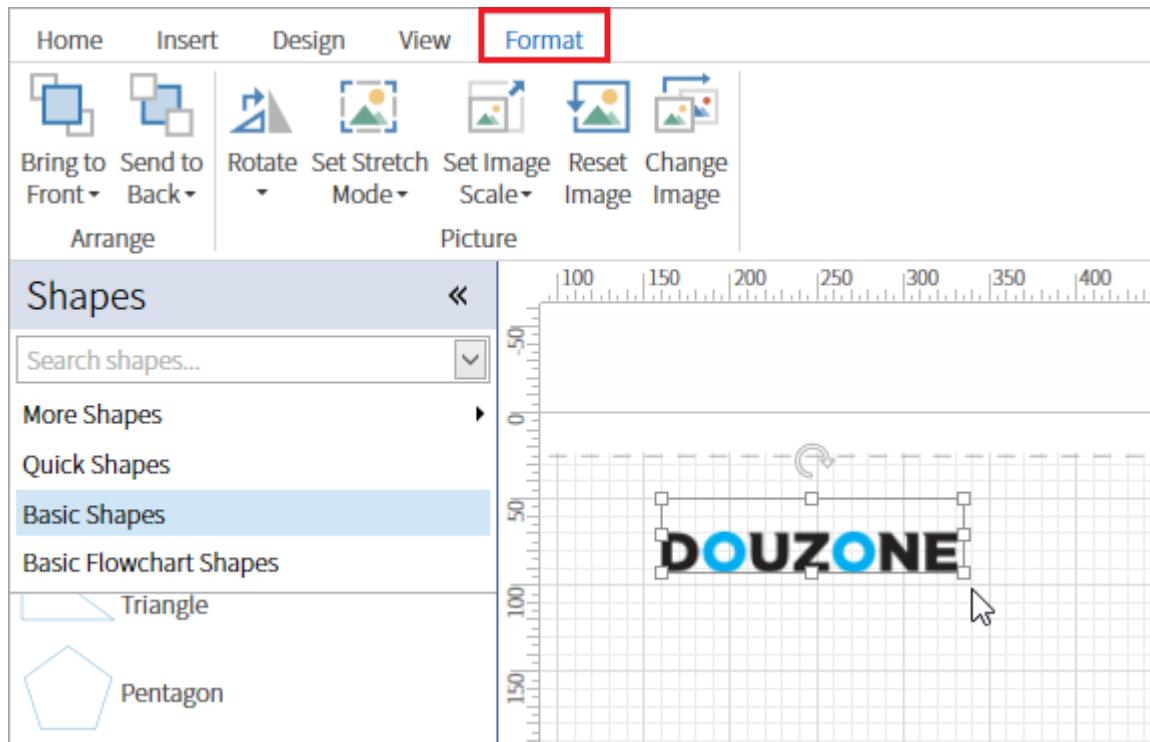


<그림 6-19. View 화면>

| 구분               | 하위 메뉴 | 설명                                |
|------------------|-------|-----------------------------------|
| Size             |       | 여백 설정                             |
| Container styles |       | 컨테이너 스타일 설정 및 타이틀 표시/감추기, 컨테이너 접음 |

<표 6-10. Diagram View 설명>

- 그림 삽입 후 선택 했을 때 Format 탭



<그림 6-20. Format 화면>

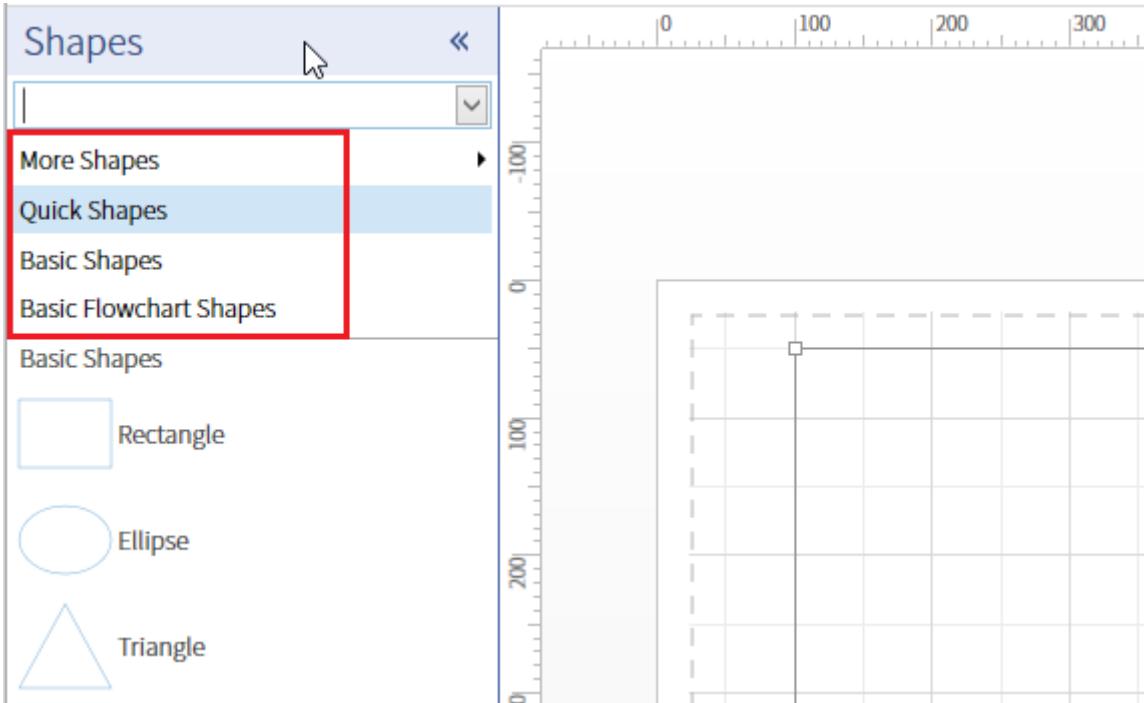
| 구분      | 하위 메뉴            | 설명          |
|---------|------------------|-------------|
| Arrange | Bring to Front   | 앞으로 가져오기    |
|         | Send to Back     | 뒤로 보내기      |
| Picture | Rotate           | 사진 회전, 뒤집기  |
|         | Set Stretch Mode | 그림 늘임 모드 설정 |
|         | Set Image Scale  | 사이즈 설정      |
|         | Reset Image      | 이미지 재설정     |
|         | Change Image     | 이미지 변경      |

<표 6-11. Diagram 그림 삽입 Format 설명>

## 2) Diagram 도형과 속성

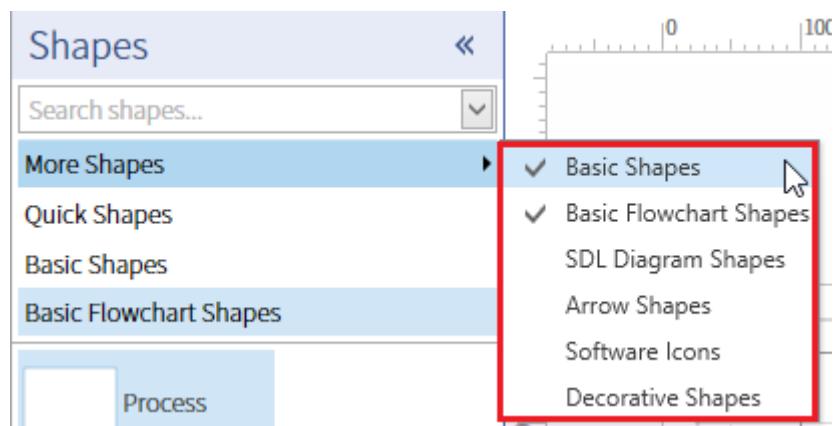
### ① 도형

- 기본적으로 4개 메뉴로 되어 있으며, More Shapes는 다른 형태의 도형을 추가, Quick Shapes는 More Shapes에서 체크된 도형 종류별로 자주 사용되는 도형을 제공한다.



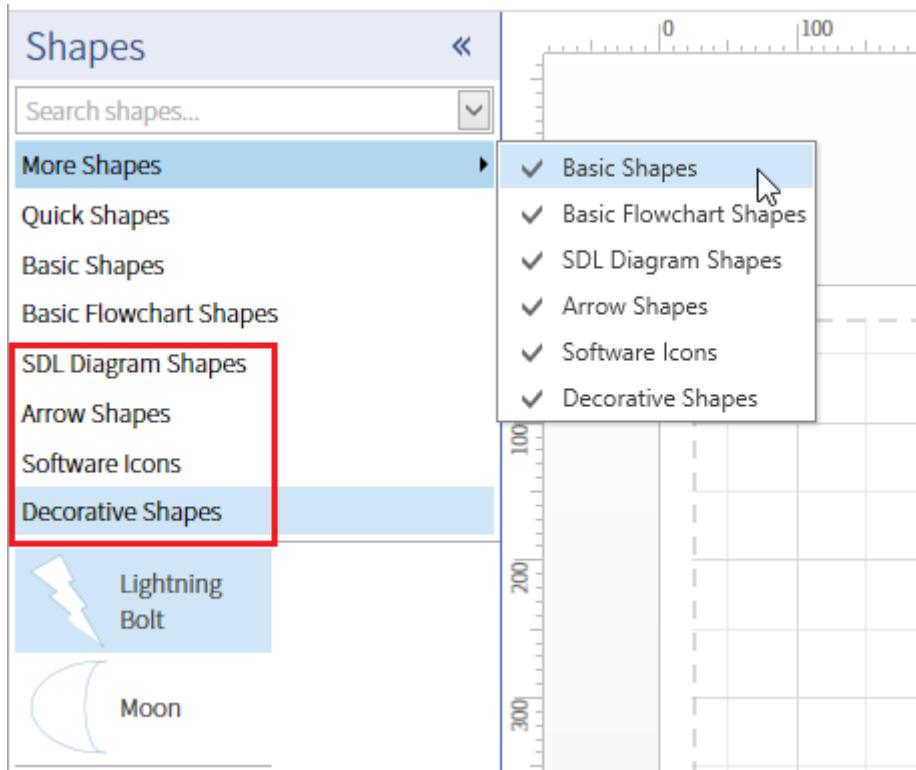
<그림 6-21. Shapes 화면>

- More Shapes 을 눌러 기본으로 제공하는 도형인 Basic Shapes 와 Basic Flowchart Shapes 외에 다양한 형태의 도형을 메뉴에 추가할 수 있다.



<그림 6-22. More Shapes 화면>

- More Shapes에서 모두 체크했을 경우 추가할 수 있는 도형 메뉴도 늘어난다.

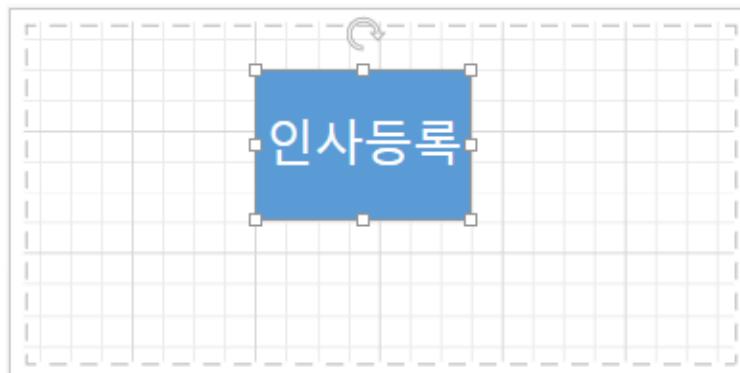


<그림 6-23. 모든 형태의 도형을 체크했을 때 메뉴 화면>

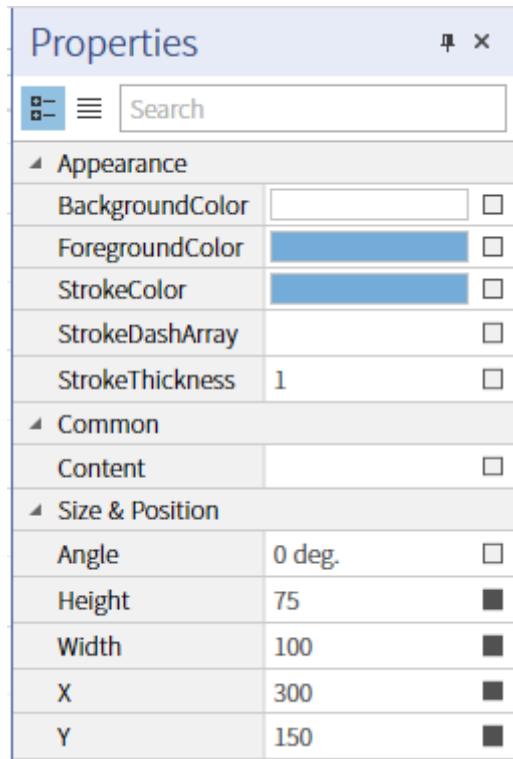
## ② 속성

- 선택한 개체(도형, 컨테이너, 화살표, 그림)에 따라서 속성 화면도 다르다.

### 가) 도형(Shape) 선택



<그림 6-24. 도형 선택 화면>

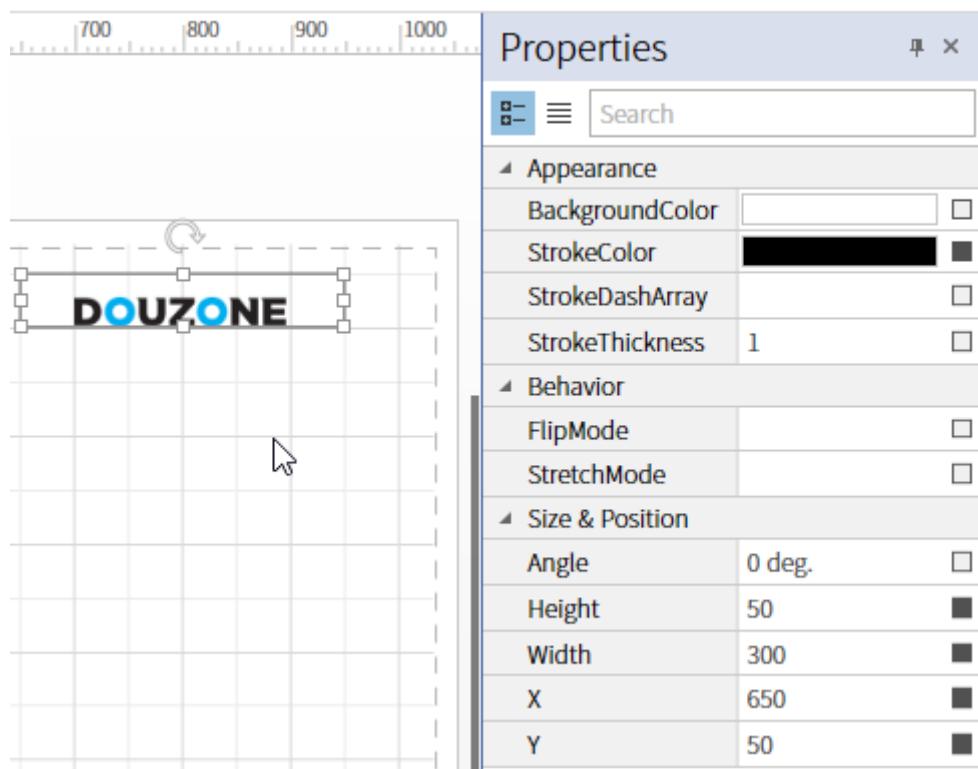


<그림 6-25. 도형 선택 시 Properties 화면>

| 구분              | 하위 메뉴           | 설명           |
|-----------------|-----------------|--------------|
| Appearance      | BackgroudColor  | 배경색          |
|                 | ForegroundColor | 전경색          |
|                 | StrokeColor     | 도형 윤곽선 색     |
|                 | StrokeDashArray | 도형 윤곽선 선 스타일 |
|                 | StrokeThickness | 도형 윤곽선 두께    |
| Common          | Content         | 텍스트 입력       |
| Size & Position | Angle           | 각도           |
|                 | Height          | 높이           |
|                 | Width           | 너비           |
|                 | X               | X 축 좌표       |
|                 | Y               | Y 축 좌표       |

<표 6-12. 도형 선택시 Properties 설명>

나) 그림 선택

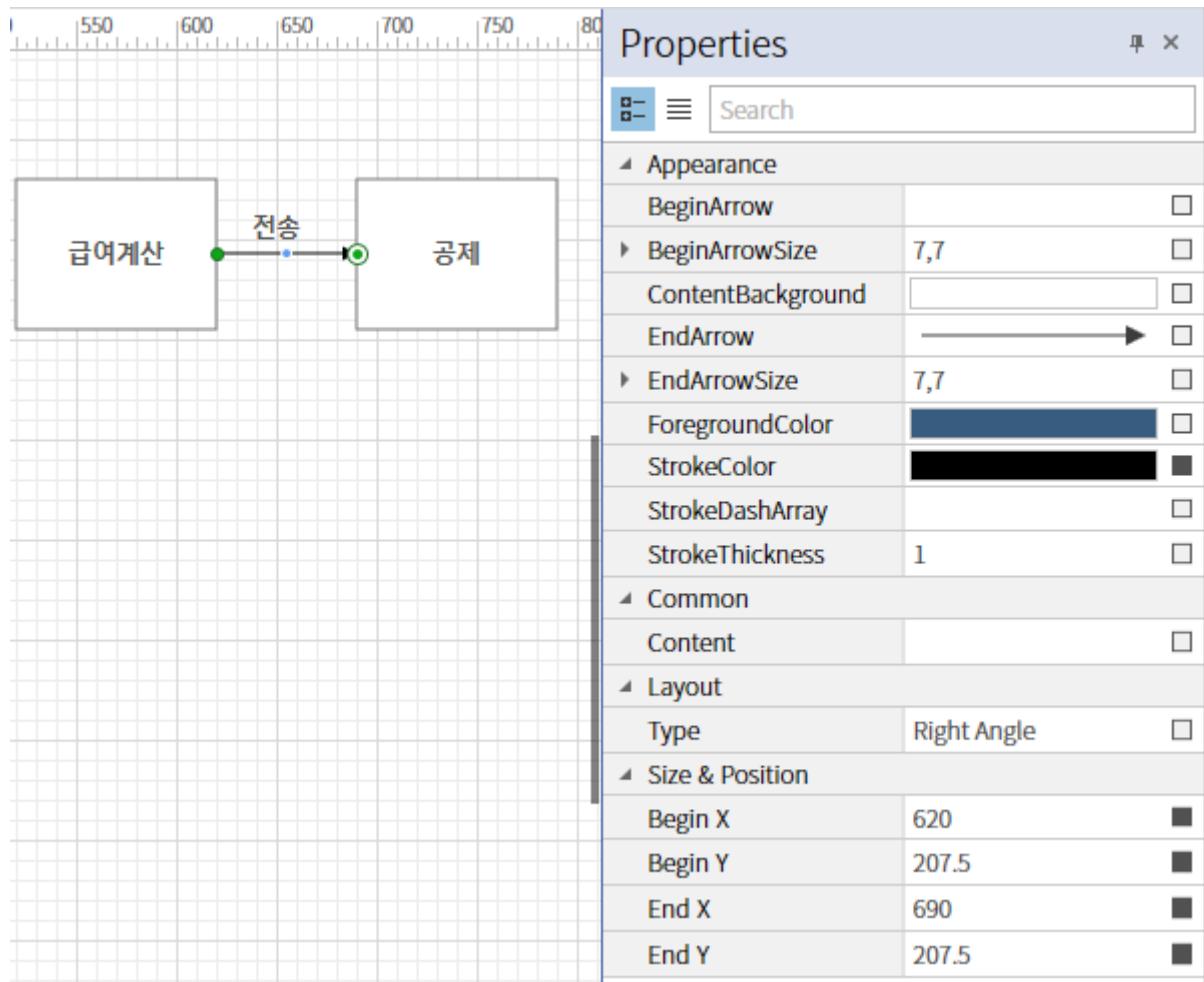


<그림 6-26. 그림 선택 시 Properties 화면>

| 구분              | 하위 메뉴           | 설명           |
|-----------------|-----------------|--------------|
| Appearance      | BackgroundColor | 배경색          |
|                 | StrokeColor     | 그림 윤곽선 색     |
|                 | StrokeDashArray | 그림 윤곽선 선 스타일 |
|                 | StrokeThickness | 그림 윤곽선 두께    |
| Behavior        | FlipMode        | 뒤집기 모드       |
|                 | StretchMode     | 늘임 모드        |
| Size & Position | Angle           | 각도           |
|                 | Height          | 높이           |
|                 | Width           | 너비           |
|                 | X               | X 축 좌표       |
|                 | Y               | Y 축 좌표       |

<표 6-13. 그림 선택시 Properties 설명>

다) Connector(연결선) 선택



<그림 6-27. Connnetor 선택 시 Properties 화면>

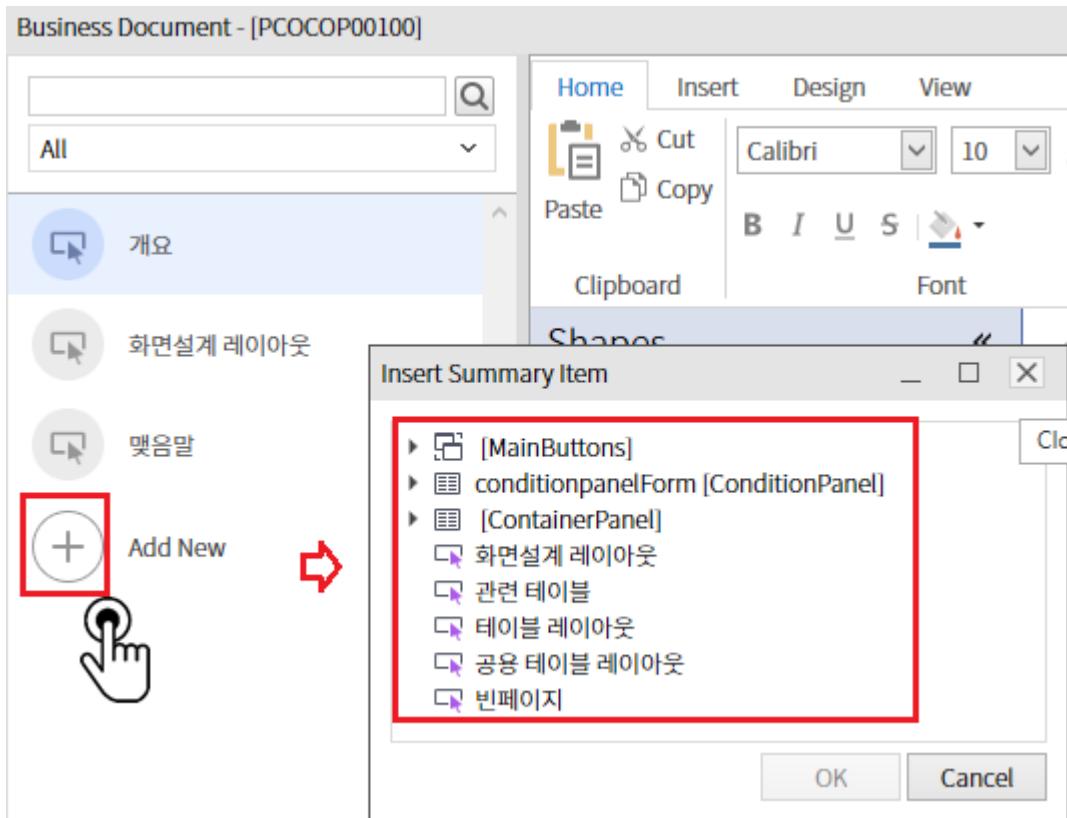
| 구분         | 하위 메뉴             | 설명                 |
|------------|-------------------|--------------------|
| Appearance | BeginArrow        | 시작 화살표 모양          |
|            | BeginArrowSize    | 시작 화살표 사이즈(높이, 너비) |
|            | ContentBackground | 텍스트 배경색            |
|            | EndArrow          | 끝 화살표 모양           |
|            | EndArrowSize      | 끝 화살표 사이즈(높이, 너비)  |
|            | ForegroundColor   | 전경색                |
|            | StrokeColor       | 화살표 색              |
|            | StrokeDashArray,  | 화살표 스타일            |
|            | StrokeThickness   | 화살표 두께             |
| Common     | Content           | 텍스트 입력             |

|                 |         |                         |
|-----------------|---------|-------------------------|
| Layout          | Type    | 타입 설정(곡선, 직각, 일직선, 조작도) |
| Size & Position | Begin X | 시작 X 축 좌표               |
|                 | Begin Y | 시작 Y 축 좌표               |
|                 | End X   | 끝 X 축 좌표                |
|                 | End Y   | 끝 Y 축 좌표                |

<표 6-14. Connector 선택시 Properties 설명>

### 3) 추가 페이지

Add New 버튼을 클릭했을 때 추가할 수 있는 페이지는 화면에 있는 모든 컨트롤들, 기능 개요 레이아웃, 화면 설계 레이아웃, 관련 테이블, 테이블 레이아웃, 공용 테이블 레이아웃, 빈 페이지 등을 추가할 수 있다.



<그림 6-28. 추가할 수 있는 페이지 선택 화면>

개요 페이지에서 기능 개요와 프로세스 흐름도를 작성했을 때 기존 설계 문서(PPT) 상단에 있는 부분인 Function 제외한 모듈, Component, 메뉴 ID의 경우 개발 도구에서 설정한 값으로 고정된다. Function 입력 부분은 Canvas의 function에서 작성한 값이 출력된다.

| 모듈    | HR, 인사               | Component | PAM, 인사관리 | Function | PAR, 급여보고서 |
|-------|----------------------|-----------|-----------|----------|------------|
| 메뉴 ID | PBMPAR01000, 보험료공제현황 |           |           |          |            |

<그림 6-29. 기존 설계 문서(ppt) 상단 부분 화면>

The screenshot shows a Microsoft Word document titled "프로그램 설계". The ribbon is visible at the top, showing tabs for Home, Insert, View, and several icons for Save, Sync, History, Diagram, Message Box, Copy, Paste, and Paste Special. A sidebar on the left lists items: "개요" (selected), "Grid1", "맵음말", and "Add New". The main content area contains two numbered steps:

1. 프로세스 개요  
급여에서 공제된 보험료공제내역을 조회
2. 프로세스 절차
  - 1) 급여계산을 통해 보험료를 공제
  - 2) 보험료 공제현황을 조회

A red arrow points from the sidebar to the main content area, indicating the connection between the design and the implementation.

The screenshot shows a Microsoft Word document titled "프로그램 설계". A table is present at the top:

|       |               |           |    |
|-------|---------------|-----------|----|
| 모듈    | CL            | Component | 개요 |
| 메뉴 ID | PBMPAR_Design |           |    |

The sidebar on the left is labeled "기능개요 및 업무흐름" and contains the same two steps as in the previous screenshot:

1. 프로세스 개요  
급여에서 공제된 보험료공제내역을 조회한다.
2. 프로세스 절차
  - 1) 급여계산을 통해 보험료를 공제
  - 2) 보험료 공제현황을 조회

<그림 6-30. 개요에 작성한 기능 개요 및 업무 흐름 및 pdf 출력 화면>

모듈, Componet, 메뉴 ID 등 항목 값을 수정해야 되는 경우 기능 개요 레이아웃이나 화면 설계 레이아웃 페이지를 선택하고 추가해서 작성한다.

Add New을 눌러 화면 설계 레이아웃을 선택하면 <그림 6-31>처럼 페이지가 추가 되며 모든 항목 수정이 가능하다.

모듈 HR, 인사 Component PAM, 인사관리

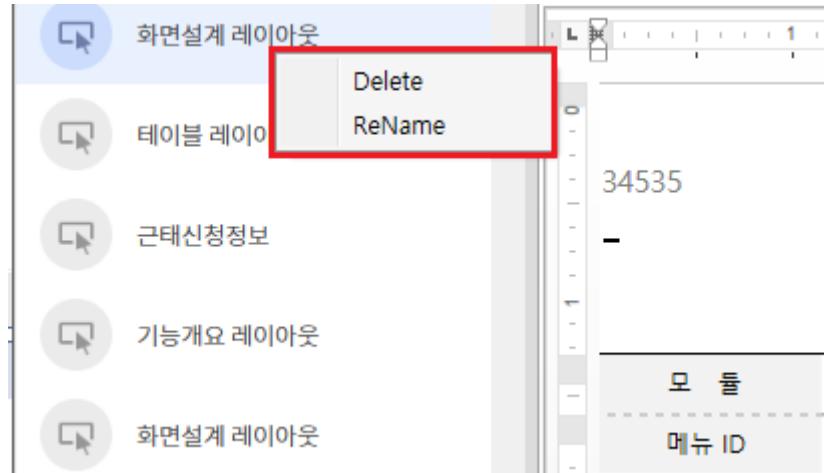
메뉴 ID PBMPAR01000, 보험료 공제 현황

화면설계 및 요구기능

1. 프로세스 개요  
급여에서 공제된 보험료공제내역을 조회한다.
2. 프로세스 절차
  - 1) 급여계산을 통해 보험료를 공제
  - 2) 보험료 공제현황을 조회

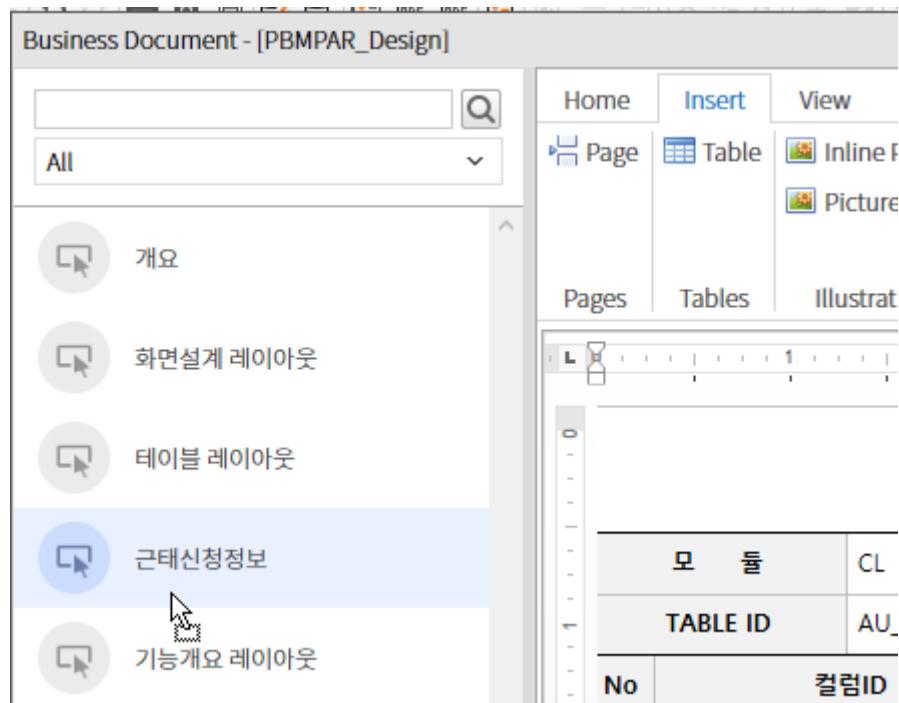
<그림 6-31. 화면 설계 레이아웃 페이지 추가 후 작성 화면>

개요 및 맷음말을 제외한 다른 페이지는 선택 후 우클릭 후 삭제(Delete)가 가능하며 사용자가 추가한 페이지 경우 이름 변경(ReName)이 가능하다. 컨트롤인 경우 개발 도구 Properties 창의 DocumentID 속성에서 변경 가능하다



<그림 6-32. 페이지 선택 후 삭제와 페이지 이름 변경 화면>

. 페이지 순서 변경은 개요 및 맷음말로 제외하고 페이지 선택 후 마우스 Drag & Drop 으로 이동이 가능하다.

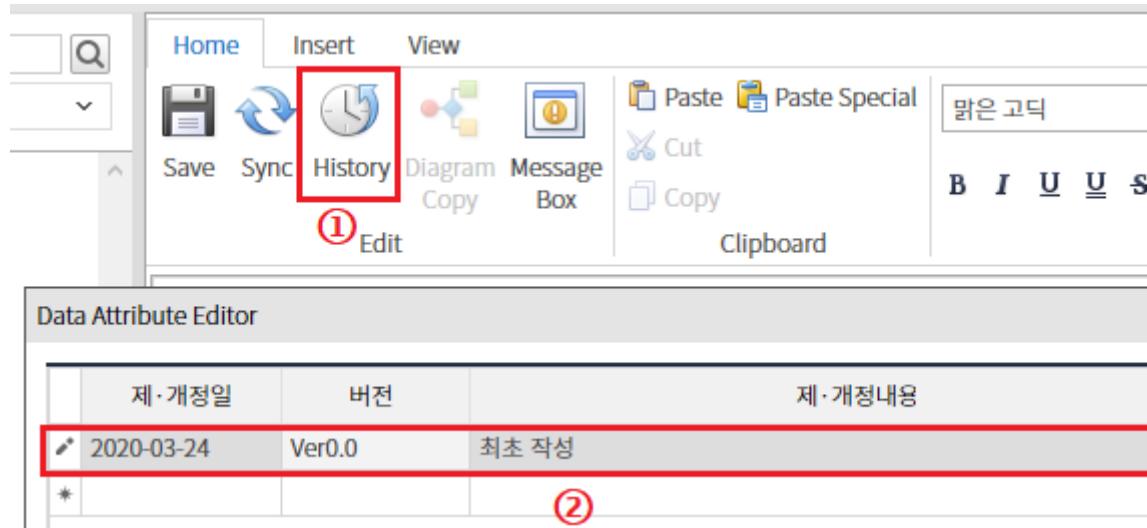


<그림 6-33. 페이지 이동 화면>

### 3. 설계 문서 작성

#### 1) 문서 개정 이력

문서 개정 이력은 Home > History 툴바를 클릭하면 작성 화면이 열리며 항목을 입력한다.



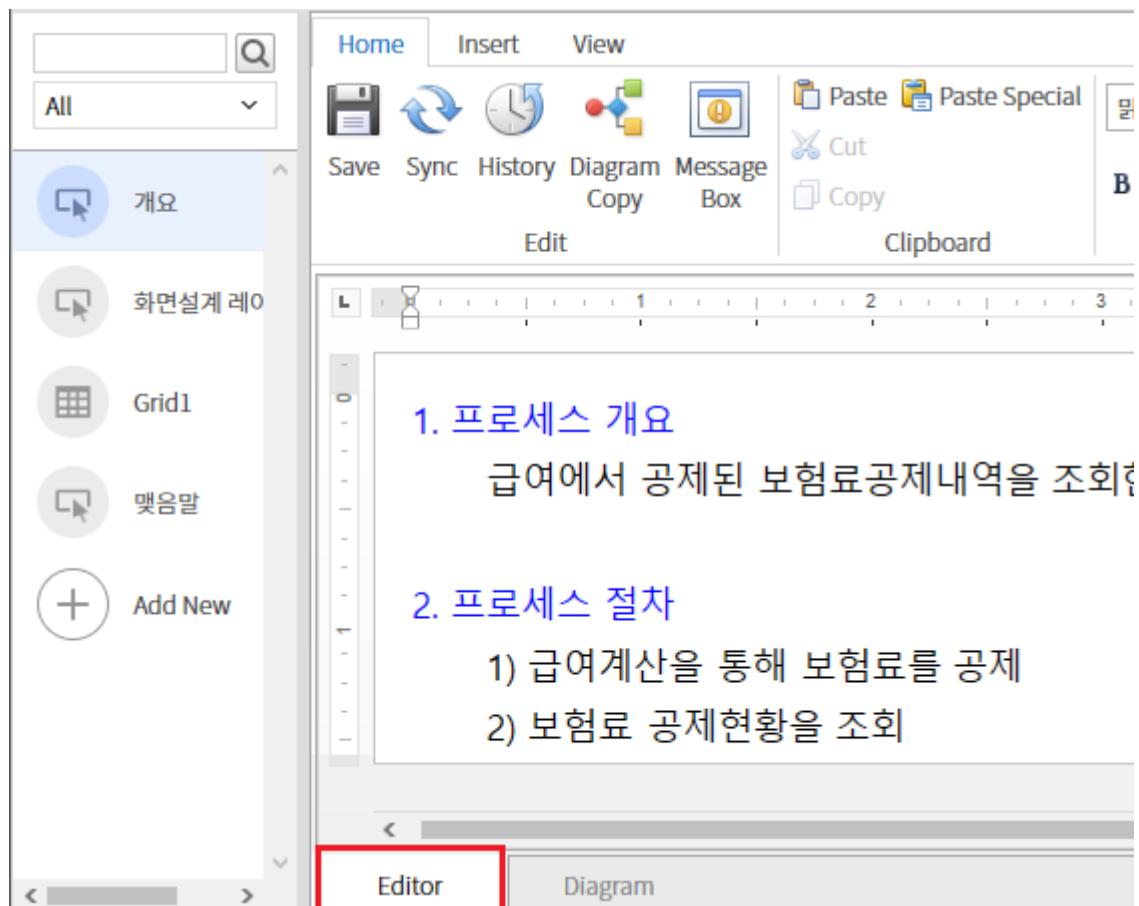
<그림 6-34. 문서 개정 이력 작성 화면>

#### 2) 기능 개요

기본적으로 기능 개요 및 업무 흐름도 중에 기능 개요는 Editor 탭에서 작성한다.

The screenshot shows a slide titled '프로그램 설계' (Program Design). It includes sections for '모듈' (Module) set to 'HR, 인사' and 'Component' set to 'PAM, 인사', and '메뉴 ID' (Menu ID) set to 'PBMPAR01000, 보험료공제현황'. Below this is a section titled '1. 기능개요 및 업무흐름' (1. Functional Requirements and Work Flow). This section has two main items: '기능 개요' (Functional Requirements) and '1. 프로세스 개요' (1. Process Overview). The '기능 개요' item is highlighted with a red box and labeled with a red circled '①'. The '1. 프로세스 개요' item is also highlighted with a red box and labeled with a red circled '②'. Under '1. 프로세스 개요', there are two sub-points: '1. 프로세스 개요' (1. Process Overview) and '2. 프로세스 절차' (2. Process Procedure), which lists '1) 급여계산을 통해 보험료를 공제' (1) Deduct insurance premium through salary calculation) and '2) 보험료 공제현황을 조회' (2) Query insurance premium deduction status).

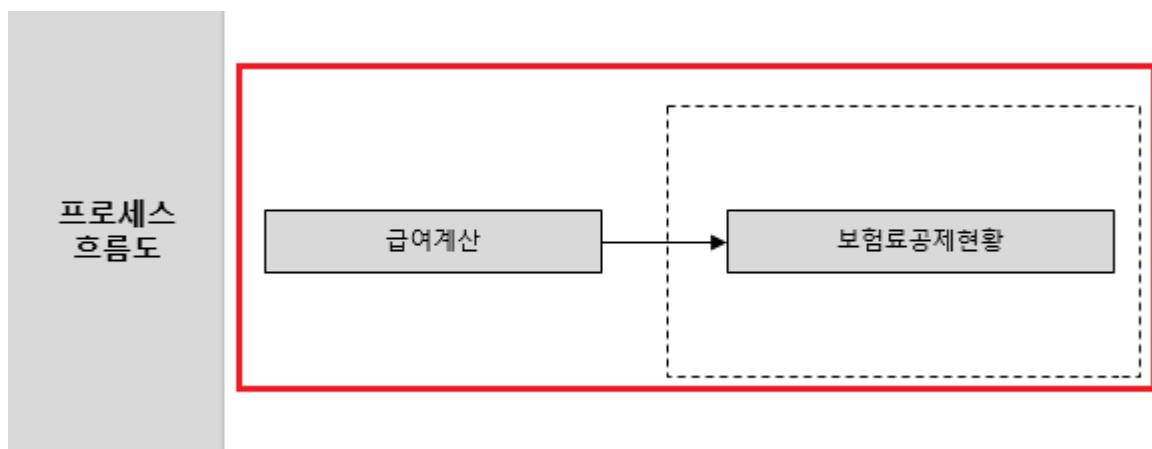
<그림 6-35. 기준 문서(ppt) 기능 개요 부분>



<그림 6-36. 설계 도구에서 기능 개요 작성 화면>

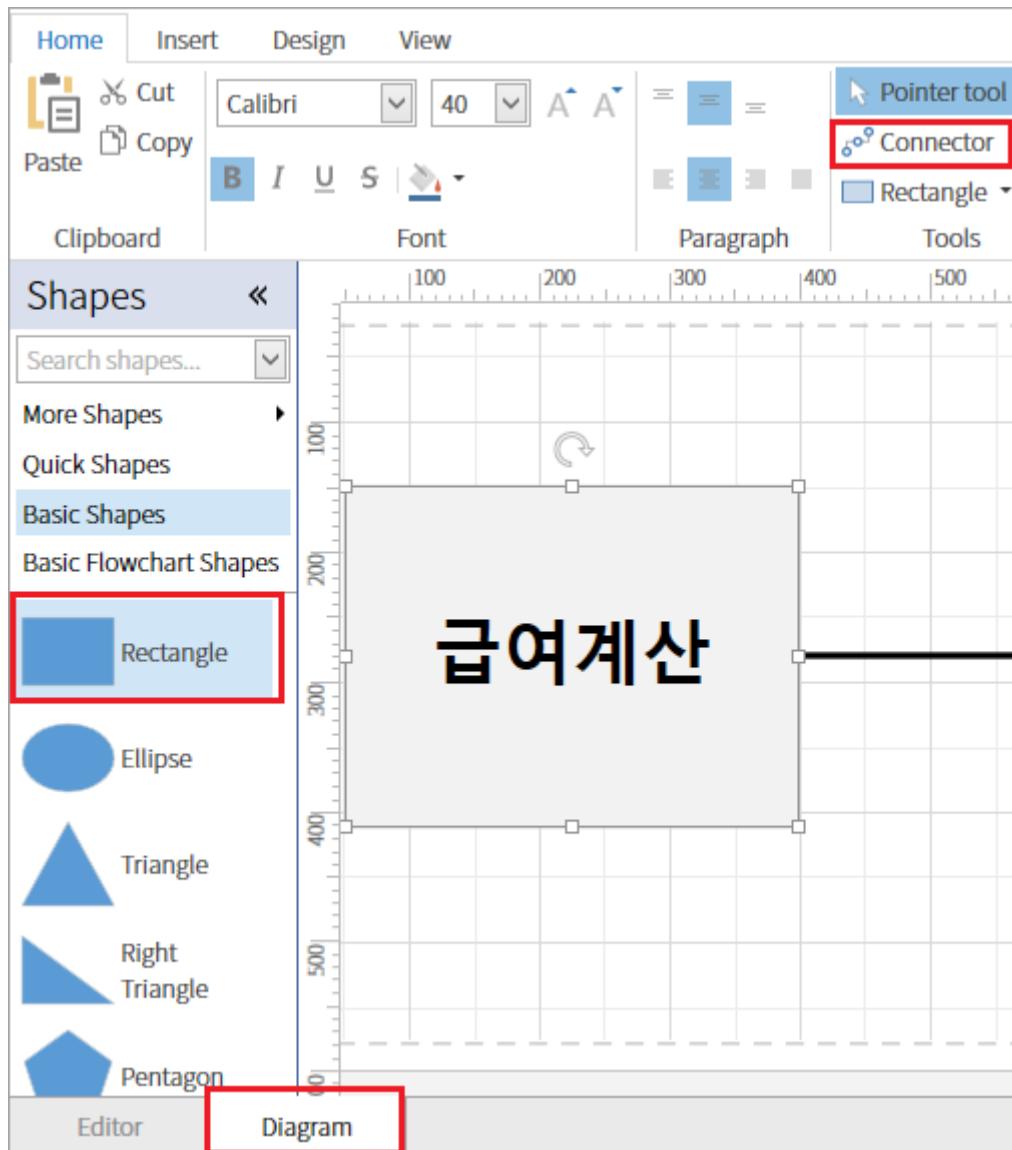
### 3) 업무 흐름도

업무 흐름도는 Diagram 탭에서 작성한다.



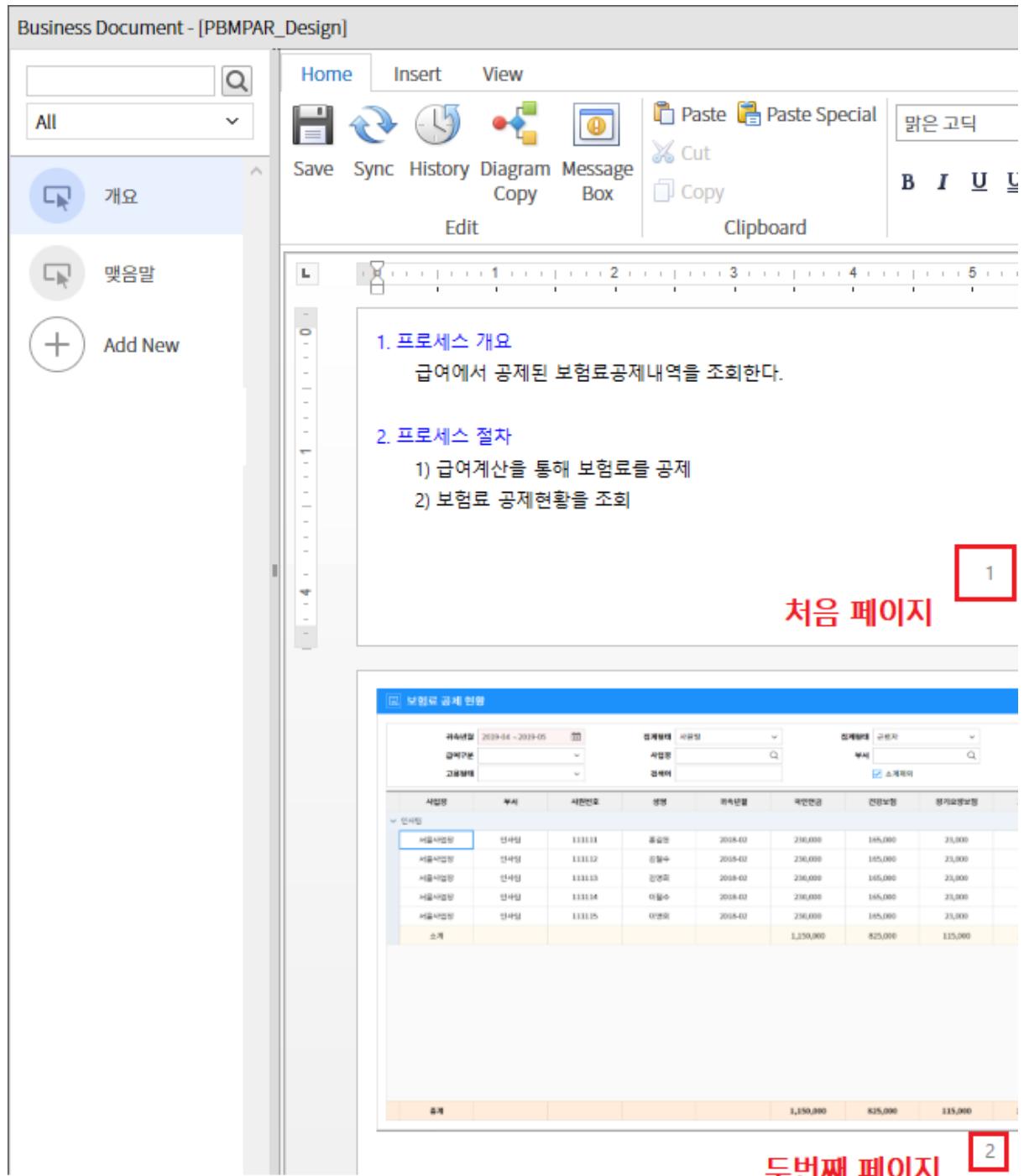
<그림 6-37. 기존 문서(ppt) 프로세스 흐름도 부분>

Diagram 탭에서 도형(Shape)과 Connector 를 사용하여 프로세스 흐름도를 작성한다.



<그림 6-38. 설계 도구에서 프로세스 흐름도 작성 화면>

개요 페이지에서 처음 페이지를 제외한 다음 페이지는 PDF나 WORD 문서 파일 생성 시 화면설계 및 요구기능 양식으로 만들어진다.



<그림 6-39. 설계 도구의 개요에서 작성한 화면>

- <그림 6-39>의 두번째 페이지는 <그림 6-40>에서 pdf 파일이나 Word 파일 생성 시 처음 페이지와 다르게 양식이 화면설계 및 요구 기능으로 표시되는 것을 확인할 수 있다.

The screenshot shows a software interface titled "프로그램 설계". At the top, there is a header with columns for 모듈 (Module), CL, Component, 개요 (Overview), Function, and stat. Below this, a sub-header reads "기능개요 및 업무흐름".

**1. 프로세스 개요**  
급여에서 공제된 보험료공제내역을 조회한다.

**2. 프로세스 절차**

- 1) 급여계산을 통해 보험료를 공제
- 2) 보험료 공제현황을 조회

On the left side, there is a vertical sidebar labeled "프로세스 흐름도" (Process Flowchart). It contains two boxes: "급여계산" and "보험료공제현황". An arrow points from "급여계산" to "보험료공제현황".

Below the sidebar, there is a large red rectangular box highlighting the section "화면설계 및 요구기능". Inside this box, there is a screenshot of a database grid titled "부록별 결과 현황". The grid has columns for 사번, 부서, 이름번호, 성명, 고지금액, 미전달금, 건강보험, 광역보건법, 고액보장, 세액증정, and 합계. The data is as follows:

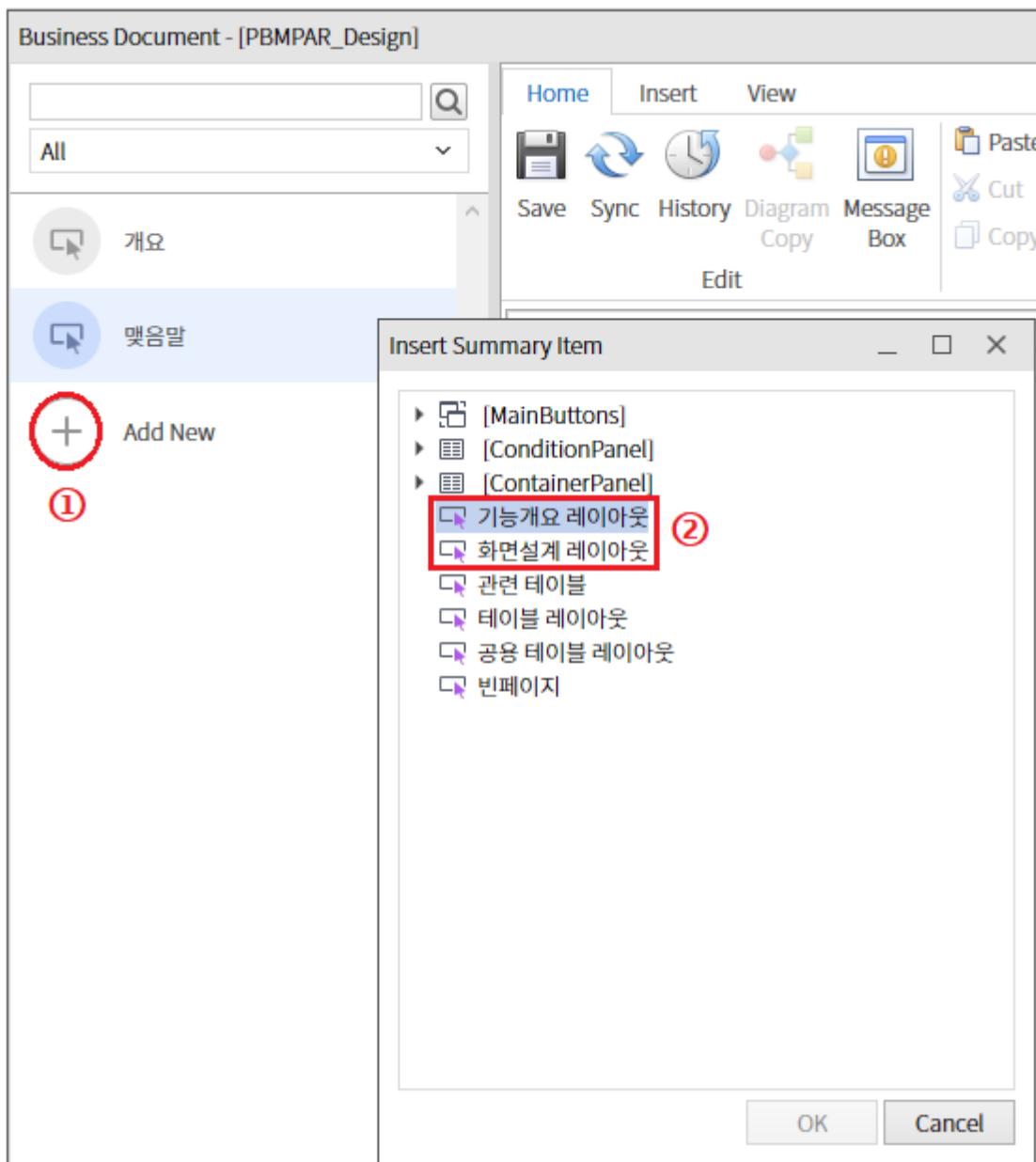
| 사번    | 부서  | 이름번호   | 성명  | 고지금액      | 미전달금    | 건강보험    | 광역보건법  | 고액보장 | 세액증정    | 합계        |
|-------|-----|--------|-----|-----------|---------|---------|--------|------|---------|-----------|
| 서울사무처 | 인사팀 | 111111 | 김영호 | 200,000   | 100,000 | 20,000  | 10,000 | 0    | 40,000  | 451,000   |
| 서울사무처 | 인사팀 | 111112 | 김민수 | 200,000   | 100,000 | 20,000  | 10,000 | 0    | 40,000  | 451,000   |
| 서울사무처 | 인사팀 | 111113 | 김민경 | 200,000   | 100,000 | 20,000  | 10,000 | 0    | 40,000  | 451,000   |
| 서울사무처 | 인사팀 | 111114 | 김민우 | 200,000   | 100,000 | 20,000  | 10,000 | 0    | 40,000  | 451,000   |
| 서울사무처 | 인사팀 | 111115 | 김민희 | 200,000   | 100,000 | 20,000  | 10,000 | 0    | 40,000  | 451,000   |
|       |     | 소계     |     | 1,000,000 | 500,000 | 100,000 | 50,000 | 0    | 200,000 | 1,250,000 |

<그림 6-40. PDF 파일 미리보기 화면>

#### 4) 신규 양식 추가

##### ① 기능개요 및 화면설계 레이아웃

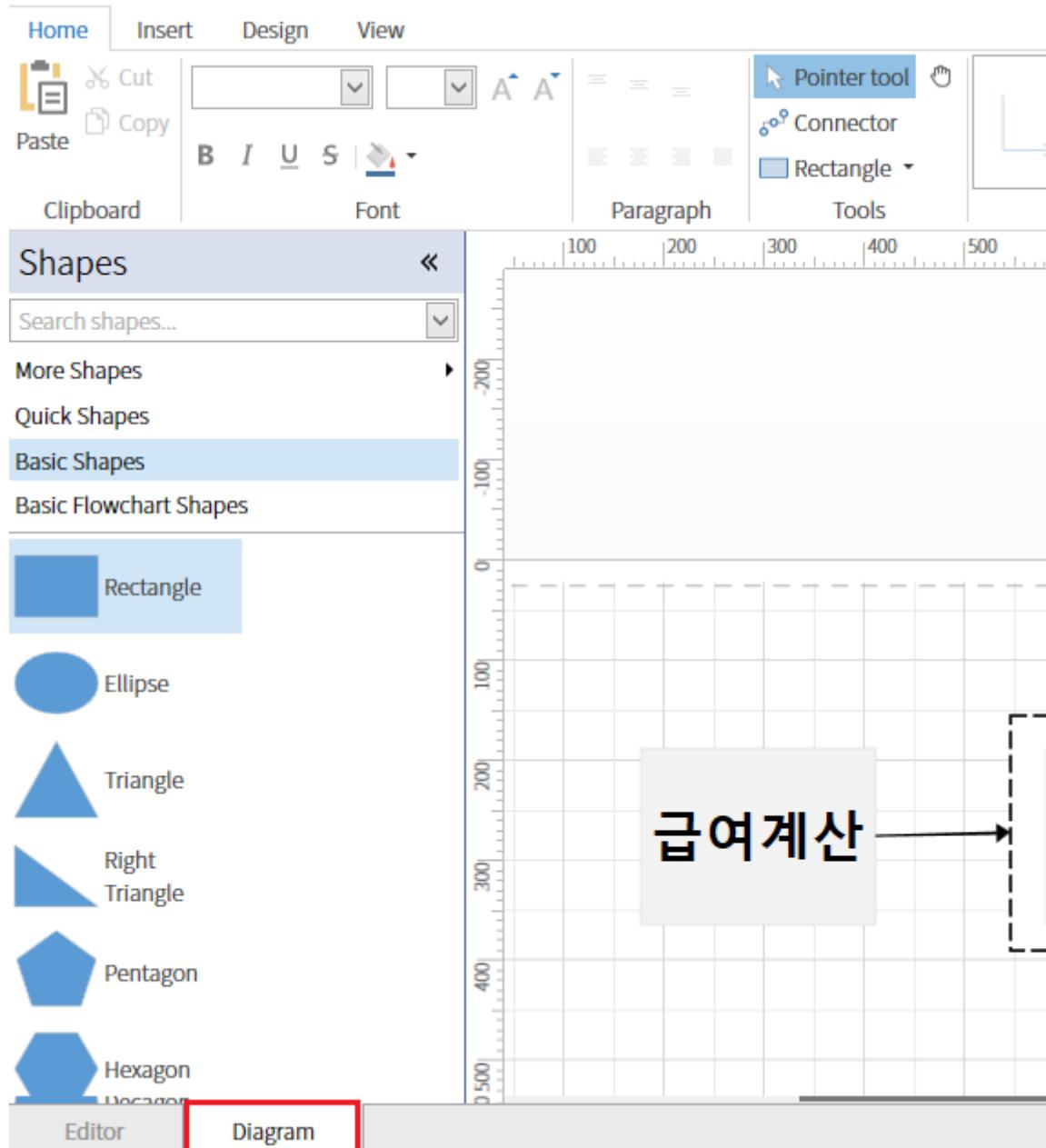
2), 3)의 개요에서 작성했을 때는 모듈, Component, 메뉴 ID 항목 값이 고정되어 있으며 수정이 되지 않으므로 편집을 해야 되는 경우 <그림 6-41>와 같이 새 페이지 추가(Add New)에서 기능개요 또는 화면설계 레이아웃을 선택하고 작성한다. 이전 [2. 화면 구성의 3\) 추가 폐이지](#)를 참조한다.



<그림 6-41. 화면 설계 레이아웃 페이지 추가 화면>

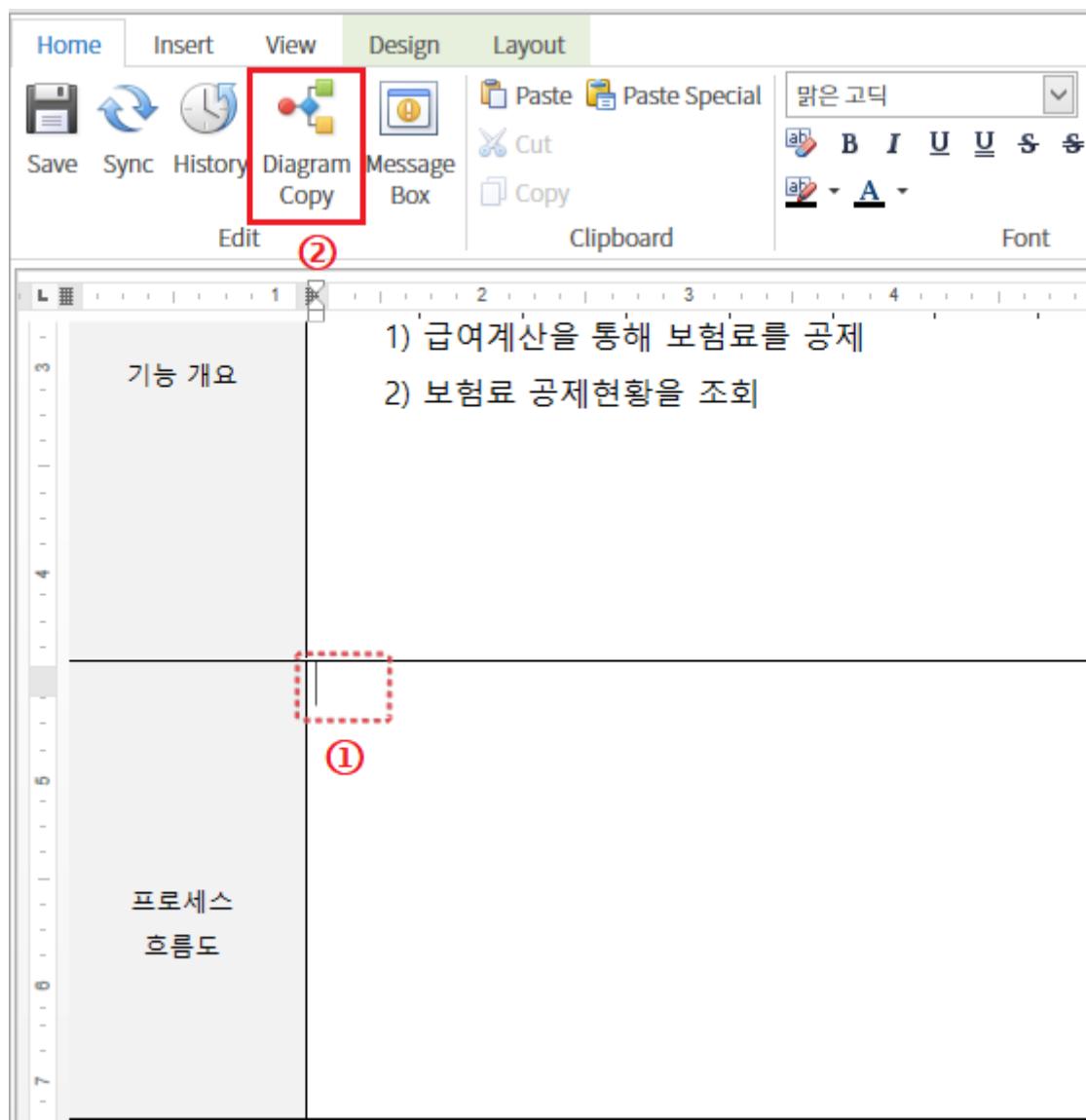
기능개요 레이아웃을 선택하고 기능 개요에 내용을 작성하고 프로세스 흐름도에 작성을 하기 위해 Diagram 탭에서 FD를 작성한 경우 pdf나 word 파일 생성 시 Editor 탭 내용만 생성되므로 Diagram 탭에서 먼저 그리고 나서 Editor 탭에서 복사를 해야 한다.

가) Diagram에서 흐름도를 작성



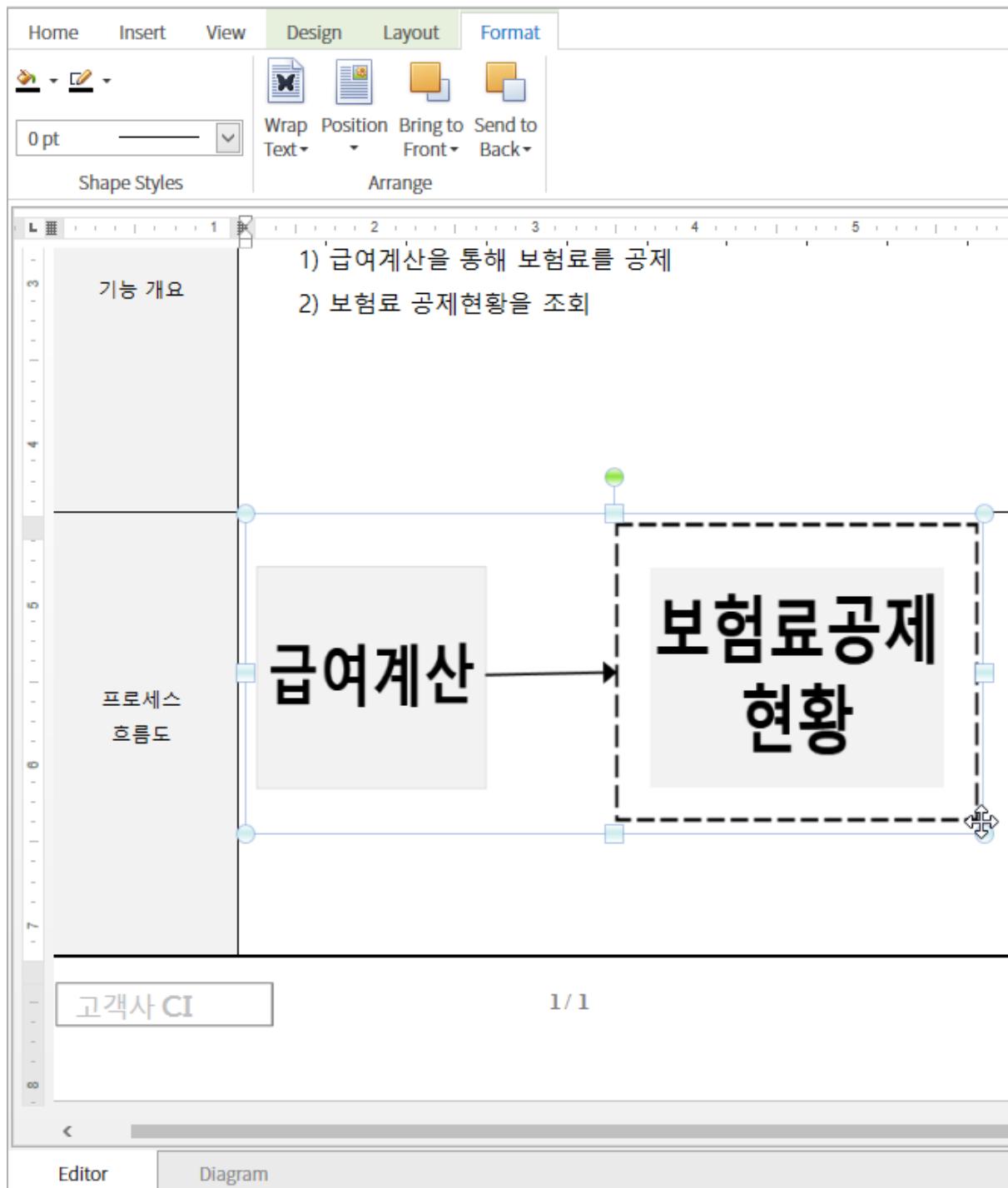
<그림 6-42 Diagram 탭에서 흐름도 작성>

나) 커서를 <그림 6-43> 처럼 놓고 메뉴 Home 탭에 Diagram Copy 를 클릭



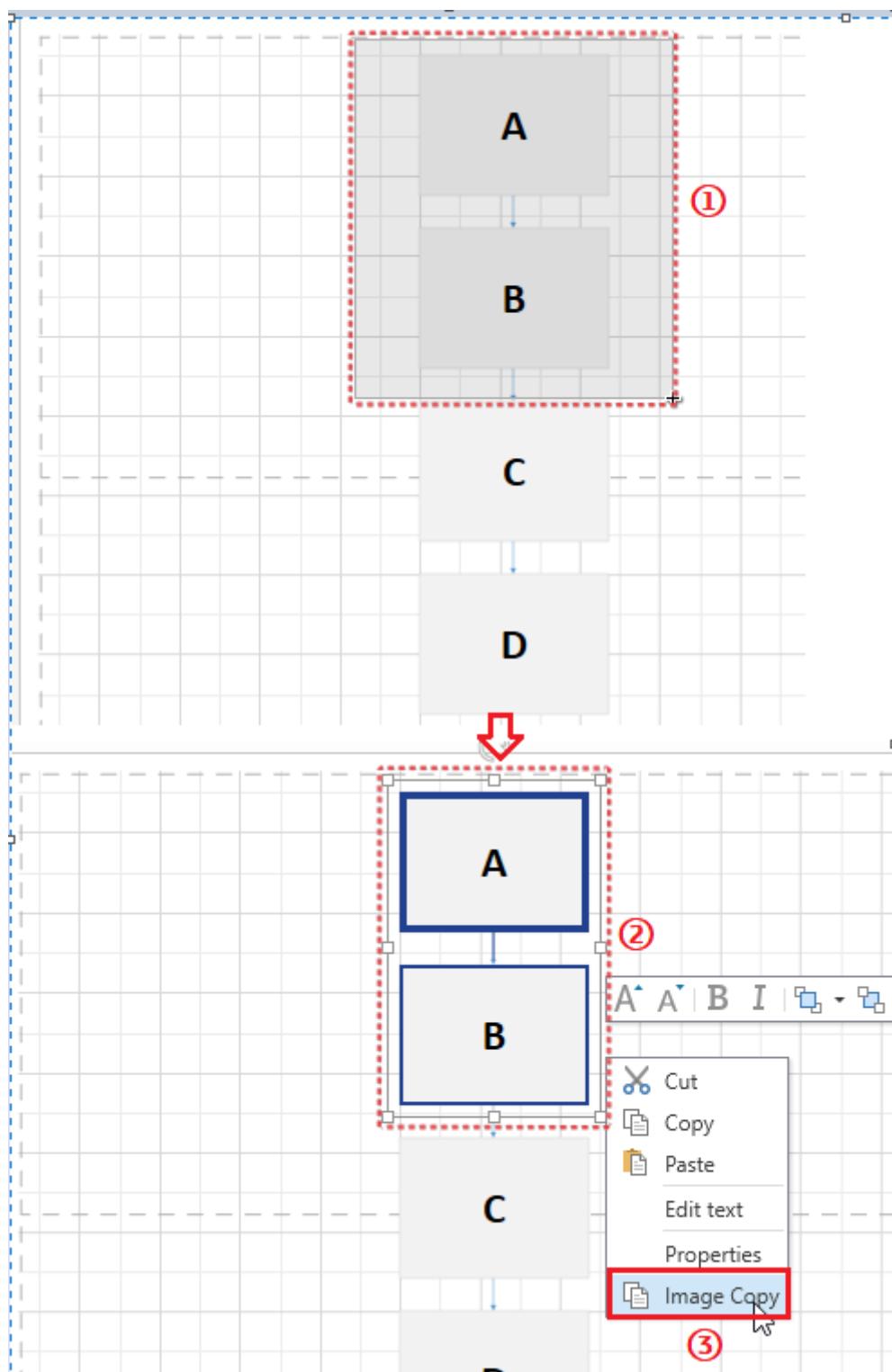
<그림 6-43. Diagram Copy 화면>

다) Diagram 탭에서 그린 흐름도가 그림처럼 복사되며 크기를 조정한다



<그림 6-44. Diagram Copy 화면>

라) 복합한 흐름도인 경우 도형을 분리하여 복사 가능. <그림 6-45> 처럼 ① 복사할 영역을 마우스로 선택하고 ② 선택이 되면 ③ 우클릭 후 Image Copy를 선택



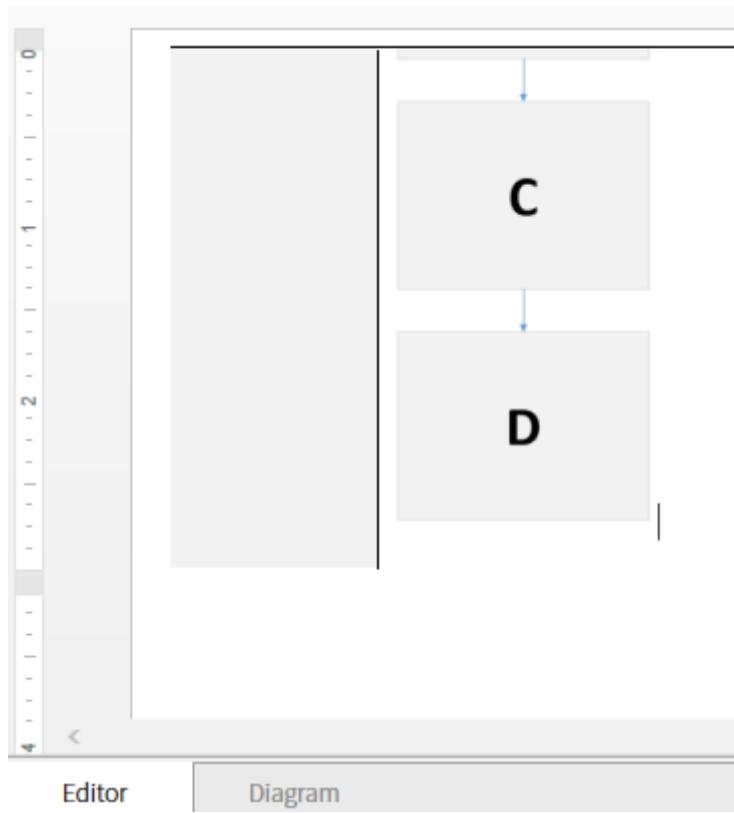
<그림 6-45. 흐름도 일부 부분 Image Copy 화면>

마) Editor 탭으로 가서 프로세스 흐름도 부분을 클릭하고 Ctrl+V 키를 눌러 붙여넣기.

| 모듈          | CL                                                                                                                                                                                                                                                                                                          | Component |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 메뉴 ID       | PBMPAR_Design                                                                                                                                                                                                                                                                                               |           |
| 기능개요 및 업무흐름 |                                                                                                                                                                                                                                                                                                             |           |
| 기능 개요       | <p><b>1. 프로세스 개요</b><br/>급여에서 공제된 보험료공제내역을 조회한다.</p>                                                                                                                                                                                                                                                        |           |
| 프로세스 흐름도    | <p><b>2. 프로세스 절차</b></p> <ol style="list-style-type: none"><li>1) 급여계산을 통해 보험료를 공제</li><li>2) 보험료 공제현황을 조회</li></ol> <pre>graph TD; subgraph A [A]; direction TB; A1(( )) --- A2[ ]; A2 --- A3(( )); end; subgraph B [B]; direction TB; B1[ ] --- B2(( )); B2 --- B3(( )); end; A1 --- B1; A3 --- B3;</pre> |           |

<그림 6-46. 흐름도 일부 부분 Image Copy 화면>

바) 나머지 도형 부분도 마찬가지로 Diagram 탭에서 영역 선택 후 우클릭하고 Image Copy를 선택하고 Editor 탭으로 와서 붙여 넣기

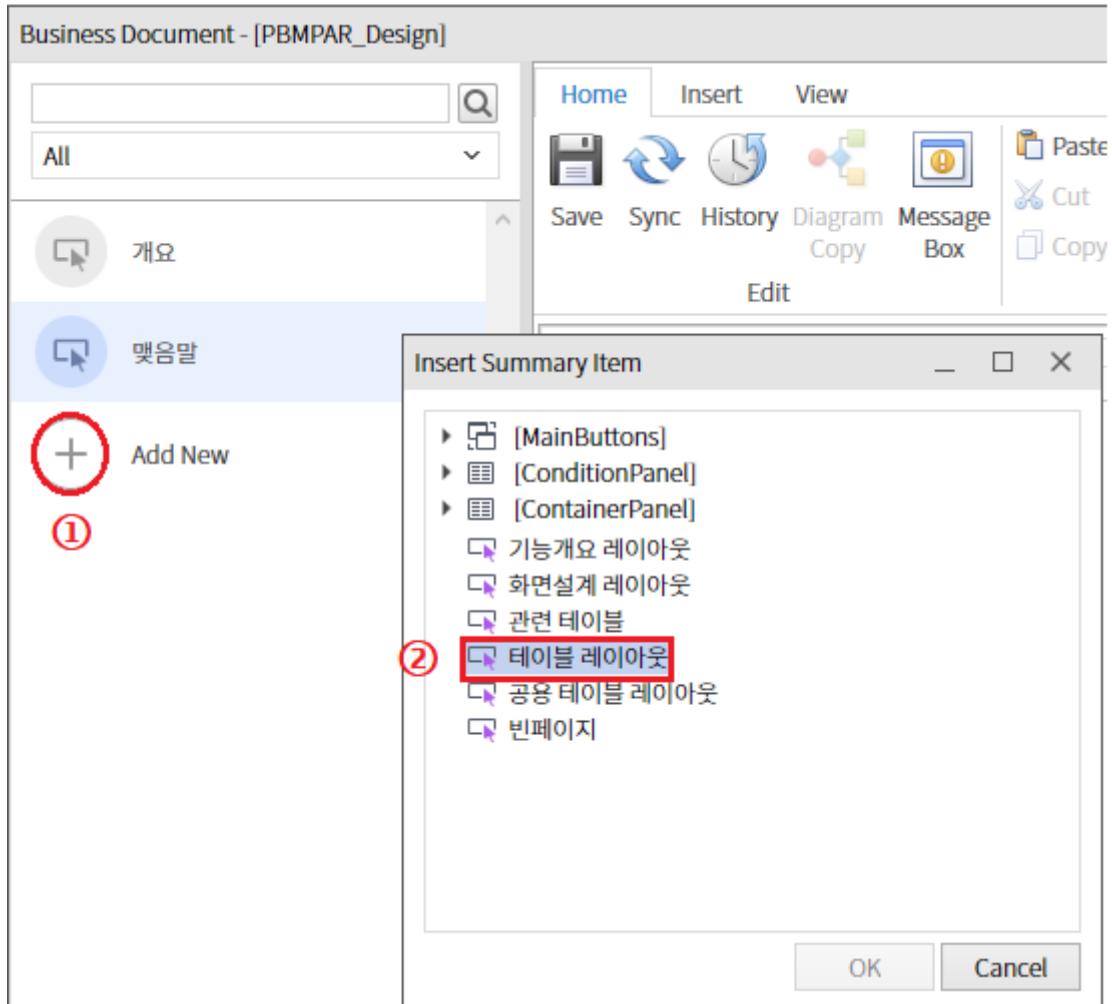


<그림 6-47. 다른 부분 복사 후 붙여넣기 화면>

이처럼 흐름도가 복잡할 경우 도형을 부분 복사하여 붙여넣기를 할 수 있다.

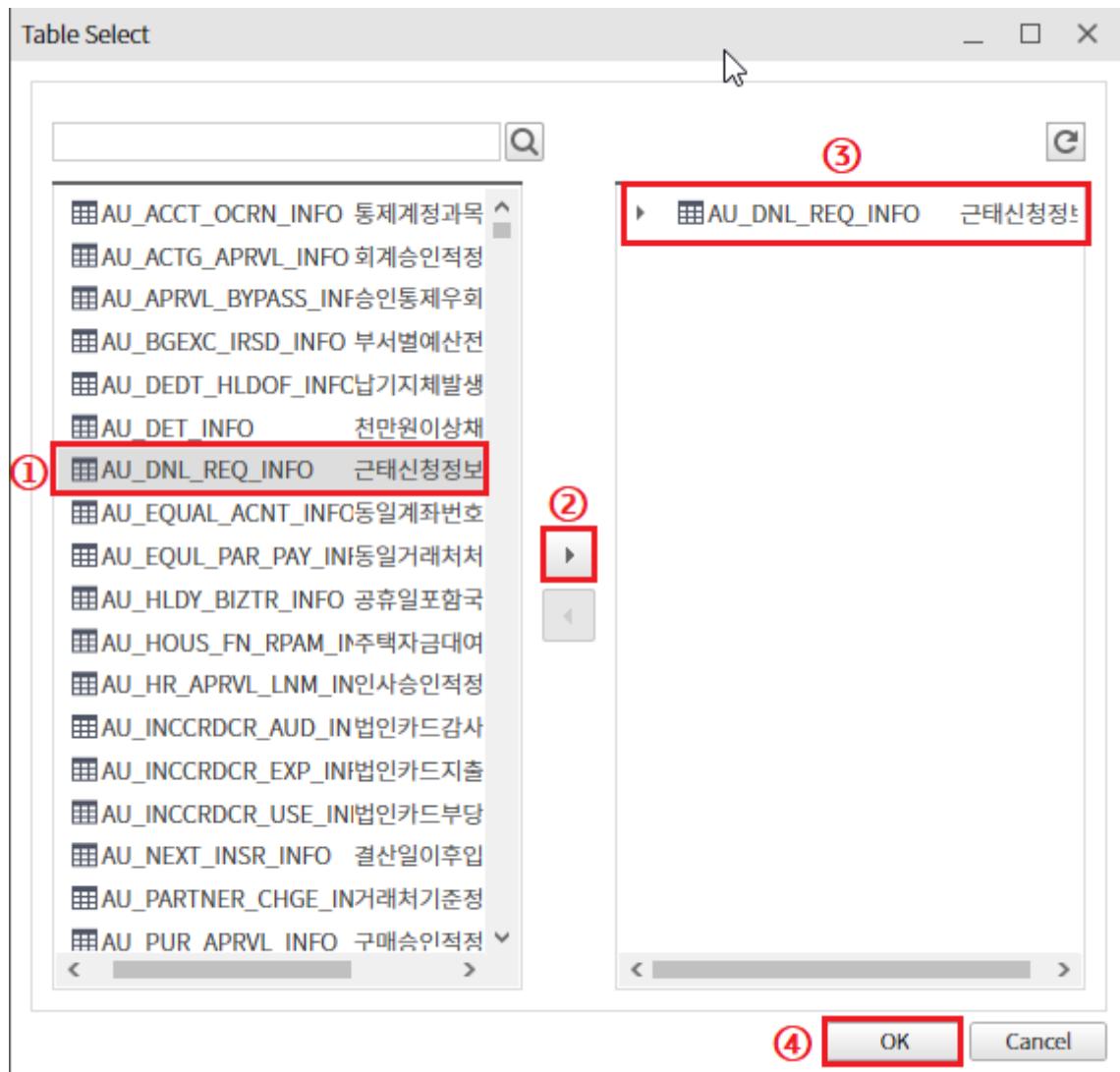
## ② 테이블 레이아웃

기존 엑셀에서 작성하여 첨부하던 테이블 레이아웃 양식을 제공하며 테이블 정보를 불러와서 선택하면 쉽고 빠르게 작성할 수 있다.



<그림 6-48. 테이블 레이아웃 선택 화면>

양식을 선택하면 Table Select 화면이 열리며 왼쪽은 테이블 정보를 보여준다. ① 검색하거나 항목을 선택하고 ② 우측 버튼을 눌러 ③ 추가할 테이블 리스트로 이동하고 ④ OK 버튼을 누른다.



<그림 6-49. 테이블 선택 화면>

페이지 이름이 테이블 한글명으로 생성되고 Table 정보를 확인할 수 있다.

The screenshot shows a software interface for creating table layouts. On the left, there is a sidebar with the following options:

- 개요
- 화면설계 레이아웃
- 테이블 레이아웃
- 근태신청정보** (highlighted with a red box)
- 맺음말
- Add New

The main area is titled "TABLE LAYOUT" and displays the following table information:

| 모 률      | CL                |           |
|----------|-------------------|-----------|
| TABLE ID | AU_DNL_REQ_INFO   |           |
| No       | 컬럼ID              | 컬럼명       |
| 1        | COMPANY_CD        | 회사코드      |
| 2        | GTHRNG_DTS        | 수집일시      |
| 3        | REQ_BWRK_DT       | 신청근무일     |
| 4        | DNL_APLT_EMP_NO   | 근태신청자사원번호 |
| 5        | REQ_BWRK_TM       | 신청근무시간    |
| 6        | DNL_APLT_NM       | 근태신청자명    |
| 7        | DNL_REQ_REASON_CD | 근태신청사유코드  |
| 8        | DNL_REQ_REASON_NM | 근태신청사유명   |

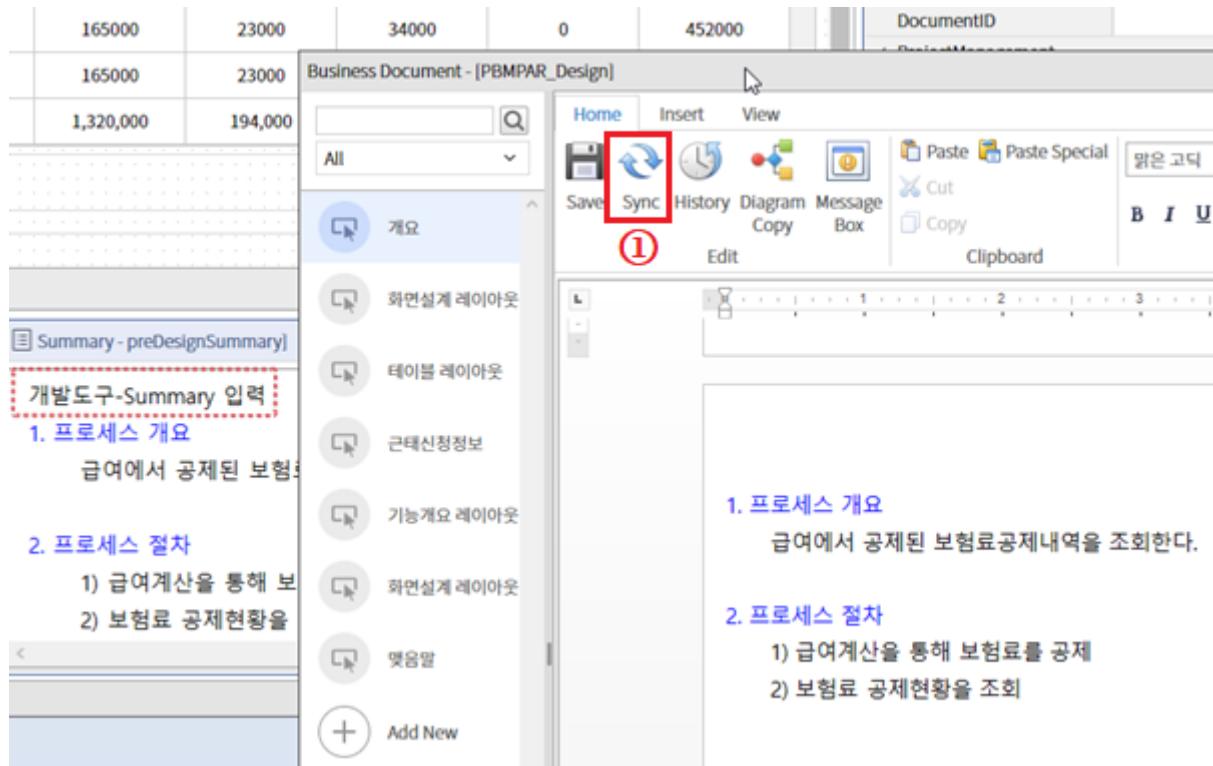
<그림 6-50. 테이블 레이아웃 페이지 생성 화면>

이 외에도 추가할 수 있는 양식에는 관련 테이블, 공용 테이블 레이아웃, 빈페이지가 있다.

## 5) 동기화 및 저장

### ① 개발 도구 동기화

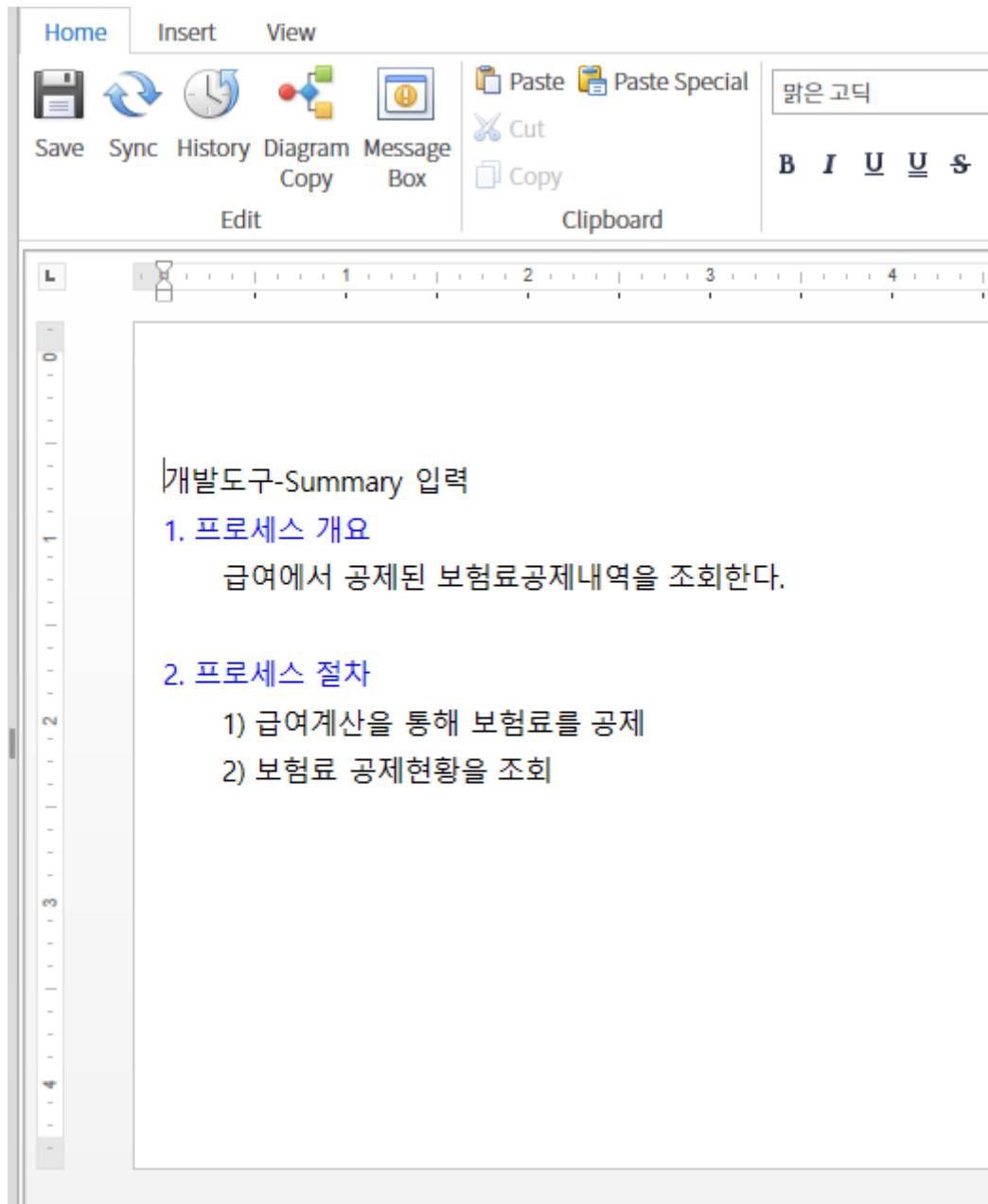
설계 도구는 모달리스<sup>1</sup> 형태로 구현되어 있으며 개발 도구의 Summary에서 작성한 내용을 설계도구에서 불러오기 위해서는 Home의 Sync 버튼을 클릭한다.



<그림 6-51. 설계 도구 Sync 툴바 화면>

<sup>1</sup> 모달리스(Modless)는 모달(Modal)과 다르게 메인창과 동시에 제어가 가능하다

개발 도구 Summary 에서 입력한 부분이 반영된다.



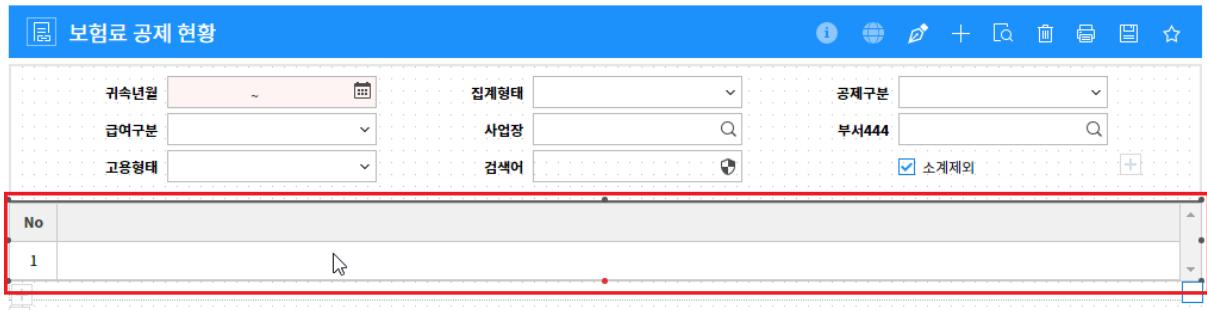
<그림 6-52. 개발도구 Summary 내용 반영>

## ② 저장

이전의 경우 설계 도구 Save 버튼을 클릭했을 때 팝업 화면처럼 임시 저장이 되고 반드시 개발 도구의 저장 버튼을 눌러야 파일에 저장되었으나 최근에는 Save 버튼을 눌렀을 때는 파일 (ewd)에도 저장되도록 변경하였다. 또한 오른쪽 하단 OK나 Cancel 버튼을 클릭했을 때 저장할지 물어보는 메세지박스가 항상 표시되어 재확인을 한다.

## 6) 개발 도구 그리드 데이터 입력 기능

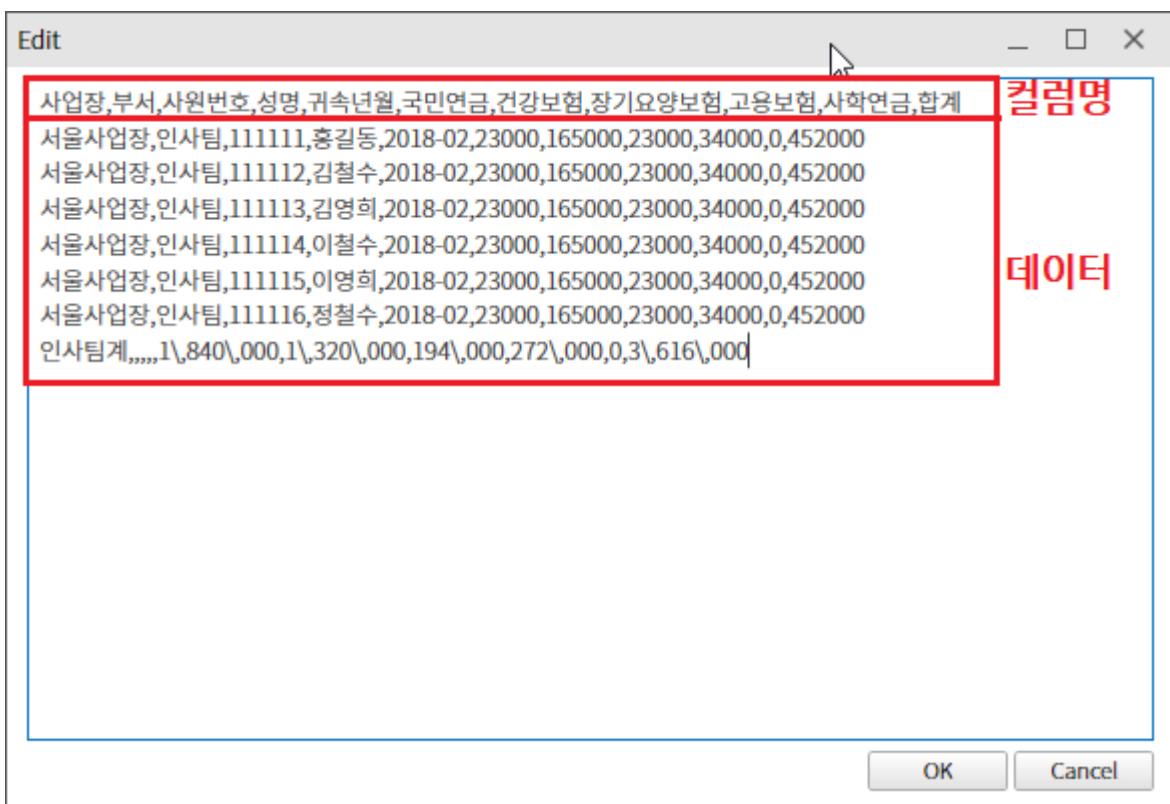
개발 도구에서 발사믹 목업(balsamiq mockups) 툴 처럼 그리드를 더블 클릭하고 데이터를 입력할 수 있는 기능이 있다. 그리드를 선택하고 더블 클릭한다.



<그림 6-53. 개발 도구에서 그리드 선택 화면>

에디터 화면이 열리면 컬럼명과 데이터를 입력한다.

- 맨 첫 행은 컬럼명이며 컬럼은 ,(콤마)로 구분 2행 부터는 데이터 입력
- 데이터에 ,(콤마)를 입력할 경우 ₩,을 입력 예) 2,000 입력 시 2₩,000



<그림 6-54. Edit 화면에서 컬럼명과 데이터 입력 화면>

OK 버튼을 누르면 데이터소스가 자동으로 생성된다. 데이터소스 탭을 누르면 자동으로 생성된 데이터소스를 확인할 수 있다.

The screenshot shows the 'DataSource' configuration interface. At the top right, it says 'dataSource1'. Below it, there are four main sections:

- General**:

|               |                              |
|---------------|------------------------------|
| Datasource ID | dataSource1                  |
| Grid Type     | Grid                         |
| Aggregate col | <input type="checkbox"/> Use |
| Group Col     | <input type="checkbox"/> Use |
| User Group    | <input type="checkbox"/> Use |
| IsPost        | <input type="checkbox"/> Use |
| Paging        | false                        |
| PagingCount   | 500                          |
- Column Schema**: A table showing columns with checkboxes for modification:

|                          | Description | Column Name | Data Type | Modify                   |
|--------------------------|-------------|-------------|-----------|--------------------------|
| <input type="checkbox"/> | 사업장         | 사업장         | string    | <input type="checkbox"/> |
| <input type="checkbox"/> | 부서          | 부서          | string    | <input type="checkbox"/> |
| <input type="checkbox"/> | 사원번호        | 사원번호        | string    | <input type="checkbox"/> |
| <input type="checkbox"/> | 성명          | 성명          | string    | <input type="checkbox"/> |
| <input type="checkbox"/> | 귀속년월        | 귀속년월        | string    | <input type="checkbox"/> |
| <input type="checkbox"/> | 국민연금        | 국민연금        | string    | <input type="checkbox"/> |
| <input type="checkbox"/> | 건강보험        | 건강보험        | string    | <input type="checkbox"/> |
| <input type="checkbox"/> | 장기요양보험      | 장기요양보험      | string    | <input type="checkbox"/> |
| <input type="checkbox"/> | 고용보험        | 고용보험        | string    | <input type="checkbox"/> |
| <input type="checkbox"/> | 사학연금        | 사학연금        | string    | <input type="checkbox"/> |
| <input type="checkbox"/> | 합계          | 합계          | string    | <input type="checkbox"/> |
- Data**:

|               |                                                                              |
|---------------|------------------------------------------------------------------------------|
| Data Provider | Local Data                                                                   |
| Data          | <input type="button"/> <input type="button"/> <input type="button"/> </> </> |
- Event**:

|       |                            |
|-------|----------------------------|
| error | <input type="button"/> </> |
|-------|----------------------------|

At the bottom, there are tabs: Designer (selected), Datasource, Script, CSS, and HTML.

<그림 6-55. 자동으로 생성된 데이터소스 화면>

또한 그리드 화면에는 컬럼과 데이터가 자동으로 입력되어 있다.

The screenshot shows a Kendo UI Grid interface titled "보험료 공제 현황". At the top, there are several search and filter input fields: "귀속년월" (Year Month), "집계형태" (Aggregation Type), "공제구분" (Exemption Category), "급여구분" (Payroll Category), "사업장" (Workplace), "부서444" (Department), "고용형태" (Employment Type), "검색어" (Search Term), and a checkbox for "소계제" (Subtotal). Below the search bar is a table with the following data:

| No | 사업장   | 부서  | 사원번호   | 성명  | 귀속년월    | 국민연금      | 건강보험      | 장기요양…   | 고용보험    |
|----|-------|-----|--------|-----|---------|-----------|-----------|---------|---------|
| 1  | 서울사업장 | 인사팀 | 111111 | 홍길동 | 2018-02 | 23000     | 165000    | 23000   | 34000   |
| 2  | 서울사업장 | 인사팀 | 111112 | 김철수 | 2018-02 | 23000     | 165000    | 23000   | 34000   |
| 3  | 서울사업장 | 인사팀 | 111113 | 김영희 | 2018-02 | 23000     | 165000    | 23000   | 34000   |
| 4  | 서울사업장 | 인사팀 | 111114 | 이철수 | 2018-02 | 23000     | 165000    | 23000   | 34000   |
| 5  | 서울사업장 | 인사팀 | 111115 | 이영희 | 2018-02 | 23000     | 165000    | 23000   | 34000   |
| 6  | 서울사업장 | 인사팀 | 111116 | 정철수 | 2018-02 | 23000     | 165000    | 23000   | 34000   |
| 7  | 인사팀계  |     |        |     |         | 1,840,000 | 1,320,000 | 194,000 | 272,000 |

At the bottom of the grid, there are three additional rows with plus signs (+) and a toolbar with tabs for "Designer", "Datasource", "Script", "CSS", and "HTML".

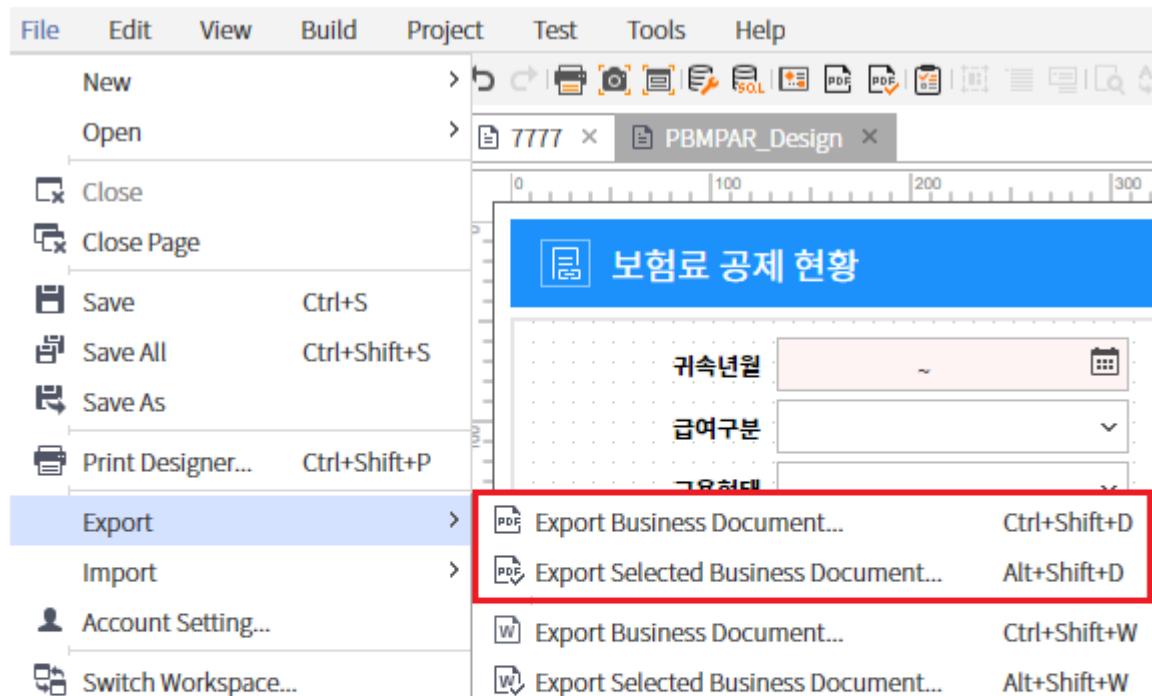
<그림 6-56. 그리드 컬럼 및 데이터 입력된 화면

## 4. 설계 문서 파일 생성

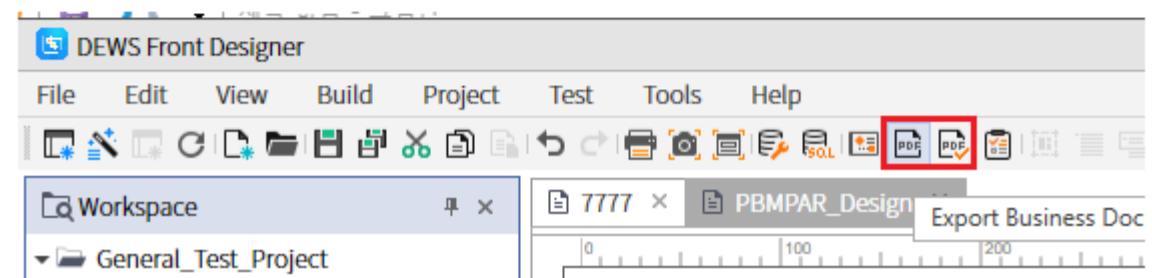
설계 문서 작성 완료 후 파일로 생성할 수 있으며 PDF와 워드(docx)로 생성된다.

### 1) PDF 파일

메뉴의 File > Export > Export Business Document , Export Selected Business Document 또는 툴바의 Export Business Document, , Export Selected Business Document를 실행한다.

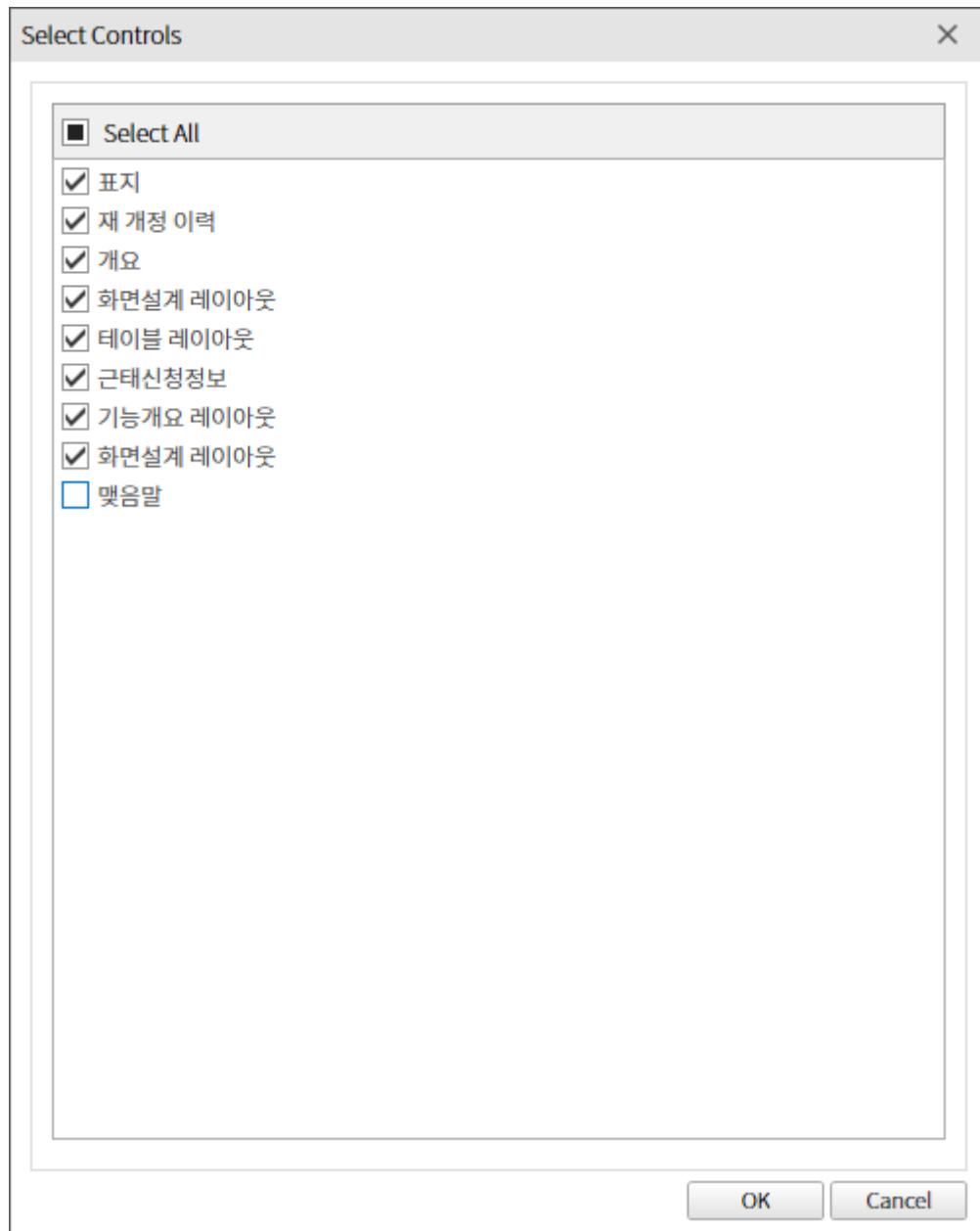


OR



<그림 6-57. PDF 파일 생성 실행 화면>

Export Business Document 는 전체 문서를 PDF로 생성하며, Export Selected Business Document 는 사용자가 선택한 페이지만 PDF 파일로 생성할 수 있다. 주로 맷음말이 필요없는 경우 선택 화면에서 체크를 해제하고 생성한다.

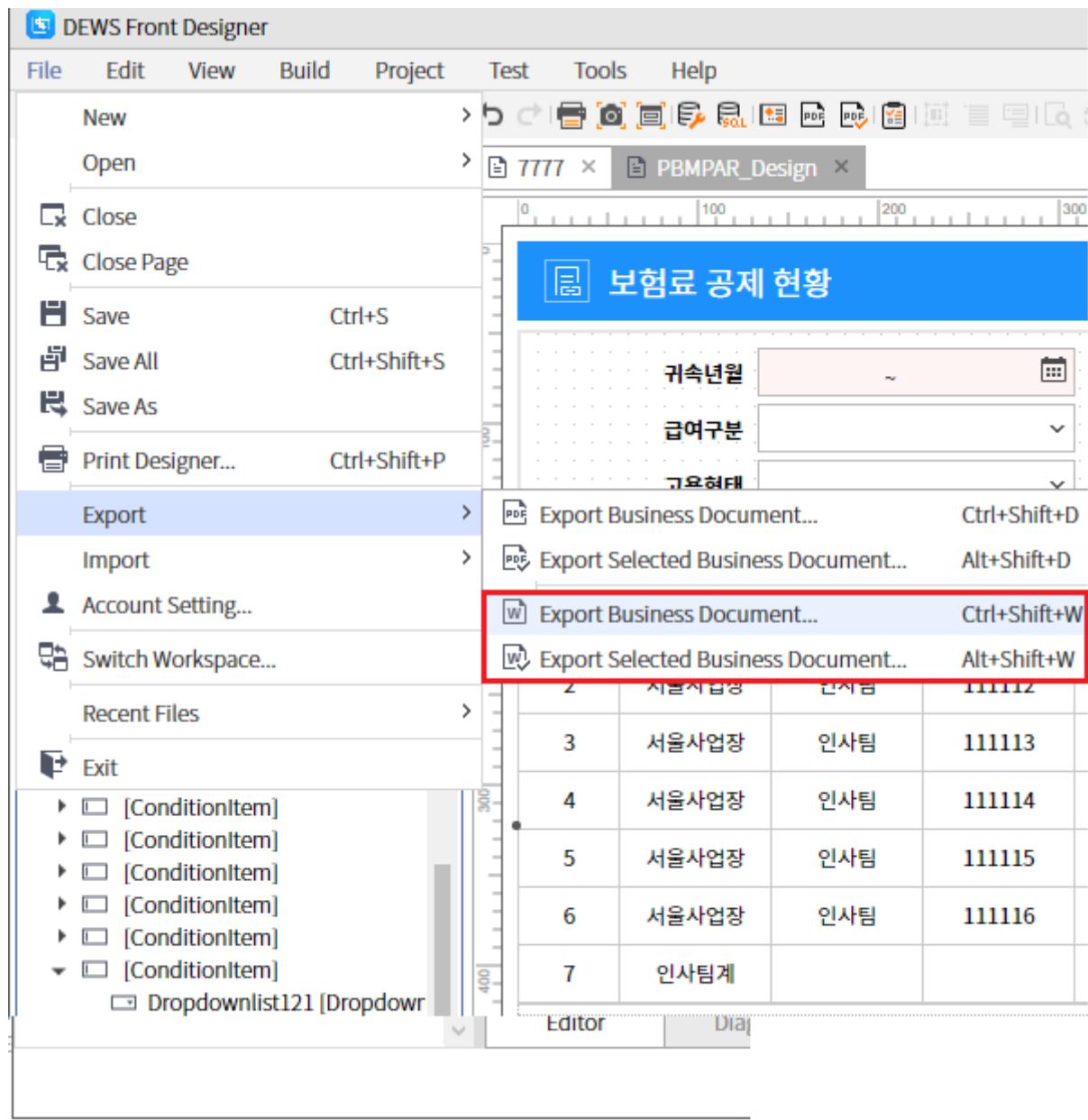


<그림 6-58. 사용자 선택 페이지 파일 생성 화면>

## 2) 워드 파일

PDF 파일은 수정이 불가하거나 수정을 해야 되는 경우 워드 파일(doc)로 생성하여 편집이 가능하다. 툴바에는 없으며 메뉴에서만 실행이 가능하다.

실행은 메뉴의 File > Export > Export Business Document , Export Selected Business Document 를 실행한다.



<그림 6-59. 워드 파일 생성 실행 메뉴 화면>

## [제7장] Wizard Project

소개 및 실행

메뉴 개발 예제

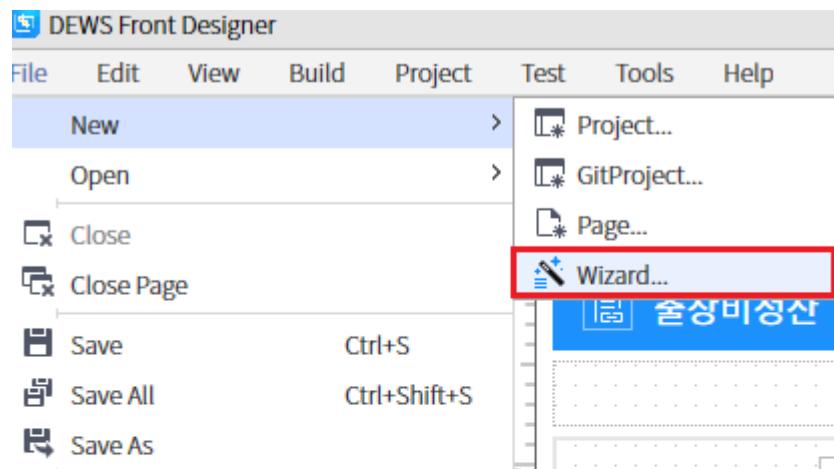
## 1. 소개 및 실행

Wizard Project는 마법사를 이용하여 메뉴를 개발하는 방법이며 Step by Step으로 화면을 보다 쉽고 빠르게 만들 수 있다.

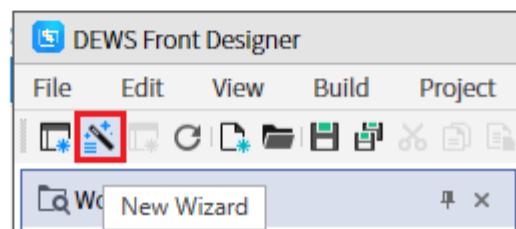
설계 단계에서는 실제 개발 구현 이전이므로 데이터 입력이나 조회, 수정, 삭제 등의 데이터 처리 작업은 할 수 없으나 Wizard를 이용할 경우 자동으로 생성된 백엔드 서비스로 미리보기에서 데이터 처리가 가능하다.

개발 구현 단계에서도 백엔드 서비스 개발 후 Wizard Project에서 제공하는 디자인 템플릿을 선택하고 서비스 API를 검색하여 빠르게 개발이 가능하다.□

Wizard Project 실행은 메뉴의 New > Wizard 나 이미지 아이콘 툴바의 New Wizard 클릭해서 실행한다.

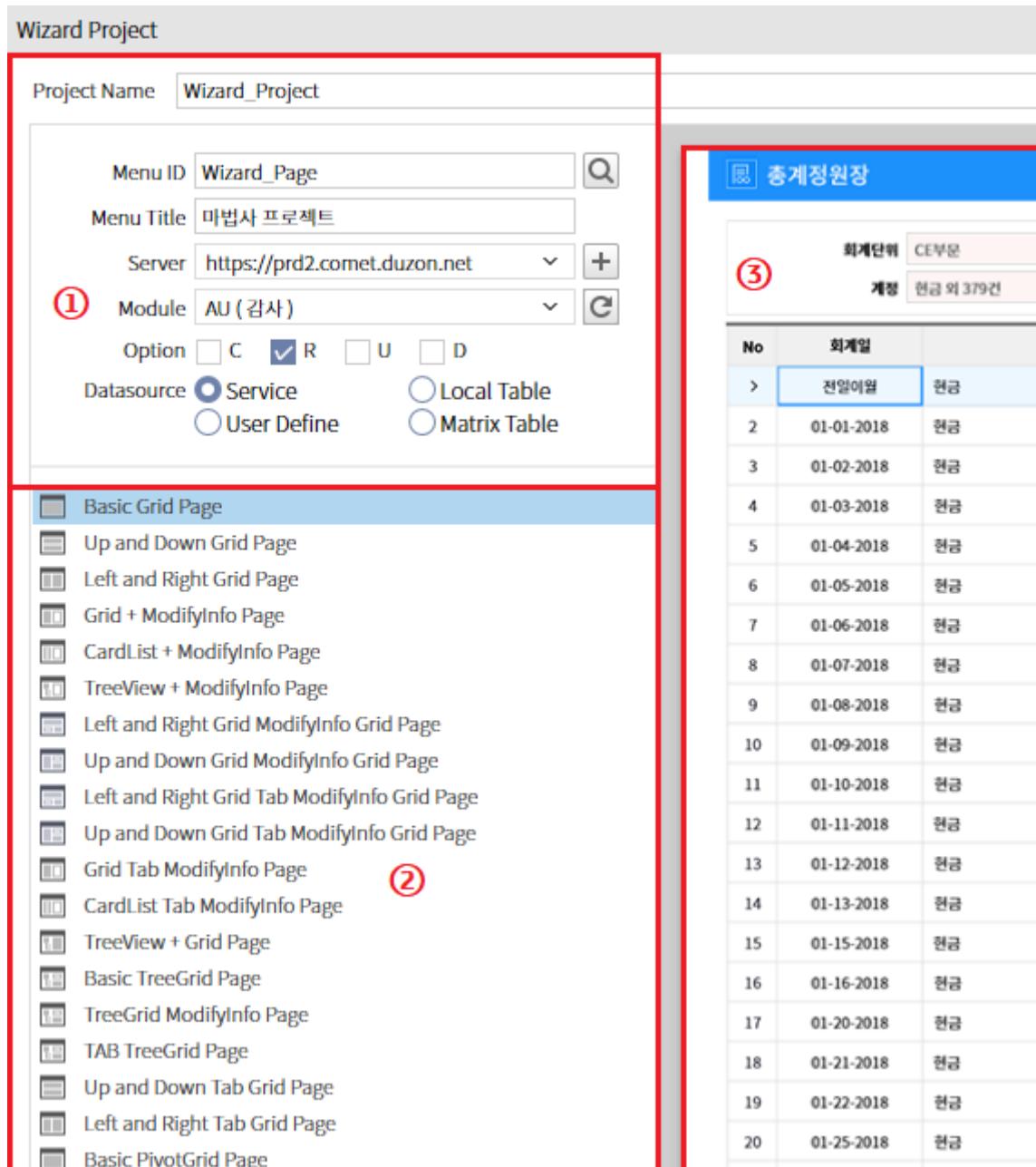


OR



<그림 7-1. Wizard Project 실행>

Wizard를 실행한 화면이다. 크게 3부분으로 되어 있으며 ①은 생성할 프로젝트 이름, 페이지, 문서 제목, 미리보기 서버, 모듈, 메뉴 옵션 선택(C:Create, R:Read, U:Update, D:Delete), 데이터소스 선택(Service, Local Table, User Define, Matrix Table) 등을 입력하거나 선택하는 영역이며, ②는 DFD에서 제공하는 디자인 템플릿을 선택하는 영역이며 ③은 ②에서 선택한 템플릿의 미리보기 화면이다.



<그림 7-2. Wizard Project 처음 화면>

## 2. 메뉴 개발 예제

Wizard Project의 Option은 만들 메뉴 특성에 따라 선택한다. 조회 메뉴의 경우 R에만 체크하고 관리 메뉴 즉 입력, 조회, 삭제, 수정 등의 데이터 처리를 하는 메뉴인 경우 CRUD 모두 체크를 한다.

Datasource는 어떤 데이터를 사용할 것인지를 지정한다. Service, Local Table, User Define, Matrix Table이 있다.

| Datasource   | 설명                                                                            |
|--------------|-------------------------------------------------------------------------------|
| Service      | - 개발자가 사용<br>- 백엔드 서비스 API가 개발되어 있는 경우 선택                                     |
| Local Table  | - 표준 또는 기존 사용자가 만든 테이블의 컬럼을 선택하여 로컬 데이터로 생성<br>- 사용자 정의 테이블을 연결한 경우 데이터 처리 가능 |
| User Define  | - 사용자가 만든 쿼리 이용                                                               |
| Matrix Table | - 공용테이블을 이용                                                                   |

<표 7-1. Datasource 유형 간단 설명>

### 1) Service

Service는 개발자가 사용하는 것으로 먼저 백엔드 서비스 API 개발 후에 진행할 수 있으며 마법사에서 서비스 API를 검색하고 정보를 가져와서 메뉴를 개발한다.

#### ① 만들 메뉴 화면

2단 그리드 구조(Master-Sub 그리드)로 된 메뉴를 Service 유형을 선택하여 만들어 보면 조회 메뉴이다.

이동평균단가조회

| No | 품목코드       | 품목명              | 기준일자       |
|----|------------|------------------|------------|
| 55 | SPARE00001 | SPARE PART 1     | 2019-01-17 |
| >  | SPARE00010 | Spare Parts #001 | 2020-03-30 |
| 57 | SPARE00020 | Spare Parts #002 | 2018-11-09 |

이력조회

| No | 기준일자       | 계산순번 | 재고수량      | 재고금액           |
|----|------------|------|-----------|----------------|
| >  | 2020-03-02 | 1    | 1,000,102 | 10,001,020,000 |

<그림 7-3. Service 유형으로 만들 메뉴 화면>

## ② 항목 입력 및 유형 선택

- Wizard를 실행하고 프로젝트 정보를 입력하고 조회 메뉴 이므로 Options은 R에 체크한다. Datasource는 Service를 선택하고 아래 화면 템플릿은 Up and Down Grid Page를 선택한다.

Wizard Project

Project Name: 이동평균단가조회

|                                                                                                                                                                                                                                                                                                  |                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| Menu ID: MLEMEIA00201                                                                                                                                                                                                                                                                            | Menu Title: 이동평균단가조회 |
| Server: https://dev.comet.duzon.net                                                                                                                                                                                                                                                              | + Module: AU (감사)    |
| Option: <input type="checkbox"/> C <input checked="" type="checkbox"/> R <input type="checkbox"/> U <input type="checkbox"/> D<br>Datasource: <input checked="" type="radio"/> Service <input type="radio"/> Local Table<br><input type="radio"/> User Define <input type="radio"/> Matrix Table |                      |
| <input type="checkbox"/> Basic Grid Page<br><input checked="" type="checkbox"/> Up and Down Grid Page<br><input type="checkbox"/> Left and Right Grid Page<br><input type="checkbox"/> Grid + ModifyInfo Page<br><input type="checkbox"/> CardList + ModifyInfo Page                             |                      |

이동평균단가조회

| No | 품목코드   |
|----|--------|
| 1  | 10000  |
| >  | 100000 |
| 3  | 100001 |
| 4  | 100002 |
| 5  | 100003 |
| 6  | 121022 |
| 7  | 124047 |

이력조회

<그림 7-4. 프로젝트 항목 입력 및 유형 선택 화면>

### ③ Back-end 서비스 API 검색

다음으로 이미 구현한 Bank-end 서비스 API를 검색하는 단계이다. 검색 조건은 기본적으로 URL Description이 선택되어 있다. URL Description은 Back-end 서비스에서 정의한 annotation 중 @DzApi 정보 중 desc 부분을 검색한다.

조회 결과 항목을 더블클릭하면 서비스 정보, 필요한 파라미터, 스키마(필드) 정보를 불러오며 처음 더블클릭한 항목은 마스터 그리드에 매핑할 데이터소스가 되며, 두번째는 디테일(서브) 그리드에 매핑할 데이터소스로 지정된다.

The screenshot shows the 'Wizard Project' interface with the following details:

- URL Description:** 이동평균
- Preview:** Parameter
- Module:** 관리회계
- URL Description:** 이동평균등록 - 마스터
- Service Description:** 기준정보관리
- Parameter:**

|          |       |             |        |
|----------|-------|-------------|--------|
| PLANT_CD | 공장코드  | QueryString | string |
| LAST_YN  | 최종값여부 | QueryString | string |
| ITEM_CD  | 품목코드  | QueryString | string |
- Module:** 관리회계
- URL Description:** 이동평균등록 - 공장리스트
- Service Description:** 기준정보관리
- Parameter:** (empty)
- Module:** 관리회계
- URL Description:** 이동평균단가조회 상세내역 조회
- Service Description:** 정보분석
- Parameter:** (empty)
- Module:** 관리회계
- URL Description:** 이동평균단가조회 조회
- Service Description:** 정보분석
- Parameter:** (empty)
- Selected Service:**

| Master | GridID | DataSourceID | URL Description  |
|--------|--------|--------------|------------------|
| Master | grid1  | dataSource1  | 이동평균단가조회 조회      |
| Sub    | grid2  | dataSource2  | 이동평균단가조회 상세내역 조회 |
- Detail Info:**

|                        |                          |
|------------------------|--------------------------|
| Grid ID                | grid2                    |
| DataSource             | dataSource2              |
| Module                 | CO                       |
| Description            | 관리회계                     |
| Service                | MaterialLedgerMIAService |
| Service Description    | 정보분석                     |
| URL Pattern            | mlemia00200_list_detail  |
| URLPattern Description | 이동평균단가조회 상세내역 조회         |
- Note:** 조회조건과 마스터 그리드와 디테일 그리드가 상호로존재하는 페이지를 생성합니다.
- Buttons:** Prev, Next

<그림 7-5. Back-end 서비스 API 검색 화면>

#### ④ 조회 조건 컨트롤 선택

다음은 조회 조건 컨트롤을 선택하는 단계이다. 이전 ③단계에서 Parameter 이름과 조회 조건 컨트롤 id와 비슷한 이름이 있을 경우 자동으로 오른쪽 조회 조건 리스트에 추가되어 있다. 필요 없는 경우 더블클릭하거나 ◀ 버튼을 눌러 리스트에서 삭제한다.

조회 조건은 Condition Explorer에 있는 조건들을 검색하고 결과 항목을 더블클릭하거나 가운데

- ▶ 버튼을 눌러 오른쪽 조회 조건 리스트로 이동한다.

| Module | Title            | Parameter                      | Description                        |
|--------|------------------|--------------------------------|------------------------------------|
| SD     | 영업품목(멀티코드)       |                                | 영업품목 - 멀티코드(H_SD_MA_ITEM_C)        |
| SD     | 영업품목(코드피커)       |                                | 영업품목 - 코드피커(H_SD_MA_ITEM_C)        |
| PU     | 구매품목 (멀티코드)      | partner_cd, plant_cd, puror... | 구매품목 도움창 - 멀티코드(H_PU_MA_ITEM_S)    |
| PU     | 구매품목 (코드피커)      | partner_cd, plant_cd, puror... | 구매품목 도움창 - 코드피커(H_PU_MA_ITEM_S)    |
| PS     | 프로젝트BOM(멀티코드)    |                                | 프로젝트BOM도움창 - 멀티코드(H_PS_BOM_MST_C)  |
| PS     | 프로젝트BOM(코드피커)    |                                | 프로젝트BOM도움창 - 코드피커(H_PS_BOM_MST_C)  |
| PP     | 공장품목 (멀티코드)      | plant_cd                       | 공장품목 도움창 - 멀티코드(H_PP_MA_PITEM_S01) |
| PP     | 공장품목 (코드피커)      | plant_cd                       | 공장품목 도움창 - 코드피커(H_PP_MA_PITEM_S01) |
| PP     | 공장품목 (멀티코드)      | plant_cd                       | 공장품목 도움창 - 멀티코드(H_PP_MA_PITEM_S)   |
| PP     | 공장품목 (코드피커)      | plant_cd                       | 공장품목 도움창 - 코드피커(H_PP_MA_PITEM_S)   |
| PP     | 공장(멀티코드)         |                                | 공장 - 멀티코드(H_PP_LTPPLANT_INFO_C)    |
| PP     | 공장(코드피커)         |                                | 공장 - 코드피커(H_PP_LTPPLANT_INFO_C)    |
| PP     | 설계변경BOM마스터(멀티코드) |                                | 설계변경BOM마스터 - 멀티코드(H_PP_ECM_INFO_C) |
| PP     | 설계변경BOM마스터(코드피커) |                                | 설계변경BOM마스터 - 코드피커(H_PP_ECM_INFO_C) |
| PP     | BOM자재목록(멀티코드)    |                                | BOM자재목록 - 멀티코드(H_PP_BOM_DTL_C2)    |
| PP     | BOM자재목록(코드피커)    |                                | BOM자재목록 - 코드피커(H_PP_BOM_DTL_C2)    |
| PP     | BOM자재목록(멀티코드)    |                                | BOM자재목록 - 멀티코드(H_PP_BOM_DTL_C1)    |
| PP     | BOM자재목록(코드피커)    |                                | BOM자재목록 - 코드피커(H_PP_BOM_DTL_C1)    |
| PP     | BOM마스터목록(멀티코드)   |                                | BOM마스터목록 - 멀티코드(H_PP_BOM_DTL_C0)   |
| PP     | BOM마스터목록(코드피커)   |                                | BOM마스터목록 - 코드피커(H_PP_BOM_DTL_C0)   |
| PM     | 품목(멀티코드)         |                                | 품목 - 멀티코드(H_PM_PITEM_C)            |
| PM     | 품목(코드피커)         |                                | 품목 - 코드피커(H_PM_PITEM_C)            |
| MA     | 공장(멀티코드)         | company_cd                     | 공장 도움창 - 멀티코드(H_MA_PLANT_MST_S)    |
| MA     | 공장(코드피커)         | company_cd                     | 공장 도움창 - 코드피커(H_MA_PLANT_MST_S)    |
| MA     | 공장품목(멀티코드)       | company_cd, plant_cd           | 공장품목 도움창 - 멀티코드(H_MA_PITEM_S)      |

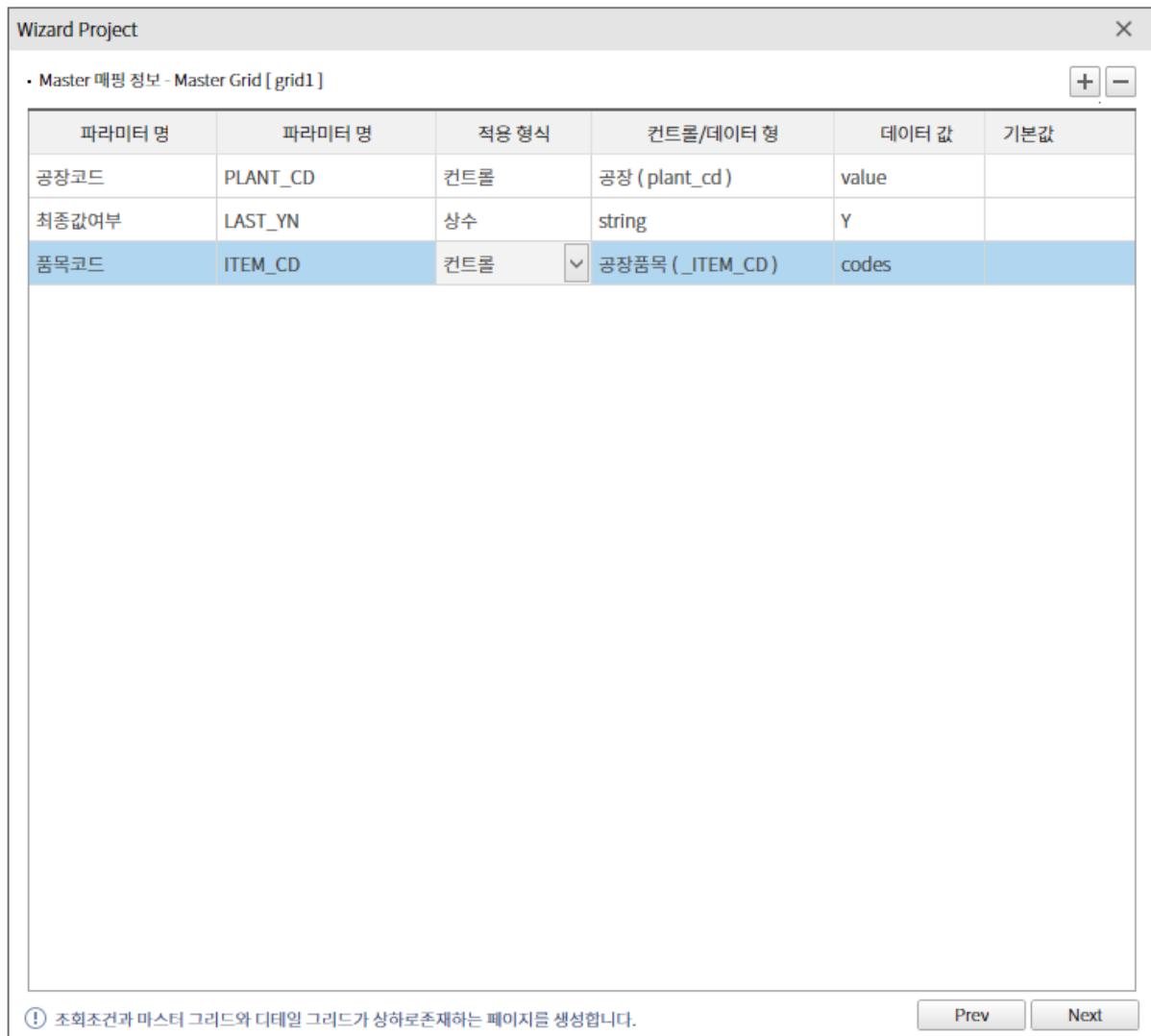
! 조회조건과 마스터 그리드와 디테일 그리드가 상호로존재하는 페이지를 생성합니다.

Prev Next

<그림 7-6. 조회 조건 컨트롤 선택 화면>

## ⑤ Master 그리드 매핑 설정

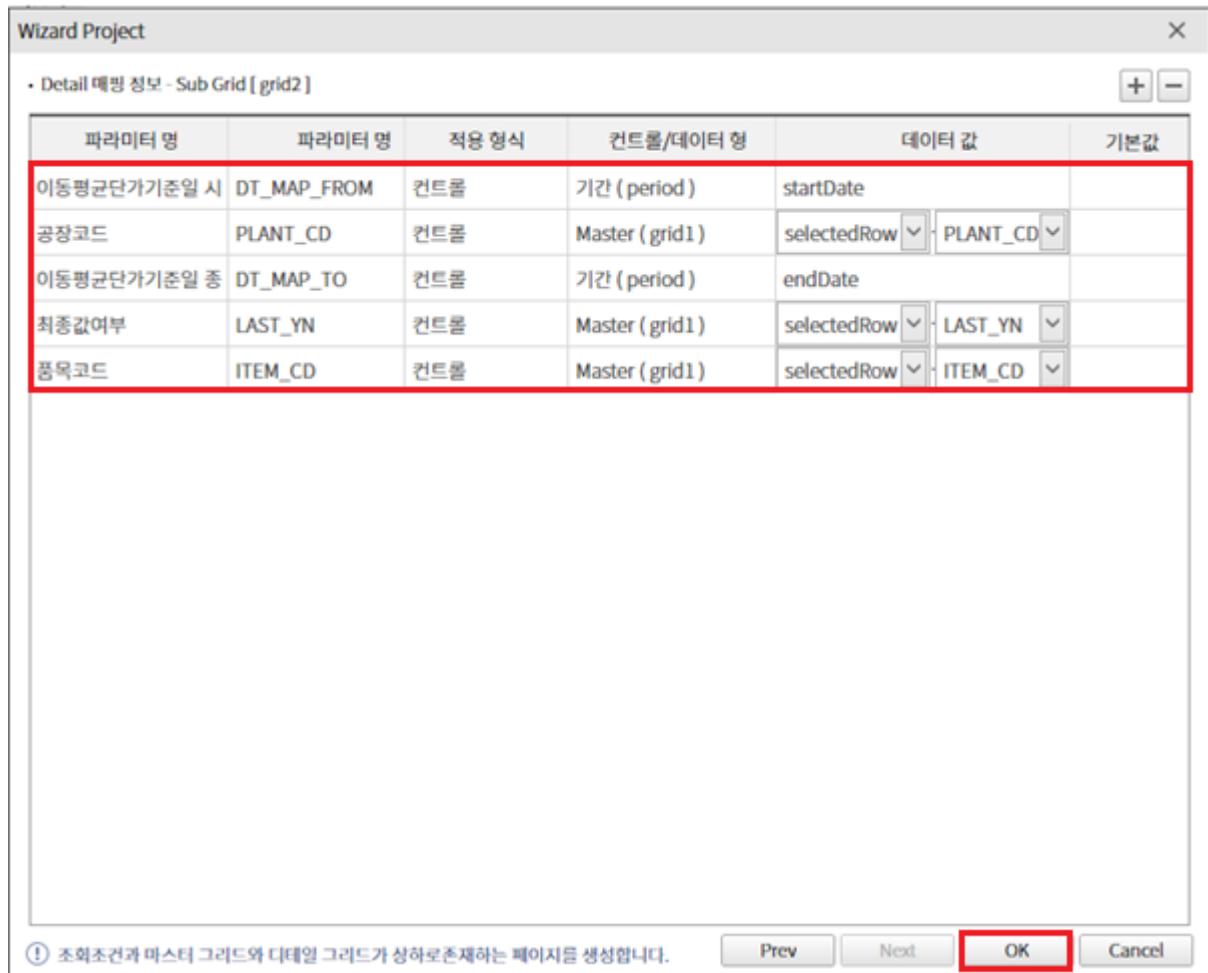
다음은 Master 그리드와 연결되는 데이터소스의 파라미터에 매핑할 조회 조건 컨트롤이나 상수를 설정하는 단계이다. ③의 오른쪽 Parameter와 조회 조건 컨트롤 id가 같거나 비슷한 경우 파라미터 값에 따라 자동으로 설정되어 있으며, 편집이 필요한 경우 수정을 하고 다음 단계로 넘어간다.



<그림 7-7. Master 그리드 매핑 화면

#### ⑥ Sub 그리드 매핑 설정 및 마법사 완료

다음은 Sub(Detail) 그리드와 연결되는 데이터소스의 파라미터에 매핑할 조회 조건 컨트롤이나 상수를 설정하는 단계이다. 마스터 그리드의 필드와 일치하는 Sub 그리드 필드가 있을 경우나 일자 파라미터의 경우 자동으로 매핑되어 있다.



<그림 7-8. Sub 그리드 매핑 화면 및 마법사 종료 화면>

마지막으로 OK 버튼을 눌러 마법사를 종료한다.

기본 화면이 완성되었으며 추가적으로 디테일 설정이 필요한 경우 화면을 수정하거나 스크립트를 작성한다.



<그림 7-9. Service로 완성된 화면>

## 2) Local Table

Local Table은 표준 테이블이나 기존 사용자가 만든 테이블의 컬럼을 선택하여 로컬 데이터로 생성하거나 사용자가 만든 테이블을 선택할 경우 데이터 처리 즉 추가, 삭제, 저장이 가능한 메뉴를 개발할 수 있다.

### ① 만들 메뉴 화면

2단 그리드 구조(Master-Sub 그리드)로 된 메뉴를 Local Table 유형을 선택하여 만들어 보며, 데이터 처리가 되는 CRUD 메뉴이다.

| No | 부서  | 부서명 |
|----|-----|-----|
| >  | 123 | 신규  |

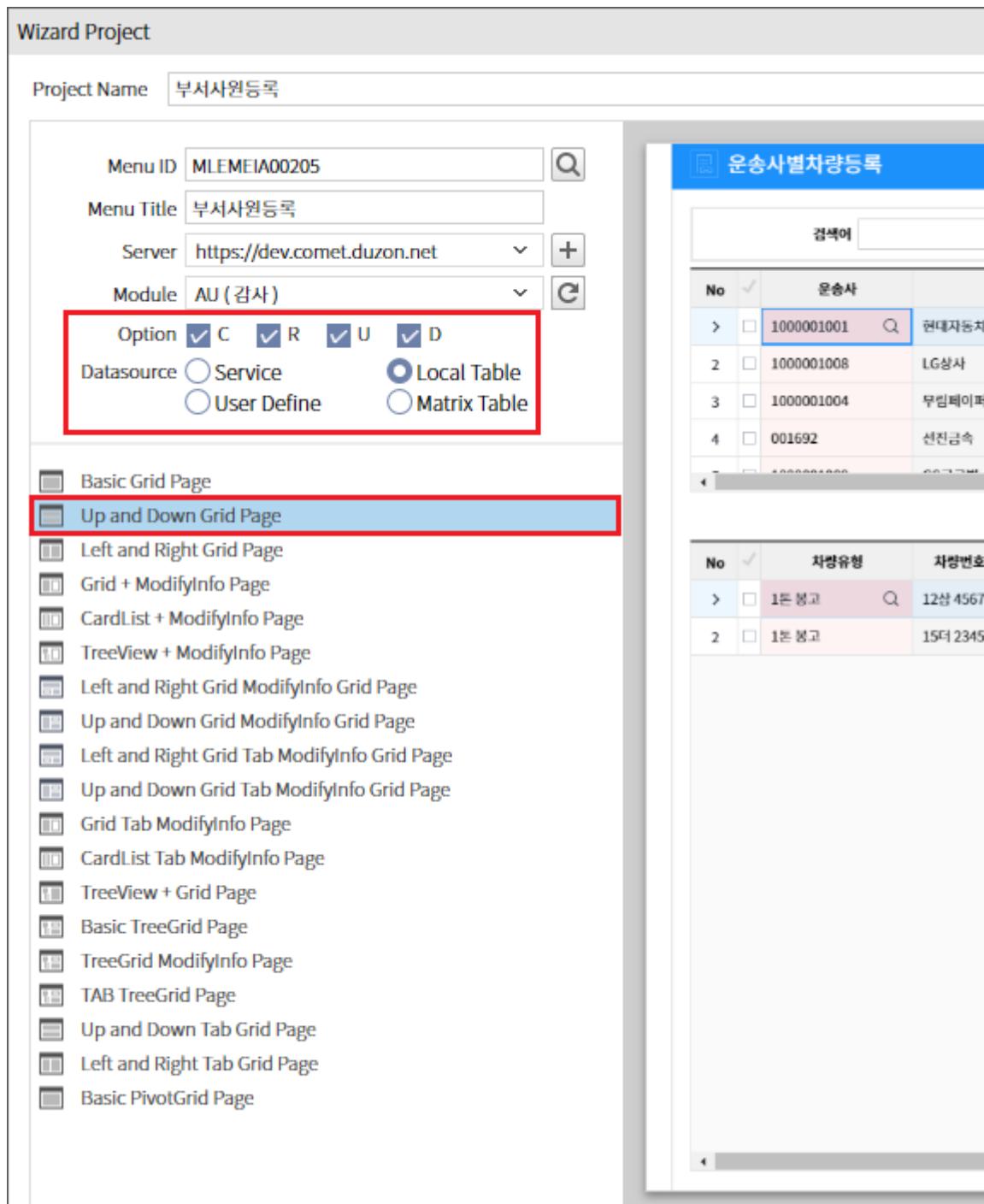
  

| No | 사원번호                 | 한글명 |
|----|----------------------|-----|
| >  | 12345678901234567890 | 1   |

<그림 7-10. Local Table 유형으로 만들 메뉴 화면>

## ② 항목 입력 및 유형 선택

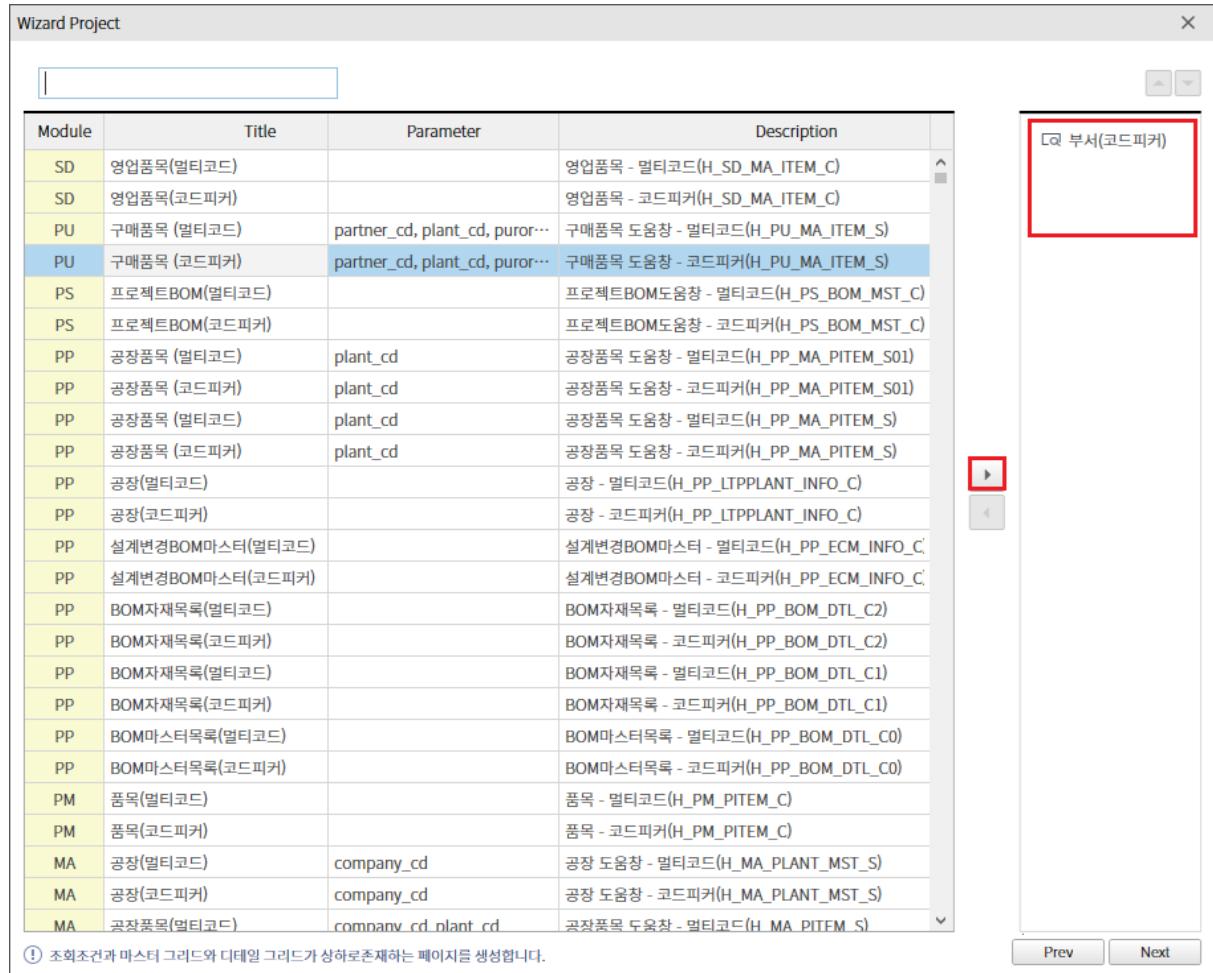
Wizard를 실행하고 프로젝트 정보를 입력하고 데이터 처리 메뉴이므로 C/R/U/D에 모두 체크한다. Datasource는 Local Table을 선택하고 아래 화면 템플릿은 Up and Down Grid Page를 선택하고 아래 Next를 눌러 다음 단계로 진행한다.



<그림 7-11. 프로젝트 항목 입력 및 유형 선택 화면>

### ③ 조회 조건 컨트롤 선택

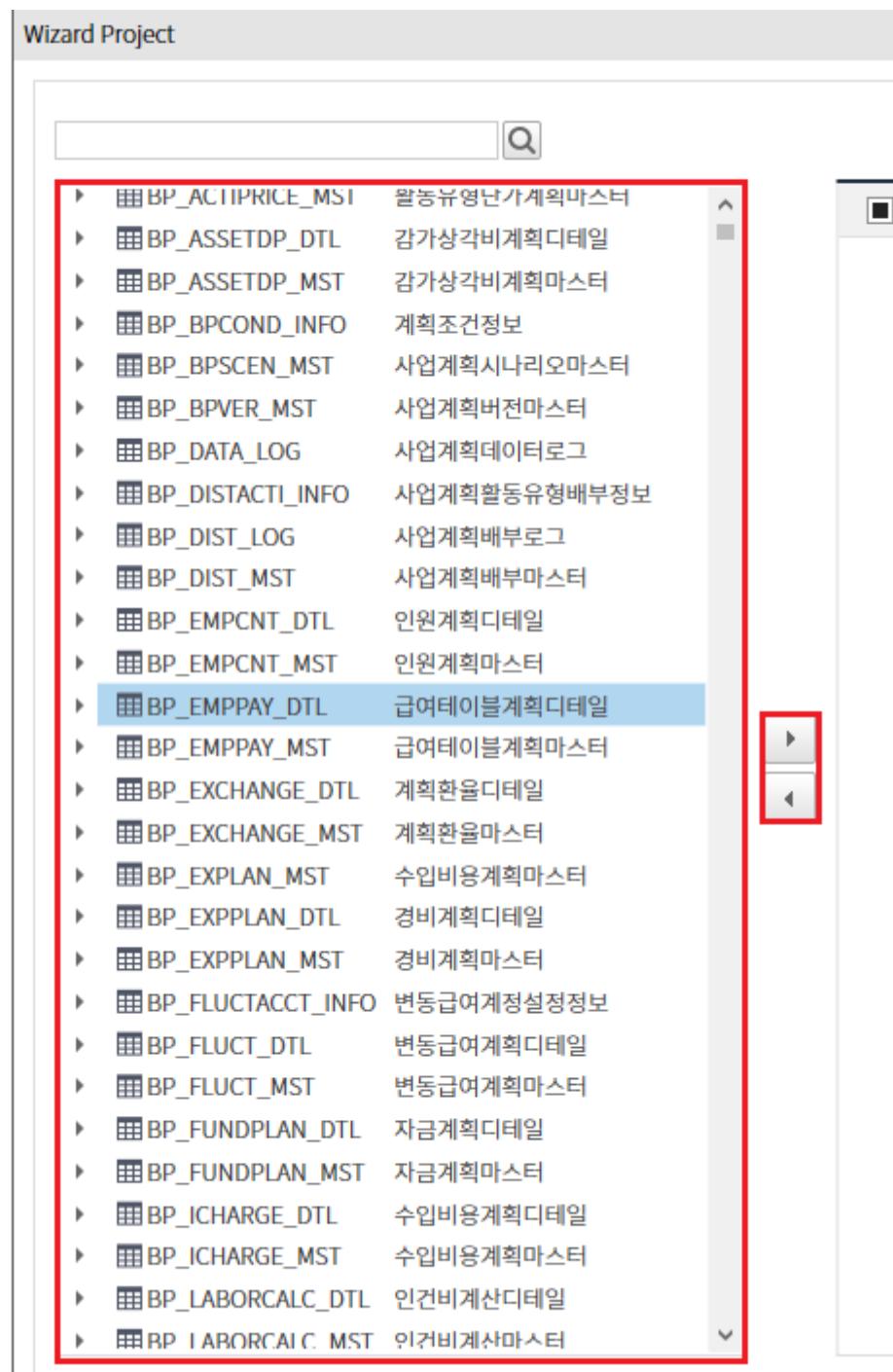
다음은 조회 조건 컨트롤을 선택하는 단계이다 ConditionExplorer에 있는 조건들을 검색하고 결과 항목을 더블클릭하거나 가운데 ▶ 버튼을 눌러 오른쪽 조회 조건 리스트로 이동한다. Local Data를 사용하므로 조회 조건과 상관없이 조회된다.



<그림 7-12. 조회 조건 컨트롤 선택 화면>

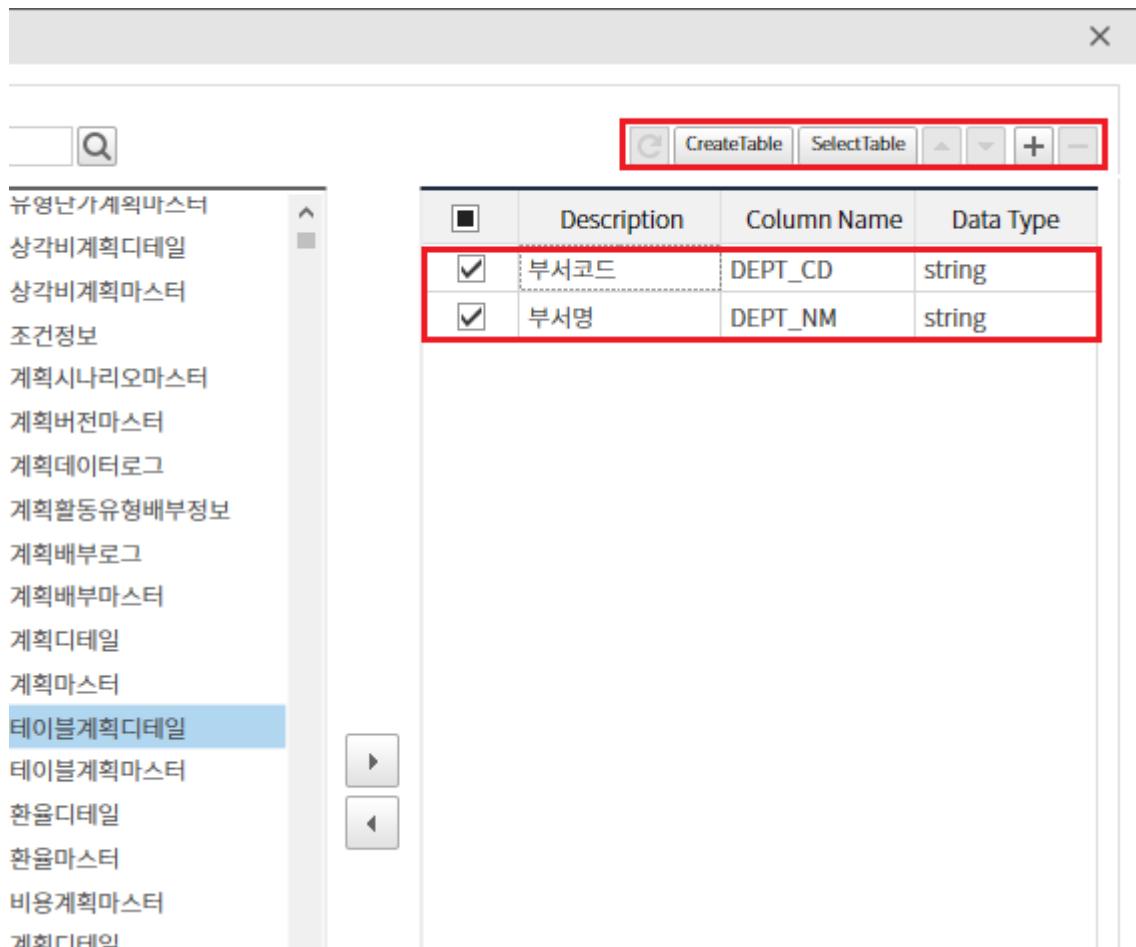
#### ④ Master 그리드 컬럼 선택

다음은 Master 그리드에 적용할 컬럼을 선택하는 단계로 왼쪽은 ②에서 설정한 서버에 테이블 정보를 모두 불러온다. 테이블을 선택하고 이동 버튼을 누르거나 테이블을 펼쳐 원하는 컬럼을 더블클릭하거나 이동 버튼을 눌러 오른쪽으로 이동시킨다.



<그림 7-13. 테이블 선택 왼쪽 화면>

오른쪽은 Local Table로 생성할 필드 리스트이며 상단은 컬럼을 테이블을 만들거나 수정할 수 있는 버튼이 있다.



<그림 7-14. 테이블 선택 오른쪽 화면>

상단 버튼 중 CreateTable과 SelectTable을 선택한 경우는 서버에서 데이터 처리가 가능하며 왼쪽 테이블 리스트에서 컬럼만 추가한 경우 기본 스크립트만 작성된다.

| 버튼 종류       | 설명                               |
|-------------|----------------------------------|
| Reset       | 컬럼 리스트 초기화                       |
| CreateTable | 컬럼 리스트를 새로운 사용자 테이블로 생성 (데이터 처리) |
| SelectTable | 기존 사용자 테이블 선택 (데이터 처리)           |
| Up          | 선택한 필드 위로 이동                     |
| Down        | 선택한 필드 아래로 이동                    |
| Add         | 하단에 새로운 필드 추가                    |
| Delete      | 선택한 필드 삭제                        |

<표 7-2. 상단 버튼 설명>

한 예로 같은 메인 저장 버튼(Save) 스크립트이지만 CreateTable 과 SelectTable 을 선택하고 생성한 메뉴와 필드만 선택하고 생성한 메뉴가 다르게 생성된다.

```
save [MainButton]
1 function() {
2
3 dews.confirm('수정된 데이터를 저장하시겠습니까 ? ', 'question').yes(function() {
4 dews.api.post(dews.url.getApiUrl('CM', 'IdeInquireService/support', {
5 async: false,
6 data: {
7 tableName : 'FD_ABCDEF_101',
8 rows: "{models:"+JSON.stringify(self.dataSource1.getDirtyData())+"
9 }
10 }).done(function (data) {
11 dews.ui.snackbar.ok('저장이 완료되었습니다.');
12 }).fail(function () {
13 dews.error('저장에 실패하였습니다.');
14 });
15 });
16
17
18 dews.api.post(dews.url.getApiUrl('CM', 'IdeInquireService/support', {
19 async: false,
20 data: {
21 tableName : 'FD_BBBBA',
22 rows: "{models:"+JSON.stringify(self.dataSource2.getDirtyData())+"
23 }
24 }).done(function (data) {
25 dews.ui.snackbar.ok('저장이 완료되었습니다.');
26 }).fail(function () {
27 dews.error('저장에 실패하였습니다.');
28 });
29 });
30 });
31}
32}
```

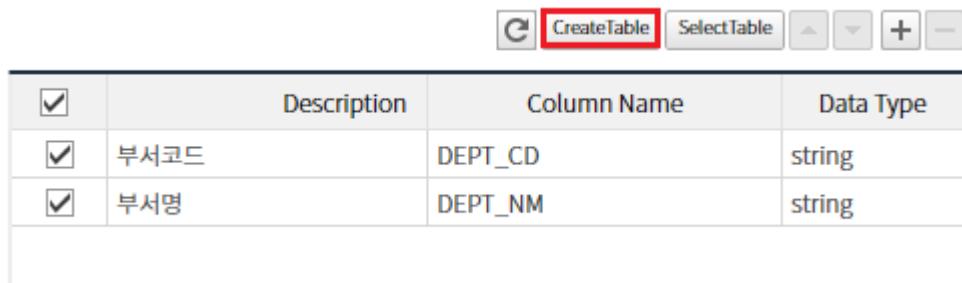
### [왼쪽 컬럼에서 이동해서 생성한 경우]

```
save [MainButton]
1 function() {
2
3 dews.confirm('수정된 데이터를 저장하시겠습니까 ? ', 'question').yes(function() {
4
5 dews.api.post(dews.url.getApiUrl('CM', 'IdeInquireService/support', {
6 async: false,
7 data: {
8 tableName : 'FD_ABCDEF_101',
9 rows: "{models:"+JSON.stringify(self.dataSource1.getDirtyData())+"
10 }
11 }).done(function (data) {
12 dews.ui.snackbar.ok('저장이 완료되었습니다.');
13 }).fail(function () {
14 dews.error('저장에 실패하였습니다.');
15 });
16
17
18 dews.api.post(dews.url.getApiUrl('CM', 'IdeInquireService/support', {
19 async: false,
20 data: {
21 tableName : 'FD_BBBBA',
22 rows: "{models:"+JSON.stringify(self.dataSource2.getDirtyData())+"
23 }
24 }).done(function (data) {
25 dews.ui.snackbar.ok('저장이 완료되었습니다.');
26 }).fail(function () {
27 dews.error('저장에 실패하였습니다.');
28 });
29 });
30 });
31}
32}
```

### [CreateTable 또는 SelectTable로 생성한 경우]

<그림 7-15. save 버튼 스크립트 차이 화면>

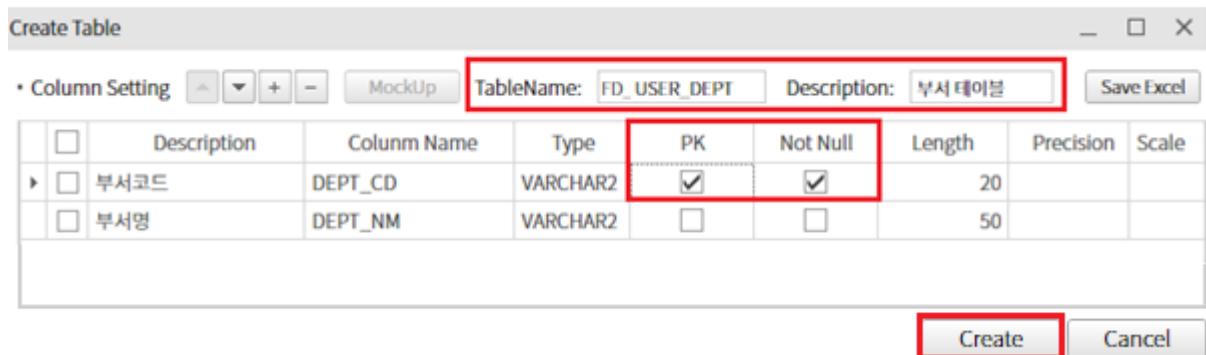
이제 오른쪽에 있는 필드들을 CreateTable 을 눌러 사용자 테이블로 추가한다.



|                                     | Description | Column Name | Data Type |
|-------------------------------------|-------------|-------------|-----------|
| <input checked="" type="checkbox"/> | 부서코드        | DEPT_CD     | string    |
| <input checked="" type="checkbox"/> | 부서명         | DEPT_NM     | string    |

<그림 7-16. Create Table 버튼>

TableName, Description을 입력하고, 기본키를 설정하기 위해서 PK와 Not Null에 체크한다 이 제 아래 Create 버튼을 눌러 테이블을 생성한다.



Create Table

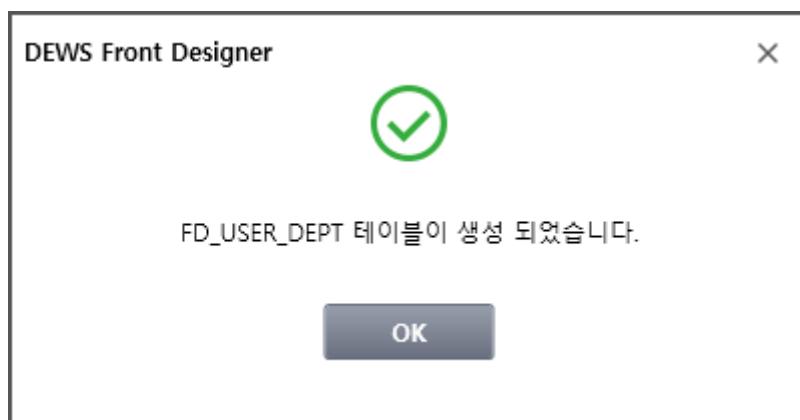
\* Column Setting MockUp TableName: FD\_USER\_DEPT Description: 부서 테이블 Save Excel

|                          | Description | Column Name | Type     | PK                                  | Not Null                            | Length | Precision | Scale |
|--------------------------|-------------|-------------|----------|-------------------------------------|-------------------------------------|--------|-----------|-------|
| <input type="checkbox"/> | 부서코드        | DEPT_CD     | VARCHAR2 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 20     |           |       |
| <input type="checkbox"/> | 부서명         | DEPT_NM     | VARCHAR2 | <input type="checkbox"/>            | <input type="checkbox"/>            | 50     |           |       |

Create Cancel

<그림 7-17. Create Table 항목 입력>

테이블 생성 메세지가 열리면 OK를 누른다.



<그림 7-18. 테이블 생성 완료 메세지 화면>

## ⑤ Sub 그리드 필드 선택 및 마법사 완료

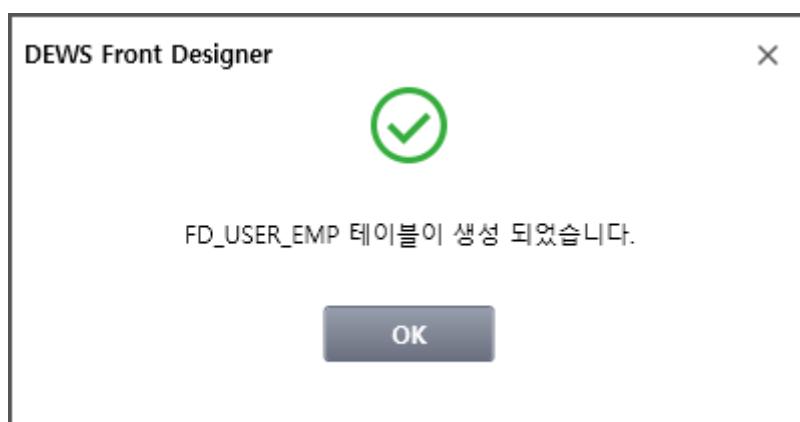
다음은 Sub 테이블을 선택하는 단계로 왼쪽은 ②에서 설정한 서버에 테이블 정보를 모두 불러온다. 테이블을 선택하고 이동 버튼을 누르거나 테이블을 펼쳐 원하는 컬럼을 더블클릭하거나 이동 버튼을 눌러 오른쪽으로 이동시킨다.

- ④ Master 그리드 필드 선택 단계 처럼 CreateTable을 눌러서 새로운 테이블을 생성한다.

|                          | Description | Column Name | Type     | PK                                  | Not Null                            | Length | Precision |
|--------------------------|-------------|-------------|----------|-------------------------------------|-------------------------------------|--------|-----------|
| <input type="checkbox"/> | 사원번호        | EMP_NO      | VARCHAR2 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 20     |           |
| <input type="checkbox"/> | 한글명         | KOR_NM      | VARCHAR2 | <input type="checkbox"/>            | <input type="checkbox"/>            | 50     |           |
| <input type="checkbox"/> | 부서코드        | DEPT_CD     | VARCHAR2 | <input type="checkbox"/>            | <input type="checkbox"/>            | 20     |           |

<그림 7-19. Sub 필드 사용자 테이블 생성 화면>

- 테이블 생성 메세지가 열린다. OK를 누른다.



<그림 7-20. 테이블 생성 완료 메세지 화면>

오른쪽 하단에 OK를 눌러 마법사를 완료한다.



<그림 7-21. OK 버튼 확인>

기본 화면이 완성되었으며 추가적으로 디테일 설정이 필요한 경우 화면을 수정하거나 스크립트를 작성한다

| No | 부서 | 부서명 |
|----|----|-----|
| 1  |    |     |

| No | 사원번호 | 한글명 | 부서     |
|----|------|-----|--------|
| 1  |      | 박종규 | 701020 |

<그림 7-22. 완성된 화면>

이제 미리보기를 통해 데이터 입력/삭제/수정 및 저장을 확인해 본다.

| No | 부서     | 부서명   |
|----|--------|-------|
| >  | 701020 | 국내구매부 |

| No | 사원번호       | 한글명 | 부서     |
|----|------------|-----|--------|
| 1  | 1010293011 | 박종규 | 701020 |
| >  | 1122334455 | 배현규 | 701010 |

<그림 7-23. 미리보기 화면>

메인 버튼들의 스크립트의 경우 실제 구현 단계에서는 사용할 수 없으며 데이터 처리 테스트 용도로 사용이 가능 하다.

### 3) User Define

DFD에서 제공하는 Query Builder를 이용하여 사용자 정의 쿼리를 만들고 User Define SQL Manager 항목에 추가하여 메뉴 개발을 할 때 사용한다. 현재는 조회(R)만 가능하므로 데이터처리 메뉴(CRUD)는 되지 않는다.

#### ① 만들 메뉴 화면

2단 그리드 구조(Master-Sub 그리드)로 된 메뉴를 Local Table 유형을 선택하여 만들어 보며 조회만 되는 메뉴이다.

| No | 부서명    |
|----|--------|
| >  | 생산기술6부 |

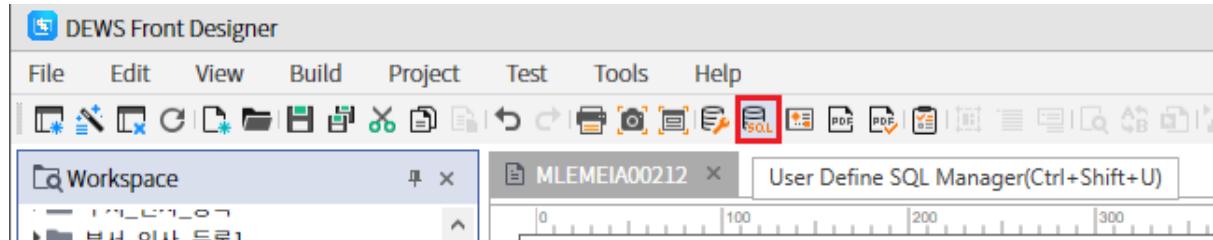
  

| No | 사원번호        | 한글명 |
|----|-------------|-----|
| 1  | 20170811001 | 박정석 |
| >  | 9000011895  | 박정석 |
| 3  | 2018091001  | 김강성 |
| 4  | 2018050110  | 김명호 |
| 5  | 9000011895  | 박정석 |
| 6  | 2018091001  | 김강성 |
| 7  | 2018050110  | 김명호 |
| 8  | 20170811001 | 박정석 |
| 9  | 9000011895  | 박정석 |

<그림 7-24. User Define 유형으로 만들 메뉴 화면>

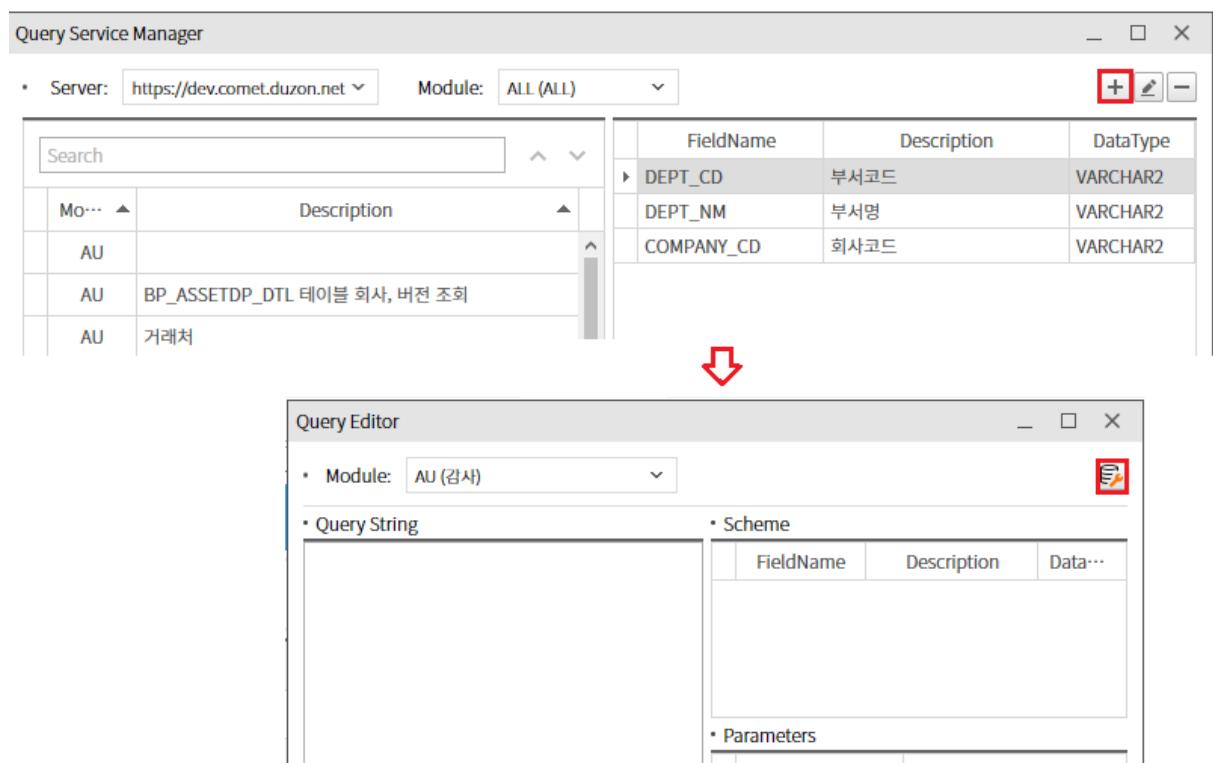
## ② User Define SQL Manager 리스트 추가

먼저 사용자 정의 쿼리를 만들기 위해서 User Define SQL Manager 실행해야 한다. 메뉴에서 실행할 수 없으며 툴바에서 실행한다.



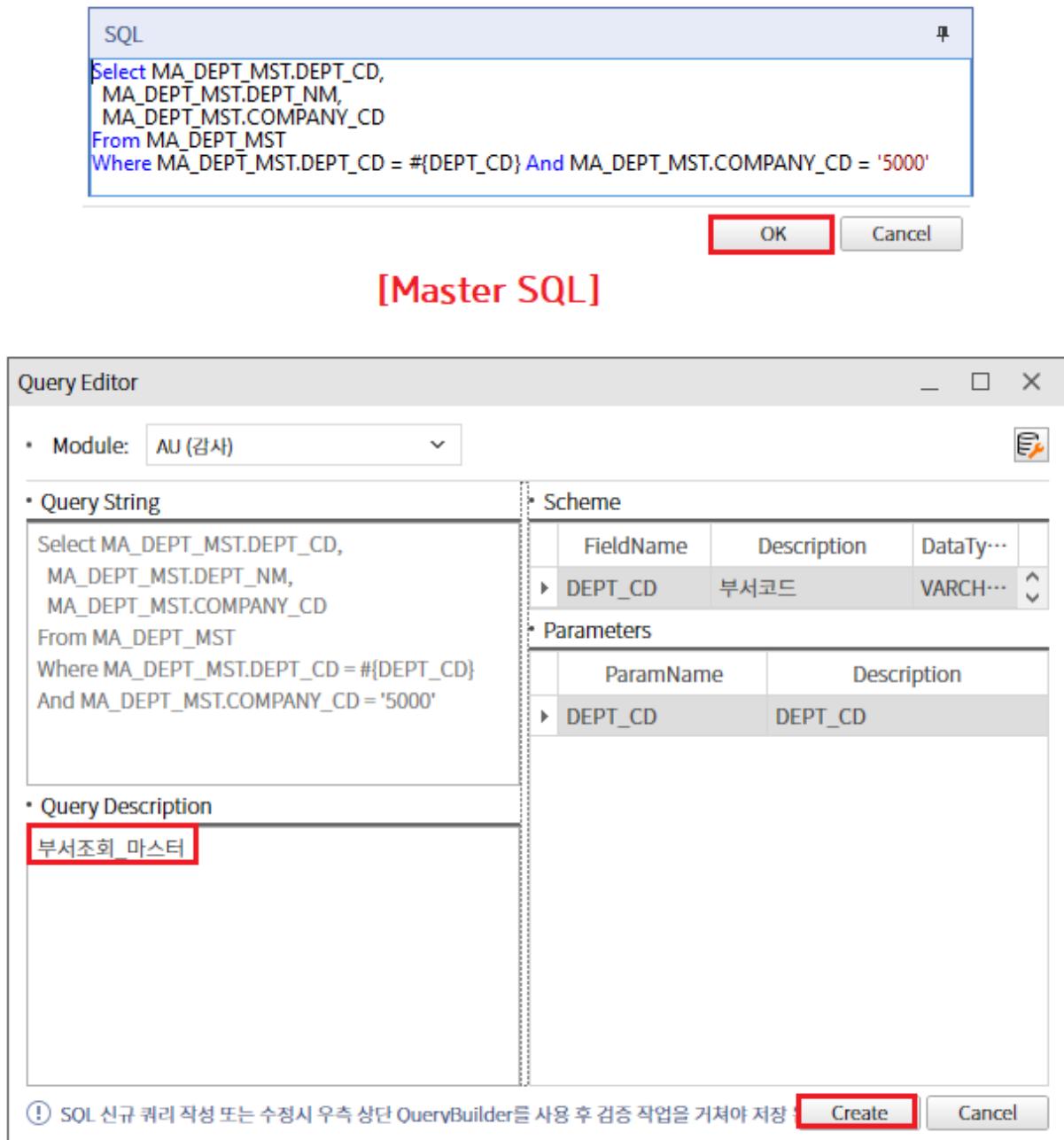
<그림 7-25 User Define SQL Manager 실행 화면>

Query Service Manager 화면이 열리며 신규 SQL 작성을 하기 위하여 상단 + 추가 버튼을 눌러 Query Editor 화면을 열고 쿼리를 작성하기 위하여 상단 쿼리 빌더를 클릭한다.



<그림 7-26. Query Service Manager 및 Query Editor 화면>

먼저 User Define SQL Manager에서 QueryBuilder를 이용하여 2개 쿼리를 생성하고 리스트에 등록한다. Master 쿼리 작성 후 실행하고 Query Editor에서 Query Description을 입력하고 Create 버튼을 눌러서 리스트에 추가한다.



<그림 7-27. Master SQL 작성 및 항목 추가>

다음으로 Sub 쿼리 작성 후 실행하고 Query Editor에서 Query Description을 입력하고 Create 버튼을 눌러서 리스트에 추가한다.

**[Sub SQL]**

The screenshot shows the Oracle SQL Developer Query Editor window. In the top-left corner, there is a separate box labeled "SQL" containing a subquery:

```
SQL
Select HR_EMP_MST.EMP_NO,
 HR_EMP_MST.KOR_NM,
 HR_EMP_MST.DEPT_CD
 From HR_EMP_MST
 Where HR_EMP_MST.DEPT_CD = #{DEPT_CD}
```

The main Query Editor window has the following sections:

- Module:** AU (감사)
- Query String:** The same subquery as above.
- Scheme:** A table showing the structure of the selected table:

| FieldName | Description | Data... |
|-----------|-------------|---------|
| EMP_NO    | 사원번호        | VARC... |
| KOR_NM    | 한글명         | VARC... |
| DEPT_CD   | 부서코드        | VARC... |
- Parameters:** A table showing the parameters used in the query:

| ParamName | Description |
|-----------|-------------|
| DEPT_CD   | DEPT_CD     |
- Query Description:** A text area containing "사원조회\_서브(디테일)" which is highlighted with a red box.

At the bottom of the window, there is a note: "① SQL 신규 쿼리 작성 또는 수정시 우측 상단 QueryBuilder를 사용 후 검증 작업을 거친 후 Create 버튼을 눌러주세요." followed by "Create" and "Cancel" buttons.

<그림 7-28. Sub SQL 작성 및 항목 추가>

Query Service Manager에 사용자가 작성한 Master와 Sub 리스트가 있다.

The screenshot shows the 'Query Service Manager' application window. At the top, there are fields for 'Server' (set to <https://dev.comet.duzon.net>) and 'Module' (set to ALL (ALL)). Below these are two tables.

The left table lists various queries categorized by type (AU, CI, FI, MF) and description. Two specific entries are highlighted with a red box:

- AU 부서조회\_마스터
- AU 사원조회\_서브(디테일)

The right table displays detailed information for the selected query '부서조회\_마스터'. It includes columns for 'FieldName', 'Description', and 'DataType'. The first three rows are shown:

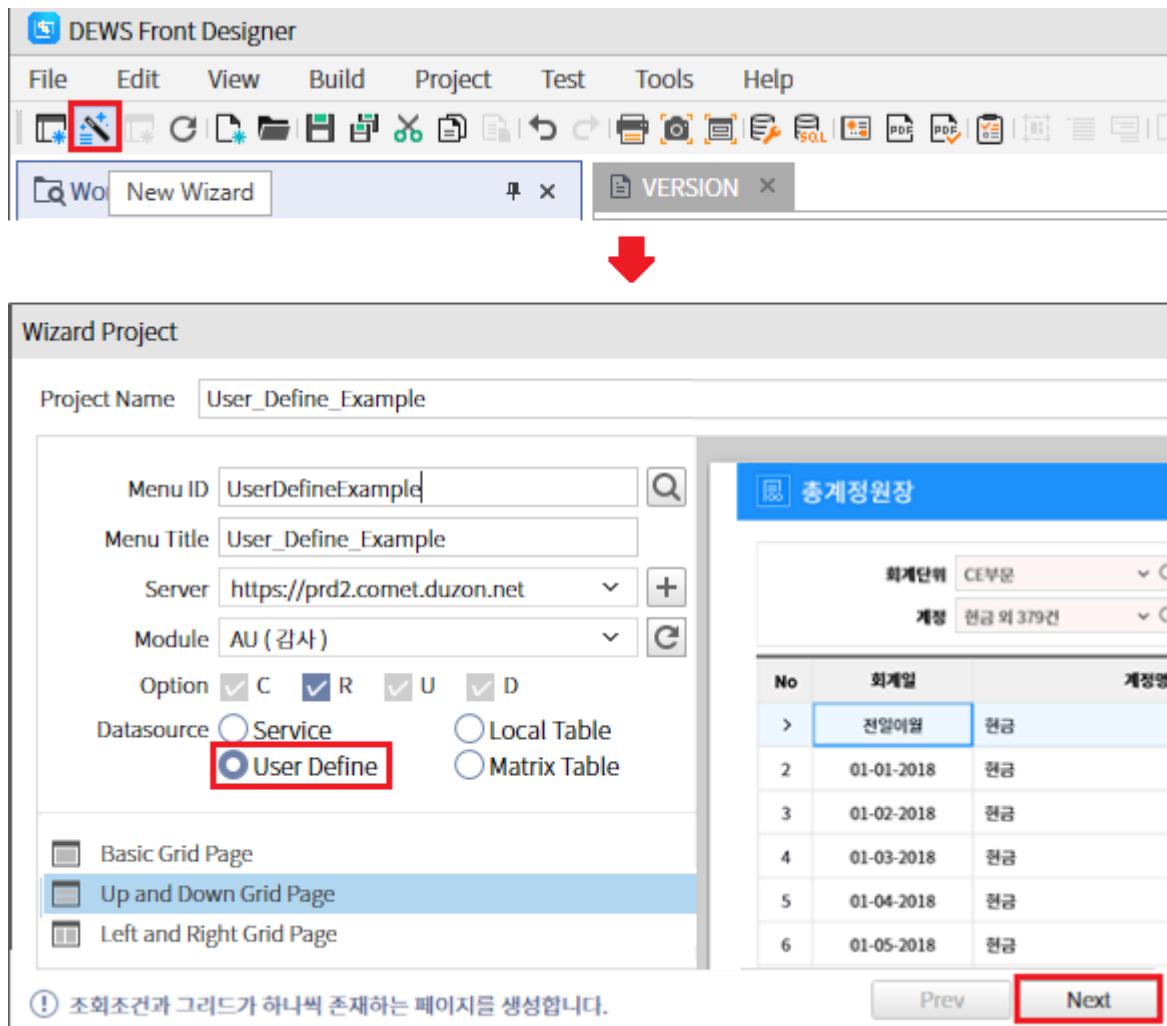
| FieldName  | Description | DataType |
|------------|-------------|----------|
| DEPT_CD    | 부서코드        | VARCHAR2 |
| DEPT_NM    | 부서명         | VARCHAR2 |
| COMPANY_CD | 회사코드        | VARCHAR2 |

At the bottom right of the main area is an 'OK' button.

<그림 7-29. 사용자가 추가한 Query Service Manager 리스트 화면>

### ③ 항목 입력 및 유형 선택

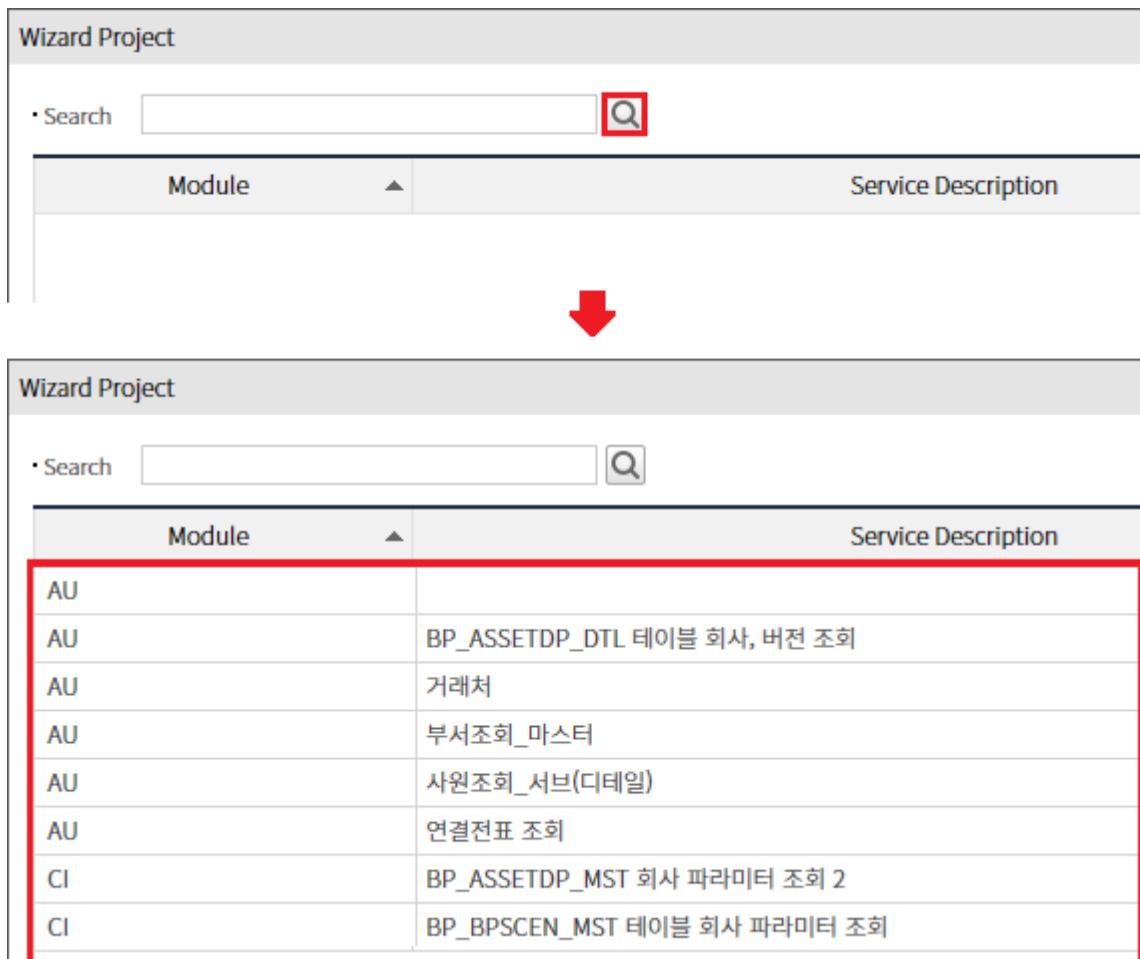
New Wizard 아이콘을 클릭하여 마법사 프로젝트를 실행한다. 항목을 각각 입력하고 User Define을 선택하고, 생성할 화면 유형을 선택한다. 현재는 조회(Read)만 가능하므로 나머지 항목들은 비활성화 된다.



<그림 7-30. 프로젝트 항목 입력 및 유형 선택 화면>

#### ④ 사용자 쿼리 선택

다음은 이전에서 생성한 User Define SQL Manager 에서 추가한 항목을 선택하는 단계이다. 검색 버튼을 누르면 이전 서버에서 사용자가 만든 쿼리 리스트(Query Service Manager List)가 표시된다.



| Module | Service Description          |
|--------|------------------------------|
| AU     |                              |
| AU     | BP_ASSETDP_DTL 테이블 회사, 버전 조회 |
| AU     | 거래처                          |
| AU     | 부서조회_마스터                     |
| AU     | 사원조회_서브(디테일)                 |
| AU     | 연결전표 조회                      |
| CI     | BP_ASSETDP_MST 회사 파라미터 조회 2  |
| CI     | BP_BPSCEN_MST 테이블 회사 파라미터 조회 |

<그림 7-31. 사용자 쿼리 리스트 화면>

리스트 항목을 더블 클릭하여 선택하면 Selected Service, Parameter, Data Schema, Detail Info를 확인할 수 있다. 처음 더블 클릭한 항목은 Master, 두 번째 더블 클릭한 항목은 Sub로 설정된다.

The screenshot shows the 'Wizard Project' window with several tabs and sections:

- Search:** A search bar with a magnifying glass icon.
- Module:** A list of service descriptions:
  - AU 부서조회\_마스터
  - AU 사원조회\_서브(디테일)** (highlighted in blue)
  - CI 공통코드-디테일
  - CI 거래처조회(MA)
  - AU 사원정보(DETAIL)
  - AU 사원을 조회하는 쿼리입니다
  - CI 공통코드-마스타
  - AU 부서정보(MASTER)
- Parameter:** A table showing parameters:
 

| Parameter | Description | Transfer Type | Data Type |
|-----------|-------------|---------------|-----------|
| DEPT_CD   | DEPT_CD     | QueryString   | string    |
- Data Schema:** A table showing column definitions:
 

| Column     | Description | Data Type |
|------------|-------------|-----------|
| DEPT_CD    | 부서코드        | string    |
| DEPT_NM    | 부서명         | string    |
| COMPANY_CD | 회사코드        | string    |
- Selected Service:** A table showing selected service details:
 

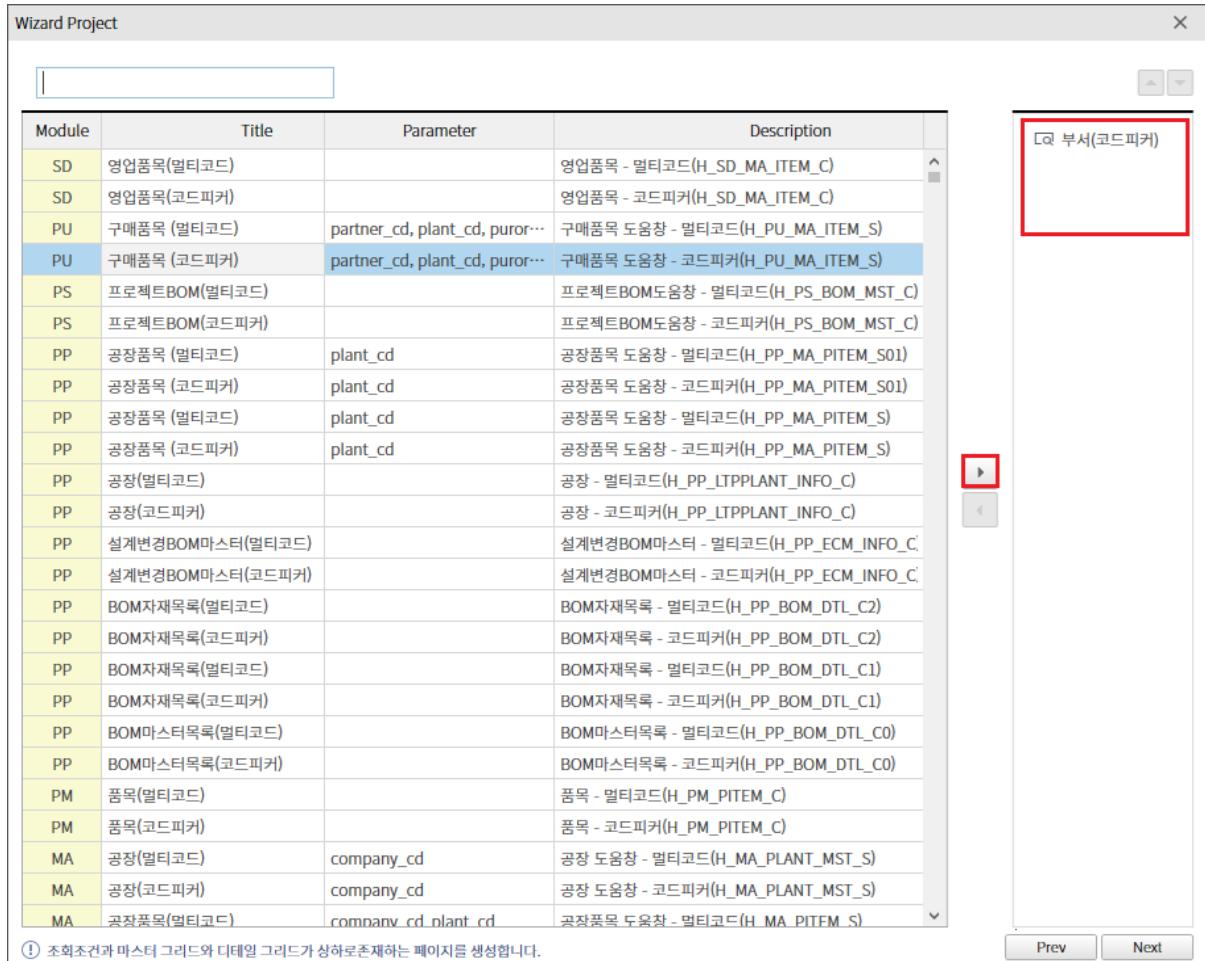
| Master | GridID | DataSourceID | Service Description |
|--------|--------|--------------|---------------------|
| Master | grid1  | dataSource1  | 부서조회_마스터            |
| Sub    | grid2  | dataSource2  | 사원조회_서브(디테일)        |
- Detail Info [ Master ]:** A table showing master service details:
 

|                         |                       |
|-------------------------|-----------------------|
| Grid ID                 | grid1                 |
| DataSource              | dataSource1           |
| Module                  | CM                    |
| Description             | CM                    |
| Service                 | CustomBusinessService |
| Service Description     | 커스텀 비즈니스 서비스          |
| URL Pattern             | bp/invoke             |
| URL Pattern Description | 쿼리 서비스 데이터 조회         |
- Note:** ⓘ 조회조건과 그리드가 하나씩 존재하는 페이지를 생성합니다.
- Buttons:** Prev and Next buttons.

<그림 7-32. User Define SQL Manager 리스트 선택 화면>

## ⑤ 조회 조건 컨트롤 선택

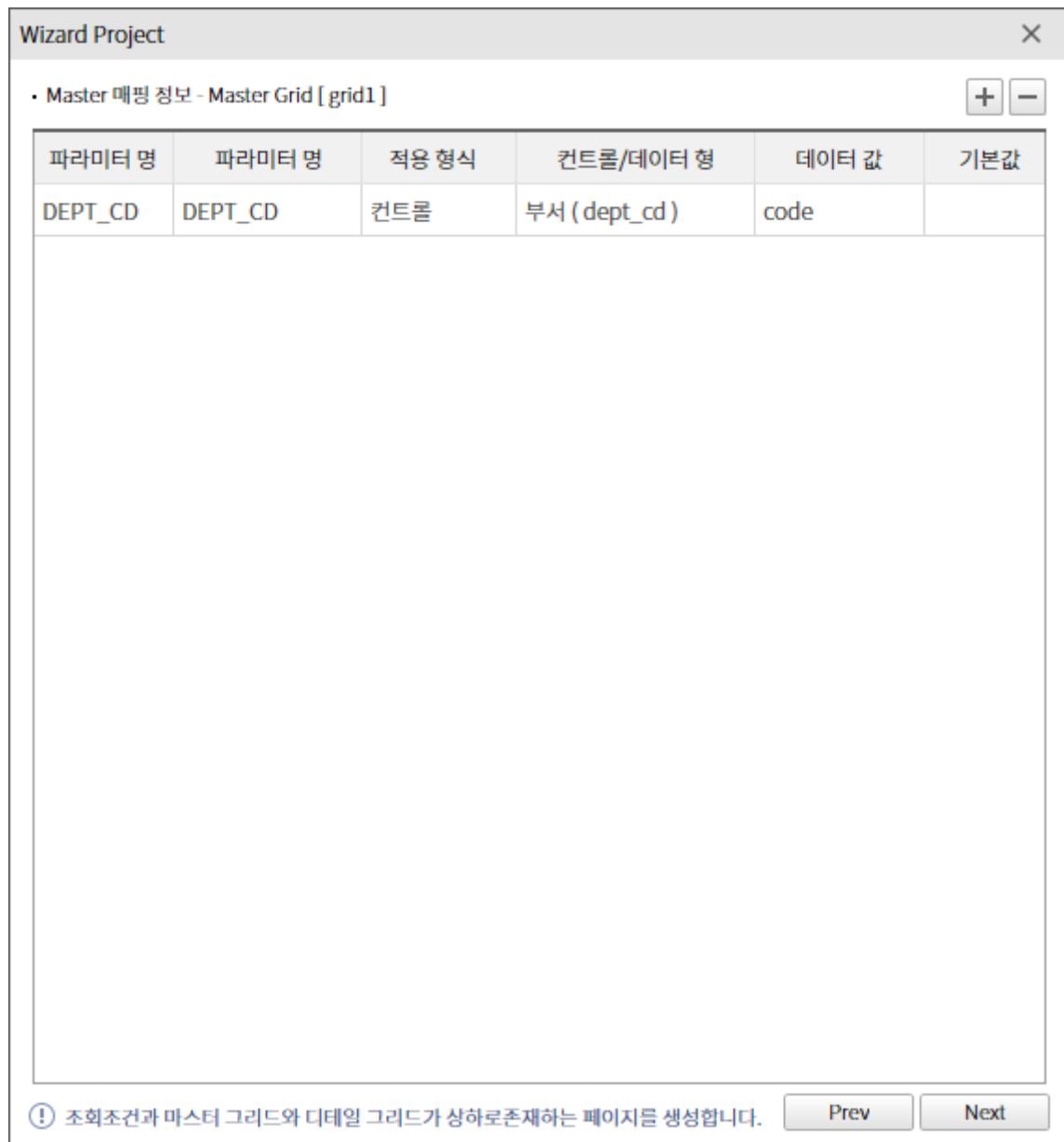
다음은 조회 조건 컨트롤을 선택하는 단계이다 Parameter 이름과 조회 조건 컨트롤 id와 비슷한 이름이 있을 경우 자동으로 오른쪽 조회 조건 리스트에 추가되어 있다. 필요없는 경우 더블클릭하거나 ◀ 버튼을 눌러 리스트에서 삭제한다. 조건을 추가할 경우 검색어 입력 후 조회하고 리스트에서 더블클릭하거나 ▶ 버튼을 눌러 오른쪽 조회 조건 리스트에 추가한다.



<그림 7-33. 조회 조건 컨트롤 선택 화면>

#### ⑥ Master 그리드 매핑 설정

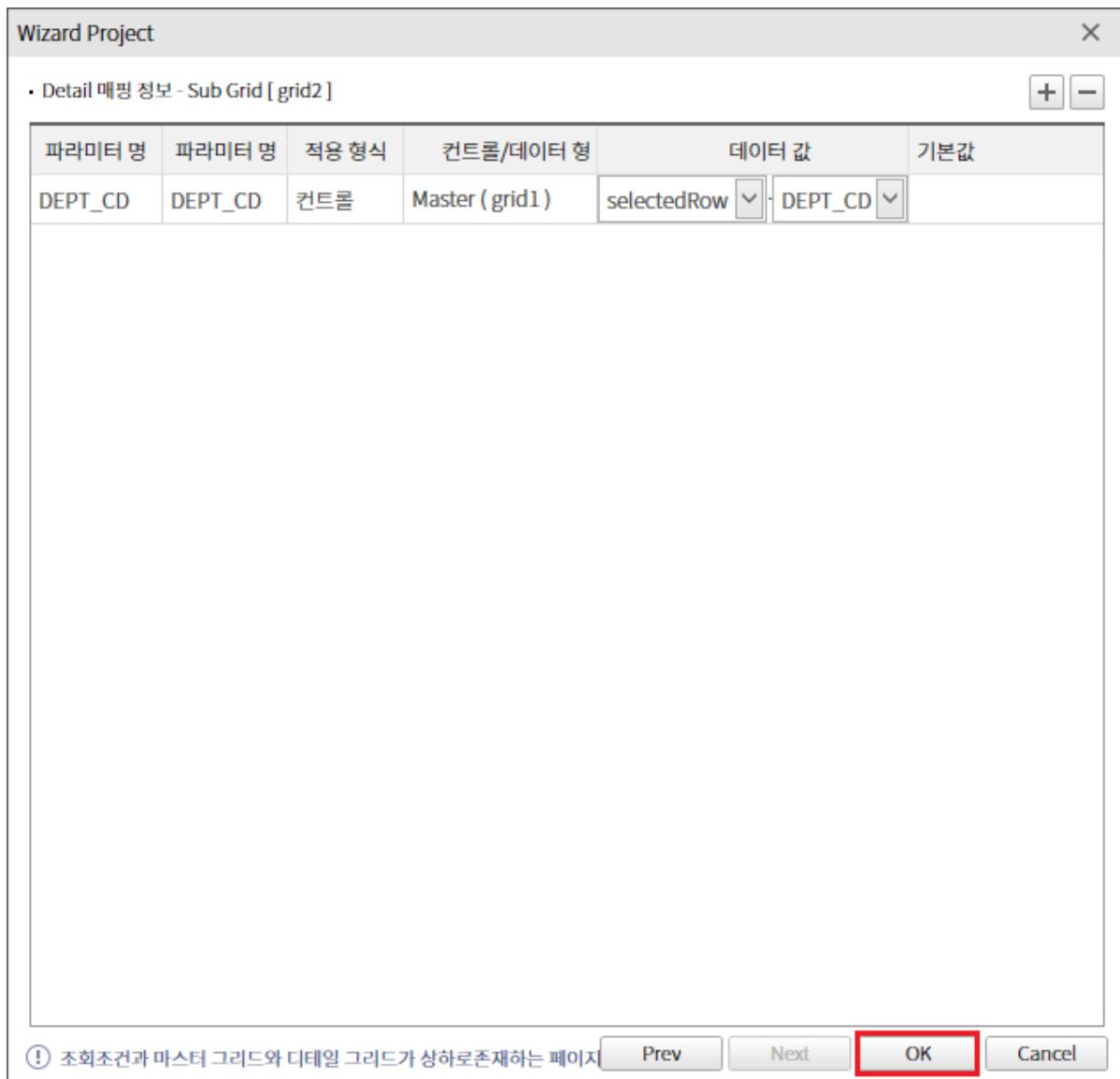
다음은 Master 그리드와 연결되는 데이터소스의 파라미터에 매핑할 조회 조건 컨트롤이나 상수를 설정하는 단계이다. 파라미터 이름과 조회 조건 컨트롤 id가 같거나 비슷한 경우 파라미터 값에 따라 자동으로 설정되어 있으며, 편집이 필요한 경우 수정을 하고 다음 단계로 넘어간다



<그림 7-34. Master 그리드 매핑 화면>

#### ⑥ Sub 그리드 매핑 설정 및 마법사 완료

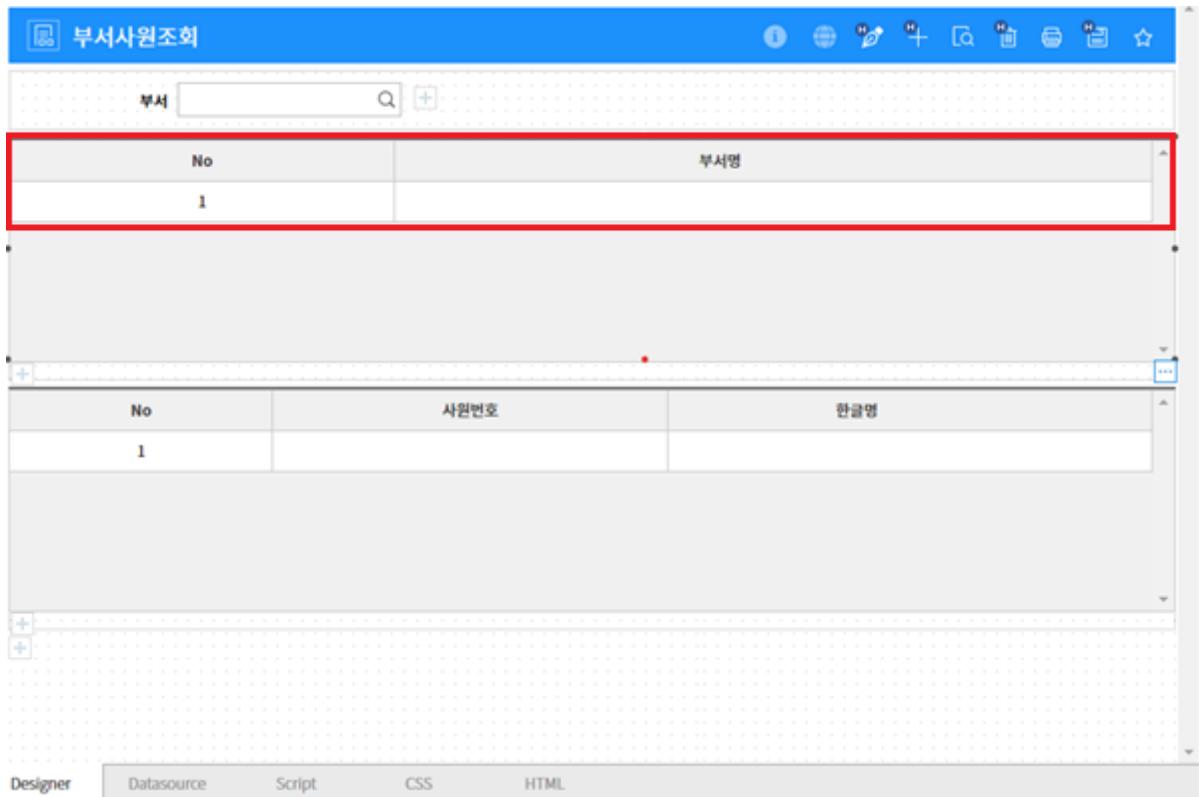
다음은 Sub(Detail) 그리드와 연결되는 데이터소스의 파라미터에 매핑할 조회 조건 컨트롤이나 상수를 설정하는 단계이다. 마스터 그리드의 필드와 일치하는 Sub 그리드 필드가 있을 경우 자동으로 매핑되어 있다.



<그림 7-35. Sub 그리드 매핑 화면 및 마법사 종료 화면>

마지막으로 OK 버튼을 눌러 마법사를 종료한다.

기본 화면이 완성되었으며 추가적으로 디테일 설정이 필요한 경우 화면을 수정하거나 스크립트를 작성한다



<그림 7-36. 완성된 화면>

Master 그리드의 경우 부서명만 표시되므로 선택 후 Properties 창 컬럼 설정을 클릭 후 부서코드도 표시 체크 한다.

| Header Caption | Column Name | Width                                  |
|----------------|-------------|----------------------------------------|
| 부서             | DEPT_CD     | <input checked="" type="checkbox"/> 80 |
| 부서명            | DEPT_NM     | <input checked="" type="checkbox"/> 90 |

<그림 7-37. Master 그리드 컬럼 설정 화면>

이제 미리보기를 해서 부서를 선택하고 조회를 해본다. 정상적으로 조회되는 것을 확인할 수 있다. .

| No | 부서     | 부서명    |
|----|--------|--------|
| >  | 995100 | 생산기술6부 |
|    |        |        |
|    |        |        |
|    |        |        |
|    |        |        |
|    |        |        |
|    |        |        |
|    |        |        |
|    |        |        |

| No | 사원번호        | 한글명 |
|----|-------------|-----|
| >  | 9000011895  | 박정석 |
| 1  | 20170811001 | 박정석 |
| 3  | 2018091001  | 김강성 |
| 4  | 2018050110  | 김명호 |
| 5  | 9000011895  | 박정석 |
| 6  | 2018091001  | 김강성 |
| 7  | 2018050110  | 김명호 |
| 8  | 20170811001 | 박정석 |
| 9  | 9000011895  | 박정석 |

<그림 7-38. 미리보기 후 조회 화면>

#### 4) Matrix Table

공용테이블(Matrix Table)은 공용으로 사용하는 테이블에 여러 메뉴들의 데이터 처리를 할 때 사용한다. 하나의 공용 API와 테이블 재사용으로 개발 생산성 재고와 운영 비용을 최소화할 수 있는 장점이 있다.

##### ① 만들 메뉴 화면

2단 그리드 구조(Master-Sub 그리드)로 된 메뉴를 Matrix Table 유형을 선택하여 만들어 보며, 데이터 처리가 되는 CRUD 메뉴이다.

| No | 부서     | 부서명  |
|----|--------|------|
| >  | 992300 | 출하6부 |

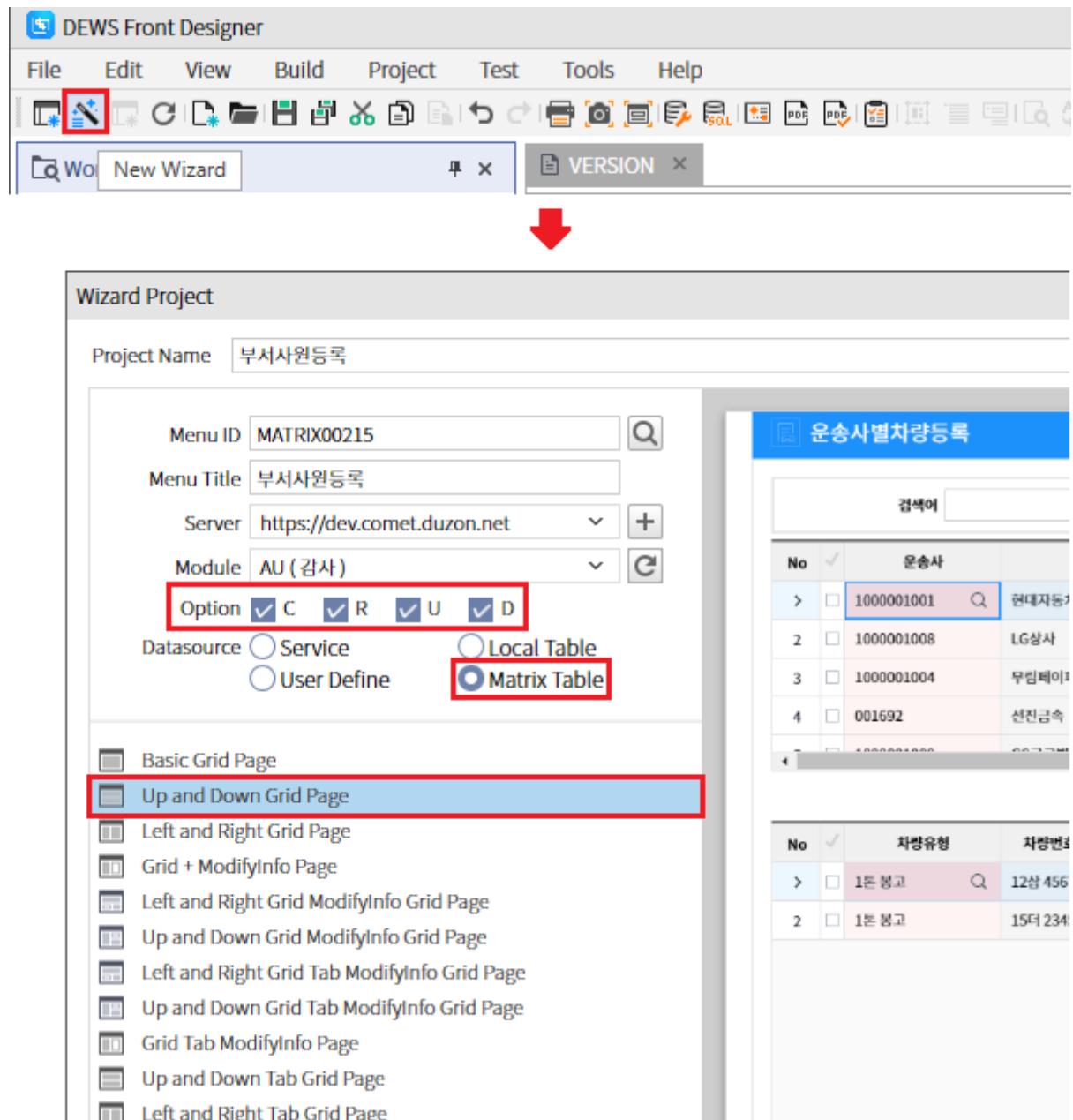
  

| No | 부서     | 사원번호       | 사원명 |
|----|--------|------------|-----|
| 1  | 992300 | 1111222233 | 김을동 |
| >  | 992300 | 1122334455 | 박을동 |

<그림 7-39. Matrix Table 유형으로 만들 메뉴 화면>

## ② 항목 입력 및 유형 선택

Wizard를 실행하고 프로젝트 정보를 입력하고 데이터 처리 메뉴이므로 C/R/U/D에 모두 체크한다. Datasource는 Matrix Table을 선택하고 아래 화면 템플릿은 Up and Down Grid Page를 선택하고 아래 Next를 눌러 다음 단계로 진행한다.



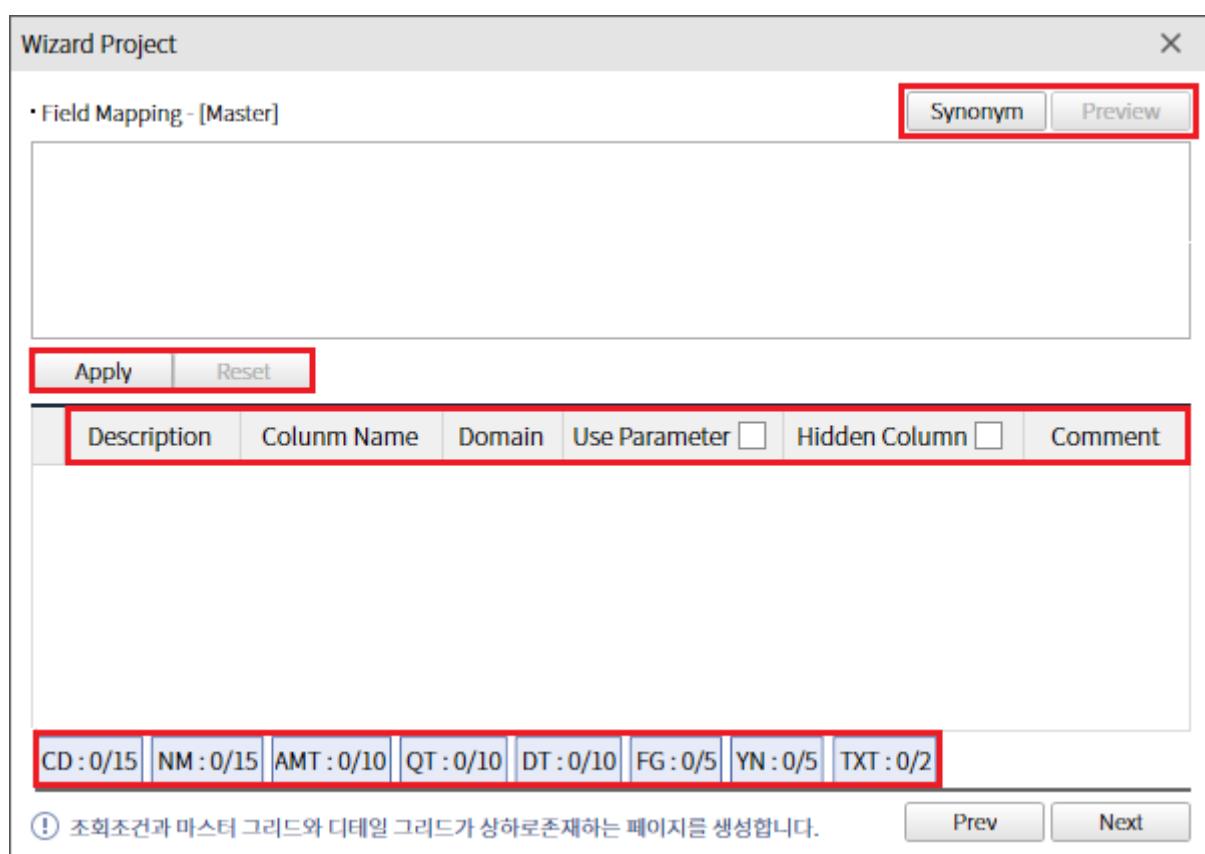
<그림 7-40. 프로젝트 항목 입력 및 유형 선택 화면>

### ③ 그리드 컬럼 입력 및 공용테이블

다음으로 공용 테이블에 사용할 컬럼을 입력한다. Description을 입력하고 Apply 버튼을 눌러 공용테이블에 적용 했을 때 매핑되는 컬럼명을 자동으로 생성해 준다. Reset은 공용테이블에 적용할 컬럼을 초기화한다. Description 입력 시 다양한 옵션을 대괄호([])를 사용하여 설정을 할 수 있다. 예를 들어 파라미터를 사용할 경우 [^]를 입력하고, h는 숨길 컬럼, t는 컬럼명에 따른 에디터 자동 미생성, o는 폼패널에 바인딩할 컬럼 설정 등을 설정할 수 있다.

우측 상단 Synonym(동의어)은 여러 Description을 가진 컬럼명을 관리하며(예를 들어, ANIC\_AMT 컬럼명은 연봉금액, 연봉, 연봉액 등 여러 의미를 가질 수 있음) 메뉴 Tools > Synonyms Dictionary에서도 편집을 할 수 있다. Preview는 공용테이블 전체 컬럼을 확인할 수 있다.

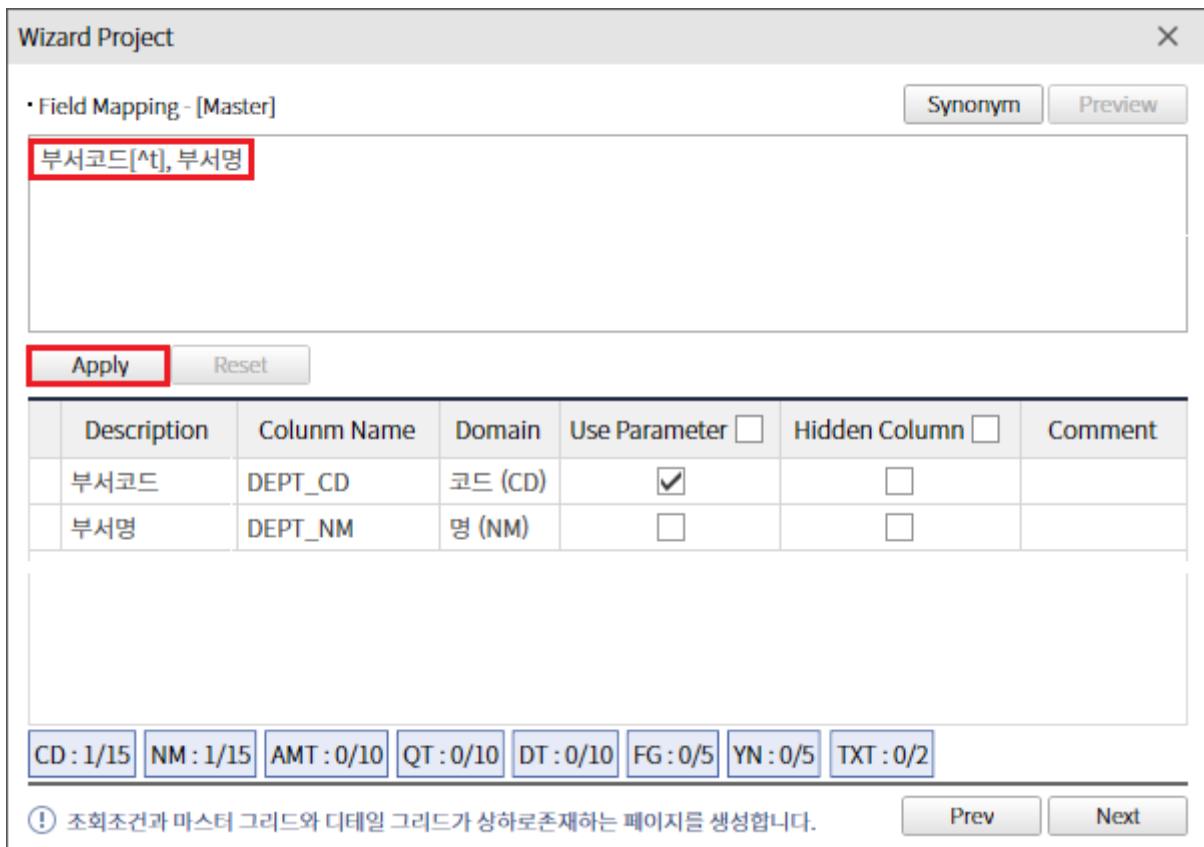
가운데 공용 테이블에 적용될 타이틀은 Description, Column Name, Domain(영역), Use Parameter(파라미터에 사용될 컬럼), Hidden Column(숨길 컬럼), Comment(비고) 등을 입력할 수 있으며 공용테이블 Domain(영역)에는 CD(코드-15개), NM(명-15개), AMT(금액-10개), QT(수량-10개), DT(일자-10개), FG(플래그-5개), YN(논리-5개), TXT(텍스트-2개)를 제공한다.



<그림 7-41. 그리드 컬럼 화면 설명>

- Master 그리드 적용 컬럼 입력

Master 그리드에 적용할 필드 Description을 입력하고 Apply을 눌러서 공용테이블에 사용될 컬럼으로 적용한다. 부서코드의 경우 대괄호를 사용하여 옵션을 입력하여 자동으로 use Parameter에 체크가 된다.

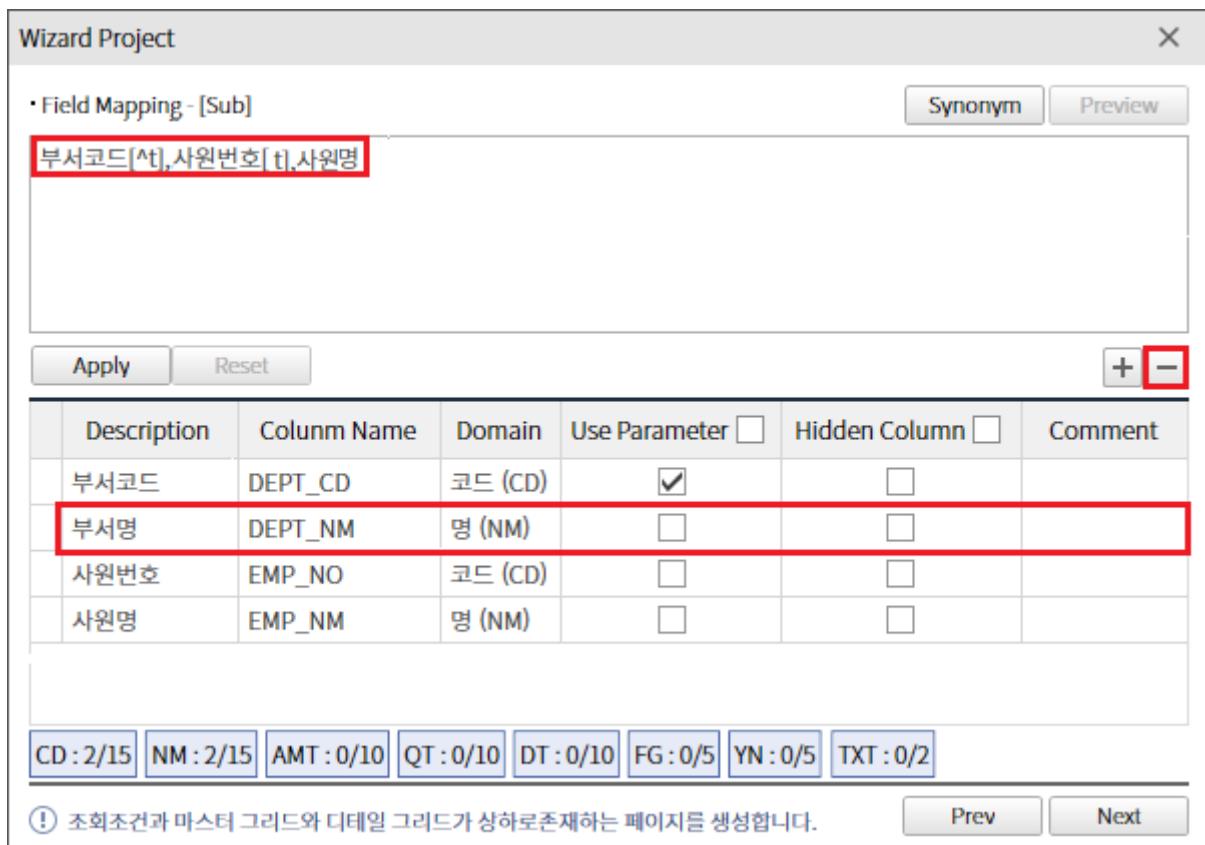


<그림 7-42. Master 그리드 적용 컬럼 입력 화면>

아래 도메인 항목을 보면 CD(코드)는 15개 중 1개를 사용하며, NM(명)도 15개 중 1개를 사용한다는 것을 확인할 수 있다.

### - Sub(디테일) 그리드 적용 컬럼 입력

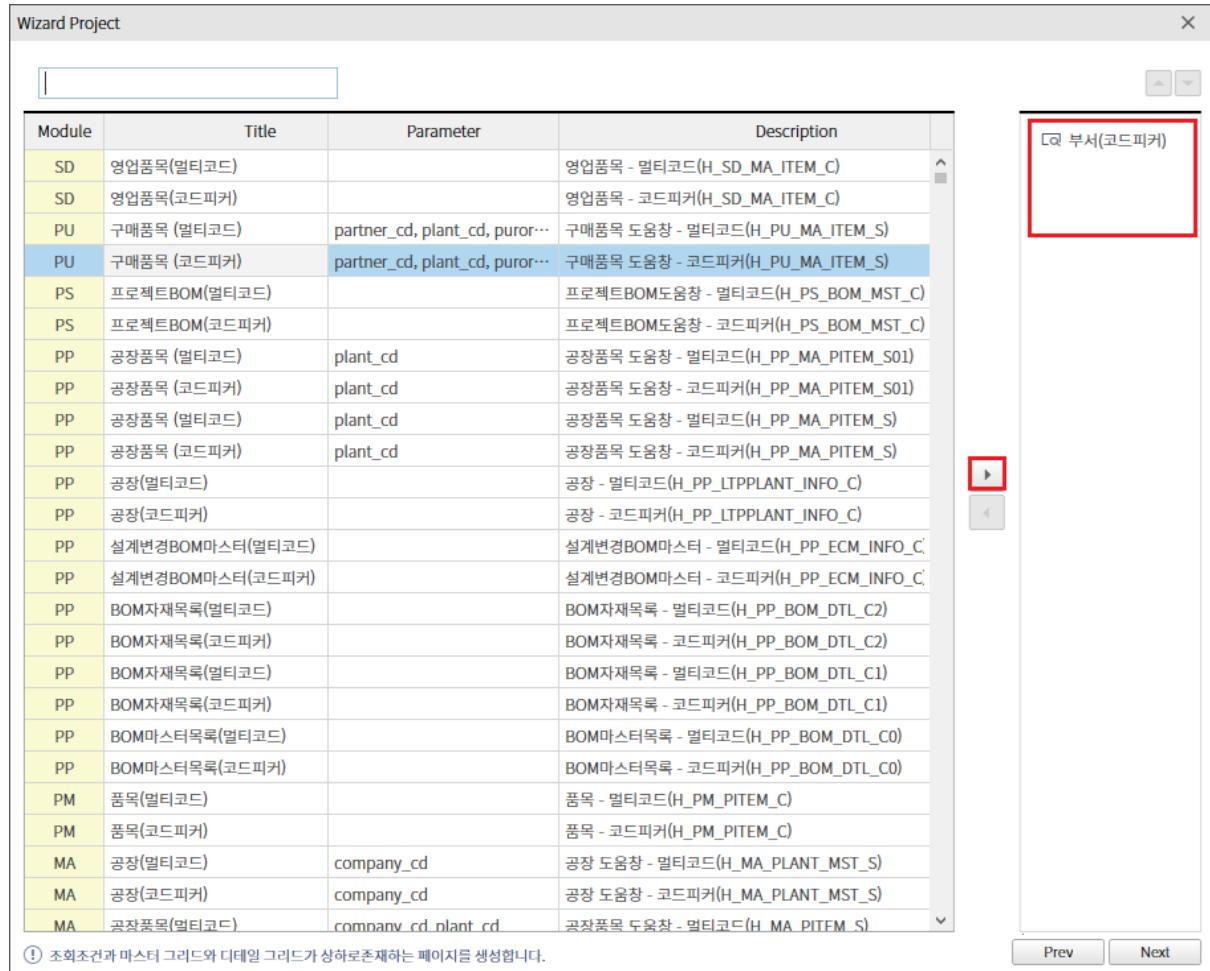
다음으로 Sub 그리드에 적용할 필드 Description을 입력하고 Apply를 눌러서 공용테이블에 사용될 컬럼으로 적용한다. Master 그리드와 마찬가지로 부서코드의 경우 대괄호를 사용하여 옵션을 입력하여 자동으로 use Parameter에 체크가 된다. 또한 부서명도 자동 생성되나 필요없는 컬럼이므로 -(삭제) 버튼을 리스트에서 삭제한다.



<그림 7-43. Sub 그리드 적용 컬럼 입력 화면>

### ③ 조회 조건 컨트롤 선택

다음은 조회 조건 컨트롤을 선택하는 단계이다. 이전 단계에서 Use Parameter로 체크한 Column Name과 조회 조건 컨트롤 id와 비슷한 이름이 있을 경우 자동으로 오른쪽 조회 조건 리스트에 추가되어 있다. 필요없는 경우 더블클릭하거나 ◀ 버튼을 눌러 리스트에서 삭제한다.



<그림 7-44. 조회 조건 컨트롤 선택 화면>

#### ④ Master 그리드 매핑 설정

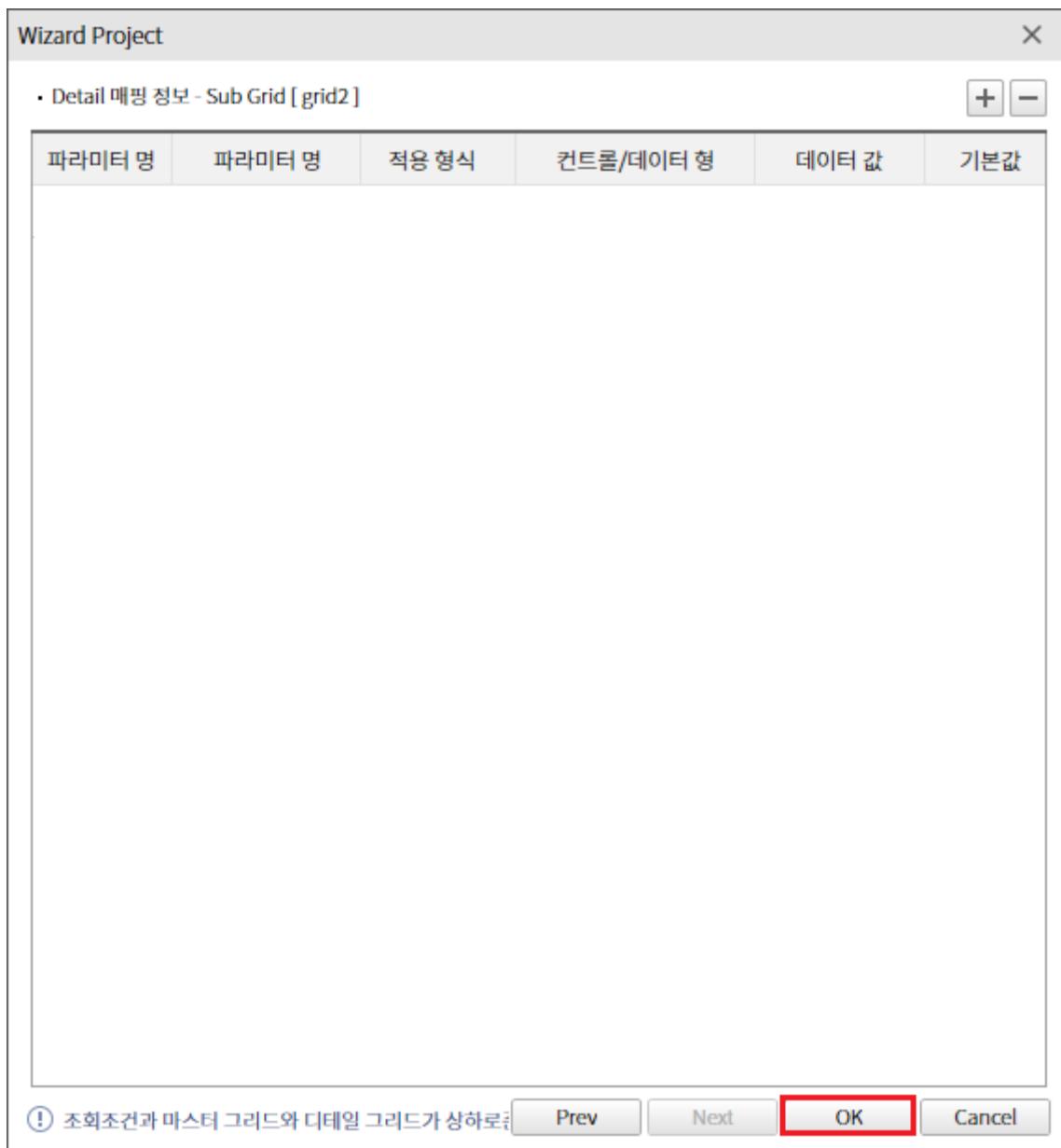
다음은 Master 그리드와 연결되는 데이터소스의 파라미터에 매핑할 조회 조건 컨트롤이나 상수를 설정하는 단계이다. Use Parameter로 체크한 Column Name과 조회 조건 컨트롤 id가 같거나 비슷한 경우 파라미터 값에 따라 자동으로 설정되어 있으며, 편집이 필요한 경우 추가하거나 수정을 하고 다음 단계로 넘어간다



<그림 7-45. Master 그리드 매핑 화면>

#### ⑥ Sub 그리드 매핑 설정 및 마법사 완료

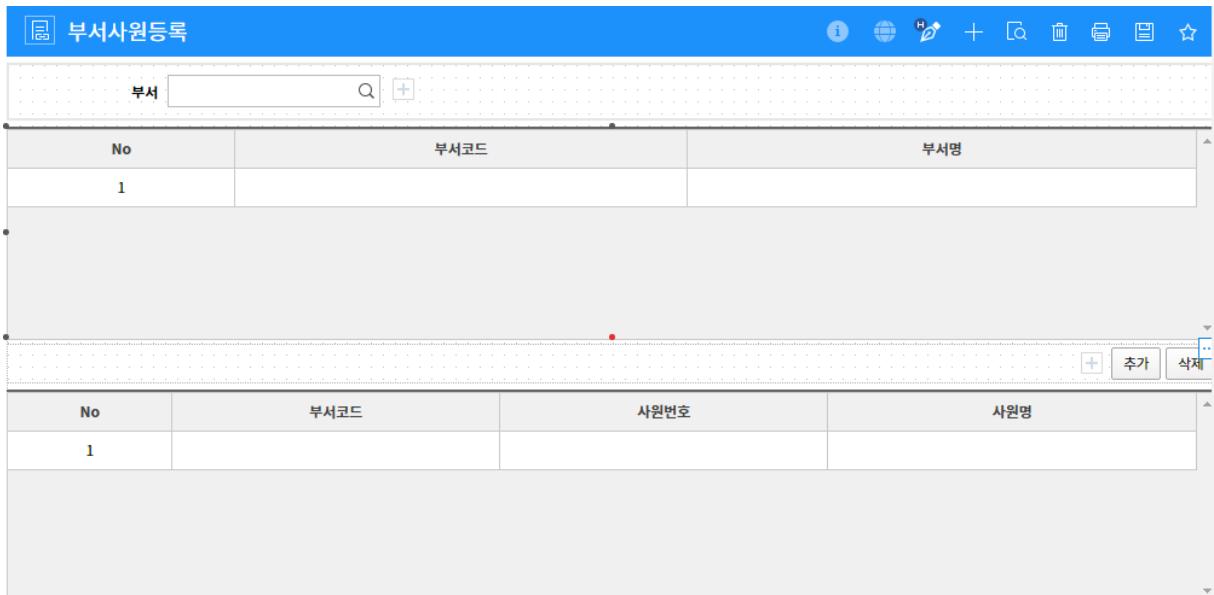
다음은 Sub(Detail) 그리드와 연결되는 데이터소스의 파라미터에 매핑할 조회 조건 컨트롤이나 상수를 설정하는 단계이다. Use Parameter에 체크한 Column Name과 마스터 그리드의 Column Name과 일치하는 경우 자동으로 매핑되어 있으므로 리스트에는 표시되지 않는다. 또한 내부적으로 공용테이블 rfr\_usrdata\_sq 컬럼을 통해 Master와 Sub 그리드 관계를 연결한다. 추가로 매핑이 필요한 경우 우측 상단 +(추가)버튼을 눌러 추가한다.



<그림 7-46. Sub 그리드 매핑 화면 및 마법사 종료 화면>

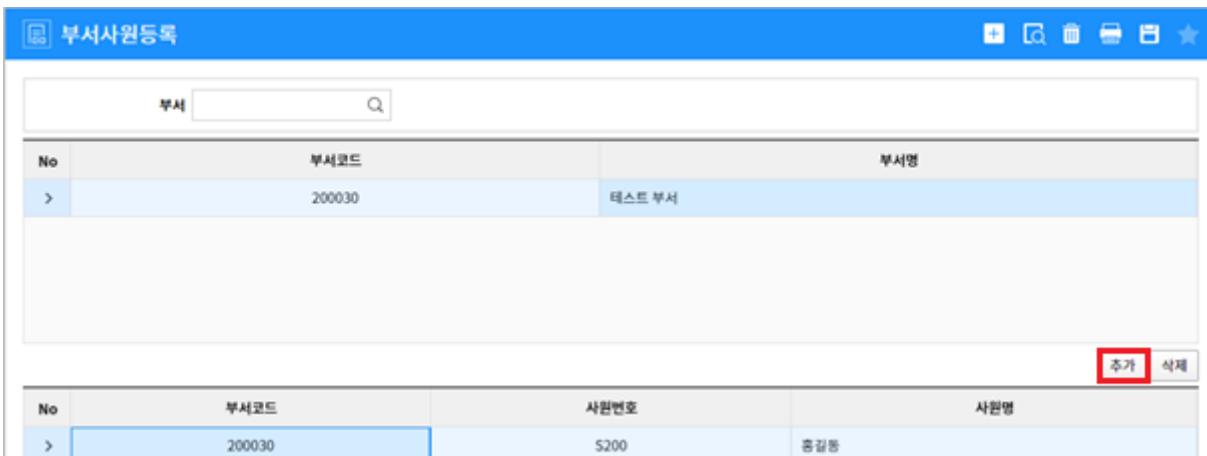
마지막으로 OK 버튼을 눌러 마법사를 종료한다.

기본 화면이 완성되었으며 추가적으로 디테일 설정이 필요한 경우 화면을 수정하거나 스크립트를 편집한다



<그림 7-47. 완성된 화면>

이제 미리보기를 통해 데이터 입력/삭제/수정 후 저장을 확인해 본다. 하단 추가 버튼의 경우 Master의 부서코드가 자동으로 입력된다.



<그림 7-48. 미리보기 화면>

## [제8장] Code Inspection

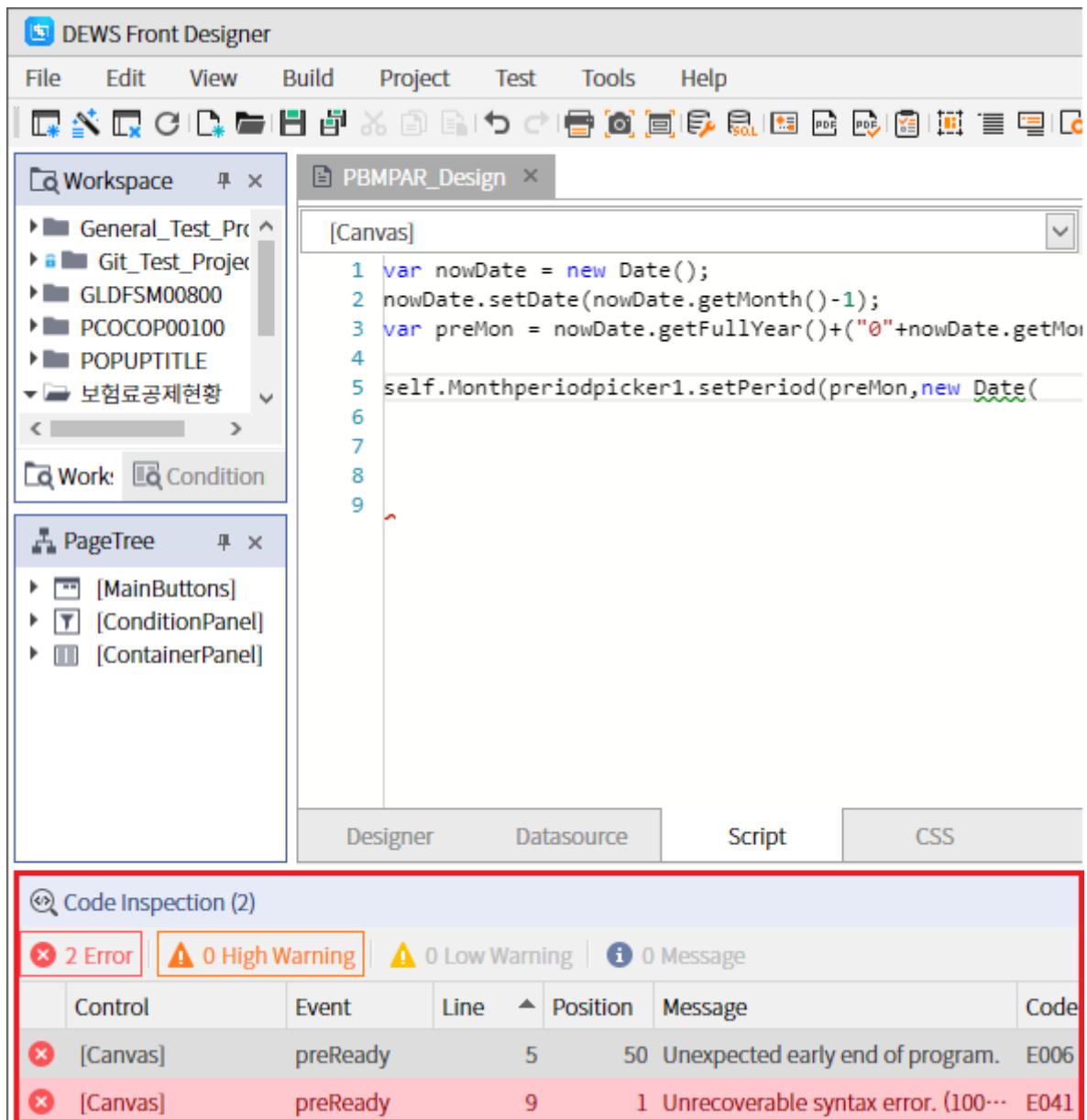
소개 및 이점

환경 설정 및 실행 방법

## 1. 소개 및 이점

Code Inspection 즉, 소스 코드 정적 분석 도구는 개발자가 작성한 소스코드를 분석하여 코드의 문법 오류나 작성 규칙에 맞지 않거나 잘못 구현된 부분들을 체크하는 작업이다.

Code Inspection을 통해 코드의 품질을 높이고 보안 취약점을 제거할 수 있다. 또한 개발자들이 같은 규칙을 사용하기 때문에 코드 분석이 용이하다.



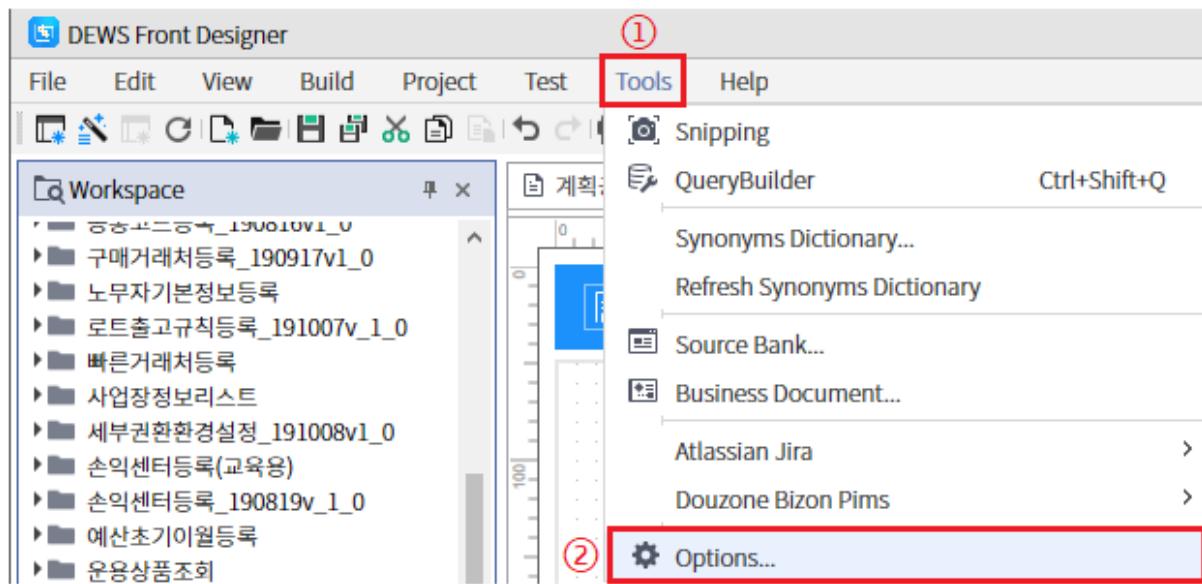
<그림 8-1. Code Inspection 화면>

## 2. 기능 설정 및 실행 화면

### 1) 기능 설정

DFD는 Code Inspection 기능이 추가되어 있으며 JSHint Open 소스를 사용한다. 기능을 활성화하기 위해서는 먼저 옵션에서 체크를 해야 한다.

메뉴의 Tools > Options 클릭한다.



<그림 8-2. Options 실행 화면>

왼쪽 System Setting의 IDE Options을 클릭하고 오른쪽 옵션에서 Enable Code Inspection을 체크 한다. 만약 Node.js<sup>2</sup>가 설치되어 있지 않은 경우 비활성되어 있으며 설치를 진행해야 한다.

- 옵션
  - 마지막 페이지 열기
  - Git 기능(다시 실행해야 반영됨)
  - 빌드 후 같은 실행 경로 지정
  - 워크스페이스 탐색기에서 활성화된 항목 추적
  - 통합 스크립트 에디터 사용(열려있는 문서는 새로 열어야 반영됨)
  - Enable Code Inspection *(Requires JsHint Installation of Node.js)*

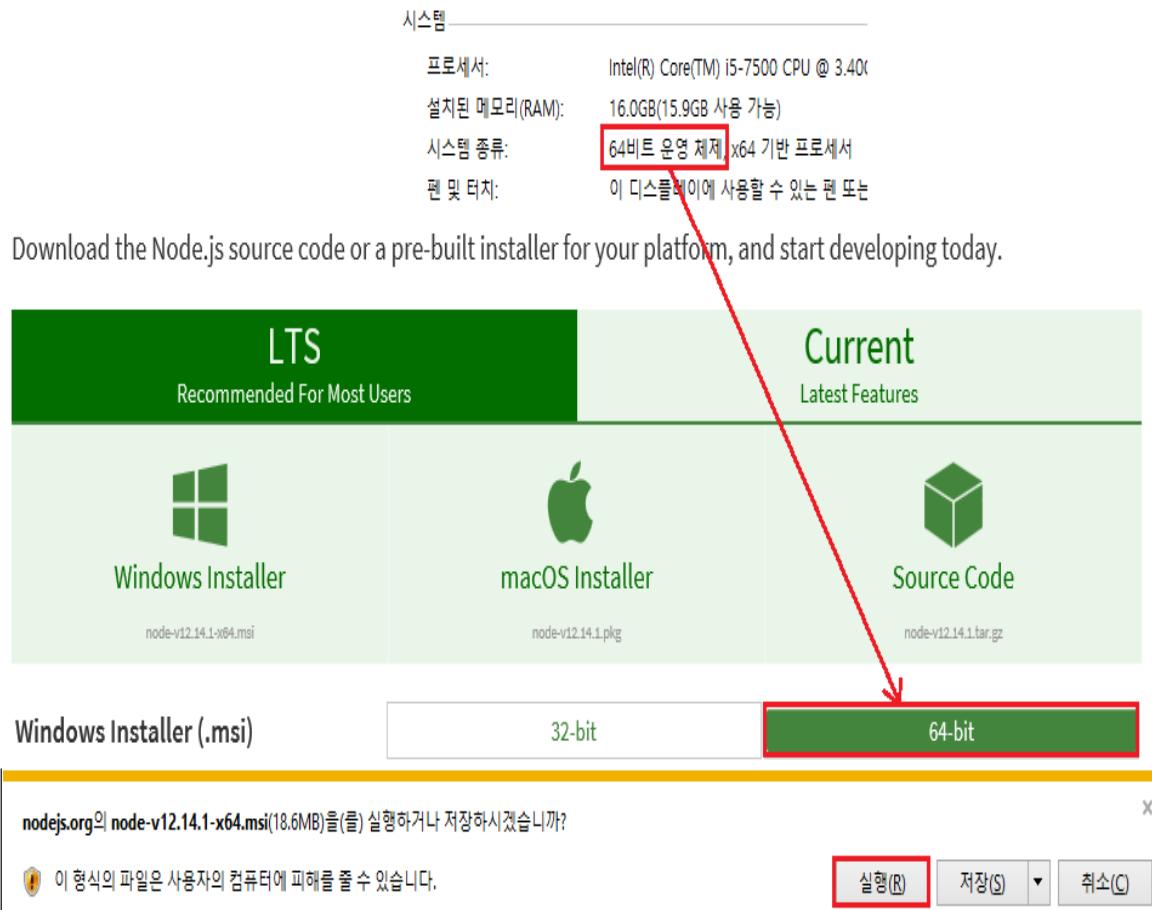
<그림 8-3. Code Inspection 기능 체크 옵션 화면>

<sup>2</sup> 서버 쪽 응용 프로그램 개발에 사용할 수 있는 자바스크립트 기반의 소프트웨어 플랫폼

## 2) Node.js 설치

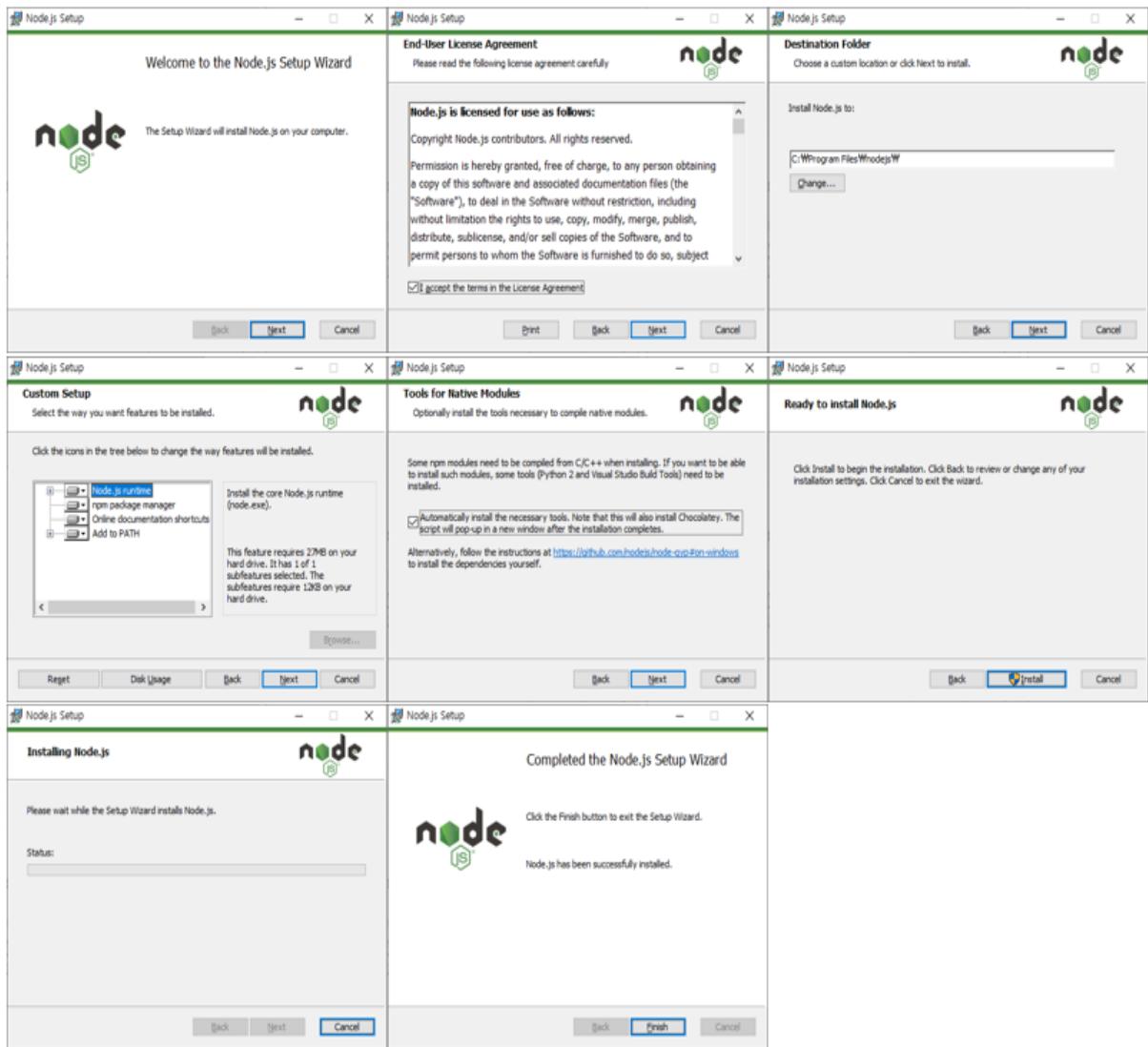
<그림 8-3>의 Requires JsHint Installation of Node.js을 클릭하면 설치 웹 페이지로 이동한다.

사용자에 맞는 윈도우 버전을 확인 후 클릭하여 다운로드가 되면 실행을 눌러 설치를 진행한다.



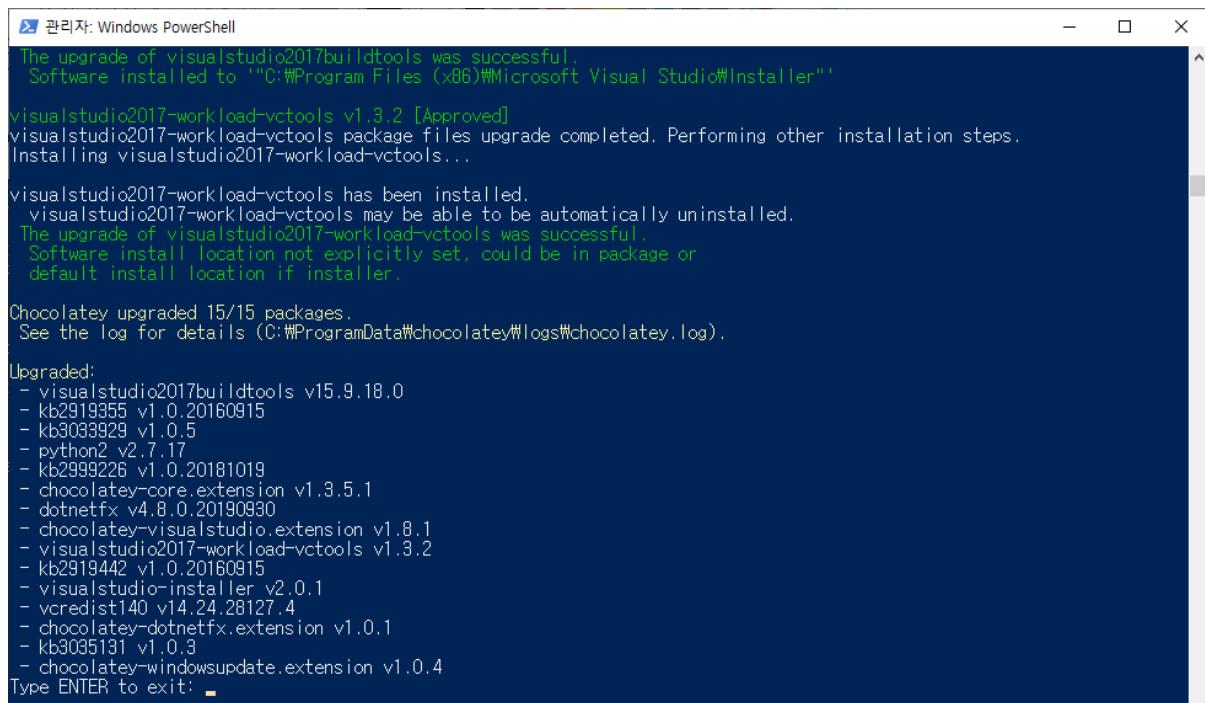
<그림 8-4> Node.js 설치 파일 다운로드 및 실행 화면>

설치 화면이 열리면 Next을 눌러 설치를 진행한다.



<그림 8-5. Node.js 설치 화면>

Windows PowerSell 화면이 열리며 자동으로 설치가 진행된다. 엔터키를 눌러 화면을 빠져 나온다.



```
관리자: Windows PowerShell
The upgrade of visualstudio2017buildtools was successful.
Software installed to "'C:\Program Files (x86)\Microsoft Visual Studio\Installer'"
visualstudio2017-workload-vctools v1.3.2 [Approved]
visualstudio2017-workload-vctools package files upgrade completed. Performing other installation steps.
Installing visualstudio2017-workload-vctools...

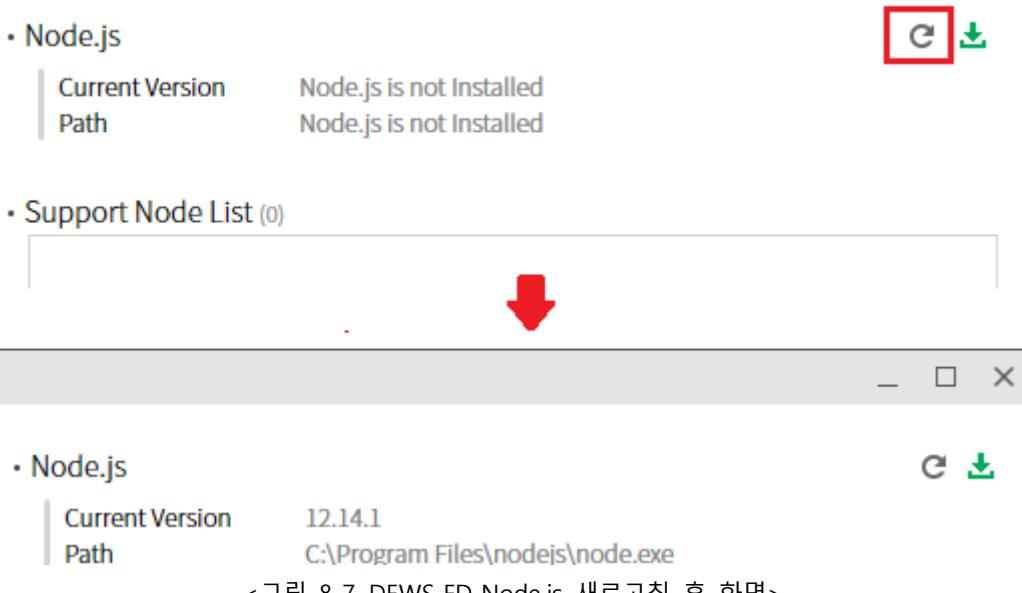
visualstudio2017-workload-vctools has been installed.
visualstudio2017-workload-vctools may be able to be automatically uninstalled.
The upgrade of visualstudio2017-workload-vctools was successful.
Software install location not explicitly set, could be in package or
default install location if installer.

Chocolatey upgraded 15/15 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Upgraded:
- visualstudio2017buildtools v15.9.18.0
- kb2919355 v1.0.20160915
- kb3033929 v1.0.5
- python2 v2.7.17
- kb2999226 v1.0.20181019
- chocolatey-core.extension v1.3.5.1
- dotnetfx v4.8.0.20190930
- chocolatey-visualstudio.extension v1.8.1
- visualstudio2017-workload-vctools v1.3.2
- kb2919442 v1.0.20160915
- visualstudio-installer v2.0.1
- vcredist140 v14.24.28127.4
- chocolatey-dotnetfx.extension v1.0.1
- kb3035131 v1.0.3
- chocolatey-windowsupdate.extension v1.0.4
Type ENTER to exit: -
```

<그림 8-6. PowerSell 화면>

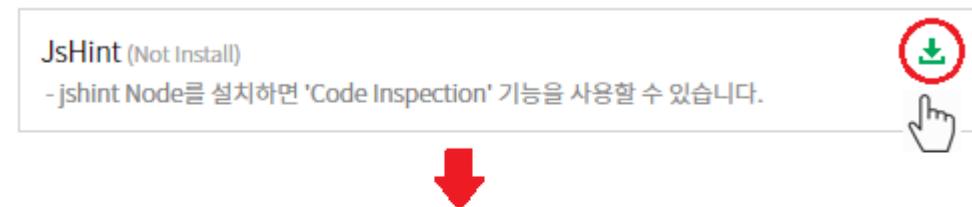
DEWS FD로 이동하여 Node.js Option을 선택하고 오른쪽에서 새로고침을 누르면 설치된 Node.js를 확인할 수 있다.



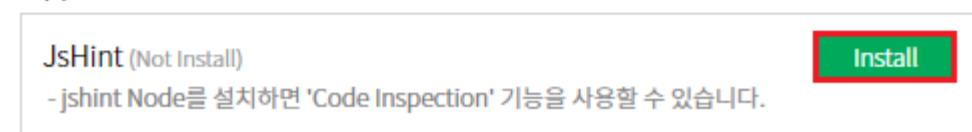
<그림 8-7. DEWS FD Node.js 새로고침 후 화면>

Node.js는 설치는 되었으니 Code Inspection 기능을 사용하기 위해서는 추가적으로 JsHint를 설치해야 한다. Support Node List 에 보면 JsHint가 있으며 설치가 되어 있지 않다(Not Install). 오른쪽 담기 그림에 마우스를 올려놓으면 Install 버튼으로 이미지가 바뀌며 클릭하여 설치를 진행한다.

- Support Node List (2)



- Support Node List (2)



<그림 8-8. JsHint 추가 설치 화면>

npm(node Package manager) 화면이 열리며 자동으로 설치한다.

The screenshot shows a terminal window titled 'npm'. The command 'extract: jshint: sill extract jshint@latest ext' is being typed into the terminal. The terminal window has a dark background and light-colored text.

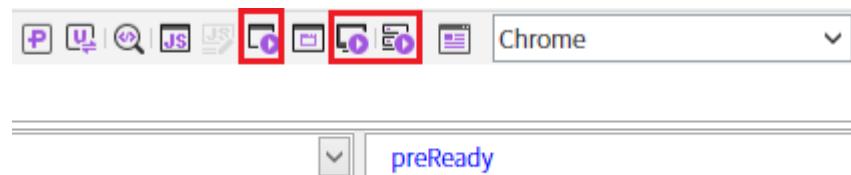
<그림 8-9. npm 화면>

다시 IDE Options을 클릭하고 오른쪽 옵션에서 Enable Code Inspection에 체크한다.



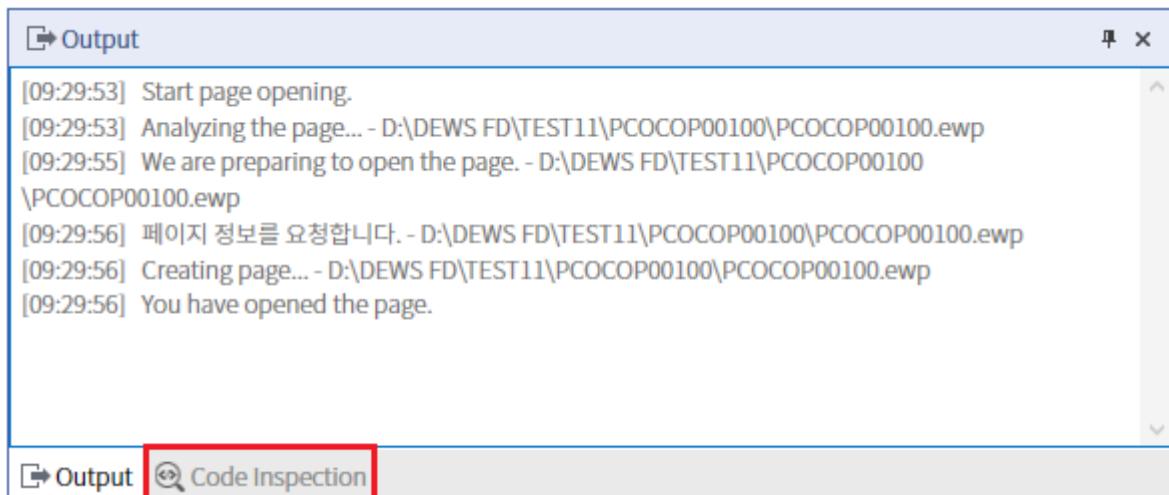
<그림 8-10. 옵션에 Enable Code Inspection 체크 화면>

Code Inspection을 사용하기 위해서는 반드시 빌드를 눌러야 한다.



<그림 8-11. 툴바 Build 실행 아이콘>

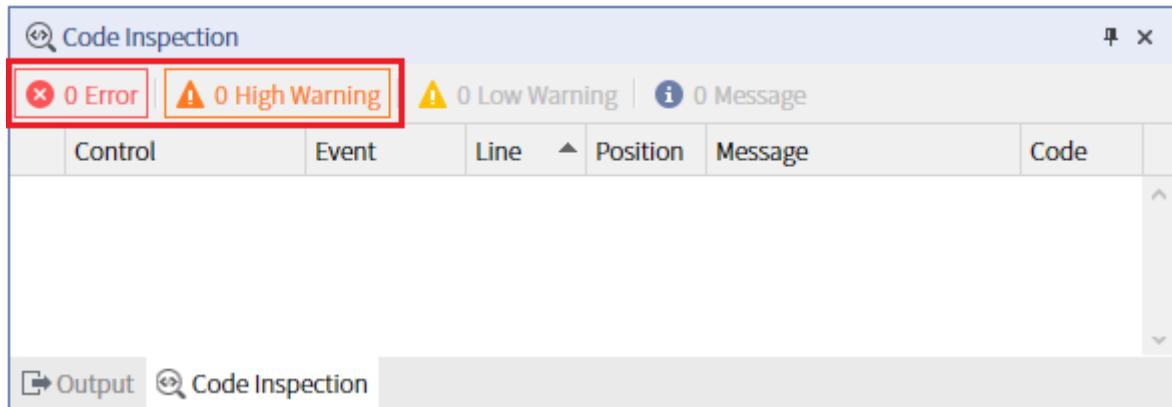
왼쪽 하단 Output 화면에 Code Inspection 탭이 생성되고, 탭을 클릭하면 Code Inspection 화면으로 이동한다.



<그림 8-12. Code Inspection 탭 화면>

### 3) Code Inspection 화면

상단은 4가지 타입의 아이콘 이미지, 오류 개수, 종류로 되어 있으며, 클릭 시 아래 결과 리스트에서 숨기거나 보이게 하는 필터링을 할 수 있다. Error와 High Warning은 디폴트로 리스트에 보여준다.



<그림 8-13. Code Inspection 탭 화면>

| 아이콘 타입       | 설명                                    |
|--------------|---------------------------------------|
| Error        | 오류. 반드시 수정 해야 함                       |
| High Warning | 높은 경고. 미리보기를 포함한 런타임에 오류가 발생할 수 있는 코드 |
| Low Warning  | 낮은 경고. 오류는 발생하지 않지만 코드 수정을 권함         |
| Message      | 메세지. 일반적인 알람                          |

<표 8-1. 아이콘 4가지 타입 설명>

아래에는 분석 결과 제목과 리스트가 있다

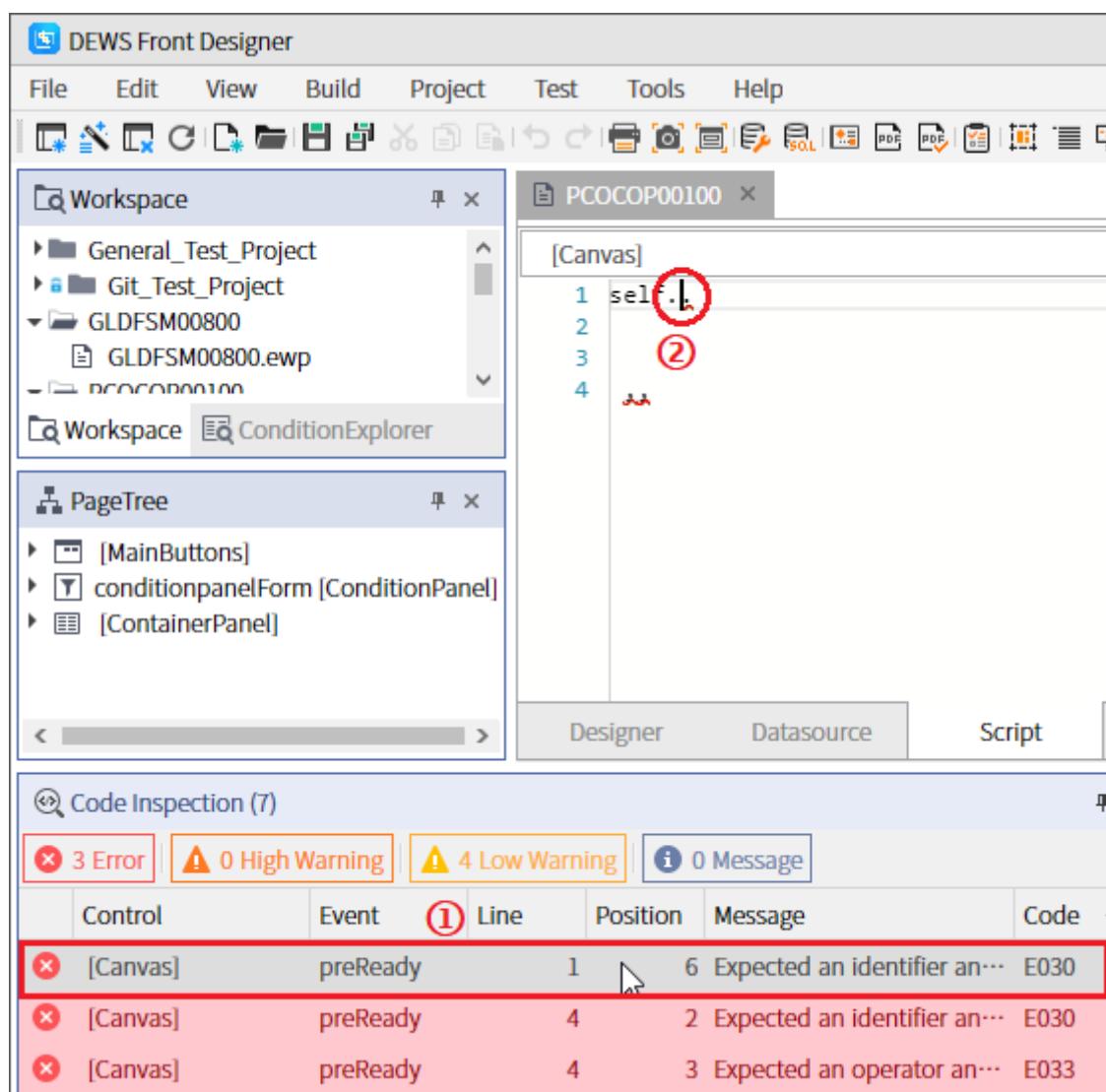


<그림 8-14. 분석 결과 화면>

| Title    | 설명        |
|----------|-----------|
| Control  | 발생한 컨트롤   |
| Event    | 발생한 이벤트   |
| Line     | 라인 위치     |
| Position | 줄 위치      |
| Messasge | 분석 결과 메세지 |
| Code     | 메세지 코드    |

<표 8-2. 분석 결과 타이틀 설명>

항목을 더블 클릭 하면 오류가 발생한 Script 부분의 라인과 줄로 이동하므로 편집을 빠르게 할 수 있다. 오류가 난 부분은 빨간색 물결 표시로 되어 있다.



<그림 8-15. 항목 더블 클릭 시 스크립트 라인 이동 화면>

## [제9장] 예제 만들어 보기

전표삭제현황(Basic Grid Page)

자금수지표(Up and Down Grid Page)

운용상품조회(CardList + ModityInfo Page)

부서인원입력(CRUD- Local)

빠른거래처 등록(CRUD)

## 1. 전표삭제현황

### 1) 개발 화면

<그림 9-1>은 개발 완료된 전표삭제현황 화면이다. 회계단위 및 회계일은 필수 조회 조건이며, 프로그램을 실행했을 때 회계단위, 작성부서, 작성자는 로그인한 정보를, 회계일은 해당월에 첫번째 일자와 오늘 일자를 기본값으로 설정되어 있다.

| No | 회계단위 | 작성부서   | 작성자 | 회계일        | 순번 | 삭제자명 | 삭제일시       | 전표번호           |
|----|------|--------|-----|------------|----|------|------------|----------------|
| 1  | CE부문 | 채널사업본부 | 김인영 | 2018-05-01 | 1  | 김인영  | 2018-04-27 | FI201805010001 |
| 2  | CE부문 | 채널사업본부 | 김인영 | 2018-05-01 | 1  | 김인영  | 2018-04-27 | FI201805010002 |

<그림 9-1. 전표삭제현황 개발화면>

<필수조회 조건>

<초기값 설정>

<그림 9-2. 전표삭제현황 필수 조회 조건 및 초기값 설정>

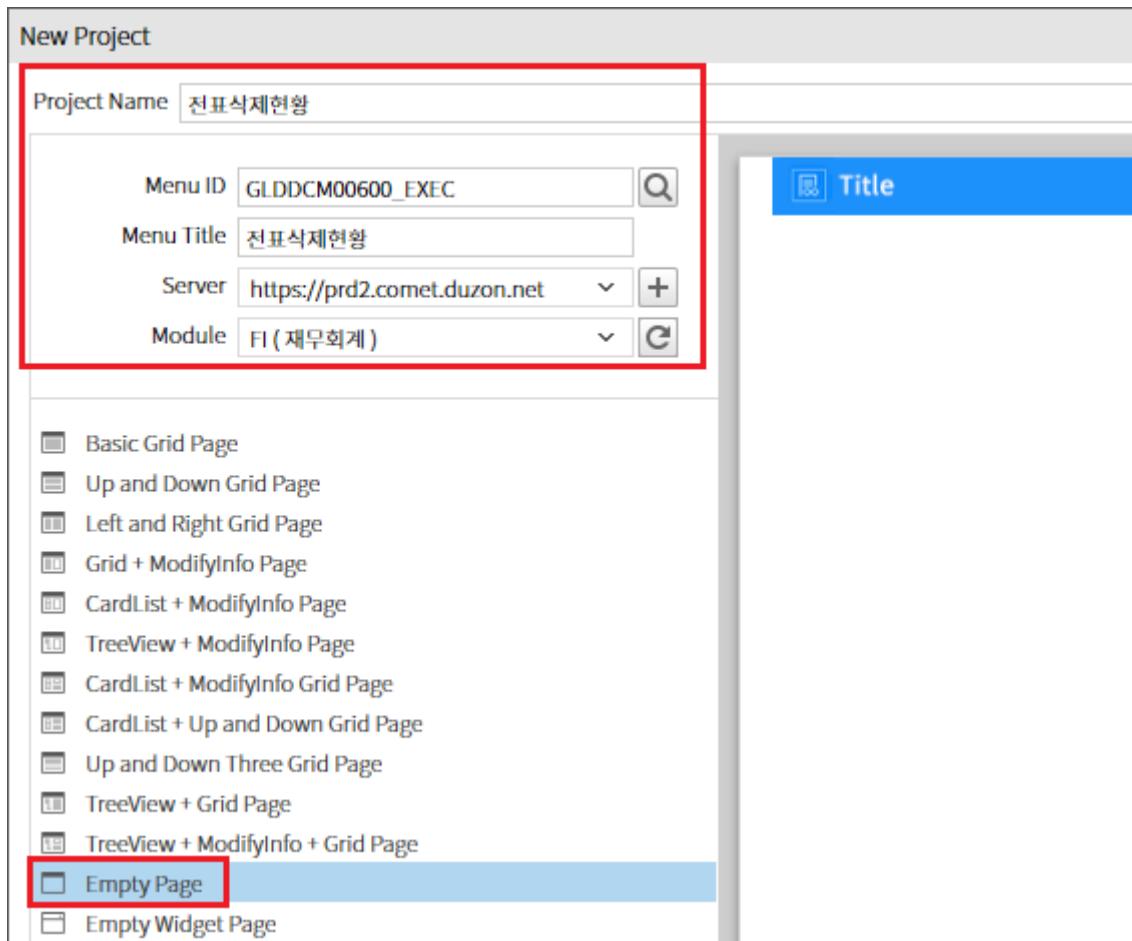
### 2) 화면 디자인

새 프로젝트를 생성한다. (File ▶ New ▶ Project... 또는 툴바의 )

|                      |                                                                            |
|----------------------|----------------------------------------------------------------------------|
| Project Name (프로젝트명) | 전표삭제현황 입력                                                                  |
| Menu ID (메뉴ID)       | GLDDCM00600_EXEC 입력                                                        |
| Menu Title (메뉴 제목)   | 전표삭제현황 입력                                                                  |
| Server (서버 선택)       | <a href="https://prd2.comet.duzon.net">https://prd2.comet.duzon.net</a> 선택 |
| Module (모듈 선택)       | FI (재무회계) 선택                                                               |
| 템플릿(template)        | Empty Page 선택                                                              |

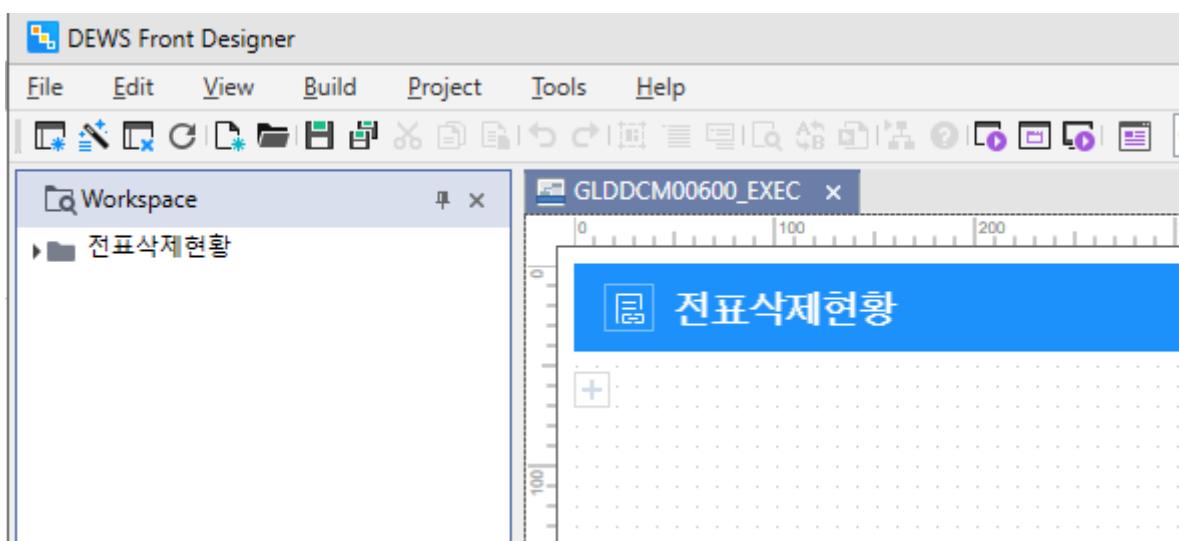
<표 9-1. 새 프로젝트 생성 입력>

<그림 9-1> 정보를 입력 후 OK 버튼을 클릭하여 프로젝트를 생성한다.



<그림 9-3. New Project 입력 후 OK 버튼 클릭>

전표삭제현황 프로젝트가 Workspace에 생성된다.

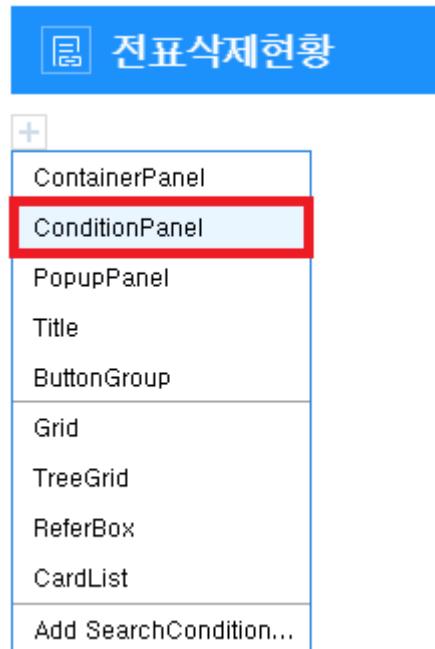


<그림 9-4. 전표삭제현황 프로젝트 생성>

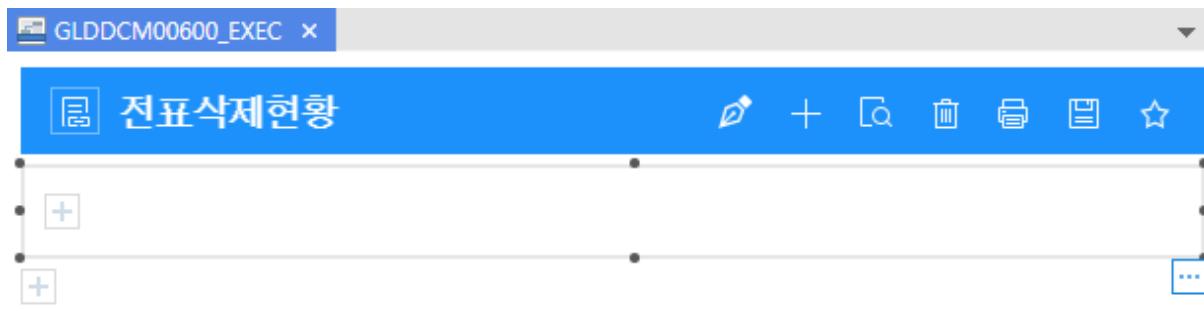
조회 조건에 해당하는 컨트롤들은 ConditionPanel, 데이터를 보여주는 그리드 컨트롤은 ContainerPanel 을 먼저 생성하고 각각 생성한다. 먼저 조회 조건에 해당하는 컨트롤을 그린다.



버튼을 누르고 ConditionPanel 을 선택하여 ConditionPanel 을 생성한다.

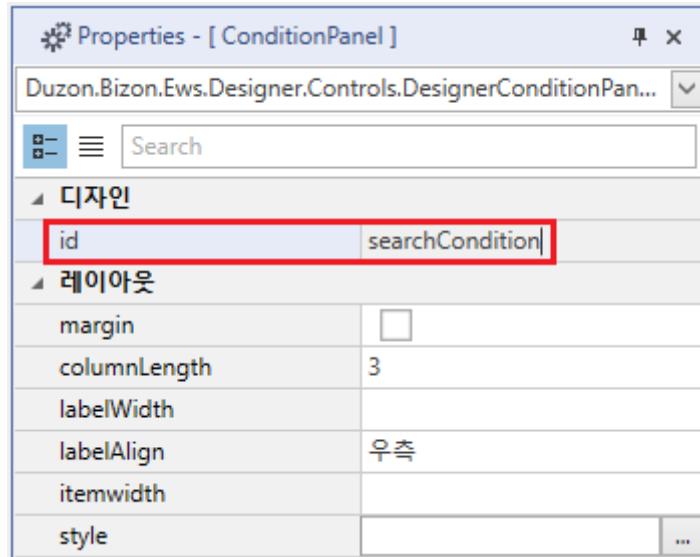


<그림 9-5. ConditionPanel 선택>



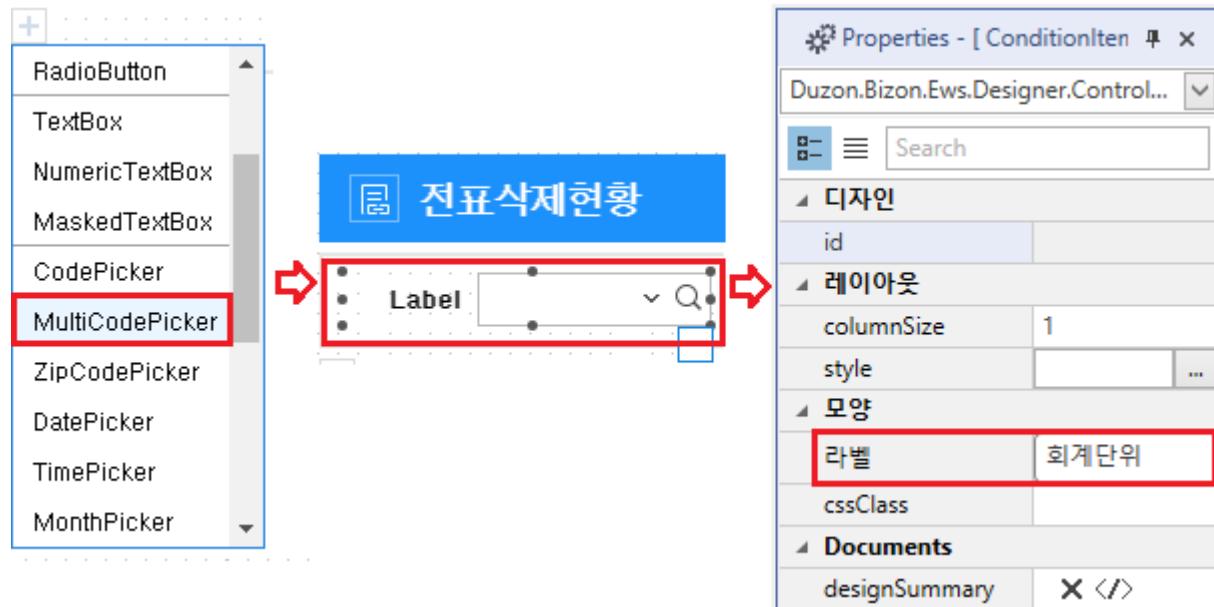
<그림 9-6. ConditionPanel 생성>

프로그램 실행 후 조회 시 필수 조회 조건 체크를 하기 위해 ConditionPanel 선택 후 ID 속성을 searchCondition 으로 입력한다.



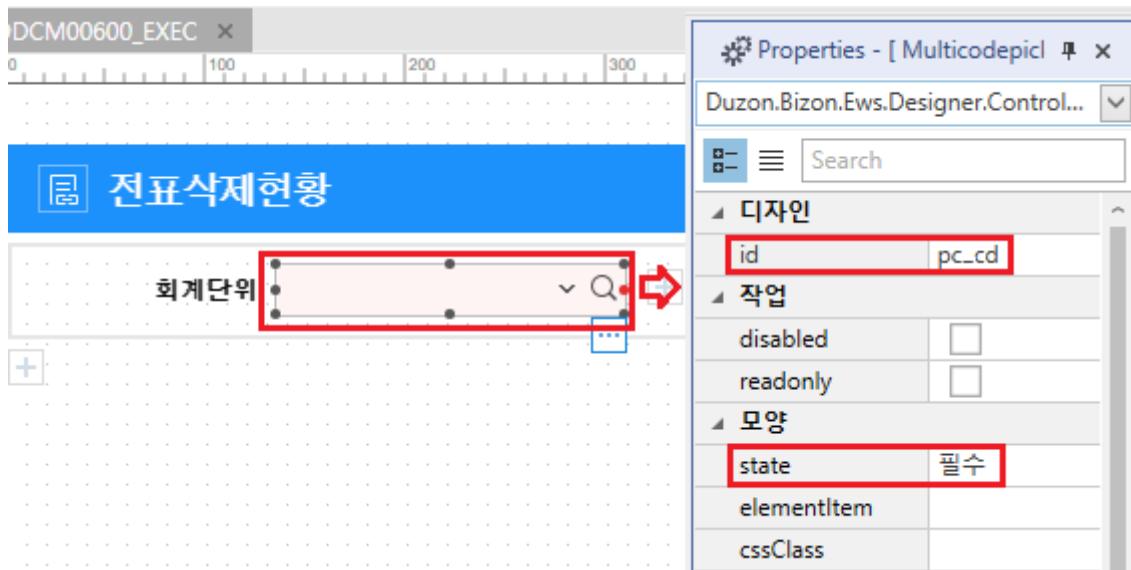
<그림 9-7. ConditionPanel ID 속성 입력>

회계단위 조회 조건을 추가하기 위해 ConditionPanel 내의 를 눌러 MultiCodePicker(다중 도움창)를 선택한다. Label 이름을 선택하면 ConditionItem 컨트롤이 선택되며, 라벨 속성에서 회계단위를 입력한다. 다른 방법으로 조회 조건 탐색기에서(ConditionExplorer) 불러올 수 있다. Add SearchCondition 을 눌러 회계단위를 검색하고 FI/회계단위(멀티코드)를 더블클릭하고 선택한다(참조).



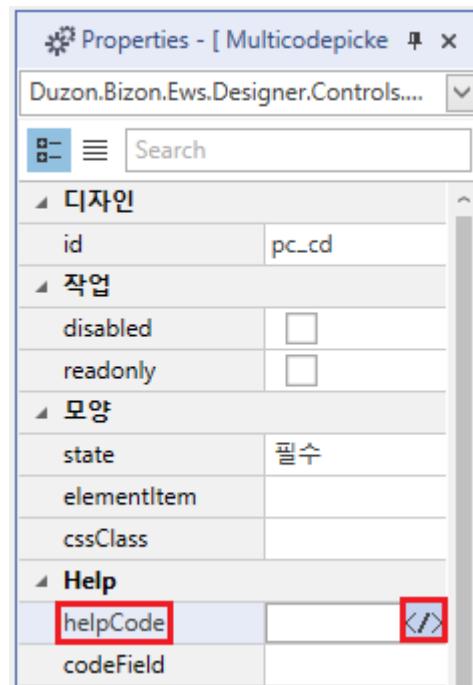
<그림 9-8. 회계단위 조회 조건 추가>

Multicodepicker 를 선택하고 id 는 pc\_cd 를 입력, state 는 필수를 선택한다.



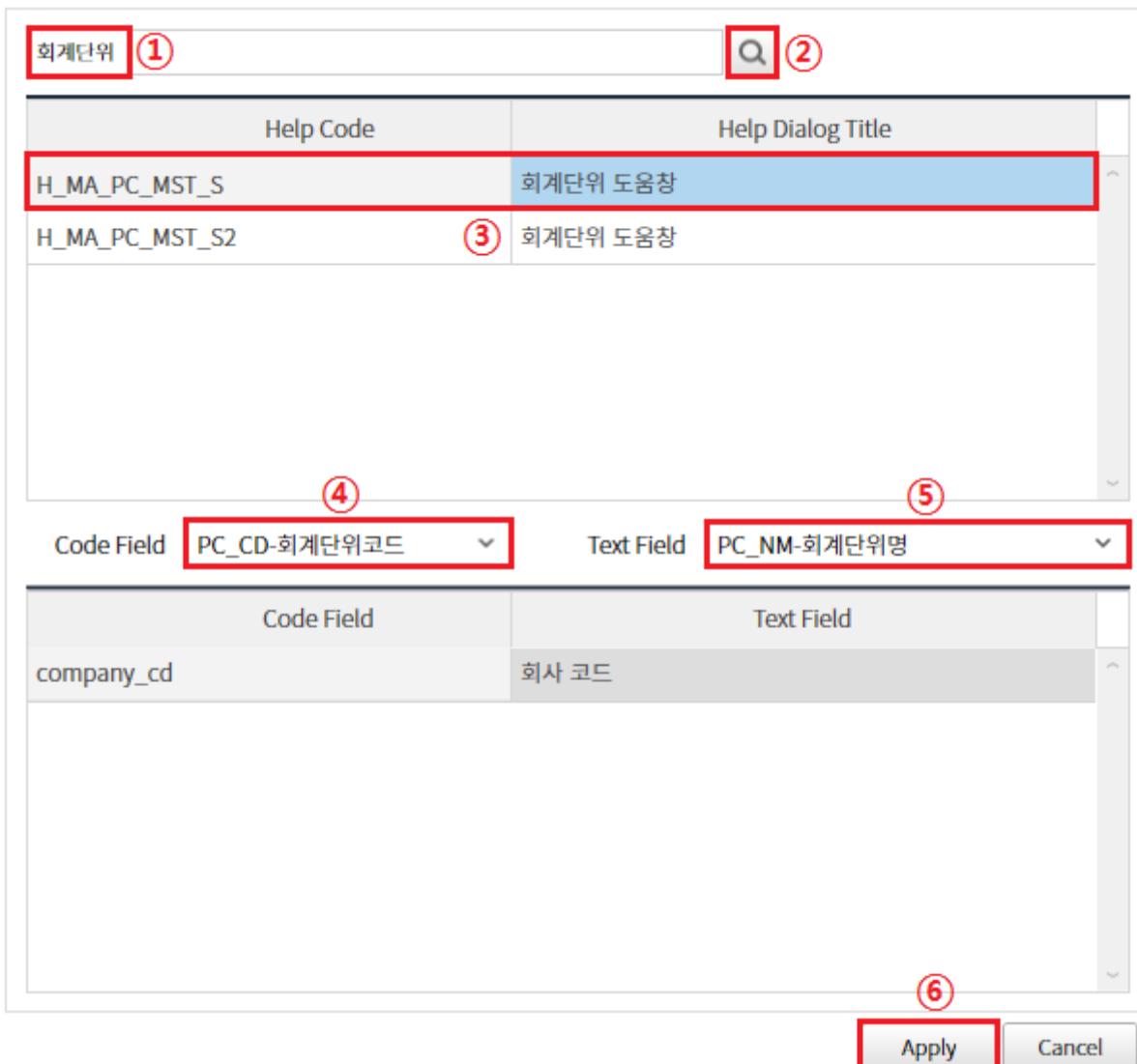
<그림 9-9. 회계단위 id, state 속성 변경>

회계단위 도움창을 선택하기 위하여 helpCode 속성에 스크립트 버튼 </>을 누른다.



<그림 9-10. helpCode 스크립트 선택>

Code Help 화면에서 검색어를 ① 회계단위로 입력하고, 엔터키나 ② Search 를 눌러 검색한다. 2 개의 Help Code 가 검색되며, 끝에 S2 는 같은 회사에 여러 개 회계 법인이 존재할 때 선택하며, 여기서는 ③ H\_MA\_PC\_MST\_S 를 선택한다. 다음 Code Field 는 ④ PC\_CD-회계단위코드를 Text Field 는 ⑤ PC\_NM-회계단위명을 선택하고 ⑥ Apply 를 클릭한다.



<그림 9-11. 도움창코드 선택>

helpCode, codeField, textField 속성에 값이 입력되어 있다.

| Help      |                   |
|-----------|-------------------|
| helpCode  | H_MA_PC_MST_S </> |
| codeField | PC_CD             |
| textField | PC_NM             |

<그림 9-12. helpCode 확인>

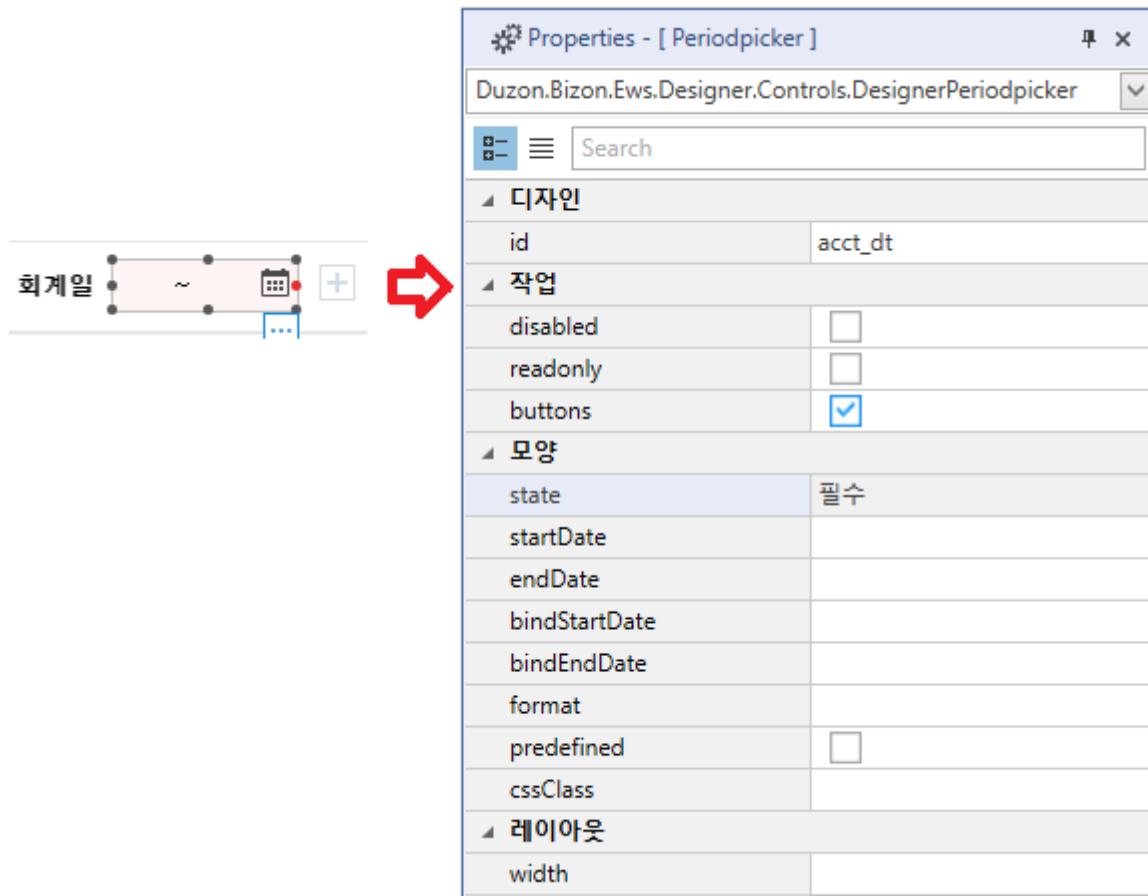
스크롤을 내려서 helpParams 의 ① 파라미터 매핑을 눌러 ② 컨트롤/데이터형을 number, 데이터 값에 self.user.companyCode(로그인한 사용자의 회사코드)를 입력하고 확인을 누른다. 기본적으로 회사코드(cd\_company)로 필요로 하는 helpCode의 경우 필수적으로 입력을 해야 조회 버튼(돋보기)을 눌렀을 때 조회가 되며, 필요없는 경우라면 파라미터를 삭제한다. number를 선택하는 이유는 string을 선택했을 때 양쪽에 문자를 나타내는 “(작은따옴표)가 붙어 문자형으로 인식하기 때문이다. 다음 ③ helpSize를 ‘대’로 선택한다.

The screenshot shows three panels illustrating the configuration of helpParams and helpSize.

- Top Panel:** A table titled "Help" with three rows: helpCode (H\_MA\_PC\_MST\_S), codeField (PC\_CD), and textField (PC\_NM). The helpCode row has a value with a double quotes icon (</>).
- Middle Panel:** A table titled "파라미터 매핑" (Parameter Mapping) with one row: helpParams. The "Mapping Information" section shows a mapping for company\_cd: 적용 형식 (Type) is number (②), and 데이터 값 (Value) is self.user.companyCode. The helpParams row has a value with a double quotes icon (</>).
- Bottom Panel:** A dropdown menu for helpSize with options: 중 (Medium) (③ selected), 사용자지정 (Custom), 소 (Small), 중 (Medium), 대 (Large) (selected), and 특대 (Extra Large). The large option 대 is highlighted with a red box.

<그림 9-13. helpSize 선택>

같은 방법으로 를 눌러 회계일은 Periodpicker(날짜(기간) 도움창)을 선택하고, Label 을 클릭하여 라벨을 회계일을 입력하고, PeriodPicker 선택 후, Periodpicker의 id는 acct\_dt, state는 필수를 선택한다.



<그림 9-14. 회계일 컨트롤 설정>

다음으로 작성부서는 를 눌러 Muticodepicker(멀티 도움창)을 선택하고, 라벨은 작성부서를 입력하고, Muticodepicker 를 선택 후 id 는 dept\_cd 를 입력한다. helpCode 속성에서 스크립트 버튼을 </> 눌러 Code Help 선택 창에서 부서를 입력하고 검색한다. 검색 결과 Help Code 에서 H\_MA\_DEPT\_MST\_S 를 선택하고 Code Field 는 DEPT\_CD-부서코드를 Text Field 는 DEPT\_NM-부서명을 각각 선택하고 Apply 를 누른다.



<그림 9-15. 작성부서 helpCode 선택>

스크롤을 내려서 helpParmas 의 파라미터 맵핑 을 눌러 company\_cd 파라미터의 경우 회계단위와 같이 컨트롤/데이터형을 number, 데이터값을 self.user.companyCode 를 입력하고 확인을 누른다. 다음 helpSize 를 '대'로 선택한다.

파라미터 매핑

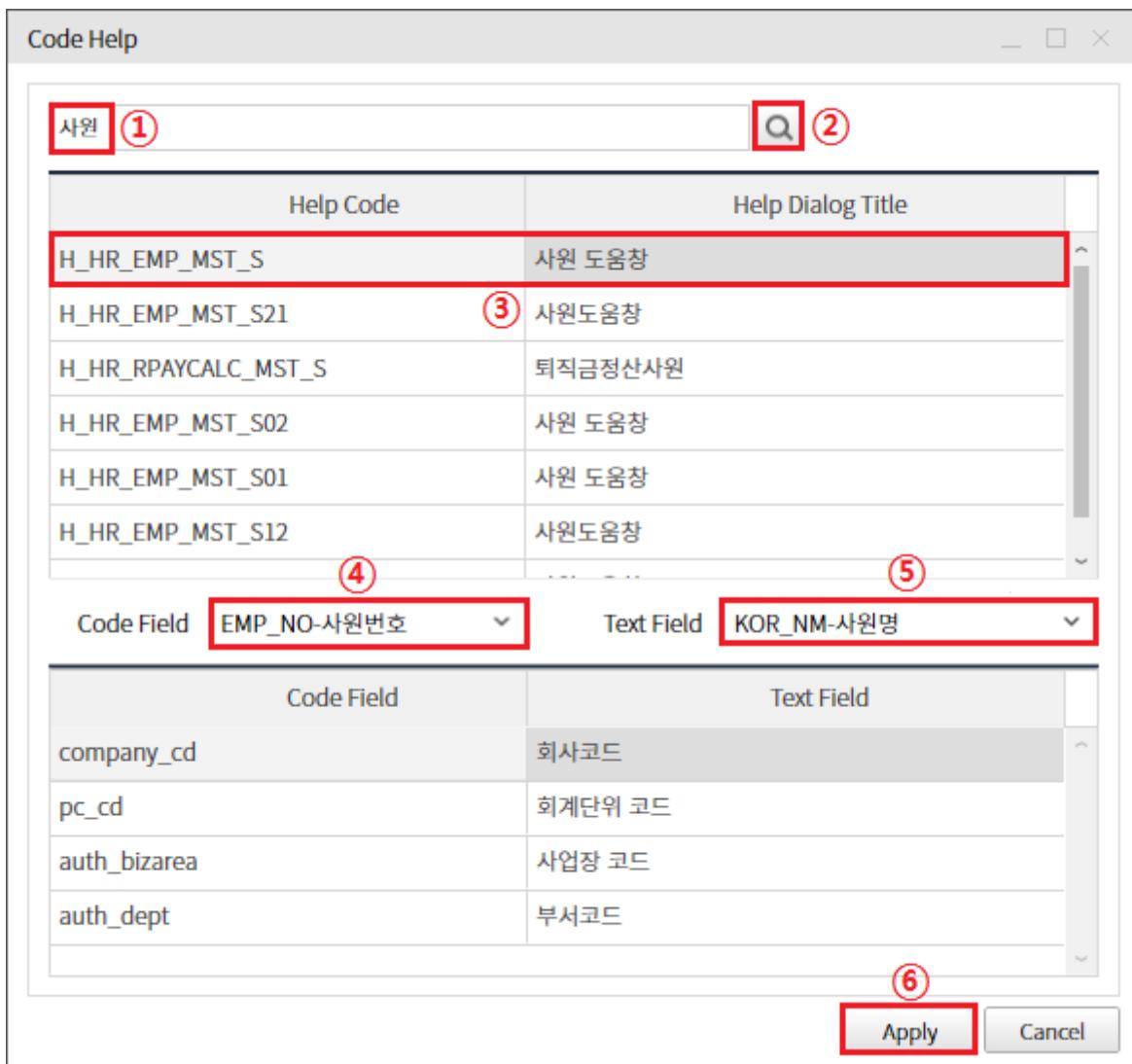
• 매핑 정보

| 파라미터 명       | 적용 형식 | 컨트롤/데이터 형 | 데이터 값                 |
|--------------|-------|-----------|-----------------------|
| company_cd   | 상수    | number    | self.user.companyCode |
| auth_bizarea | 상수    |           |                       |
| auth_pc      | 상수    |           |                       |
| auth_dept    | 상수    |           |                       |

helpSize ③ 대

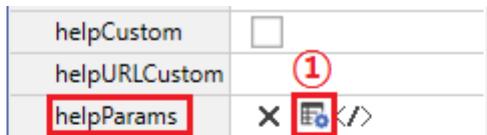
<그림 9-16. helpSize 선택>

마지막 조회 조건인 작성자는 CodePicker(단일 도움창)을 선택한다. 조회 조건의 경우 3 열이 최대이므로 아래 행으로 자동 배치 된다. 라벨은 작성자를 입력하고, CodePicker를 선택 후 id를 emp\_no로 입력한다. helpCode 속성에서 스크립트 버튼을 </>눌러 Code Help 선택 창에서 사원을 입력하고 검색한다. 검색 결과 Help Code에서 H\_HR\_EMP\_MST\_S를 선택하고 Code Field는 EMP\_NO-사원번호를 Text Field는 KOR\_NM-사원명을 각각 선택하고 Apply를 누른다.



<그림 9-17. 작성자 helpCode 선택>

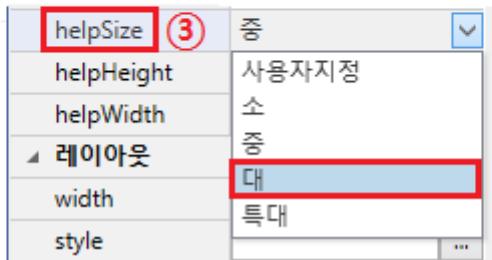
helpCode, codeField, textField 속성에 값이 입력되어 있다. 스크롤을 내려서 helpParams 의 파라미터 매핑  을 눌러 company\_cd 파라미터의 경우 이전 조건과 같이 컨트롤/데이터형을 number, 데이터값을 self.user.companyCode 를 입력하고 auth\_dept 파라미터는 조회조건에 있는 작성부서코드 값을 받아서 해당 부서에 해당하는 직원만 보여주므로 적용형식을 컨트롤, 컨트롤/데이터형을 dept\_cd, 데이터값은 codes 를 선택한다. 다음 helpSize 를 '대'로 선택한다.



**파라미터 매핑**

• 매핑 정보

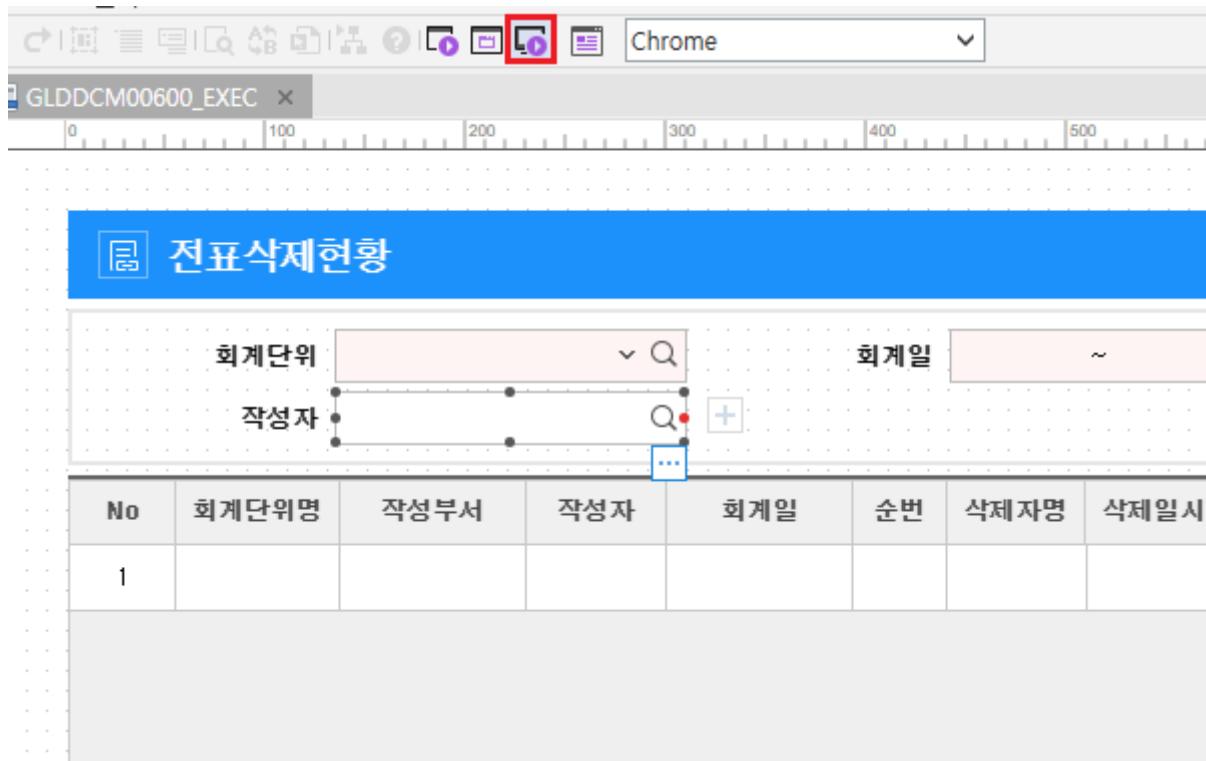
| 파라미터 명       | 적용 형식 | 컨트롤/데이터 형 | 데이터 값                 |
|--------------|-------|-----------|-----------------------|
| company_cd   | 상수    | number    | self.user.companyCode |
| pc_cd        | 상수    |           |                       |
| auth_bizarea | 상수    |           |                       |
| auth_dept    | 컨트롤   | dept_cd   | codes                 |

helpSize ③ 중  
helpHeight 사용자지정  
helpWidth 소  
레이아웃 중  
width 대  
style 특대

<그림 9-18. helpCode 확인 및 helpSize 설정>

조회 최종화면이다. 툴바에서  Build & Preview(Ctrl+B)버튼을 눌러 미리보기를 실행한다.



<그림 9-19. Built & Preview 실행>

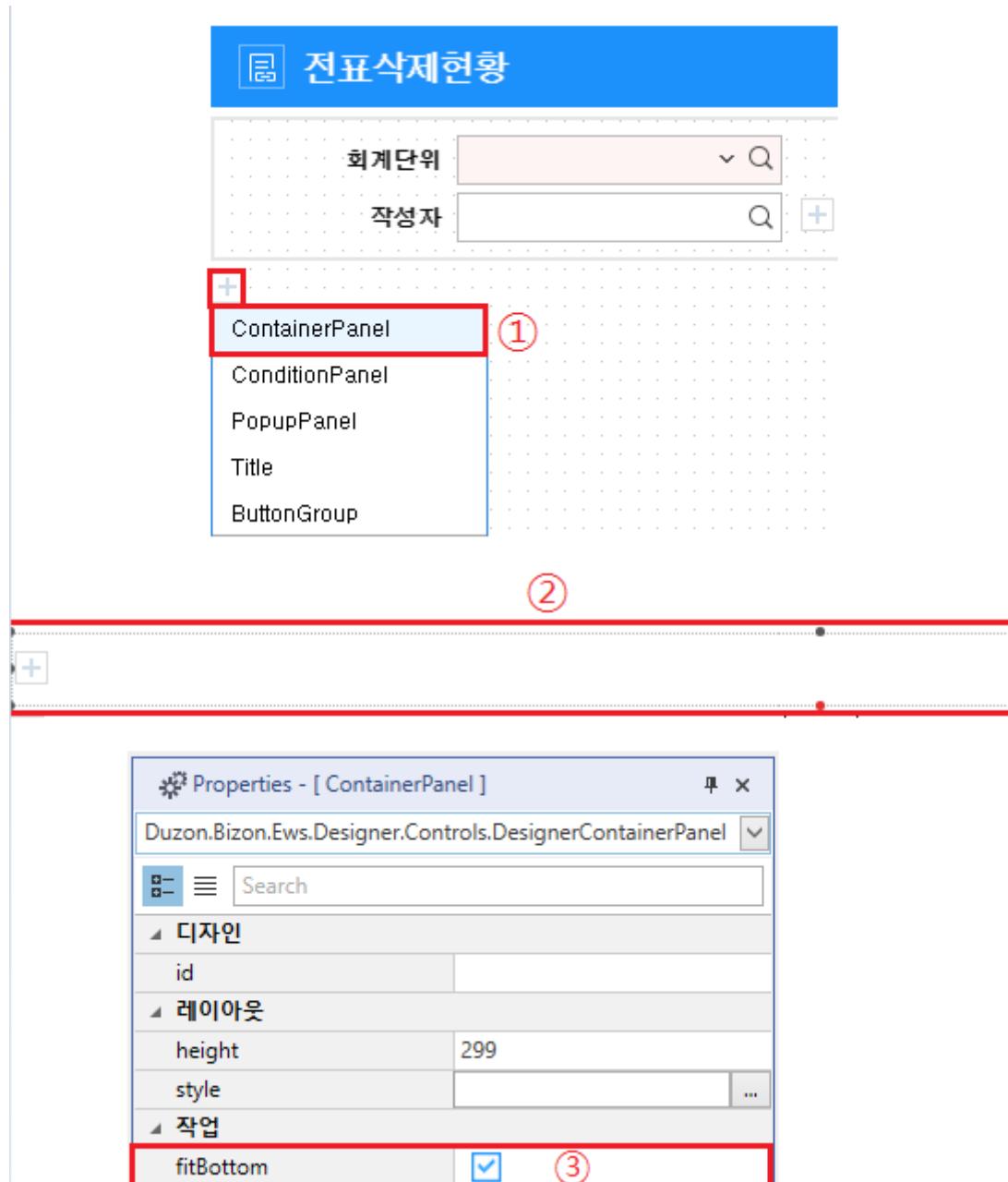
웹 브라우저가 실행되며, 조회 조건들을 열어 확인한다.

This screenshot shows the same search interface as in Figure 9-19, but with a modal dialog box overlaid. The modal is titled '회계단위 도움창' and contains a date input field '종료일' set to '2018-08-28' with a calendar icon. Below this is a table with columns: No, 코드, 회계단위, and 부서명. The table rows are: 1. > 1000 CE부문, 2. 2000 IM부문, 3. 3000 DS부문 반도체. The cell containing '1000' is highlighted with a blue border.

<그림 9-20. Chrome 웹 브라우저 Preview 화면>

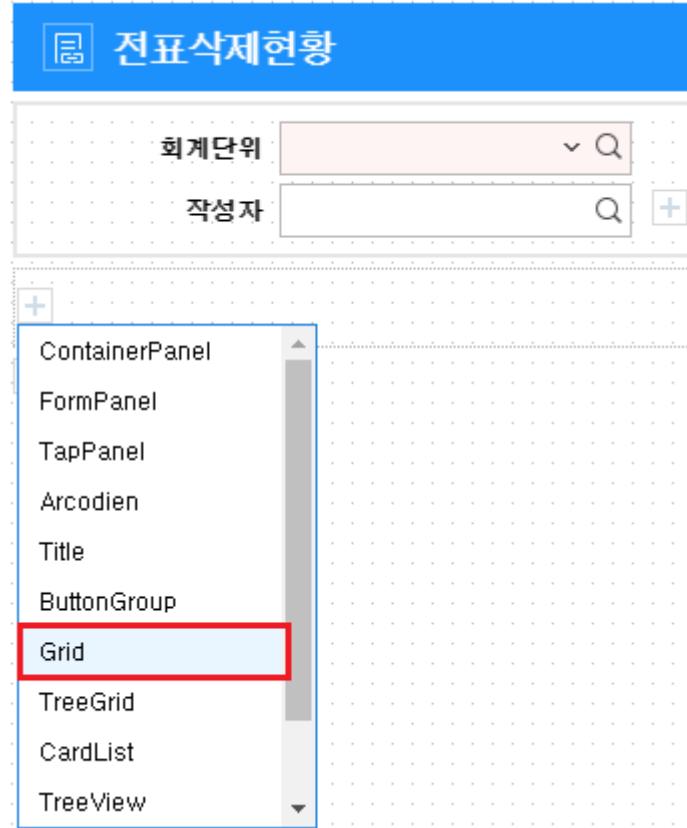
다음으로 그리드(Grid) 컨트롤을 추가한다.

- Canvas  을 눌러 ① ContainerPanel 을 추가하고, 추가한 ② ContainerPanel 을 선택한다.  
③ 브라우저 크기에 따른 하단 자동 높이를 설정하기 위해 속성 중 fitBottom 속성을 체크한다.



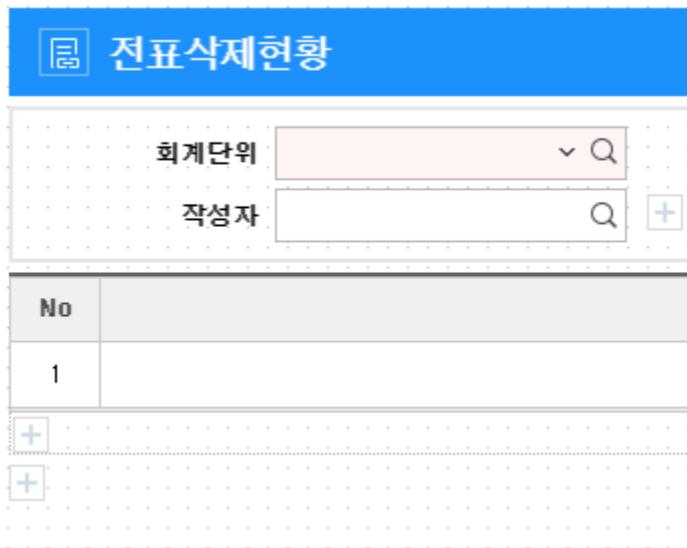
<그림 9-21. ContainerPanel 및 fitBottom 속성 변경>

이제  를 눌러 Grid 를 추가한다.



<그림 9-22. Grid 추가>

그리드가 추가된 화면이다.

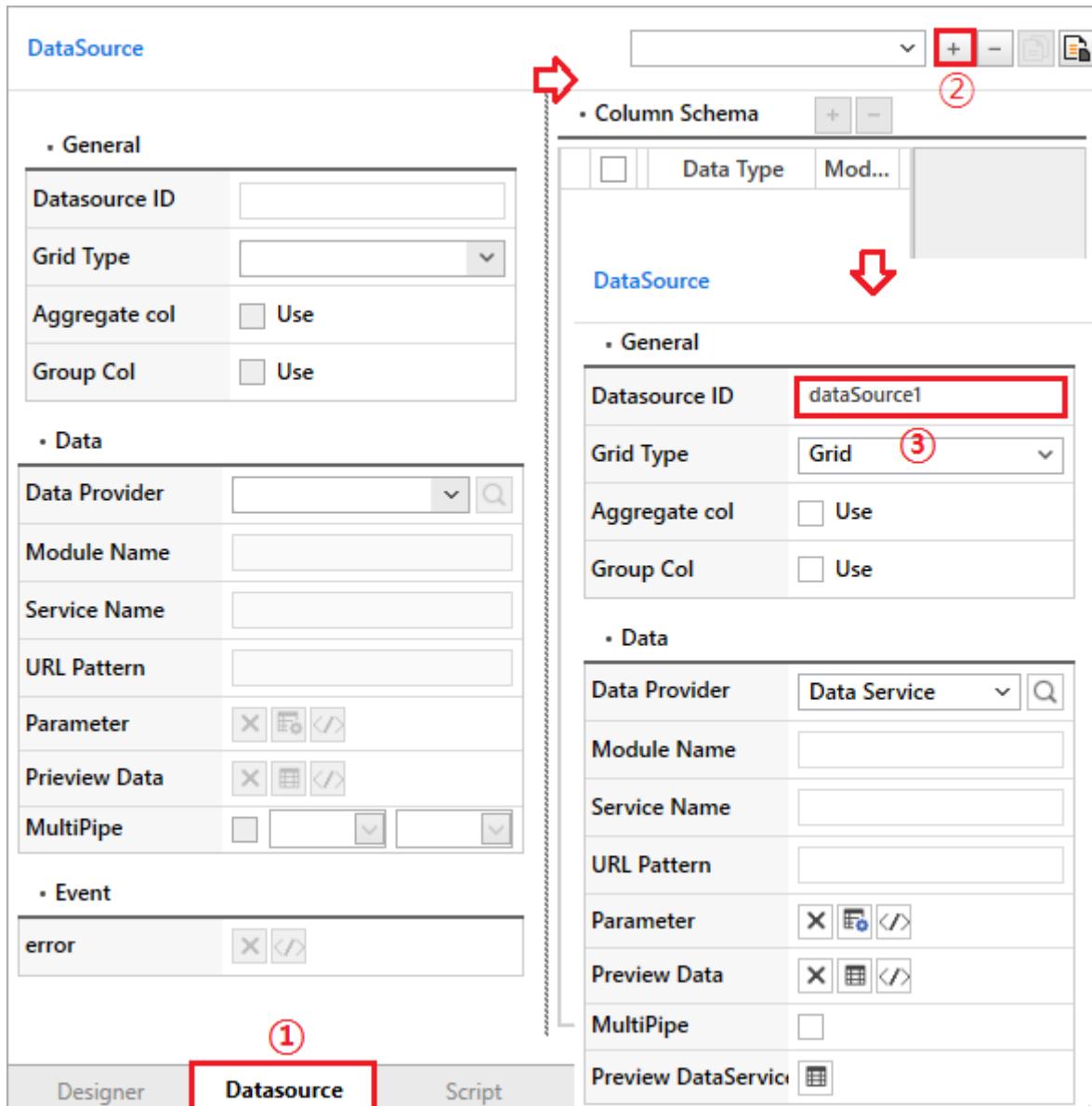


<그림 9-22. Grid 를 추가한 화면>

### 3) 데이터소스 추가 및 바인딩

그리드에 컬럼을 설정하고 데이터를 가져오기 위해 데이터소스를 추가한다.

- ① Datasource 탭을 클릭한다.
- ② 우측 상단에서 을 눌러 Datasource를 추가한다.
- ③ datasource1이 자동으로 추가되어 있으며, ID 변경시 반드시 엔터키를 눌러야 반영된다.



<그림 9-23. Datasource 추가>

다음으로 Data Provider(데이터 제공자)에서 데이터 서비스를 검색하며, Data Service가 기본값으로 선택되어 있다. ① 검색 버튼을 클릭한다.

• Data

|               |                                  |                                     |                                   |
|---------------|----------------------------------|-------------------------------------|-----------------------------------|
| Data Provider | Data Service                     | <input type="button" value="🔍"/>    | (1)                               |
| Module Name   |                                  |                                     |                                   |
| Service Name  |                                  |                                     |                                   |
| URL Pattern   |                                  |                                     |                                   |
| Parameter     | <input type="button" value="X"/> | <input type="button" value="grid"/> | <input type="button" value="⟨⟩"/> |
| Prievew Data  | <input type="button" value="X"/> | <input type="button" value="grid"/> | <input type="button" value="⟨⟩"/> |
| MultiPipe     | <input type="checkbox"/>         |                                     |                                   |

<그림 9-24. Data Provider 화면 열기>

Search for Data Services 화면이 열리며, 검색 기준이 URL Description으로 선택되어 있다. 전표삭제를 입력하고 Search(검색) 한다.

Search for Data Services

|                 |                                  |              |                 |             |  |
|-----------------|----------------------------------|--------------|-----------------|-------------|--|
| URL Description | <input type="button" value="🔍"/> |              |                 |             |  |
| • DataService   |                                  |              |                 |             |  |
| Module Desc     | URL Description                  | Service Name | Service Descrip | URL Pattern |  |
|                 |                                  |              |                 |             |  |

<그림 9-25. URL Description 검색 조건 입력>

검색되는 ① DataService를 선택하고 ② Apply를 클릭하고, 알림창이 뜨면 ③ 예를 선택한다.



<그림 9-26. Data Services 선택>

다음 Parameter에서 ① Mapping 버튼을 눌러 나오는 파라미터 맵핑 화면에서 적용 형식을 모두 컨트롤로 선택하고 나머지는 아래 <표 9-2>처럼 선택한다.

• Data

|               |                                                                               |
|---------------|-------------------------------------------------------------------------------|
| Data Provider | Data Service                                                                  |
| Module Name   | FI                                                                            |
| Service Name  | GeneralLedgerDocumentI                                                        |
| URL Pattern   | glldoc00900_list                                                              |
| Parameter     | X <input type="button" value="edit"/> <input type="button" value="remove"/> ① |

↓

파라미터 매핑

• 매핑 정보

| 파라미터 명       | 적용 형식 | 컨트롤/데이터 형 |
|--------------|-------|-----------|
| to_actg_dt   | 컨트롤   | acct_dt   |
| pc_cd        | 컨트롤   | pc_cd     |
| wrt_dept_cd  | 컨트롤   | dept_cd   |
| from_actg_dt | 컨트롤   | acct_dt   |
| wrt_emp_no   | 컨트롤   | emp_no    |

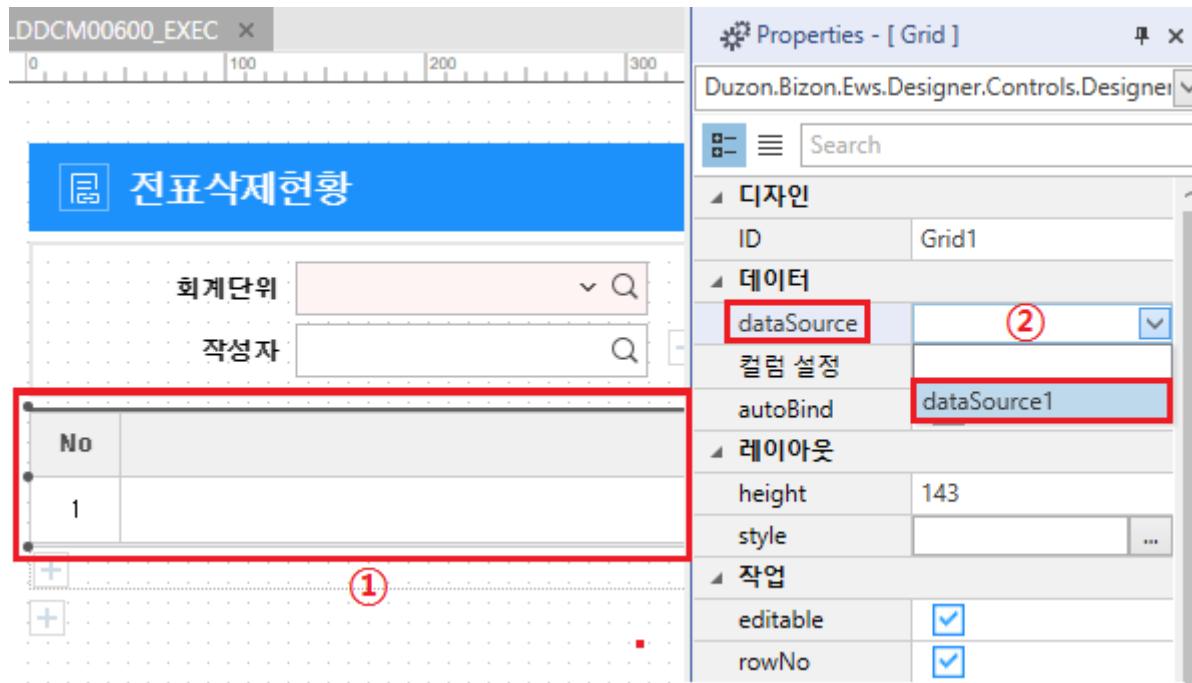
②

<그림 9-27. 파라미터 매핑>

| 파라미터 명       | 설정   |                                 |           |
|--------------|------|---------------------------------|-----------|
|              | 적용형식 | 컨트롤/데이터형                        | 데이터 값     |
| to_actg_dt   | 컨트롤  | acct_dt (회계일 Periodpicker)      | endDate   |
| pc_cd        | 컨트롤  | pc_cd (회계 단위 Multicodepicker)   | codes     |
| wrt_dept_cd  | 컨트롤  | dept_cd (작성부서 Multicodepicker ) | codes     |
| from_actg_dt | 컨트롤  | acct_dt (회계일 Periodpicker)      | startDate |
| wrt_emp_no   | 컨트롤  | emp_no (작성자 Codepicker)         | code      |

<표 9-2. 파라미터 매핑 설정>

다시 Designer로 가서 Grid를 선택하고 DataSource 속성에서 dataSource1은 선택한다.



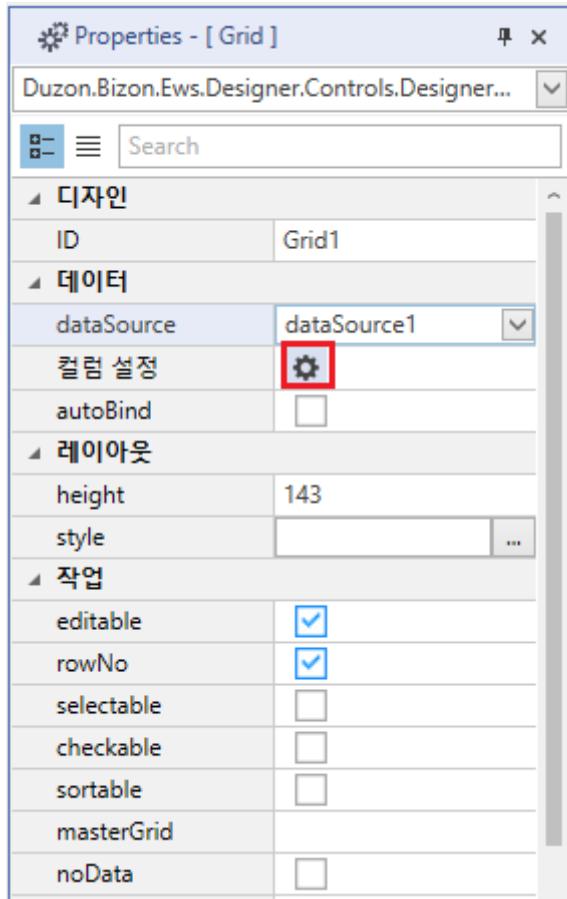
<그림 9-28. Grid 의 dataSource 속성 설정>

선택하면 Datasource 탭에서 가져온 Column Schema 에서 정의한 컬럼들이 그리드 컬럼으로 바인딩된 것을 볼 수 있다.

| No | 회계단위명 | NM_WDEPT | 작성사원번호 | NM_EMP_W... | 회계일 |
|----|-------|----------|--------|-------------|-----|
| 1  |       |          |        |             |     |

<그림 9-29. 바인딩된 그리드 컬럼>

컬럼별로 속성을 정해주기 위해 컬럼설정 을 클릭한다.



<그림 9-30. 컬럼 설정 열기>

그리드-컬럼설정 화면에서 컬럼별 속성을 <표 9-3>과 같이 설정한다. (출력 컬럼만 설정)

| 컬럼명           | 설정     |       |                   |
|---------------|--------|-------|-------------------|
|               | 헤더타이틀  | Width | 텍스트정렬/출력포맷/포맷 문자열 |
| PC_NM         | 회계단위명  | 70    | 왼쪽/(빈칸)/(빈칸)      |
| WRT_DEPT_NM   | 작성부서   | 80    | 왼쪽/(빈칸)/(빈칸)      |
| WRT_EMP_NM    | 작성자    | 60    | 왼쪽/(빈칸)/(빈칸)      |
| ACTG_DT       | 회계일    | 80    | 가운데/날짜/yyyy-MM-dd |
| ACTG_NO       | 순번     | 40    | 왼쪽/(빈칸)/(빈칸))     |
| DELETE_EMP_NM | 삭제자명   | 60    | 왼쪽/(빈칸)/(빈칸)      |
| DELETE_DTS    | 삭제일시   | 80    | 가운데/날짜/yyyy-MM-dd |
| DOCU_NO       | 전표번호   | 120   | 왼쪽/(빈칸)/(빈칸)      |
| NOTE_DC       | 적요     | 270   | 왼쪽/(빈칸)/(빈칸)      |
| DR_AMT        | 차변장부금액 | 120   | 오른쪽/숫자포맷/#,###    |
| CR_AMT        | 대변장부금액 | 120   | 오른쪽/숫자포맷/#,###    |

<표 9-3. 컬럼별 속성(출력컬럼만)>

+ Add Header    X Remove Header          C

| 회계단위 | 작성부서명 | 작성자 | 회계일 | 순번 | 삭제자명 | 삭제일시 | 전 |
|------|-------|-----|-----|----|------|------|---|
|      |       |     |     |    |      |      |   |

• Column Information

| Column Name    | Header Caption | <input type="checkbox"/>            | Width |   |
|----------------|----------------|-------------------------------------|-------|---|
| COMPANY_CD     | 회사코드           | <input type="checkbox"/>            | 80    | ^ |
| PC_CD          | 회계단위코드         | <input type="checkbox"/>            | 80    |   |
| PC_NM          | 회계단위           | <input checked="" type="checkbox"/> | 70    |   |
| WRT_DEPT_CD    | 작성부서코드         | <input type="checkbox"/>            | 80    |   |
| WRT_DEPT_NM    | 작성부서명          | <input checked="" type="checkbox"/> | 80    |   |
| WRT_EMP_NO     | 작성사원번호         | <input type="checkbox"/>            | 60    |   |
| WRT_EMP_NM     | 작성자            | <input checked="" type="checkbox"/> | 60    |   |
| ACTG_DT        | 회계일            | <input checked="" type="checkbox"/> | 80    |   |
| ACTG_NO        | 순번             | <input checked="" type="checkbox"/> | 40    |   |
| DELETE_EMP_NO  | 삭제사원코드         | <input type="checkbox"/>            | 60    |   |
| DELETE_EMP_NM  | 삭제자명           | <input checked="" type="checkbox"/> | 80    |   |
| TMP_DELETE_DTS | 삭제일시           | <input type="checkbox"/>            | 60    |   |
| DELETE_DTS     | 삭제일시           | <input checked="" type="checkbox"/> | 60    |   |
| DOCU_NO        | 전표번호           | <input checked="" type="checkbox"/> | 120   |   |
| NOTE_DC        | 적요             | <input checked="" type="checkbox"/> | 270   |   |
| DR_AMT         | 차변장부금액         | <input checked="" type="checkbox"/> | 120   |   |
| CR_AMT         | 대변장부금액         | <input checked="" type="checkbox"/> | 120   |   |

| field            | CR_AM                               |
|------------------|-------------------------------------|
| 헤더 타이틀           | 대변장!                                |
| visible          | <input checked="" type="checkbox"/> |
| Width            | 120                                 |
| 텍스트 정렬           | 오른쪽                                 |
| sortable         | <input checked="" type="checkbox"/> |
| style            |                                     |
| prefix           |                                     |
| suffix           |                                     |
| attributes       | none                                |
| hideChildHeaders | <input type="checkbox"/>            |
| merge            | <input type="checkbox"/>            |
| grouping         | <input type="checkbox"/>            |
| movable          | <input checked="" type="checkbox"/> |
| ▶ footer         |                                     |
| ▶ 출력 포맷          | 숫자 포                                |
| 에디터              |                                     |
| designSummary    |                                     |

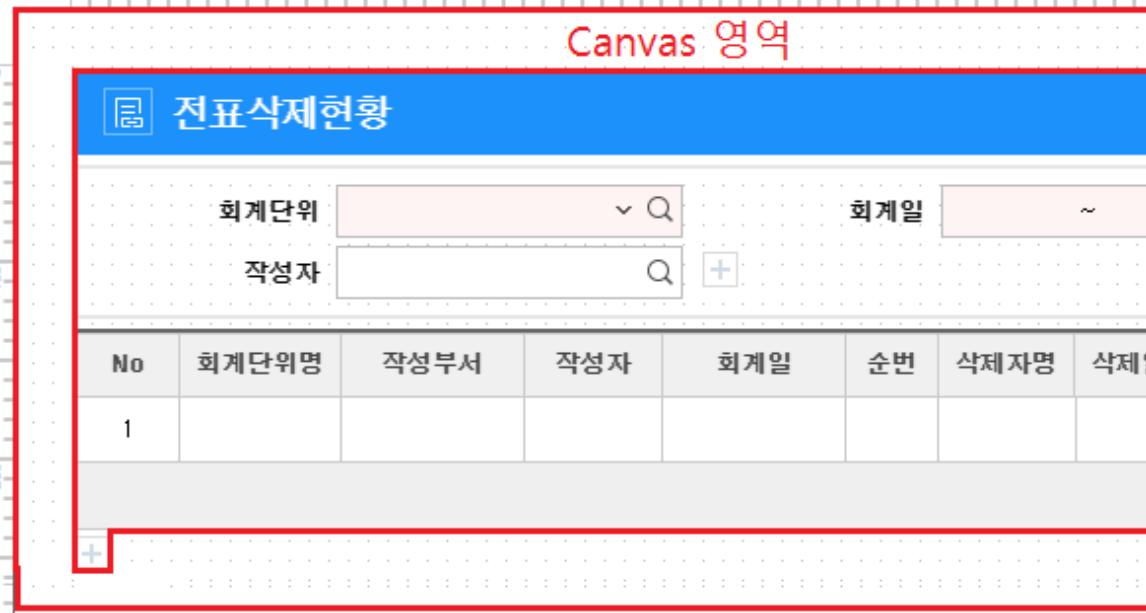
<그림 9-31. 컬럼별 속성 설정>

#### 4) 이벤트 스크립트 설정

페이지가 로드될 때 실행되는 초기값 설정과 필수 조회 조건을 체크하고 조회하는 이벤트 스크립트를 작성한다.

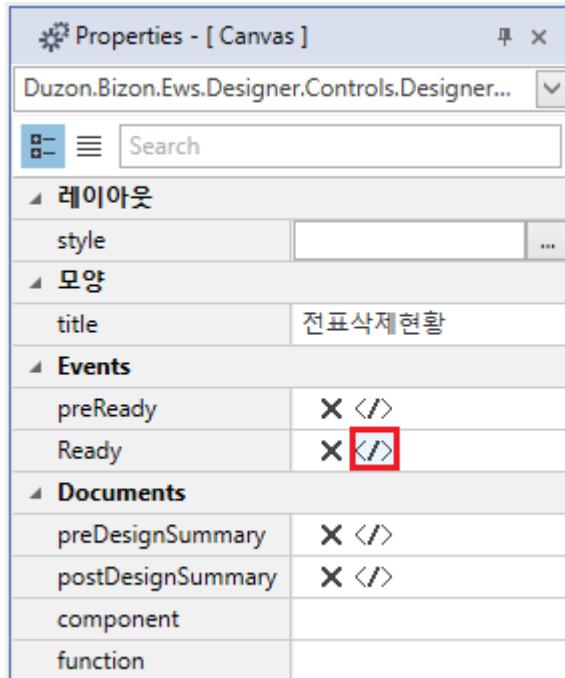
먼저 초기값을 설정하는 스크립트를 작성한다. 초기값 설정은 최상위 Canvas 의 Events에서 구현하며 이벤트는 preReady 와 Ready 가 있다. preReady 는 Datasource 와 그리드 설정 전에, Ready 는 이후에 실행된다.

Canvas 선택은 문서 제목 위 상단이나 아래 하단 부분을 클릭한다.



<그림 9-32. Cavas 선택 방법>

이제 이벤트 Ready 스크립트 버튼</>을 눌러 CodeEditor 화면에서 4개의 컨트롤 초기값을 설정하는 스크립트를 작성한다.



<그림 9-33. Ready 이벤트 스크립트 버튼 실행>

```
// 회계 단위 초기값 설정
self.pc_cd.setData([
 // Code
 PC_CD:self.user.profitCenterCode,
 // Text
 PC_NM:self.user.profitCenterName
]);

// 회계일 초기값 설정(이번달 첫번째 일자와 오늘 일자)
self.acct_dt.setupToday();

// 작성부서 초기값 설정

self.dept_cd.setData([
 // Code
 DEPT_CD : self.user.deptCode,
 // Text
 DEPT_NM : self.user.deptName
]);

// 작성자 초기값 설정

self.emp_no.setData([
 // Code
 EMP_NO : self.user.empCode,
 // Text
 KOR_NM : self.user.username
]);
```

```
[DesignerCanvas] ready
1 // 회계 단위 초기값 설정|
2 self.pc_cd.setData([{|
3 // Code|
4 PC_CD:self.user.profitCenterCode,|
5 // Text|
6 PC_NM:self.user.profitCenterName|
7 }]);|
8|
9 // 회계일 초기값 설정(이번달 첫번째 일자와 오늘 일자)|
10 self.acct_dt.setupToday();|
11|
12 // 작성부서 초기값 설정|
13|
14 self.dept_cd.setData([{|
15 // Code|
16 DEPT_CD : self.user.deptCode,|
17 // Text|
18 DEPT_NM : self.user.deptName|
19 }]);|
20|
21 // 작성자 초기값 설정|
22|
23 self.emp_no.setData([{|
24 // Code|
25 EMP_NO : self.user.empCode,|
26 // Text|
27 KOR_NM :self.user.username|
28 }]);
```

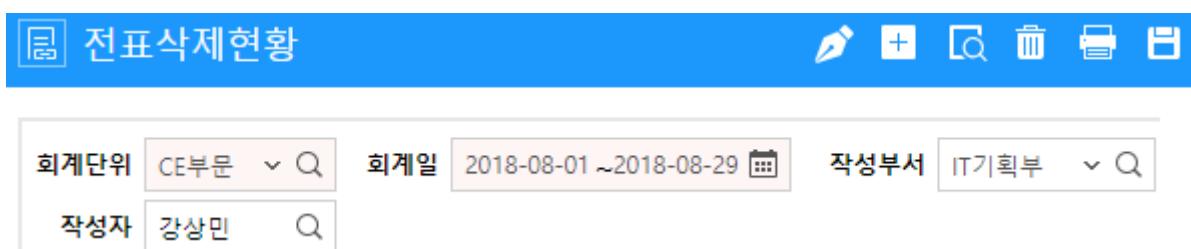
<그림 9-34. Cavas Ready Event 스크립트 작성>

스크립트에서 입력한 self.user 아래 세션 정보들은 크롬 브라우저 개발자 도구에서 Break Point 지정 후 새로 고침한 후 Console에서 self.user를 입력하여 확인할 수 있다.

```
> self.user
< ▼ {jti: "688d5905-bb65-4e73-96b5-6618fade3b57", deptName: "IT기획부",
 bizAreaCode: "1000"
 bizAreaName: "서울본사"
 companyCode: "5000"
 companyName: "더존전자(주)"
 companyNumber: "3215646121"
 deptCode: "505010"
 deptName: "IT기획부"
 dutyCode: "19"
 dutyName: "과장"
 dutyRankCode: "19"
 dutyRankName: "과장"
 empCode: "G2017122000003"
 gEmpCode: "H2017122000003"
 gaapCode: "1"
 grant_type: "password"
 groupCode: "1000"
 groupName: "ERP관리자(GERP)"
 guid: "60BF3A760E4E11C4E0530100007FCDD8"
 isSystemUser: "false"
 isUserMenu: "false"
 jti: "688d5905-bb65-4e73-96b5-6618fade3b57"
 language: "ko"
 profitCenterCode: "1000"
 profitCenterName: "CE부문"
 timezone: "KOR10"
 txCrypt: "6bknalwbFy5Yce7sj20DxGwRqBTjVCQnKlyiHrPm7a/PMu6zow=="
 userGbn: "2"
 userGbnName: "내부"
 userid: "sunsetmin"
 username: "강상민"
 userpic: "../../../../images/user-no-pic.png"
```

<그림 9-35. self.user 아래 속성들>

툴바에서  Build & Preview(Ctrl+B)버튼을 눌러 미리보기를 실행하면 조회 조건 컨트롤들의 초기값이 설정되어 있다.



The screenshot shows a search interface with the following fields and their initial values:

- 회계단위: CE부문
- 회계일: 2018-08-01 ~ 2018-08-29
- 작성부서: IT기획부
- 작성자: 강상민

Below the search bar, there are standard toolbar icons for edit, add, search, delete, print, and refresh.

<그림 9-36. 미리보기에서 초기값 확인>

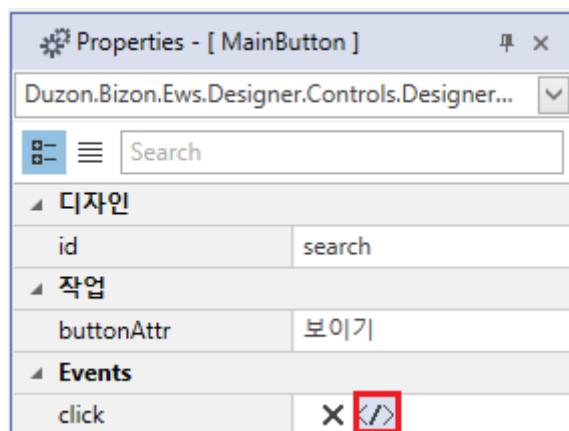
다음으로 필수 조회 조건을 체크하고 조회하는 이벤트 스크립트를 추가한다.

필수 조회 조건 체크는 조회 조건들을 포함하고 있는 ConditionPanel에서 체크하며, 조회는 그리드와 바인딩되는 Datasource의 read 함수(데이터소스를 호출)를 작성한다. 조회버튼을 선택한다.



<그림 9-37. 조회 버튼 선택>

프로퍼티 창에서 click 이벤트를 클릭한다.



<그림 9-38. Click 이벤트 선택>

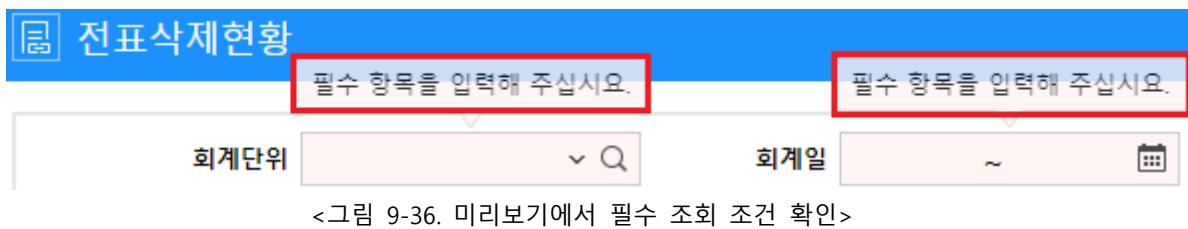
CodeEditor 화면에서 다음과 같이 입력하고 OK를 누른다.

```
// 필수 조회 조건 체크(컨디션 패널 ID)
if (self.searchCondition.validate({tooltip:true,message:'필수 항목을 입력해 주십시오.'}))
{
 // 데이터소스 읽기
 self.dataSource1.read();
}

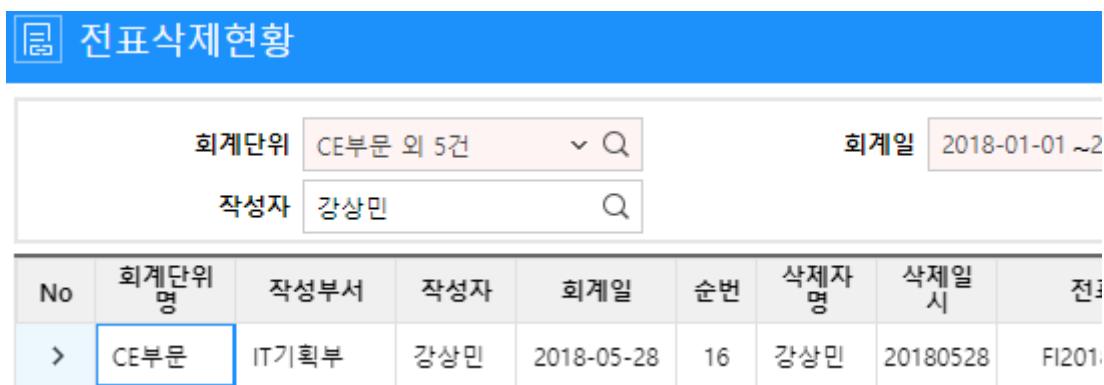
search [DesignerMainButton] click
30 function() {
31 // TODO: 데이터소스를 조회합니다.
32 // 필수 조회 조건 체크(컨디션 패널 ID)
33 if (self.searchCondition.validate({tooltip:true,message:'필수 항목을 입력해 주십시오.'}))
34 {
35 // 데이터소스 읽기
36 self.dataSource1.read();
37 }
38 }
```

<그림 9-39. 조회 버튼 Click 이벤트에 스크립트 입력 화면>

툴바에서  Build & Preview(Ctrl+B)버튼을 눌러 미리보기를 실행하여 회계단위와 회계일을 초기값을 지우고 조회하면 '필수 항목을 입력해 주십시오'라는 메시지가 나타난다.



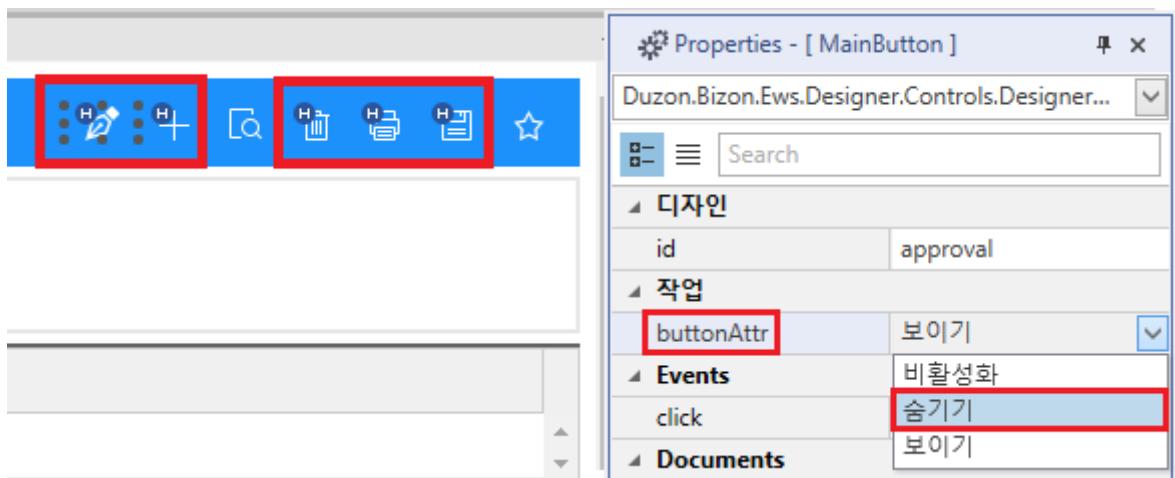
회계단위와 회계일을 입력하고 조회하면 그리드에 데이터가 표시된다.



| No | 회계단위<br>명 | 작성부서  | 작성자 | 회계일        | 순번 | 삭제자<br>명 | 삭제일<br>시 | 전     |
|----|-----------|-------|-----|------------|----|----------|----------|-------|
| >  | CE부문      | IT기획부 | 강상민 | 2018-05-28 | 16 | 강상민      | 20180528 | FI201 |

<그림 9-37. 필수 입력 조건 입력 후 조회한 화면>

상단 우측 나머지 버튼들은 사용하지 않기 때문에 buttonAttr 속성을 숨기기로 설정한다.



## 2. 자금수지표

### 1) 개발 화면

<그림 9-39>는 개발된 자금수지표 화면이다. 회계단위와 회계일은 필수 조회 조건이며 페이지가 로드될 때 초기값은 회계단위는 로그인한 정보, 회계일은 오늘 일자, 조회구분은 금액 0 미포함이며, 필수 영역에 위치하고 있다. 다음 행인 자금그룹 1과 자금그룹 2는 옵션 영역에 위치한다.

금융상품과 계정 등 2개의 탭 패널로 구성되어 있고, 아래에 각각 상세내역이 있다. 전체 구조는 헤더 그리드와 디테일 그리드로 구분되는 Vertical Grid Page이다.

금융상품 | 계정

| No | 자금그룹1 | 자금그룹2 | 금융상품 | 기초(개시)잔액 |
|----|-------|-------|------|----------|
|    |       |       |      |          |

상세정보

| No | 금융기관 | 계좌명 | 계좌번호 | 기초(개시)잔액 | 입 |
|----|------|-----|------|----------|---|
|    |      |     |      |          |   |

<그림 9-39. 자금수지표 개발 화면>

필수

초기값

<그림 9-40. 자금수지표 필수 조회 조건 및 초기값 설정>

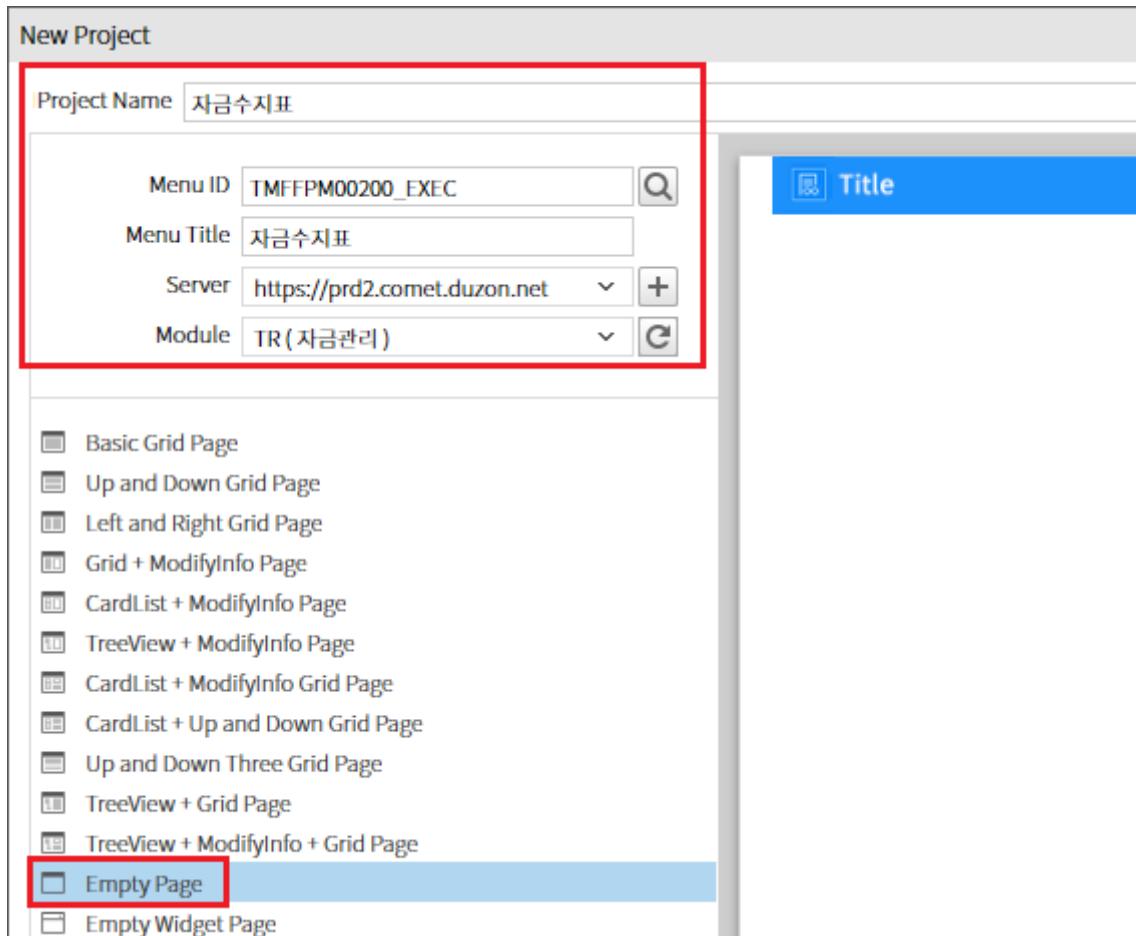
## 2) 화면 디자인

새 프로젝트를 생성한다. (File ▶ New ▶ Project... 또는 툴바의 

|                       |                                                                            |
|-----------------------|----------------------------------------------------------------------------|
| Project Name (프로젝트 명) | 자금수지표 입력                                                                   |
| Menu ID (메뉴ID)        | TMFFPM00200_EXEC 입력                                                        |
| Menu Title (메뉴 제목)    | 자금수지표 입력                                                                   |
| Server (서버 선택)        | <a href="https://prd2.comet.duzon.net">https://prd2.comet.duzon.net</a> 선택 |
| Module (모듈 선택)        | TR (자금관리) 선택                                                               |
| 템플릿(template)         | Empty Page 선택                                                              |

<표 9-4. 새 프로젝트 생성 입력>

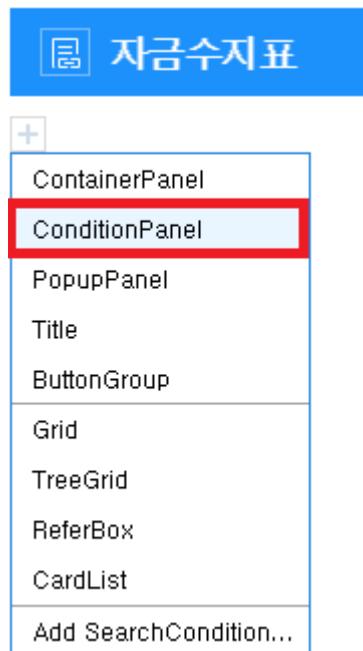
위 정보를 입력 후 OK 버튼을 클릭하여 프로젝트를 생성한다.



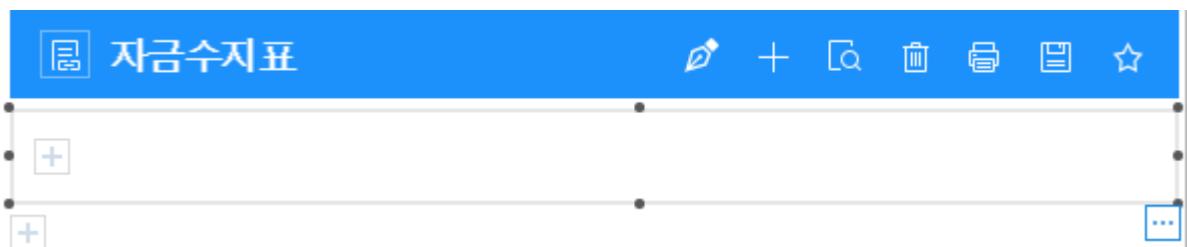
<그림 9-41. New Project 생성 화면 >

조회 조건에 해당하는 컨트롤들은 ConditionPanel, 데이터를 보여주는 컨트롤은 ContainerPanel에서 먼저 생성한 다음 ConditionPanel에 조회 조건에 해당하는 컨트롤을 생성한다.

 버튼을 누르고 ConditionPanel을 선택한다.



<그림 9-42. ConditionPanel 선택>



<그림 9-43. ConditionPanel 생성>

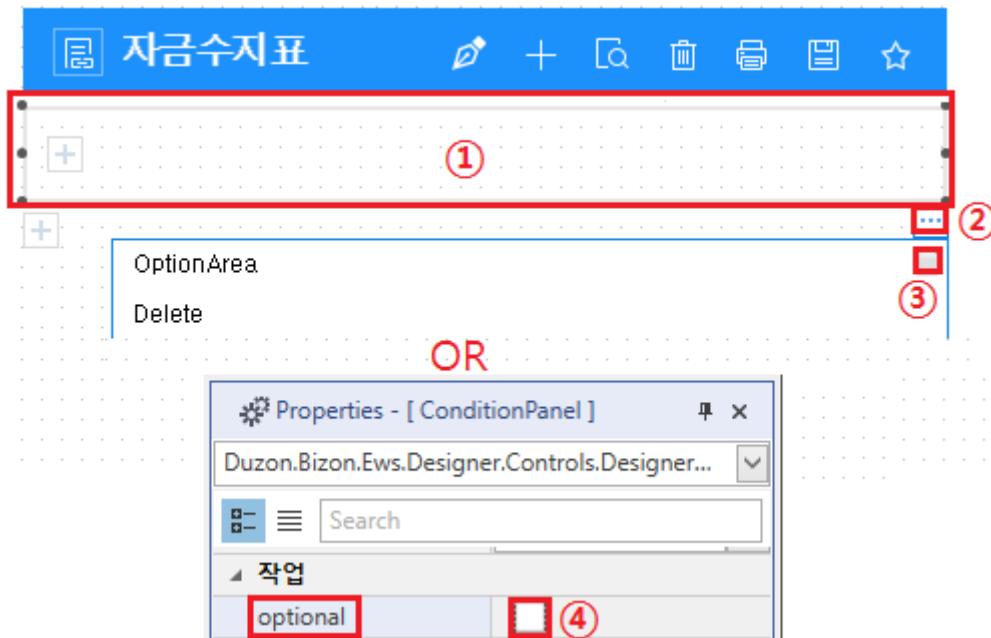
프로그램 실행 후 조회 시 필수 조회 조건 체크를 하기 위해 ConditionPanel 선택 후 ID 속성을 searchCondition으로 입력한다.

ConditionPanel 내의  를 눌러 <표 9-5>와 같이 필수 영역 3개 컨트롤을 추가하고 속성을 변경한다.

| 컨트롤 종류          | 컨트롤 설정        |         |       |                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|---------------|---------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | ConditionItem | ID      | state | Help / 데이터                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Multicodepicker | 회계단위          | pc_cd   | 필수    | helpCode 스크립트 버튼  누른 후<br>Code Help 화면에서 회계단위 검색<br>Help Code : H_MA_PC_MST_S 선택<br>Code Field : PC_CD 선택<br>Text Field : PC_NM 선택 후 Apply<br>helpParmas:  (파라미터 매핑) 클릭<br>후 company_cd 파라미터명은 컨트롤/<br>데이터형은 number, 데이터값은<br>self.user.companyCode 입력<br>helpSize : 대 |
| Periodpicker    | 회계일           | acct_dt | 필수    |                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Dropdownlist    | 조회구분          | ty_zero | 기본    | DataSource 생성 후 재설정                                                                                                                                                                                                                                                                                                                                                                                                                          |

<표 9-5. 필수 영역 컨트롤 추가 설정>

옵션 영역에 위치한 자금그룹 1과 자금그룹 2 컨트롤을 추가하기 위해서 ConditionPanel 선택 후 작업의 ④ optional 속성을 체크하거나 ① ConditionPanel 선택 후 ②  (더보기) 버튼을 눌러서 나오는 화면에서 ③ OptionArea을 체크한다.



<그림 9-44. 옵션 영역 선택>

다음으로 ① 아래 방향을 눌러 ② 옵션 영역을 표시한다



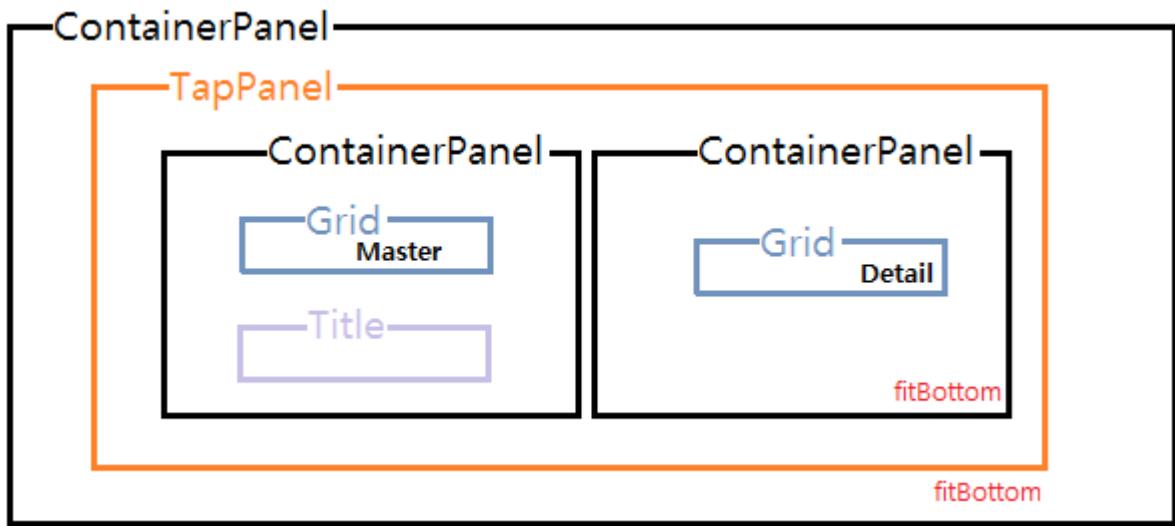
<그림 9-45. 옵션 영역 표시>

ConditionPanel 내의 를 눌러 <표 9-5>와 같이 옵션 영역에 2 개 컨트롤을 추가하고 속성을 변경한다.

| 컨트롤 종류          | ConditionItem | 컨트롤 설정 |    |                                                                                                                                                                                                                                                                                          |
|-----------------|---------------|--------|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |               | Label  | ID | state                                                                                                                                                                                                                                                                                    |
| Multicodepicker | 자금그룹1         | fg1_cd | 기본 | helpCode 스크립트 버튼  누른 후<br>Code Help 화면에서 자금그룹으로 검색<br>Help Code : H_FL_FUNDGROUP_MST_S 선택<br>Code Field : FUNDGRP1_CD 선택<br>Text Field : FUNDGRP1_NM 선택 후 Apply<br>helpSize : 대                                                                                                          |
| Multicodepicker | 자금그룹2         | fg2_cd | 기본 | helpCode 스크립트 버튼  누른 후<br>Code Help 화면에서 자금그룹으로 검색<br>Help Code : H_FL_FUNDGROUP_MST_S1 선택<br>Code Field : FUNDGRP2_CD 선택<br>Text Field : FUNDGRP2_NM 선택 후 Apply<br>helpParmas:  (파라미터 매핑) 클릭 후<br>fundgrp1_cd 파라미터명은 적용형식은<br>컨트롤, 컨트롤/데이터형은 fg1_cd ,<br>데이터값은 codes 선택<br>helpSize : 대 |

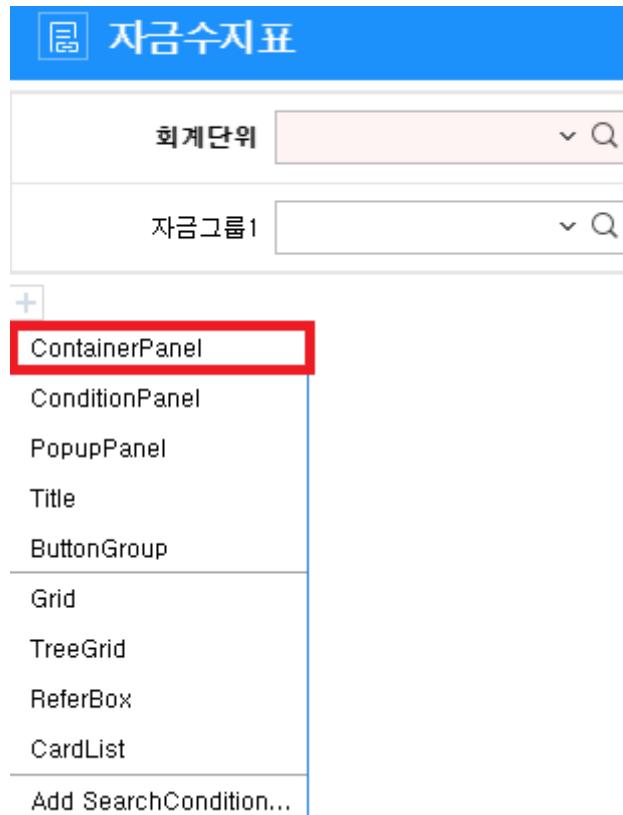
<표 9-6. 옵션 영역 컨트롤 추가 설정>

다음으로 조회 조건 아래 부분을 디자인한다. 전체 화면 구조는 <그림 9-46>같이 되어 있다.



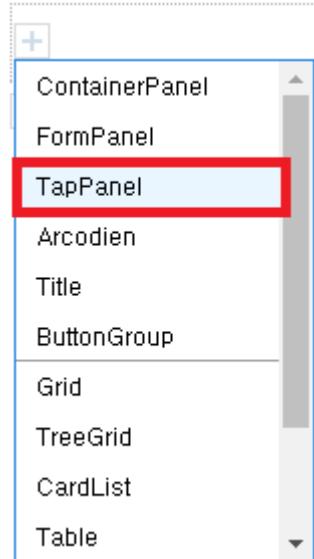
<그림 9-46. 데이터 부분 전체 화면 구조>

Canvas 을 눌러 ① ContainerPanel 을 추가한다.



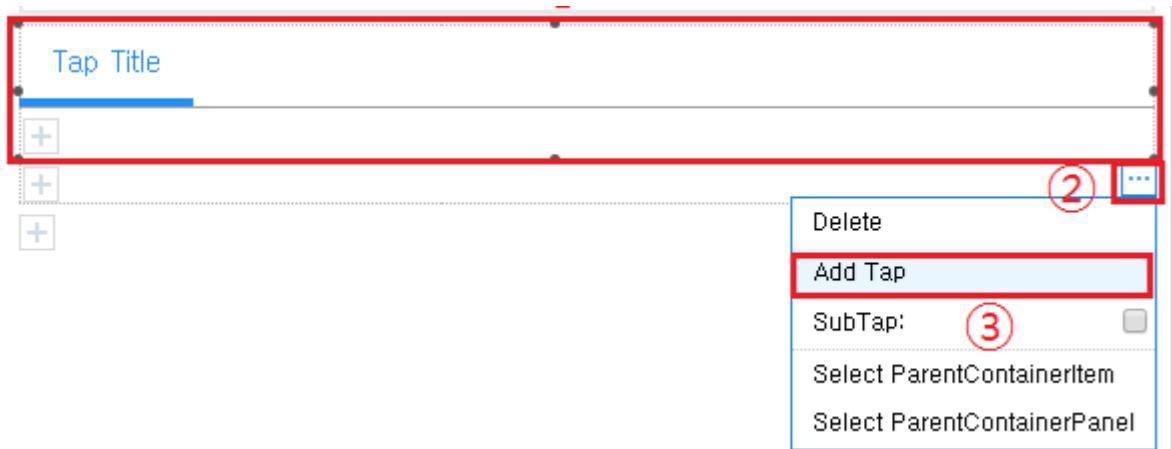
<그림 9-47. ContainerPanel 선택>

추가한 ② ContainerPanel 을 선택하고 을 눌러 TapPanel 을 선택한다.

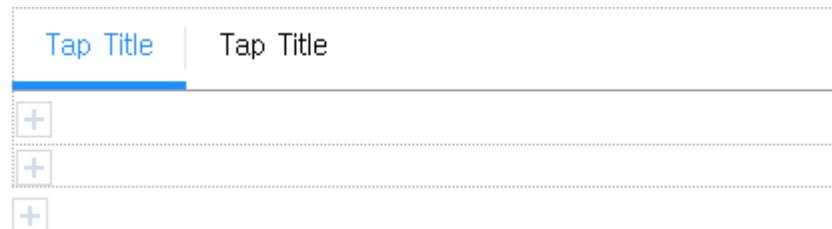


<그림 9-48. TapPanel 선택>

TapPanel 을 선택하고 ② (더보기)를 눌러 Add Tap 을 선택하여 탭을 추가한다.

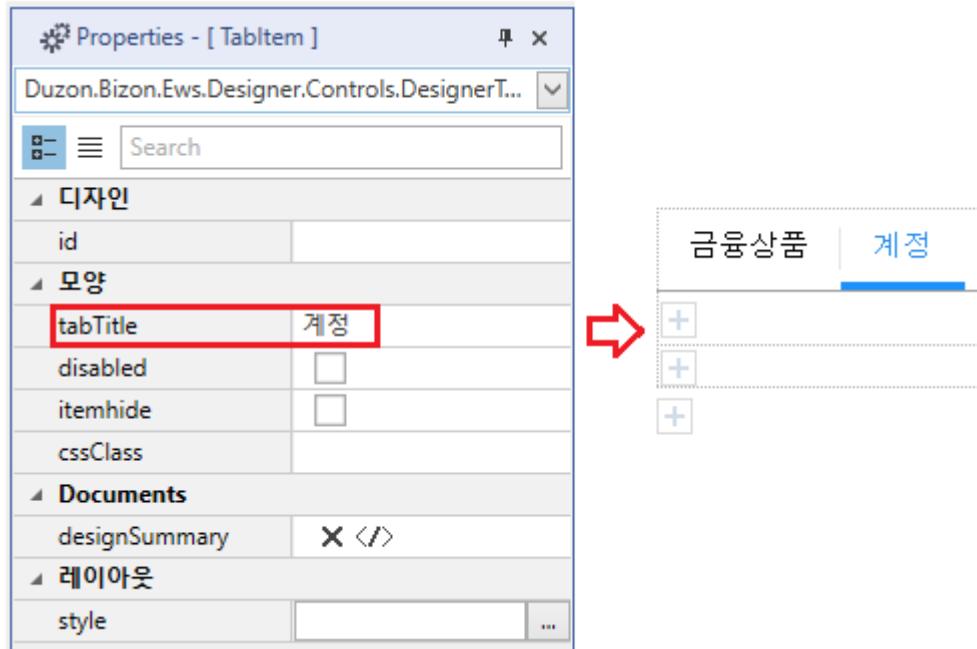


<그림 9-49. Tap 추가>



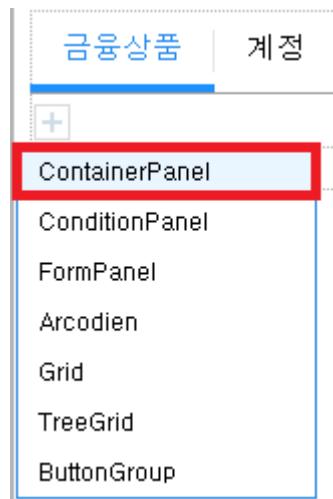
<그림 9-50. 추가된 Tap 화면>

탭 제목을 수정한다. 각각의 탭을 선택하고 모양의 tabTitle 속성을 금융상품과 계정으로 입력한다.



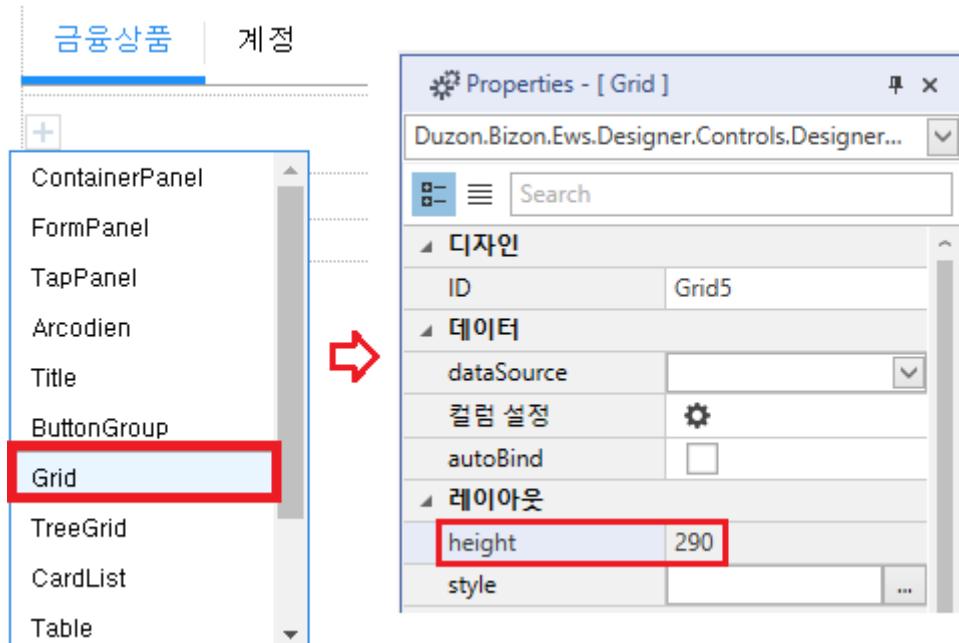
<그림 9-51. Tab 제목 변경>

이제 TabPanel 의 을 눌러 ContainerPanel 을 추가한다.



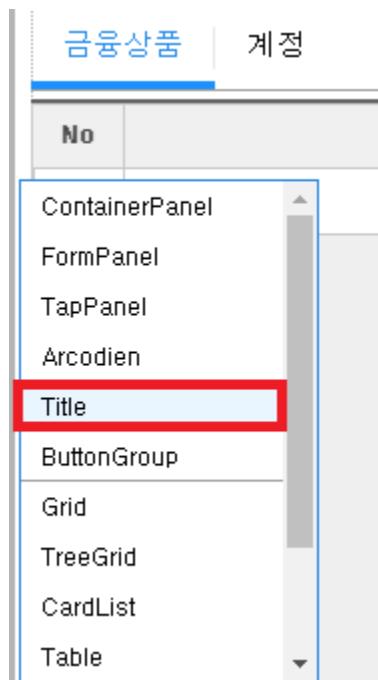
<그림 9-52. TabPanel에 ContainerPanel 추가>

ContainerPanel 의 을 눌러 Grid 를 추가 하며 Grid 의 height(높이)를 290 으로 설정한다.



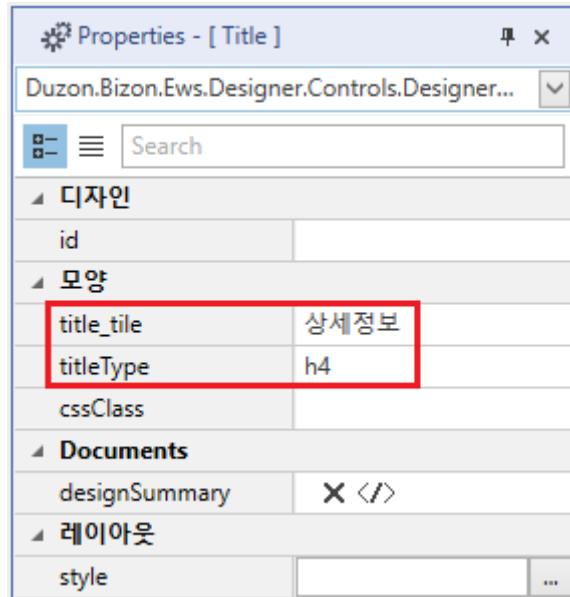
<그림 9-53. Grid 추가 후 height 설정>

그리드를 선택하고 을 눌러 Title 을 추가한다.



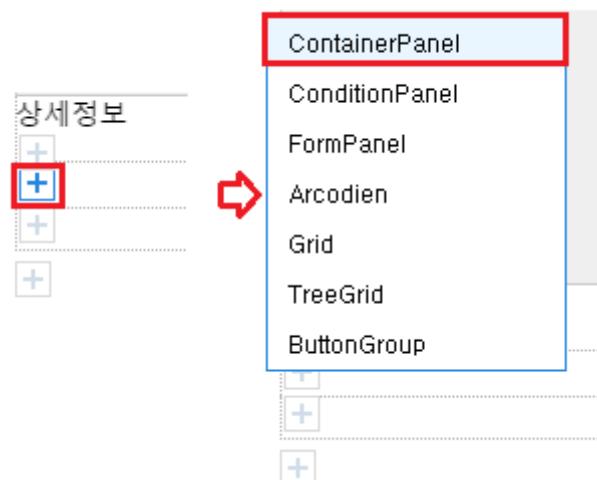
<그림 9-54. Title 추가>

Title 프로퍼티에서 title\_tile 을 상세정보를 입력하고 titleType 을 h4 로 선택한다.



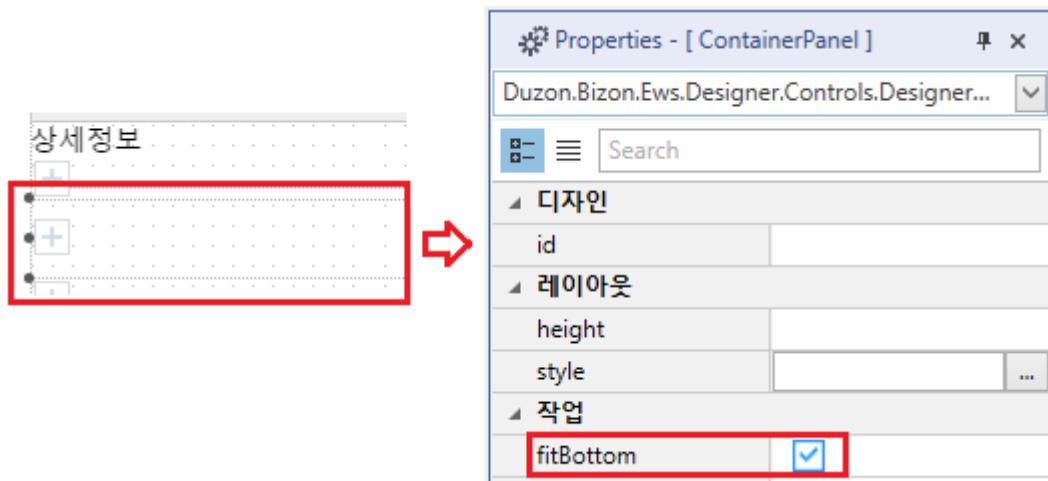
<그림 9-55. Title 속성 변경>

을 눌러 ContainerPanel 을 추가한다.



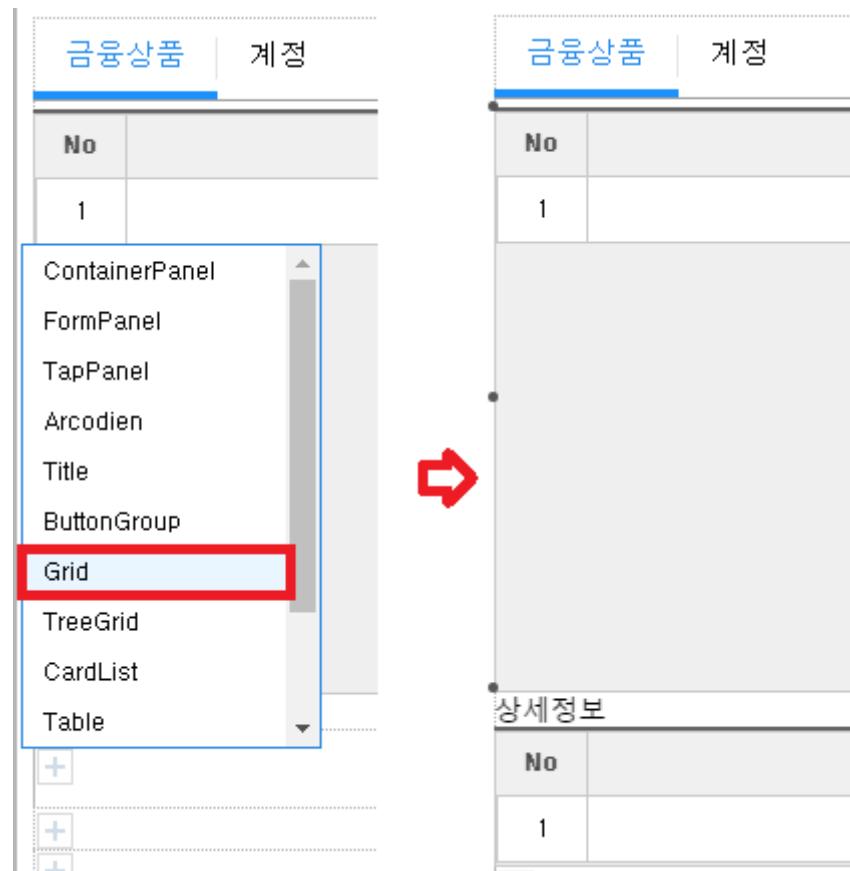
<그림 9-56. ContainerPanel 추가>

ContainerPanel 을 선택하고 브라우저 크기에 따른 자동 높이를 설정하기 위해 fitBottom 속성을 체크한다



<그림 9-57. ContainerPanel 의 fitBottom 속성 체크>

ContainerPanel 의 을 눌러 Grid 를 추가한다.



<그림 9-58. 상세정보 Grid 추가>

최종 디자인 화면이다. 계정 탭도 금융상품 탭처럼 생성한다.

Figure 9-59 displays two screenshots of a financial statement interface. Both screenshots feature a blue header bar with the text '자금수지표' and a small icon. Below the header is a search bar with dropdown menus for '회계단위' and '회계일'. The main content area contains two tabs: '금융상품' and '계정'. In the first screenshot, the '금융상품' tab is highlighted with a red box. In the second screenshot, the '계정' tab is highlighted with a red box. Both tabs have a blue underline. Below the tabs is a table with columns 'No' and '1'. The entire interface has a light gray background.

<그림 9-59. 최종 디자인 화면>

### 3) 데이터소스 추가 및 바인딩

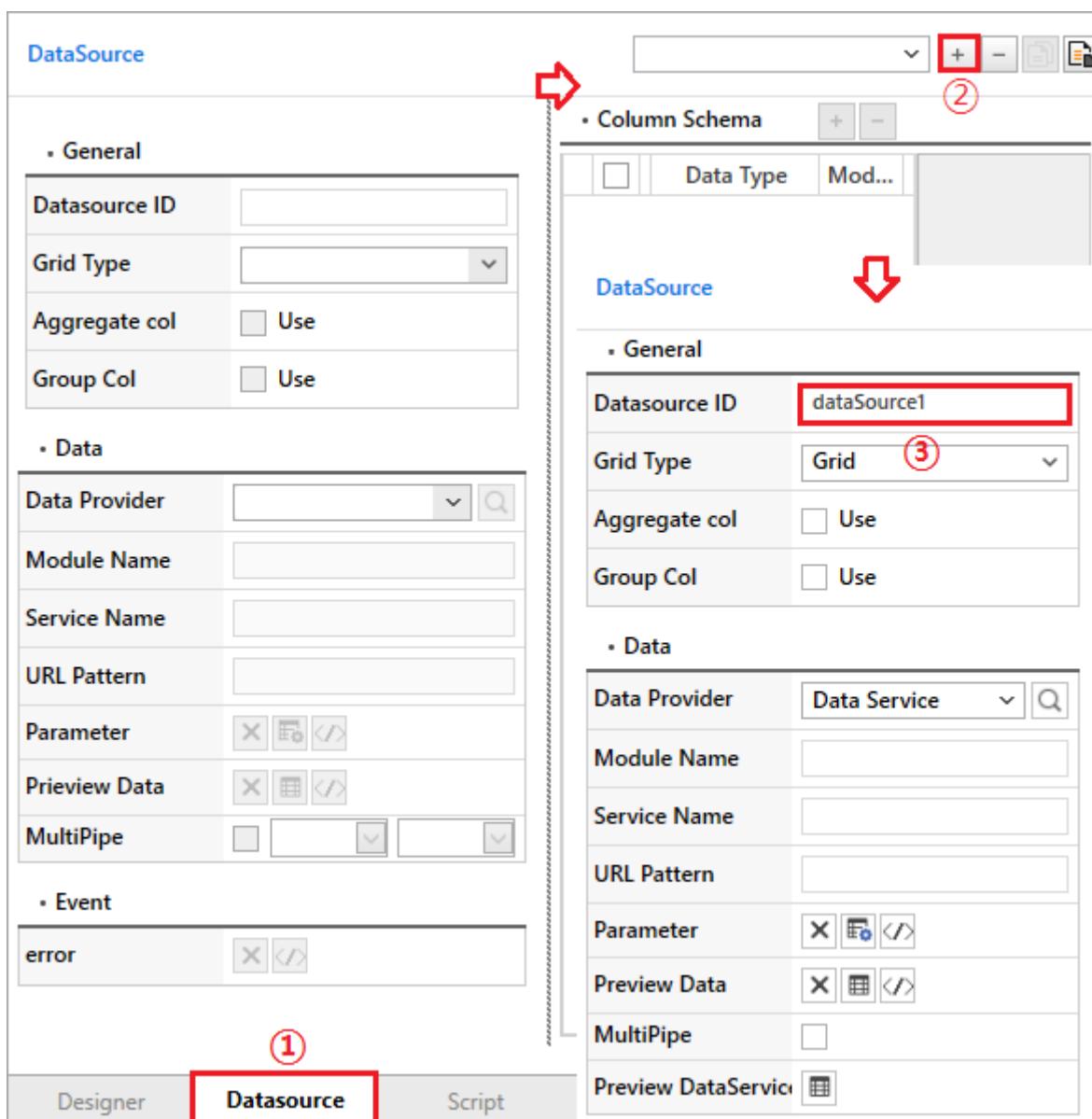
자금수지표에서 필요한 데이터소스는 조회 조건에 있는 조회구분, 금융상품과 계정 템에 각각 헤더/디테일 그리드 등 총 5개 이다.

먼저 조회 구분에 바인딩 할 데이터소스를 생성한다.

① Datasource 탭을 클릭한다.

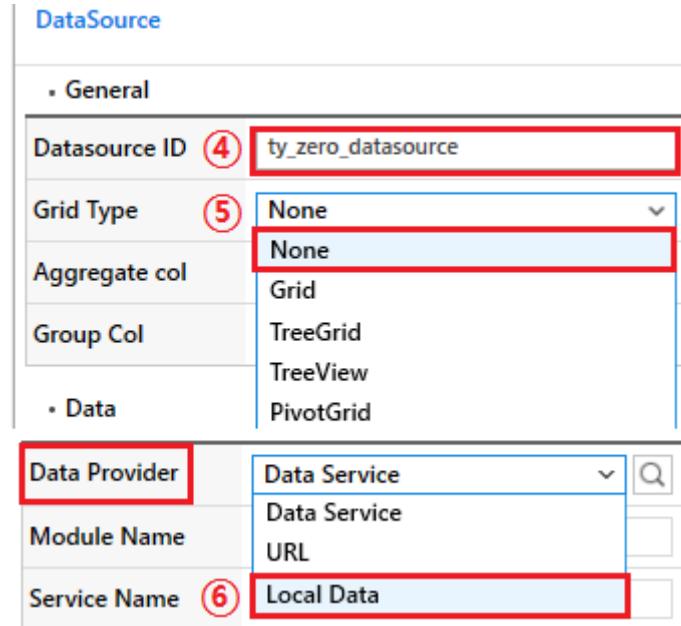
② 우측 상단에서 을 눌러 Datasource를 추가한다

③ datasource1이 자동으로 추가되어 있다. ID 변경시 반드시 엔터키를 눌러야 반영된다.



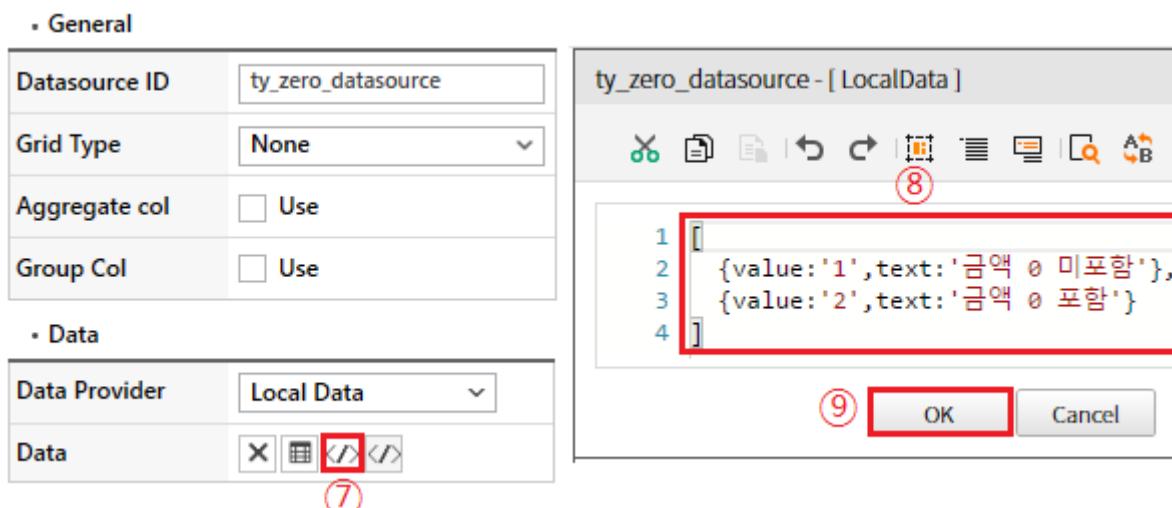
<그림 9-60. 데이터소스 추가>

- ④ Datasource ID 를 ty\_zero\_datasource 변경하고 엔터키를 누른다.
- ⑤ 그리드(Grid)에 사용하지 않기 때문에 Grid Type 을 None 으로 선택한다.
- ⑥ 데이터를 DataService 에서 선택하지 않고 직접 생성을 하기 때문에 Local Data 를 선택한다.



<그림 9-61. 데이터소스 수정>

- ⑦ 스크립트 입력 버튼 </>을 누른다.
- ⑧ [{value:'1',text:'금액 0 미포함'}, {value:'2',text:'금액 0 포함'}]을 입력한다.
- ⑨ OK 를 누른다.



<그림 9-62. 데이터소스 스크립트 작성 및 적용>

- ⑩ Column Schema 에서 를 눌러서 ⑪ value, text 를 추가한다. (데이터 입력은 ⑦~⑧ 방법  
뿐만 아니라 스크립트 버튼 옆에 JsonConverter 버튼에서 AddRow 를 눌러 쉽게 입력할 수 있다)

• Column Schema

|  | Column Name <span style="color:red;">(10)</span> | Data Type | Mod... |
|--|--------------------------------------------------|-----------|--------|
|  |                                                  |           |        |
|  |                                                  |           |        |
|  |                                                  |           |        |

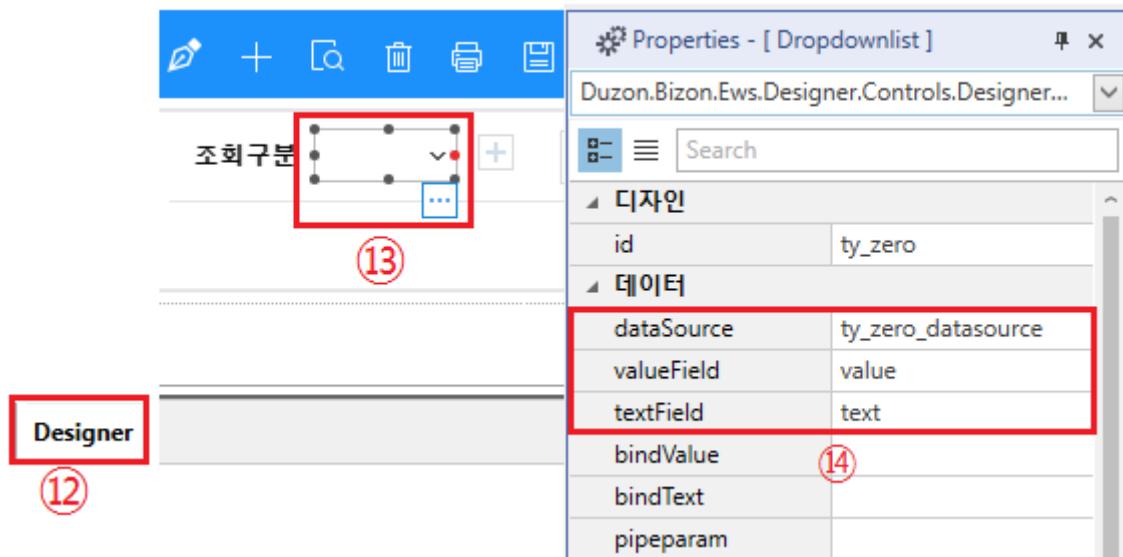
  

• Column Schema

|  | Column Name                                | Data Type | Mod... |
|--|--------------------------------------------|-----------|--------|
|  | value <span style="color:red;">(11)</span> | string    |        |
|  | text                                       | string    |        |

<그림 9-63. 데이터소스 column 추가>

- ⑫ Designer 탭을 클릭한다
- ⑬ 조회구분 Dropdownlist 를 선택한다.
- ⑭ dataSource 에 ty\_zero\_datasouce, valueField 에 value, textField 에 text 를 선택한다.



<그림 9-64. Dropdownlist 의 dataSource 설정>

툴바에서 Build & Preview(Ctrl+B)버튼을 눌러 미리보기를 실행하면 조회구분에 초기값이 설정되어 있다.

회계단위  ~

회계일

자금그룹1  ~

자금그룹2

금융상품 | 계정

데이터가 없습니다.

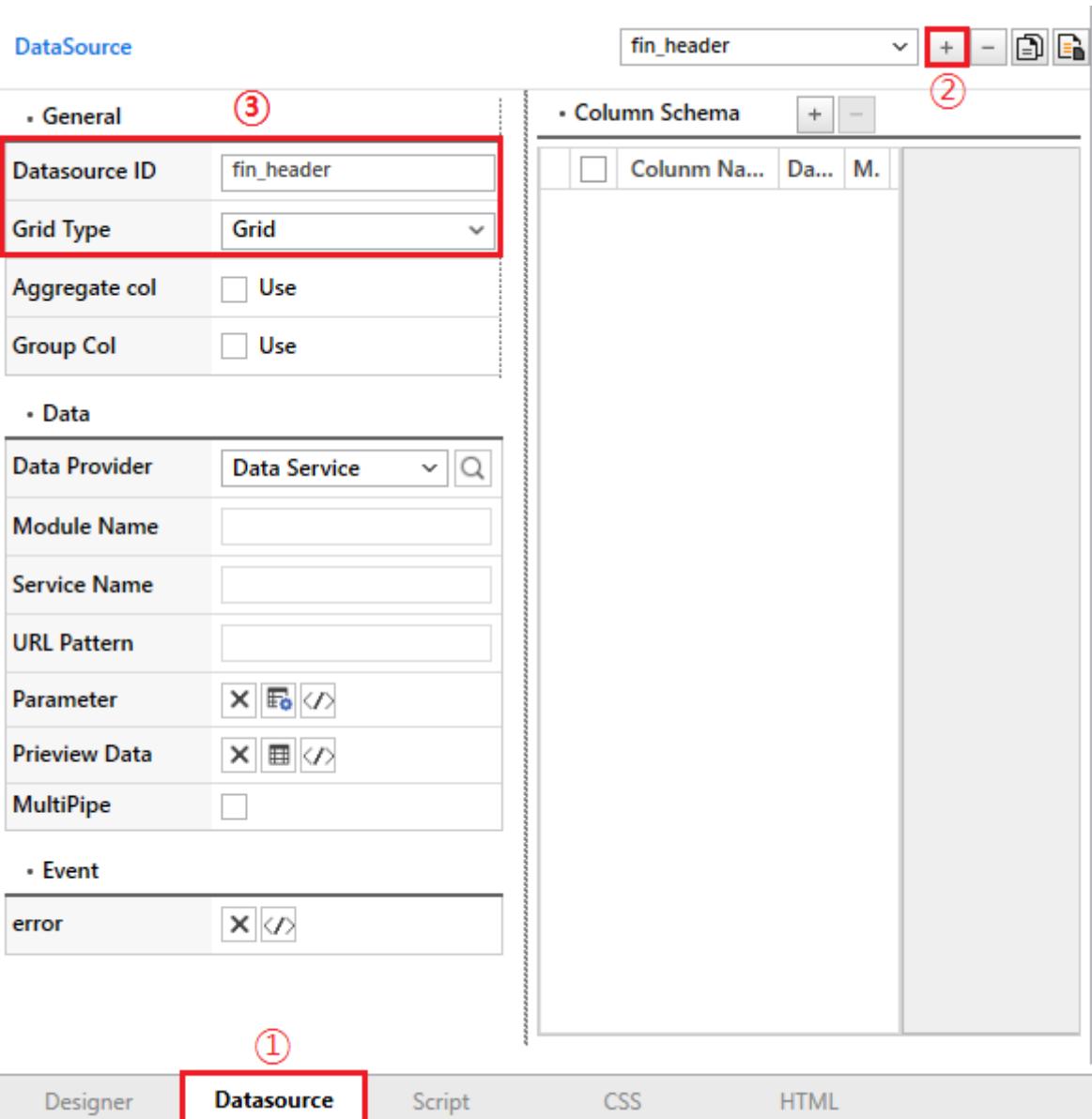
RealGridJS v1.0

상세정보

데이터가 없습니다.

<그림 9-65. 웹 브라우저 미리보기>

다음으로 금융상품 Header(상단) 그리드에 컬럼을 설정하고 데이터를 가져오기 위해 데이터소스를 추가한다. ① Datasource 탭을 선택하고 ② 을 눌러 데이터소스를 추가한다. ③ Datasource ID 를 fin\_header, Grid 에 사용되므로 Grid Type 을 Grid 로 선택한다. (기본).



<그림 9-66. 금융상품 Header(상단) 그리드 데이터소스 추가>

Data Provider(데이터 제공자)에서 데이터 서비스를 검색한다. Data Service가 기본값으로 선택되어 있다. ① 검색 버튼을 클릭한다.

f

**DataSource**

- General

|               |                              |
|---------------|------------------------------|
| Datasource ID | fin_header                   |
| Grid Type     | Grid                         |
| Aggregate col | <input type="checkbox"/> Use |
| Group Col     | <input type="checkbox"/> Use |

- Data

①

|               |                                  |                                     |
|---------------|----------------------------------|-------------------------------------|
| Data Provider | Data Service                     | <input type="button" value="🔍"/>    |
| Module Name   |                                  |                                     |
| Service Name  |                                  |                                     |
| URL Pattern   |                                  |                                     |
| Parameter     | <input type="button" value="X"/> | <input type="button" value="🔗"/>    |
| Prieview Data | <input type="button" value="X"/> | <input type="button" value="grid"/> |
| MultiPipe     | <input type="checkbox"/>         |                                     |

- Event

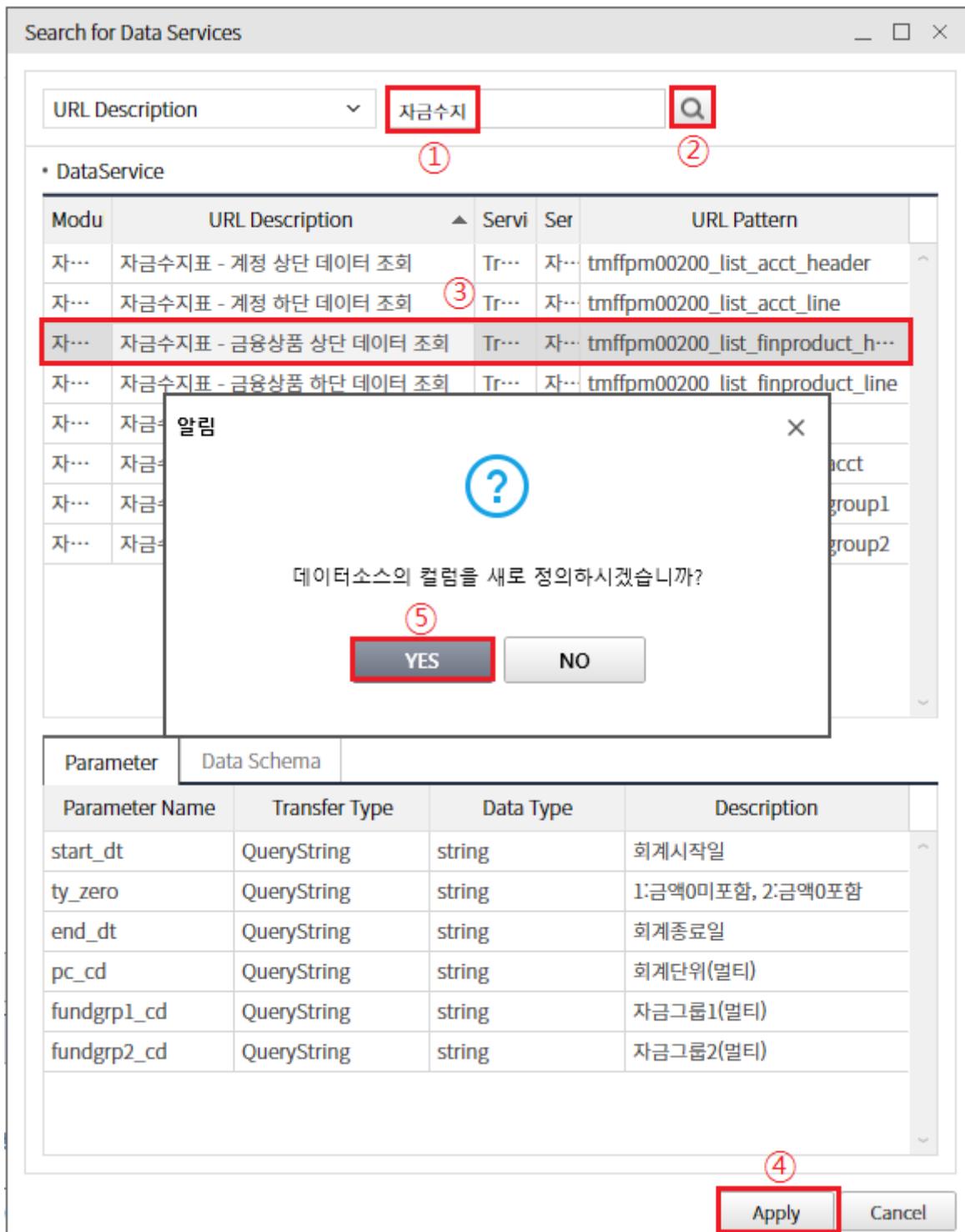
|       |                                  |                                  |
|-------|----------------------------------|----------------------------------|
| error | <input type="button" value="X"/> | <input type="button" value="🔗"/> |
|-------|----------------------------------|----------------------------------|

Designer
Datasource
Script
CSS

<그림 9-67. Data Provider 화면 열기>

Search for Data Services 화면이 열리며, 검색 기준이 URL Description으로 선택되어 있다.

- ① 자금수지를 입력하고 엔터키를 누르거나 ② Search(검색)한다. ③ URL Description 필드에 자금수지표 – 금융상품 상단 데이터 조회를 선택하고 ④ Apply를 클릭한다. ⑤ 알림창이 뜨면 예를 선택한다.



<그림 9-68. URL Description 검색 조건 입력>

다음 Parameter 에서 ① Mapping... 버튼을 눌러 나오는 ② 파라미터 매핑  화면에서 적용 형식, 컨트롤/데이터 형을 <그림 9-69>처럼 선택한다.

• Data

|               |                                                                                                                                                             |                                                                                     |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Data Provider | Data Service                                                                                                                                                |  |
| Module Name   | TR                                                                                                                                                          |                                                                                     |
| Service Name  | TreasuryManagementFundFPMService                                                                                                                            |                                                                                     |
| URL Pattern   | tmffpm00200_list_finproduct_header                                                                                                                          |                                                                                     |
| Parameter     | <input type="button" value="X"/>  <input type="button" value="&lt;/&gt;"/> | ①                                                                                   |
| Prievew Data  | <input type="button" value="X"/>  <input type="button" value="&lt;/&gt;"/> |                                                                                     |
| MultiPipe     | <input type="checkbox"/>                                                                                                                                    |                                                                                     |

↓

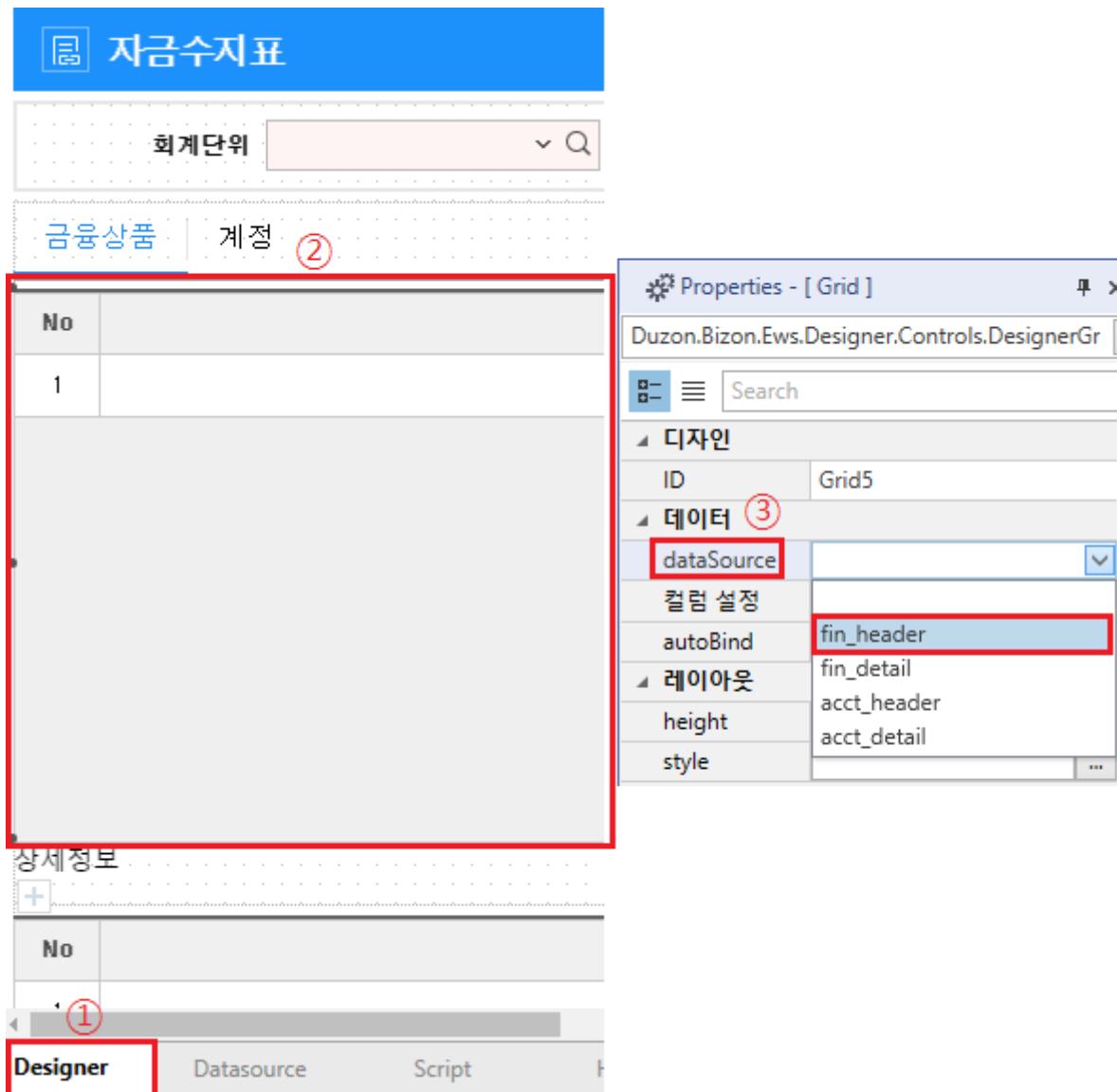
파라미터 매핑

• 매핑 정보

| 파라미터 명      | 적용 형식 | 컨트롤/데이터 형 | 데이터 값     |
|-------------|-------|-----------|-----------|
| start_dt    | 컨트롤   | acct_dt   | startDate |
| ty_zero     | 컨트롤   | ty_zero   | value     |
| end_dt      | 컨트롤   | acct_dt   | endDate   |
| pc_cd       | 컨트롤   | pc_cd     | codes     |
| fundgrp1_cd | 컨트롤   | fg1_cd    | codes     |
| fundgrp2_cd | 컨트롤   | fg2_cd    | codes     |

<그림 9-69. 금융 상품 Header 파라미터 매핑>

다시 ① Designer 탭으로 가서 금융상품 탭에 상위 ② Grid를 선택한 후에, ③ DataSource 속성에서 fin\_header를 선택한다.



<그림 9-70. 금융 상품 Header Grid 의 dataSource 속성 설정>

선택하면 Datasource 탭에서 가져온 Column Schema에서 정의한 컬럼들이 그리드 컬럼으로 바인딩된 것을 볼 수 있다.

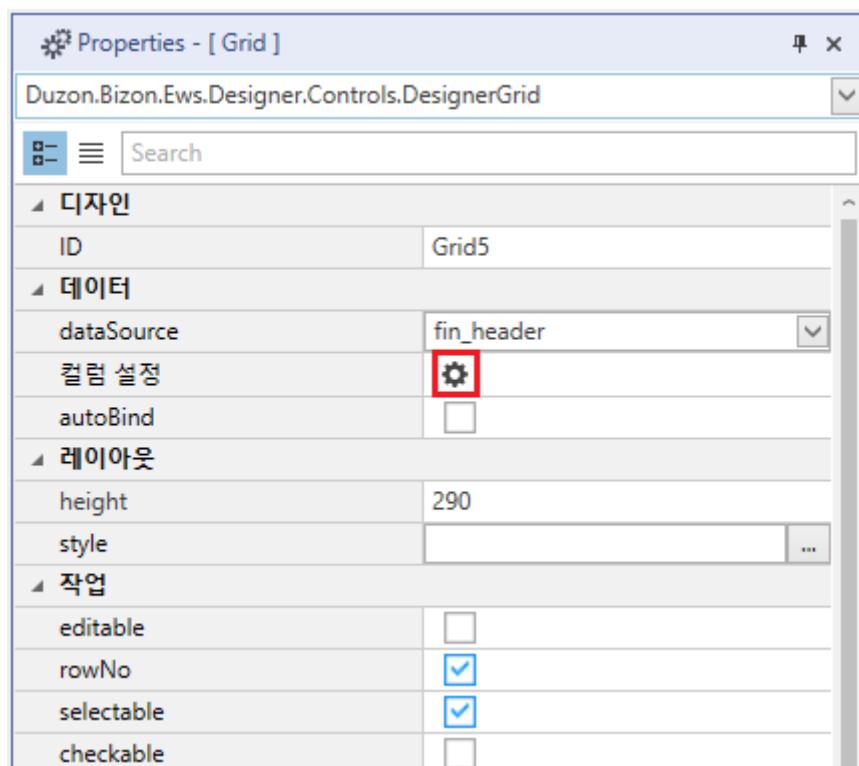
회계단위  회계일  ~  조회구조

금융상품 | 계정

| No | 자금그룹1 | 자금그룹2 | 계정명 | 금융상품 | 계좌번호 | 기초(개시)잔액 |
|----|-------|-------|-----|------|------|----------|
| 1  |       |       |     |      |      |          |

<그림 9-71. 바인딩된 그리드 컬럼>

컬럼별로 속성을 정해주기 위해 컬럼설정 을 클릭한다.



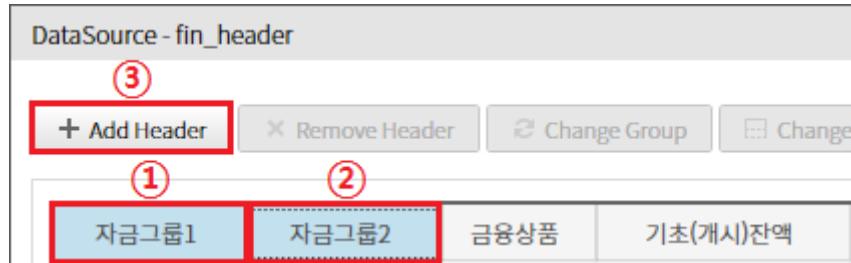
<그림 9-72. 금융 상품 Header Grid 컬럼 설정 열기>

그리드-컬럼설정 화면에서 컬럼별 속성을 <표 9-7>과 같이 설정한다(출력 컬럼만 설정)

| 컬럼명                 | 설정       |       |                    |
|---------------------|----------|-------|--------------------|
|                     | 헤더타이틀    | Width | 텍스트정렬/출력포맷/포맷 문자열  |
| FUNDGRP1_NM         | 자금그룹1    | 90    | 왼쪽/(빈칸)/(빈칸)       |
| FUNDGRP2_NM         | 자금그룹2    | 90    | 왼쪽/(빈칸)/(빈칸)       |
| NM_TP_DEPOSIT       | 금융상품     | 70    | 왼쪽/(빈칸)/(빈칸)       |
| AM_BASIC            | 기초(개시)잔액 | 120   | 오른쪽/숫자포맷/ #,##0.## |
| OTHACCTALT_AMT      | 입금       | 120   | 오른쪽/숫자포맷/ #,##0.## |
| OTHACCTALT_SALE_AMT | 출금       | 120   | 오른쪽/숫자포맷/ #,##0.## |
| AM_FINAL            | 기말(마감)잔액 | 120   | 오른쪽/숫자포맷/ #,##0.## |
| BLNC_AMT            | 계좌잔액     | 120   | 오른쪽/숫자포맷/ #,##0.## |

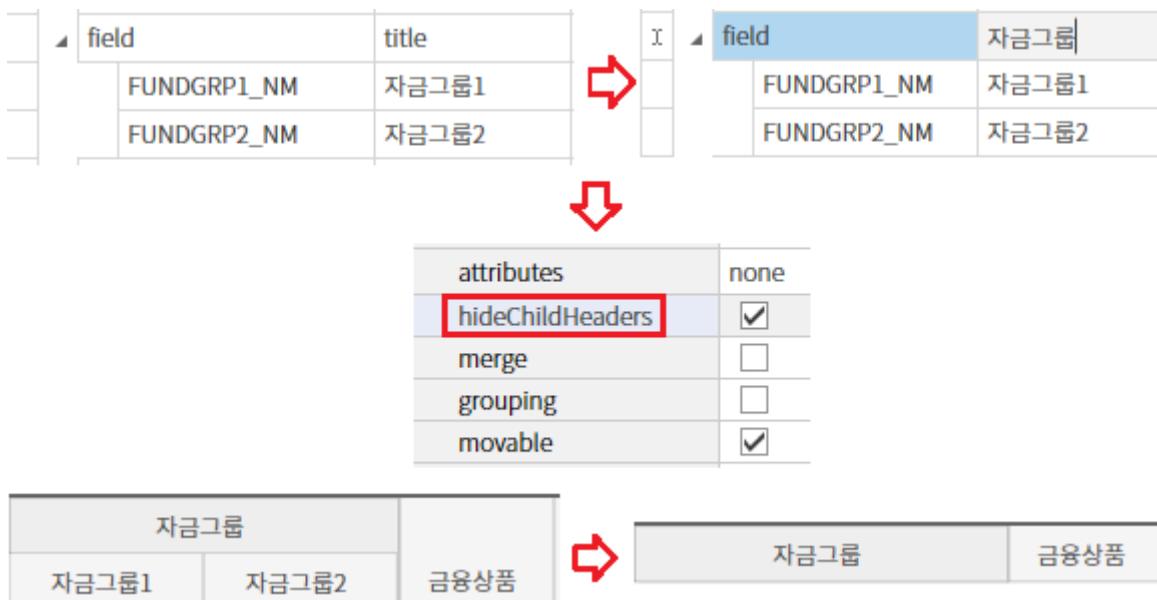
<표 9-7. 컬럼별 속성(출력컬럼만)>

- 자금그룹1과 자금그룹2를 하나의 타이틀로 병합할려면 미리보기 화면에서 ① 자금그룹1과  
 ② 자금그룹2를 다중선택(컨트롤키를 누른채 마우스 클릭) 후에 위에 +Add Header 버튼을  
 누른다.



<그림 9-73. 멀티 타이틀>

새로 생성된 field의 Header Caption을 자금그룹으로 변경하고 오른쪽 속성에서 hideChildHeader(그리드의 하위컬럼 출력여부 속성)을 체크하면 자금그룹1과 자금그룹2의 타이틀이 보이지 않는다.

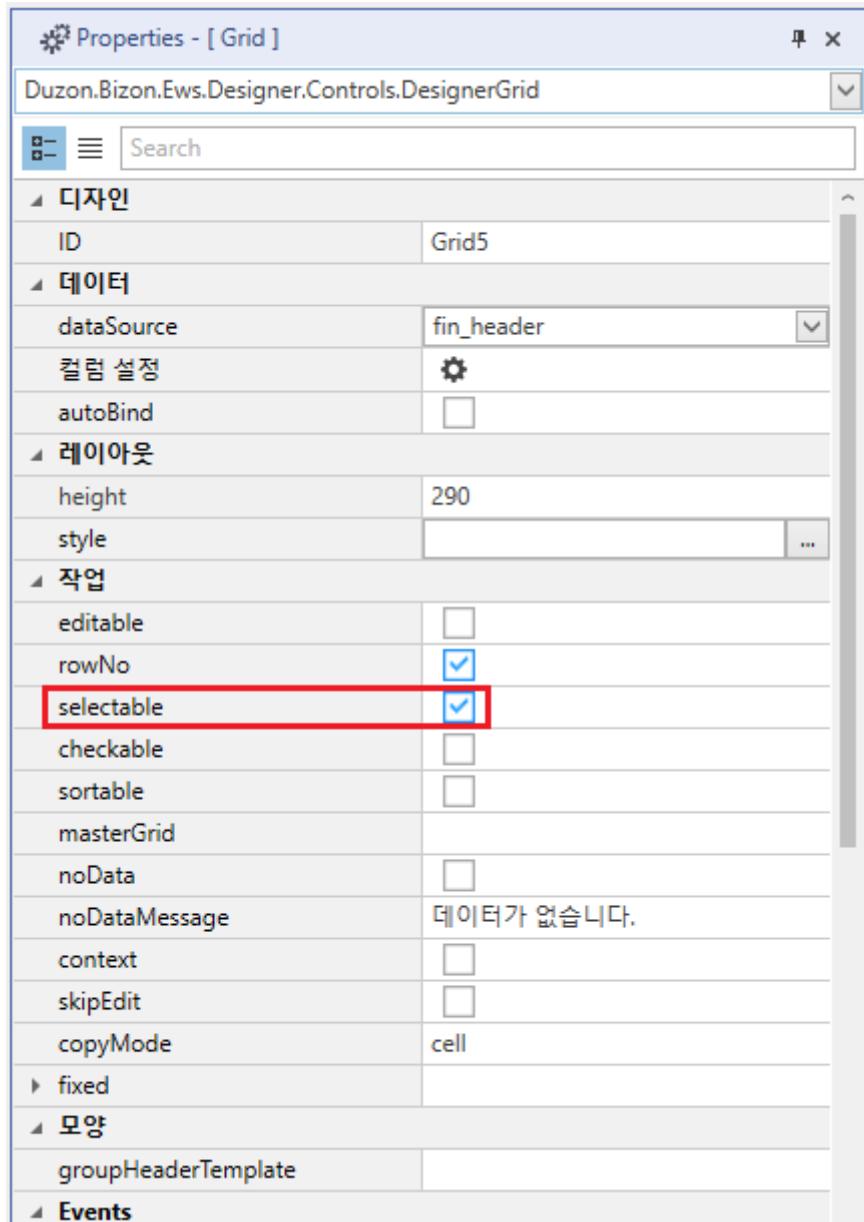


<그림 9-74. 멀티타이틀 속성 변경>

<그림 9-75>은 설정을 완료한 금융상품 Header(상단) 그리드 화면이다.

<그림 9-75. 금융상품 Header 그리드 화면>

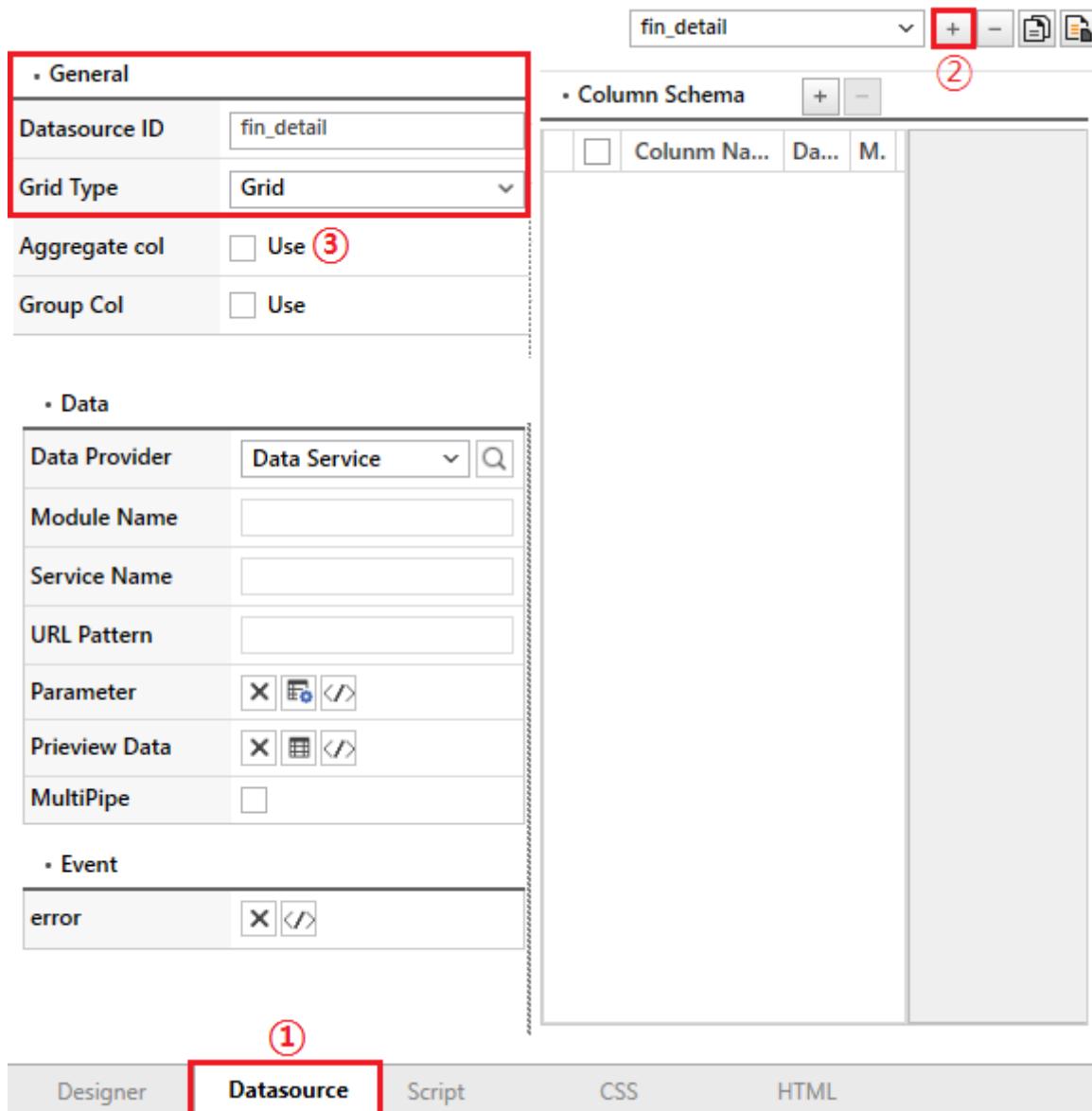
다음 금융상품 Detail(하단) 그리드의 데이터를 가져오기 위한 데이터소스를 추가하기 전에 Detail(하단) 그리드의 경우 Header(상단) 그리드를 선택할 때 해당 행의 값을 파라미터로 매핑하기 때문에 Header(상단) 그리드 속성 중 selectable(사용자가 그리드의 행을 선택할 수 있는지 여부)이 반드시 체크가 되어 있어야 한다.



<그림 9-76. 금융 상품 Header 그리드 selectable 속성 체크>

Detail(하단) 그리드에 데이터를 가져오기 위해 데이터소스를 추가한다. ① Datasource 탭을 선택하고 ② 을 눌러 데이터소스를 추가한다.

③ Datasource ID 를 fin\_detail, Grid 에 사용되므로 Grid Type 을 Grid 로 선택한다.



<그림 9-77. 금융상품 Detail Grid에 매핑할 데이터소스 추가>

Data Provider(데이터 제공자)에서 데이터 서비스를 검색한다. Data Service가 기본값으로 선택되어 있다. ① 검색 버튼을 클릭한다.

1

| General       |                              |
|---------------|------------------------------|
| Datasource ID | fin_detail                   |
| Grid Type     | Grid                         |
| Aggregate col | <input type="checkbox"/> Use |
| Group Col     | <input type="checkbox"/> Use |

| Data          |                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------|
| Data Provider | Data Service <span style="border: 1px solid red; padding: 2px;">①</span> <input type="button" value="Search"/> |
| Module Name   |                                                                                                                |
| Service Name  |                                                                                                                |
| URL Pattern   |                                                                                                                |
| Parameter     | <input type="button" value="X"/> <input type="button" value="E"/> <input type="button" value="C"/>             |
| Prievew Data  | <input type="button" value="X"/> <input type="button" value="B"/> <input type="button" value="C"/>             |
| MultiPipe     | <input type="checkbox"/>                                                                                       |

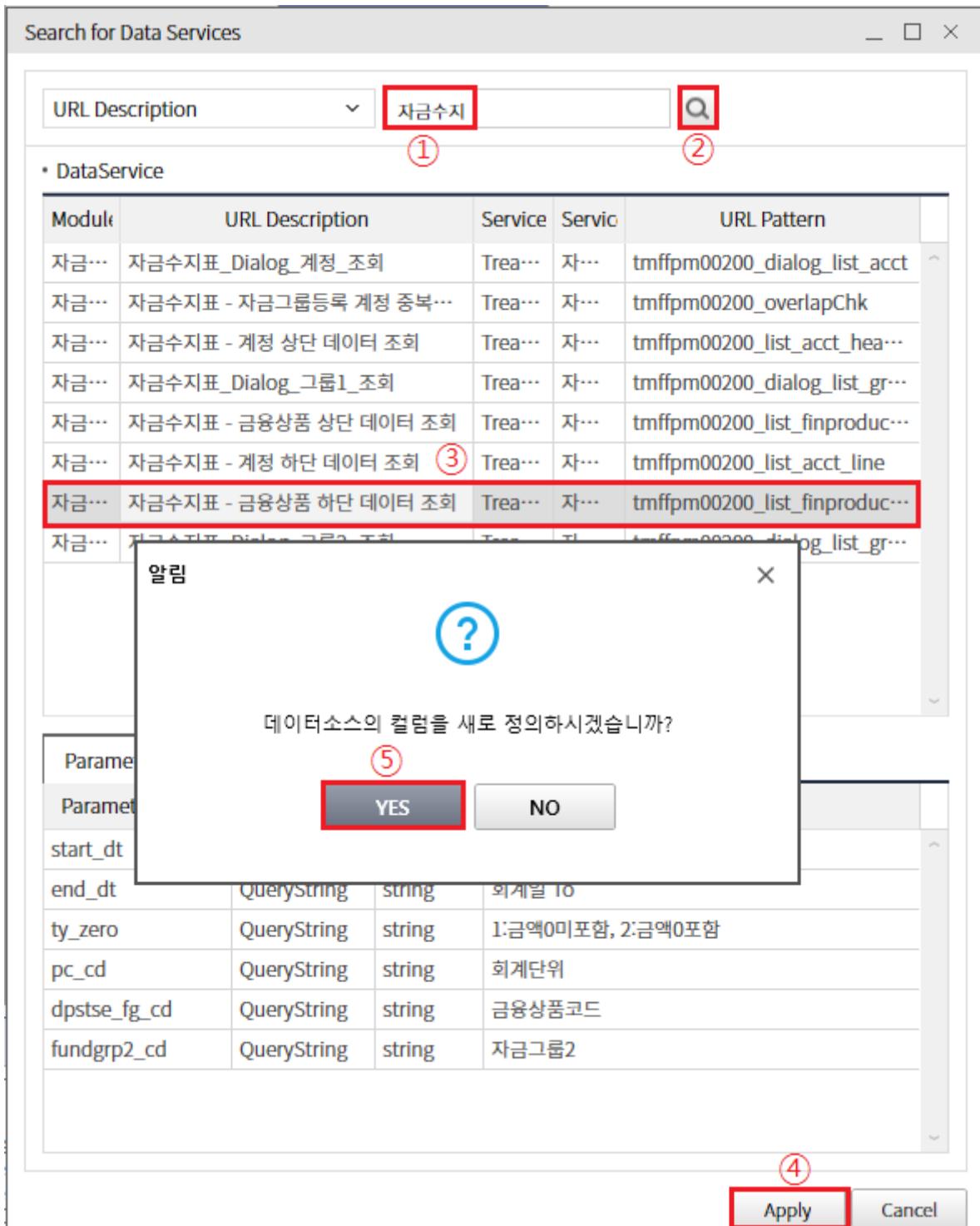
| Event |                                                                   |
|-------|-------------------------------------------------------------------|
| error | <input type="button" value="X"/> <input type="button" value="C"/> |

|          |            |        |     |
|----------|------------|--------|-----|
| Designer | Datasource | Script | CSS |
|----------|------------|--------|-----|

<그림 9-78. Data Provider 화면 열기>

Search for Data Services 화면에서 ① 자금수지를 입력하고 엔터키나 ② Search(검색)한다.  
 ③ URL Description 필드에 자금수지표 – 금융상품 하단 데이터 조회를 선택하고 ④ Apply 를 클릭한다. ⑤ 알림창이 뜨면 예를 선택한다.



<그림 9-79. URL Description 검색 조건 입력>

다음 Parameter에서 ① Mapping... 버튼을 눌러 나오는 ② 파라미터 매핑  화면에서 적용 형식, 컨트롤/데이터 형, 데이터 값은 <그림 9-80>처럼 선택한다. cd\_pc, cd\_tp\_deposit, cd\_fg2 파라미터의 컨트롤은 Header 그리드 ID인 Grid5를 선택하고 데이터 값은 선택된 행(selectedRow)의 해당 컬럼 선택하고, 나머지 파라미터는 조회조건에 있는 컨트롤들을 매핑한다.

• Data

|               |                                                                                         |                                                                                     |
|---------------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Data Provider | Data Service                                                                            |  |
| Module Name   | TR                                                                                      |                                                                                     |
| Service Name  | TreasuryManagementFundFPMService                                                        |                                                                                     |
| URL Pattern   | tmffpm00200_list_finproduct_line                                                        | ①                                                                                   |
| Parameter     |  </> |                                                                                     |
| Prievew Data  |  </> |                                                                                     |
| MultiPipe     |                                                                                         |                                                                                     |

↓

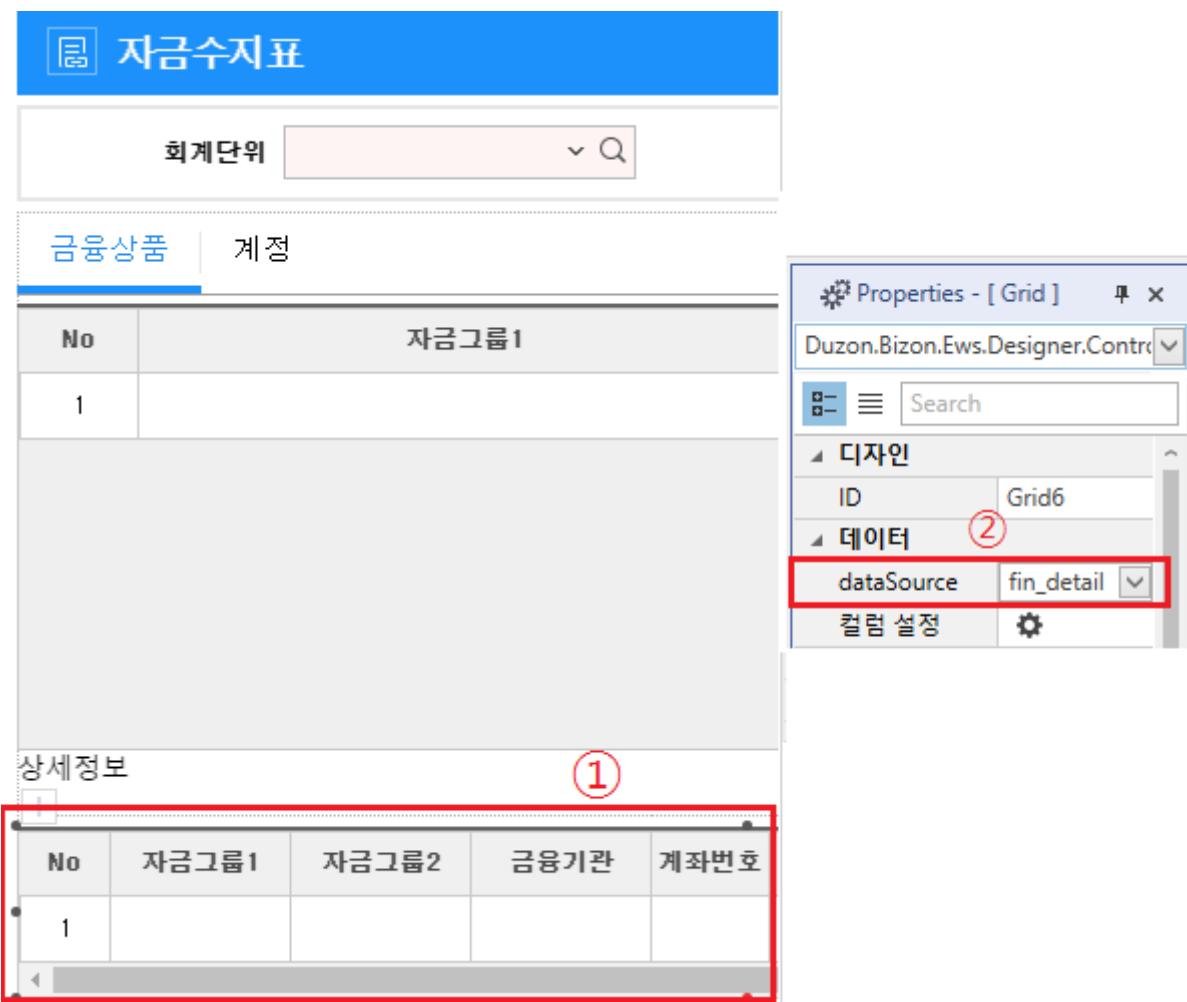
파라미터 매핑

• 매핑 정보

| 파라미터 명       | 적용 형식 | 컨트롤/데이터 형 | ②           | 데이터 값        |
|--------------|-------|-----------|-------------|--------------|
| start_dt     | 컨트롤   | acct_dt   | selectedRow | PC_CD        |
| end_dt       | 컨트롤   | acct_dt   | selectedRow | DPSTSE_FG_CD |
| ty_zero      | 컨트롤   | ty_zero   | selectedRow | FUNDGRP2_CD  |
| pc_cd        | 컨트롤   | Grid5     | selectedRow |              |
| dpstse_fg_cd | 컨트롤   | Grid5     | selectedRow |              |
| fundgrp2_cd  | 컨트롤   | Grid5     | selectedRow |              |

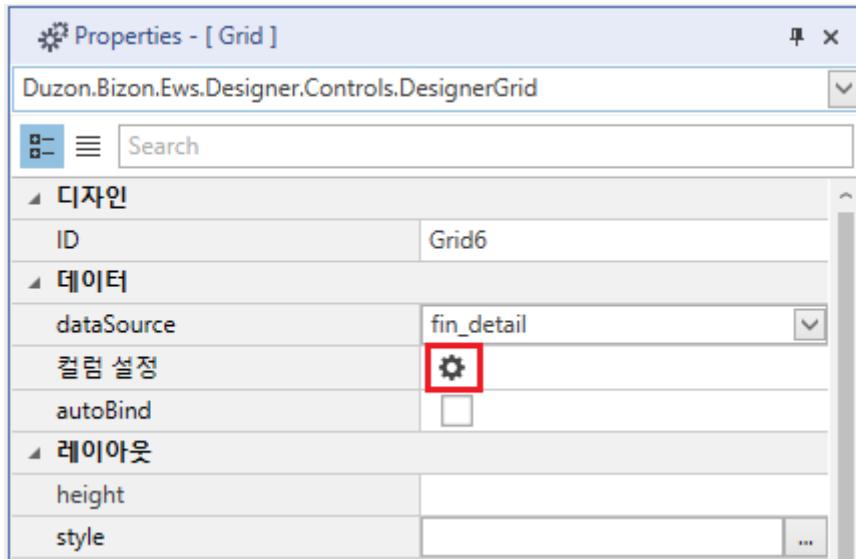
<그림 9-80. 금융 상품 Detail 파라미터 매핑>

다시 Designer 탭으로 가서 금융상품 탭에 하단 ② Grid를 선택한 후에, ③ DataSource 속성에서 fin\_detail을 선택한다.



<그림 9-81. 금융 상품 Detail Grid 의 dataSource 속성 설정 >

컬럼별로 속성을 정해주기 위해 컬럼설정  을 클릭한다.



<그림 9-82. 금융 상품 Detail Grid 컬럼 설정 열기>

그리드-컬럼설정 화면에서 컬럼별 속성을 <표 9-7>과 같이 설정한다. (출력 컬럼만 설정)

| 컬럼명                 | 설정       |       |                                 |                      |
|---------------------|----------|-------|---------------------------------|----------------------|
|                     | 헤더타이틀    | Width | footer                          | 텍스트정렬/출력포맷/포맷<br>문자열 |
| PARTNER_NM          | 금융기관     | 90    | (빈칸)                            | 왼쪽/(빈칸)/(빈칸)         |
| FINPRODUCT_NM       | 계좌명      | 140   | text : 계                        | 왼쪽/(빈칸)/(빈칸)         |
| FINPRODUCT_NO       | 계좌번호     | 130   | (빈칸)                            | 오른쪽/숫자포맷/ #,##0.##   |
| AM_BASIC            | 기초(개시)잔액 | 120   | type : sum<br>format : #,##0.## | 오른쪽/숫자포맷/ #,##0.##   |
| OTHACCTALT_AMT      | 입금       | 120   | type : sum<br>format : #,##0.## | 오른쪽/숫자포맷/ #,##0.##   |
| OTHACCTALT_SALE_AMT | 출금       | 120   | type : sum<br>format : #,##0.## | 오른쪽/숫자포맷/ #,##0.##   |
| AM_FINAL            | 기말(마감)잔액 | 120   | type : sum<br>format : #,##0.## | 오른쪽/숫자포맷/ #,##0.##   |
| BLNC_AMT            | 계좌잔액     | 120   | type : sum<br>format : #,##0.## | 오른쪽/숫자포맷/ #,##0.##   |

<표 9-7. 컬럼별 속성(출력컬럼만)>

DataSource - fin\_detail

**Add Header** **Remove Header** **Change Group** **Change Hidechild Headers** **C**

| 금융기관 | 계좌명 | 계좌번호 | 기초(개시)잔액 |  |
|------|-----|------|----------|--|
|------|-----|------|----------|--|

\* Column Information

|  | Column Name   | Header Caption | <input type="checkbox"/>            | Width |   |
|--|---------------|----------------|-------------------------------------|-------|---|
|  | FUNDGRP1_CD   | 자금그룹1 코드       | <input type="checkbox"/>            | 80    | ^ |
|  | FUNDGRP1_NM   | 자금그룹1          | <input type="checkbox"/>            | 90    |   |
|  | FUNDGRP2_CD   | 자금그룹2 코드       | <input type="checkbox"/>            | 80    |   |
|  | FUNDGRP2_NM   | 자금그룹2          | <input type="checkbox"/>            | 90    |   |
|  | PARTNER_CD    | 금융기관코드         | <input type="checkbox"/>            | 80    |   |
|  | PARTNER_NM    | 금융기관           | <input checked="" type="checkbox"/> | 111   |   |
|  | DPSTSE_FG_CD  | 금융상품구분코드       | <input type="checkbox"/>            | 80    |   |
|  | NM_TP_DEPOSIT | 금융상품구분명        | <input type="checkbox"/>            | 90    |   |
|  | ACCT_CD       | 계정코드           | <input type="checkbox"/>            | 80    |   |
|  | ACCT_NM       | 계정명            | <input type="checkbox"/>            | 90    |   |
|  | FINPRODUCT_CD | 금융상품코드         | <input type="checkbox"/>            | 80    |   |
|  | FINPRODUCT_NM | 계좌명            | <input checked="" type="checkbox"/> | 140   |   |
|  | FINPRODUCT_NO | 계좌번호           | <input checked="" type="checkbox"/> | 130   |   |
|  | AM_BASIC      | 기초(개시)잔액       | <input checked="" type="checkbox"/> | 120   |   |

| field            | BLNC_AMT                            |
|------------------|-------------------------------------|
| 헤더 타이틀           | 계좌잔액                                |
| visible          | <input checked="" type="checkbox"/> |
| Width            | 120                                 |
| 텍스트 정렬           | 가운데                                 |
| sortable         | <input checked="" type="checkbox"/> |
| style            |                                     |
| prefix           |                                     |
| suffix           |                                     |
| attributes       | none                                |
| hideChildHeaders | <input type="checkbox"/>            |
| merge            | <input type="checkbox"/>            |
| grouping         | <input type="checkbox"/>            |
| movable          | <input checked="" type="checkbox"/> |
| ▶ footer         |                                     |
| ▶ 출력 포맷          | 숫자 포맷                               |

<그림 9-83. 금융상품 Detail 그리드 그리드-컬럼 설정 화면>

<그림 9-84>은 설정을 완료한 금융상품 Detail(하단) 그리드 화면이다.

| No | 자금그룹  |       | 금융상품 | 기초(개시)잔액 |  |
|----|-------|-------|------|----------|--|
|    | 자금그룹1 | 자금그룹2 |      |          |  |
| 1  |       |       |      |          |  |

| No | 금융기관 | 계좌명 | 계좌번호 | 기초(개시)잔액 |
|----|------|-----|------|----------|
| 1  |      |     |      |          |

<그림 9-84. 금융상품 Detail(하단) 그리드 화면>

이제 계정탭의 Header 그리드의 데이터를 가져오기 위하여 데이터소스를 추가하고  
<표 9-8>과 같이 설정한다

| 구분        | General                                            | Data                                                                                               |
|-----------|----------------------------------------------------|----------------------------------------------------------------------------------------------------|
| 계정 Header | Datasource ID : acct_header<br>Grid Type : Grid 선택 | Data Provider : Data Service<br>URL Description : 자금수지표 - 계정 상단 데이터 조회<br>Parameter : <그림 9-83> 참조 |

<표 9-8. 계정 Header 그리드에 바인딩 할 데이터소스 설정>

파라미터 매핑

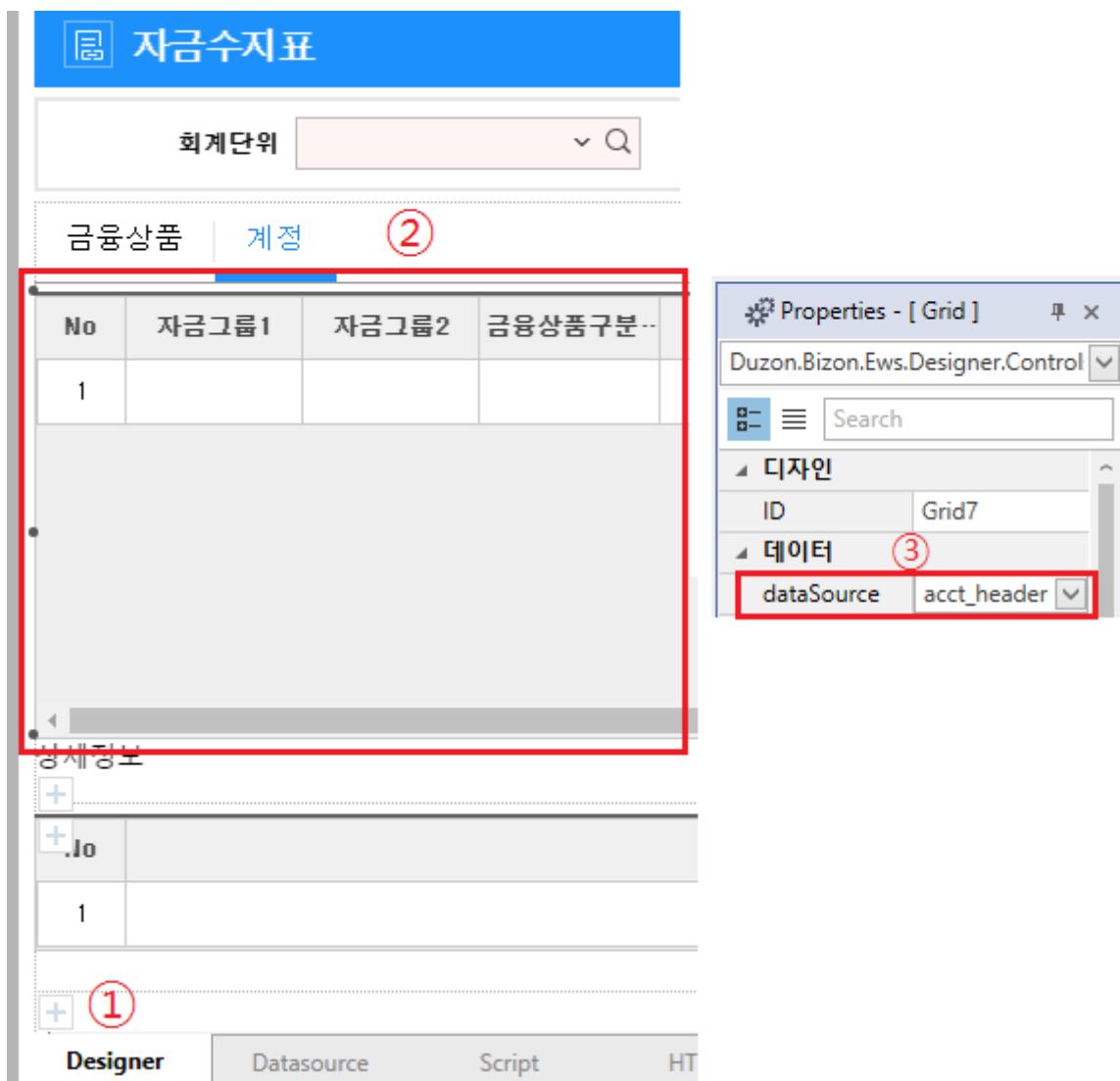
• 매핑 정보

| 파라미터 명      | 적용 형식 | 컨트롤/데이터 형 | 데이터 값     |
|-------------|-------|-----------|-----------|
| start_dt    | 컨트롤   | acct_dt   | startDate |
| ty_zero     | 컨트롤   | ty_zero   | value     |
| end_dt      | 컨트롤   | acct_dt   | endDate   |
| pc_cd       | 컨트롤   | pc_cd     | codes     |
| fundgrp1_cd | 컨트롤   | fg1_cd    | codes     |
| fundgrp2_cd | 컨트롤   | fg2_cd    | codes     |

OK

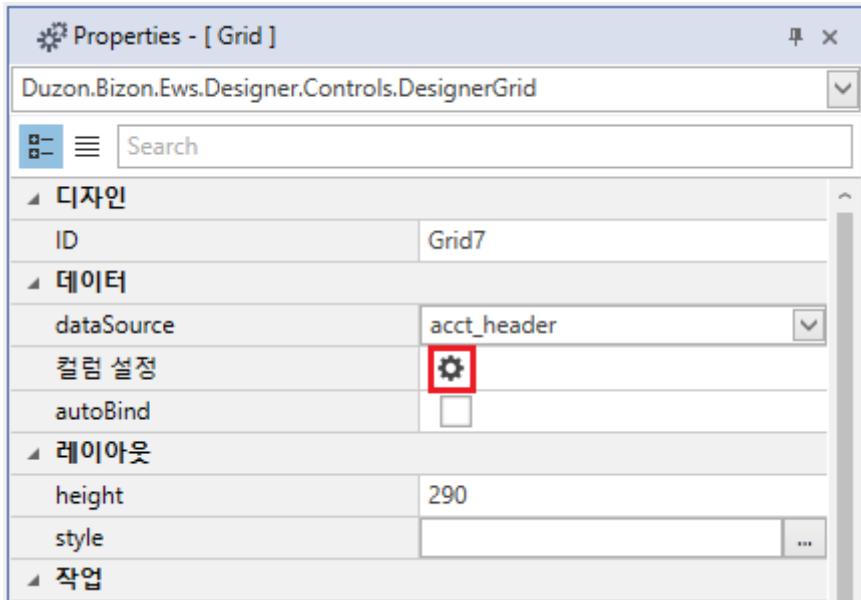
<그림 9-85. 계정 Header 데이터소스 파라미터 매핑>

다시 ① Designer 탭으로 가서 계정 탭에 상위 ② Grid를 선택한 후에, ③ DataSource 속성에서 acct\_header를 선택한다.



<그림 9-86. 계정 Header Grid 의 dataSource 속성 설정>

컬럼별로 속성을 정해주기 위해 컬럼설정 을 클릭한다.



<그림 9-87. 계정 Header Grid 컬럼 설정 열기>

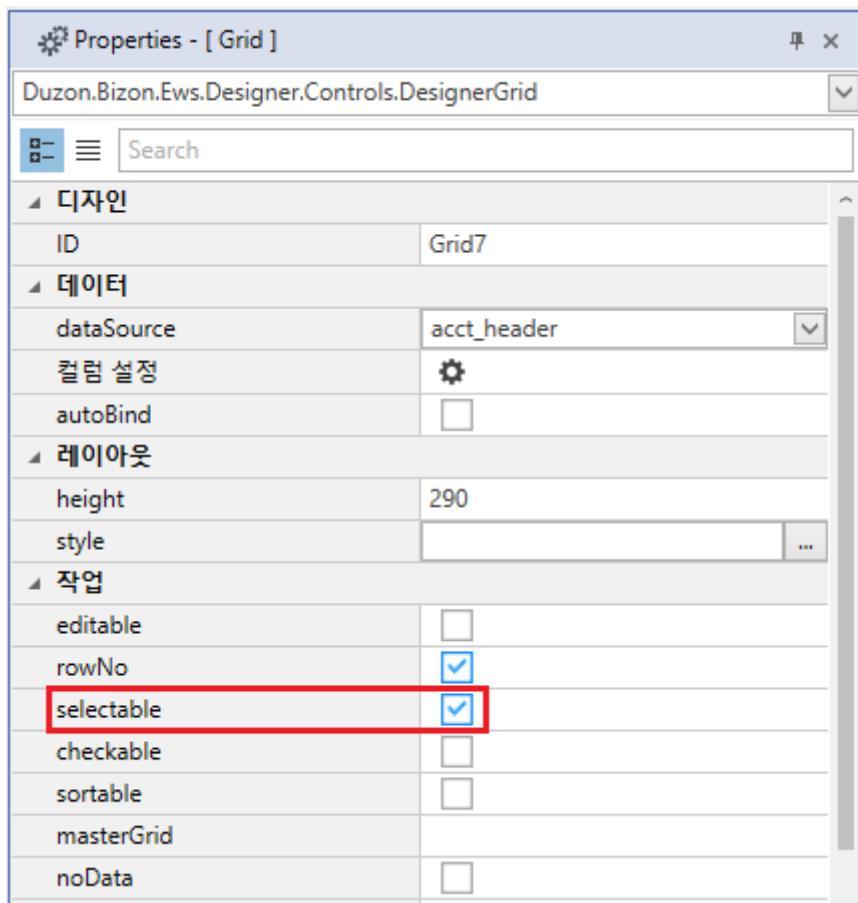
그리드-컬럼설정 화면에서 컬럼별 속성을 <표 9-9>과 같이 설정한다. (출력 컬럼만 설정)

| 컬럼명                 | 설정       |       |                    |
|---------------------|----------|-------|--------------------|
|                     | 헤더타이틀    | Width | 텍스트정렬/출력포맷/포맷 문자열  |
| FUNDGRP1_NM         | 자금그룹1    | 70    | 왼쪽/(빈칸)/(빈칸)       |
| FUNDGRP2_NM         | 자금그룹2    | 70    | 왼쪽/(빈칸)/(빈칸)       |
| ACCT_NM             | 계정       | 70    | 왼쪽/(빈칸)/(빈칸)       |
| AM_BASIC            | 기초(개시)잔액 | 120   | 오른쪽/숫자포맷/ #,##0.## |
| OTHACCTALT_AMT      | 입금       | 120   | 오른쪽/숫자포맷/ #,##0.## |
| OTHACCTALT_SALE_AMT | 출금       | 120   | 오른쪽/숫자포맷/ #,##0.## |
| AM_FINAL            | 기말(마감)잔액 | 120   | 오른쪽/숫자포맷/ #,##0.## |
| BLNC_AMT            | 계좌잔액     | 120   | 오른쪽/숫자포맷/ #,##0.## |

<표 9-9. 컬럼별 속성(출력컬럼만)>

다음 계정탭의 Detail 그리드의 데이터를 가져오기 위한 데이터소스를 추가하기 전에 금융상품 디테일 그리드와 같이 계정 Detail 그리드의 경우도 계정 Header 그리드를 선택할 때 해당 행의 값을 파라미터로 매핑하기 때문에 계정 Header 그리드 속성 중 selectable(사용자가 그리드의 행을 선택할 수 있는지 여부)이 반드시 체크가 되어 있어야 한다.

금융상품 탭의 Master 그리드 처럼 계정탭의 Master 그리드 경우도 <그림 9-73>과 <그림 9-74>를 참조하여 자금그룹 1과 자금그룹 2를 병합하여 자금그룹 타이틀을 추가한다.



<그림 9-88. 계정 Header 그리드 selectable 속성 체크>

이제 계정탭의 Detail 그리드의 데이터를 가져오기 위하여 데이터소스를 추가하고 <표 9-8>과 같이 설정한다.

| 구분        | General                                            | Data                                                                                                 |
|-----------|----------------------------------------------------|------------------------------------------------------------------------------------------------------|
| 계정 Detail | Datasource ID : acct_detail<br>Grid Type : Grid 선택 | Data Provider : Data Service<br>URL Description : 자금수지표 - 계정<br>하단 데이터 조회<br>Parameter: <그림 9-89> 참조 |

<표 9-10. 계정 Detail 그리드에 바인딩 할 데이터소스 설정>

파라미터 매핑

• 매핑 정보

| 파라미터 명   | 적용 형식 | 컨트롤/데이터 형 | 데이터 값               |
|----------|-------|-----------|---------------------|
| start_dt | 컨트롤   | acct_dt   | startDate           |
| end_dt   | 컨트롤   | acct_dt   | endDate             |
| ty_zero  | 컨트롤   | ty_zero   | value               |
| pc_cd    | 컨트롤   | Grid7     | selectedRow PC_CD   |
| acct_cd  | 컨트롤   | Grid7     | selectedRow ACCT_CD |

<그림 9-89. 계정 Detail 데이터소스 파라미터 매핑>

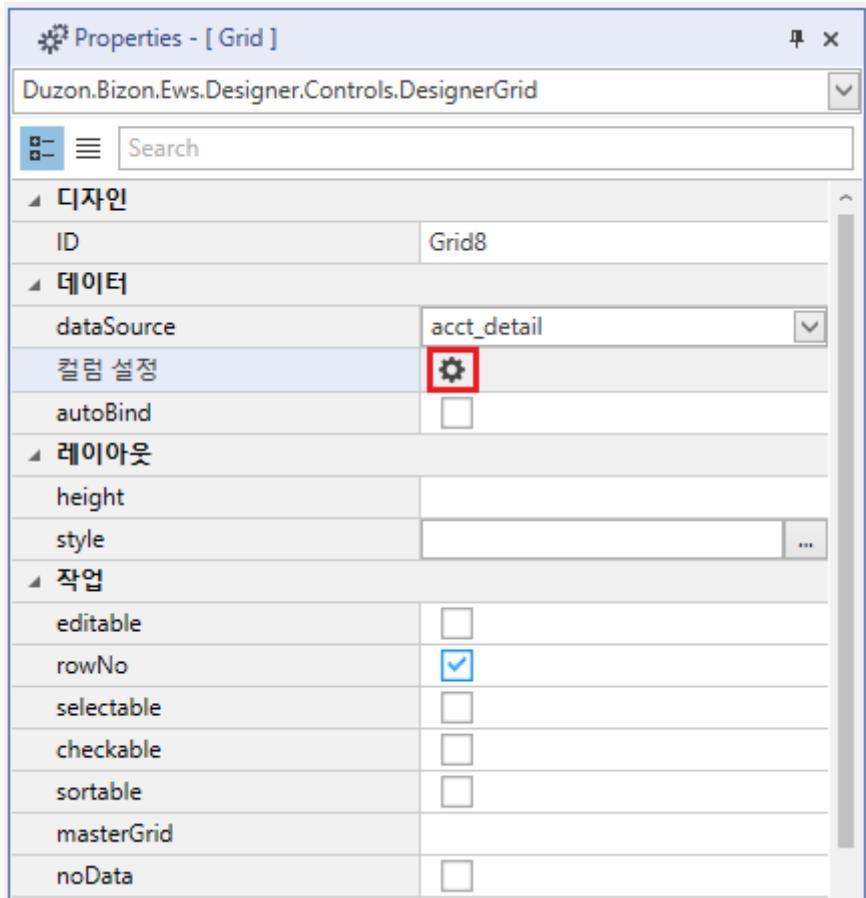
다시 ① Designer 탭으로 가서 계정 탭에 하단 ② Grid를 선택한 후에, ③ DataSource 속성에서 acct\_detail을 선택한다.

The screenshot shows the Bizon EWS Designer interface with the following details:

- Top Bar:** 자금수지표 (Title), 회계단위 (Account Unit) dropdown, and a search icon.
- Left Navigation:** 금융상품 (Financial Product) and 계정 (Account) tabs. The 계정 tab is selected and highlighted in blue.
- Main Content:** A table titled "자금그룹" (Funding Group) with one row containing "1". Below it is a section titled "상세정보" (Detailed Information) containing a table with columns: No, 자금그룹1, 자금그룹2, 금융기관, 계좌번호. This table is also highlighted with a red box.
- Properties Panel:** Properties - [Grid] window showing the following settings:
  - ID: Grid8
  - 데이터 (Data): dataSource (highlighted with a red box)
  - .dataSource value: acct\_detail (highlighted with a red box and circled with a red circle)
- Bottom Tab Bar:** Designer (highlighted with a red box and circled with a red circle), Datasource, Script, and HTML.

<그림 9-90. 계정 Detail Grid 의 dataSource 속성 설정>

컬럼별 속성을 정해주기 위해 컬럼설정 을 클릭한다.



<그림 9-91. 계정 Header Grid 컬럼 설정 열기>

그리드-컬럼설정 화면에서 컬럼별 속성을 <표 9-11>과 같이 설정한다(출력 컬럼만 설정)

| 컬럼명                 | 설정       |       |                                 |                      |
|---------------------|----------|-------|---------------------------------|----------------------|
|                     | 헤더타이틀    | Width | footer                          | 텍스트정렬/출력포맷/포맷<br>문자열 |
| PARTNER_NM          | 금융기관     | 90    | (빈칸)                            | 왼쪽/(빈칸)/(빈칸)         |
| FINPRODUCT_NM       | 계좌명      | 140   | text : 계                        | 왼쪽/(빈칸)/(빈칸)         |
| FINPRODUCT_NO       | 계좌번호     | 130   | (빈칸)                            | 오른쪽/숫자포맷/ #,##0.##   |
| AM_BASIC            | 기초(개시)잔액 | 120   | type : sum<br>format : #,##0.## | 오른쪽/숫자포맷/ #,##0.##   |
| OTHACCTALT_AMT      | 입금       | 120   | type : sum<br>format : #,##0.## | 오른쪽/숫자포맷/ #,##0.##   |
| OTHACCTALT_SALE_AMT | 출금       | 120   | type : sum<br>format : #,##0.## | 오른쪽/숫자포맷/ #,##0.##   |

|          |          |     |                                 |                    |
|----------|----------|-----|---------------------------------|--------------------|
| AM_FINAL | 기말(마감)잔액 | 120 | type : sum<br>format : #,##0.## | 오른쪽/숫자포맷/ #,##0.## |
| BLNC_AMT | 계좌잔액     | 120 | type : sum<br>format : #,##0.## | 오른쪽/숫자포맷/ #,##0.## |

<표 9-11. 컬럼별 속성(출력컬럼만)>

설정을 완료한 계정 탭 화면이다.

| No | 자금그룹  | 계정    | 기초(개시)잔액 | 입금 | 출금 |
|----|-------|-------|----------|----|----|
|    | 자금그룹1 | 자금그룹2 |          |    |    |
| 1  |       |       |          |    |    |

| No | 금융기관 | 계좌명 | 계좌번호 |
|----|------|-----|------|
| 1  |      |     |      |

<그림 9-92. 완료된 계정 탭 화면>

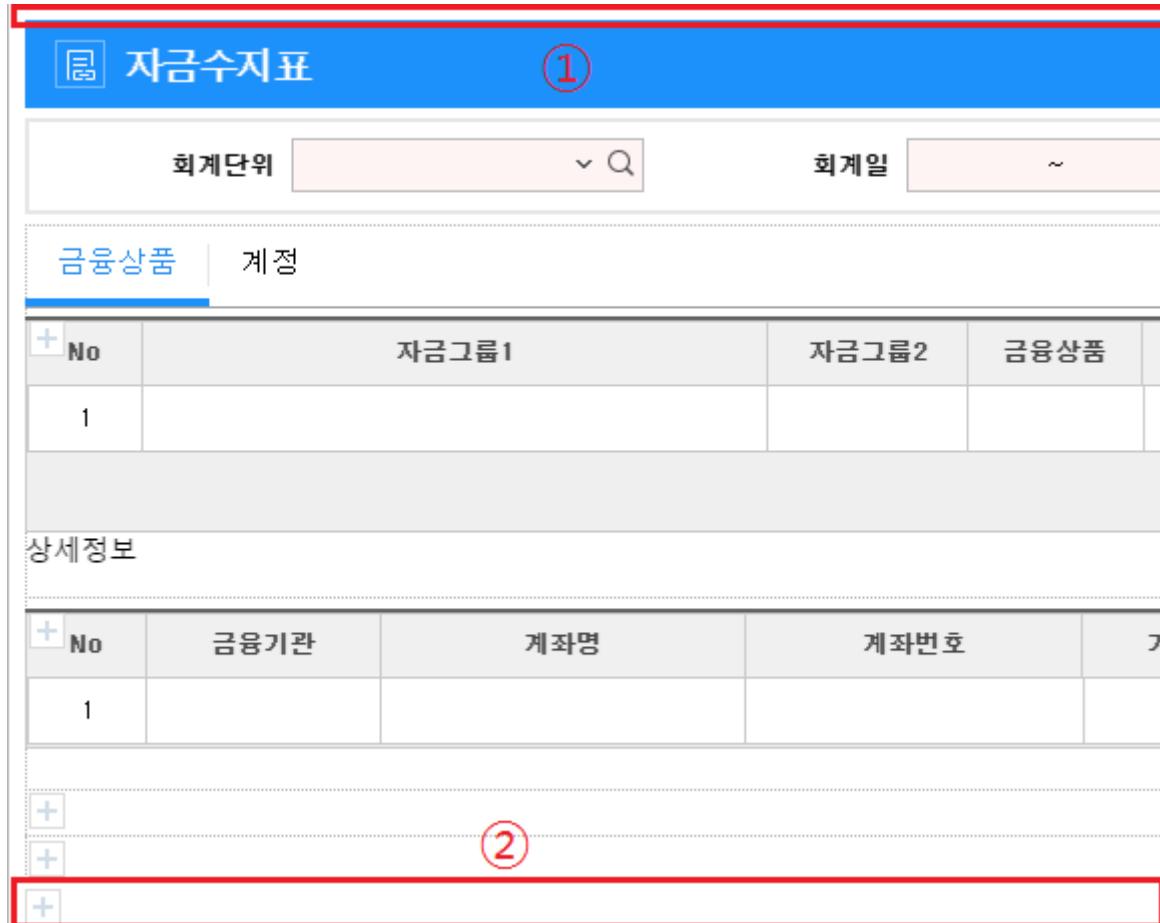
#### 4) 이벤트 스크립트 설정

자금수지표에서 사용되는 이벤트 스크립트 이다.

| 컨트롤(객체)             | 이벤트      | 설명                         |
|---------------------|----------|----------------------------|
| Cavas               | preReady | 페이지가 로드될 때 초기값 설정          |
| MainButton - Search | click    | 조회 버튼 눌렀을 때 필수 조회값 체크 및 조회 |
| 금융상품 – Header Grid  | change   | 금융상품 Detail Grid 조회        |
| 계정 – Header Grid    | change   | 계정 Detail Grid 조회          |

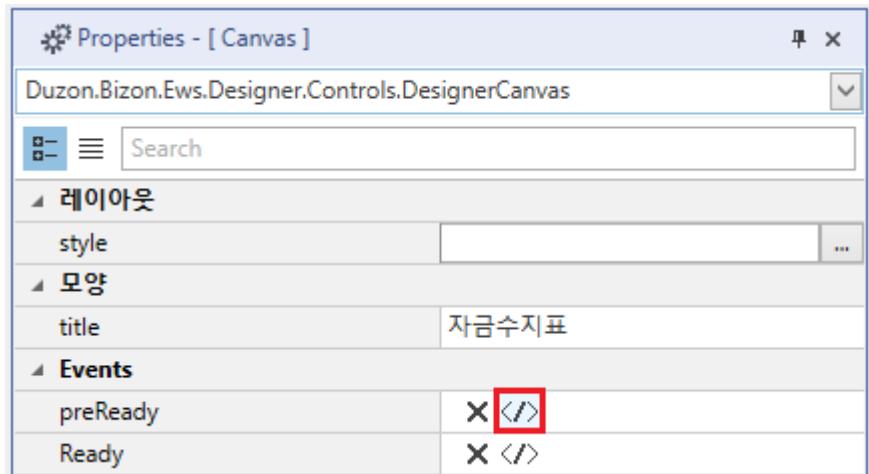
<표 9-12. 자금수지표 이벤트>

먼저 초기값을 설정하는 스크립트를 작성한다. 자금수지표는 회계단위와 회계일에 초기값을 설정하며, 최상위 Cavaus 의 Events 에서 구현한다. Cavaus 는 문서 제목 위에 ① 상단 또는 아래 ② 빙 하단 부분을 클릭하여 선택한다.



<그림 9-93. Cavaus 선택>

이제 이벤트 preReady 스크립트 버튼 </>을 눌러 CodeEditor 화면에서 2 개의 컨트롤을 초기값을 설정하는 스크립트를 작성한다.



<그림 9-94. preReady 이벤트 스크립트 버튼 실행>

```
// 회계 단위 초기값 설정
self.pc_cd.setData([
 // Code
 PC_CD:self.user.profitCenterCode,
 // Text
 PC_NM:self.user.profitCenterName
]);
```

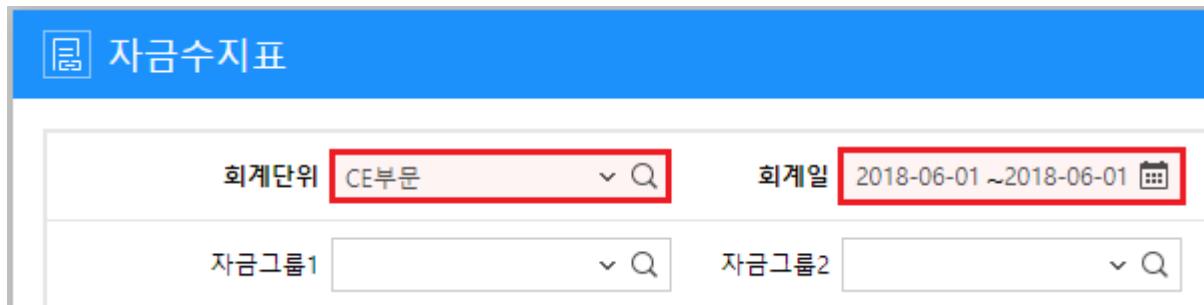
  

```
// 회계일 초기값 설정(오늘 일자)
self.acct_dt.setStartDate(new Date());
self.acct_dt.setEndDate(new Date());
```

| [DesignerCanvas] |                                        |
|------------------|----------------------------------------|
| 1                | // 회계 단위 초기값 설정                        |
| 2                | self.pc_cd.setData([         // Code   |
| 3                | PC_CD:self.user.profitCenterCode,      |
| 4                | // Text                                |
| 5                | PC_NM:self.user.profitCenterName       |
| 6                | ]);                                    |
| 7                |                                        |
| 8                |                                        |
| 9                | // 회계일 초기값 설정(오늘 일자)                   |
| 10               | self.acct_dt.setStartDate(new Date()); |
| 11               | self.acct_dt.setEndDate(new Date());   |

<그림 9-95. Cavaus preReady Event 스크립트 작성>

툴바에서 Build & Preview(Ctrl+B)버튼을 눌러 미리보기를 실행하면 조회 조건 컨트롤들의 초기값이 설정되어 있다.



<그림 9-96. 미리보기에서 초기값 확인>

다음으로 필수 조회 조건을 체크하고 조회하는 이벤트 스크립트를 추가한다.

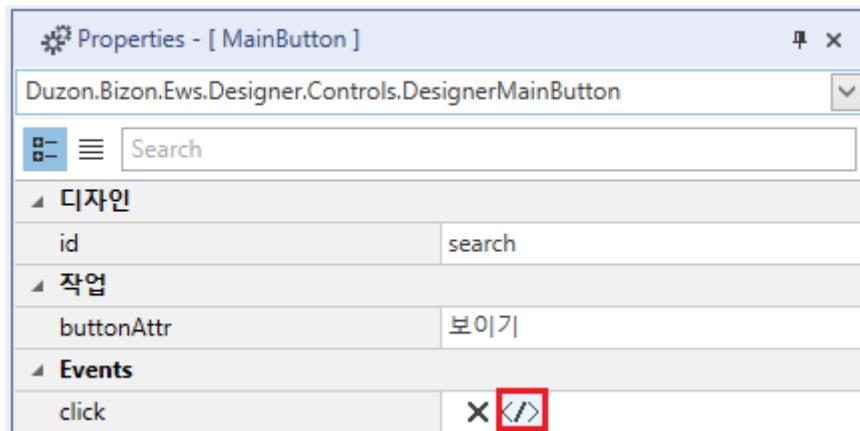
필수 조회 조건 체크는 조회 조건들을 포함하고 있는 ConditionPanel에서 체크하며, 조회는 금융상품 Header Grid와 계정 Header Grid와 바인딩되는 Datasource의 read 함수를 호출한다.

조회버튼을 선택한다.



<그림 9-97. 조회 버튼 선택>

프로퍼티 창에서 click 이벤트를 클릭한다.



<그림 9-98. Click 이벤트 선택>

CodeEditor 화면에서 다음과 같이 입력하고 OK를 누른다.

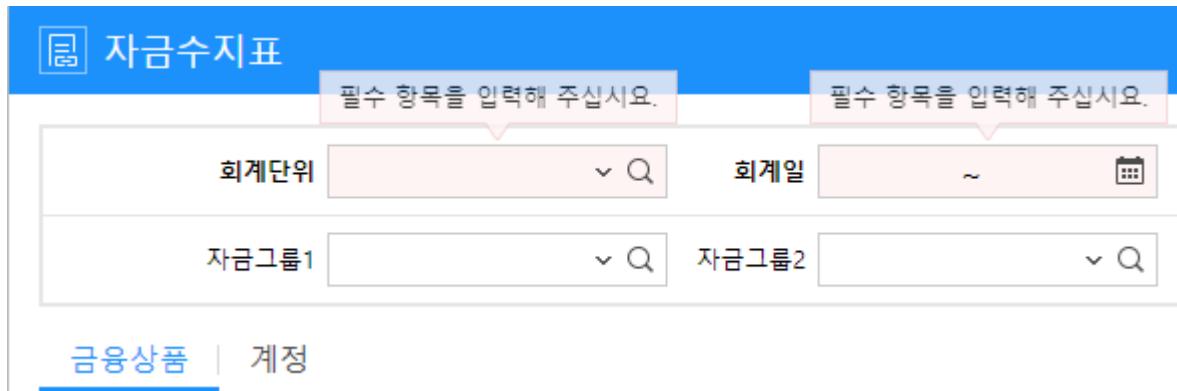
```
// 필수 조회 조건 체크(컨디션 패널 ID)
if (self.searchCondition.validate({tooltip:true,message:'필수 항목을 입력해 주십시오.'})) {
 // 금융상품 Header 데이터소스 읽기
 self.fin_header.read();
 // 계정 Header 데이터소스 읽기
 self.acct_header.read(); }
```

### search [DesignerMainButton]

```
1 function() {
2 // TODO: 데이터소스를 조회합니다.
3 // 필수 조회 조건 체크(컨디션 패널 ID)
4 if (self.searchCondition.validate({tooltip:true,message:'필수 항목을 입력해 주십시오.'})) {
5 // 금융상품 Header 데이터소스 읽기
6 self.fin_header.read();
7 // 계정 Header 데이터소스 읽기
8 self.acct_header.read();
9 }
10 }
11 }
```

<그림 9-99. 조회 버튼 Click 이벤트에 스크립트 입력 화면>

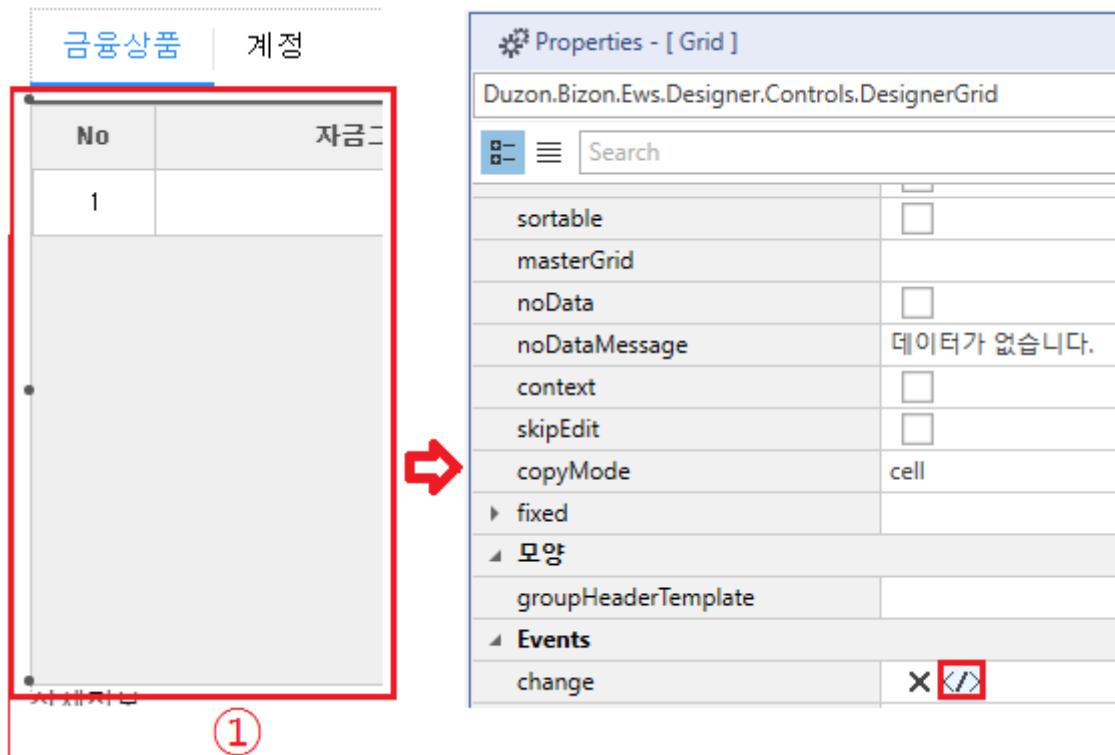
툴바에서  Build & Preview(Ctrl+B)버튼을 눌러 미리보기를 실행하여 회계단위와 회계일을 초기값을 지우고 조회하면 '필수 항목을 입력해 주십시오'라는 메시지가 나타난다..



<그림 9-100. 미리보기에서 필수 조회 조건 확인>

다음으로 조회된 금융상품 Header 그리드를 선택했을 때 Detail 그리드가 조회되는 이벤트 스크립트를 설정한다. 2 가지 방법으로 설정할 수 있으며, 첫번째는 detail 그리드 프로퍼티 화면에서 masterGrid 속성에 Header 의 Grid 를 설정해주는 방법이며 두번째는 Header 그리드 프로퍼티 화면에서 change 이벤트를 추가하는 방법이다. 여기에서는 두번째 방법으로 진행한다.

금융상품 Header 그리드를 선택하고 Events 의 change 스크립트 버튼 을 누른다



<그림 9-101. 금융상품 Header 그리드 change 스크립트 실행>

CodeEditor 화면에서 다음과 같이 입력하고 OK를 누른다.

```
function(e) {
 self.fin_detail.read();
}
```

```
1 function(e)
2 self.fin_detail.read();
3 }
```

<그림 9-102. 금융상품 Header 그리드 change 스크립트 작성>

같은 방법으로 계정 Header 그리드의 Change 이벤트 스크립트에서 다음과 같이 입력한다.

```
function(e) {
 self.acct_detail.read();
}
```

```

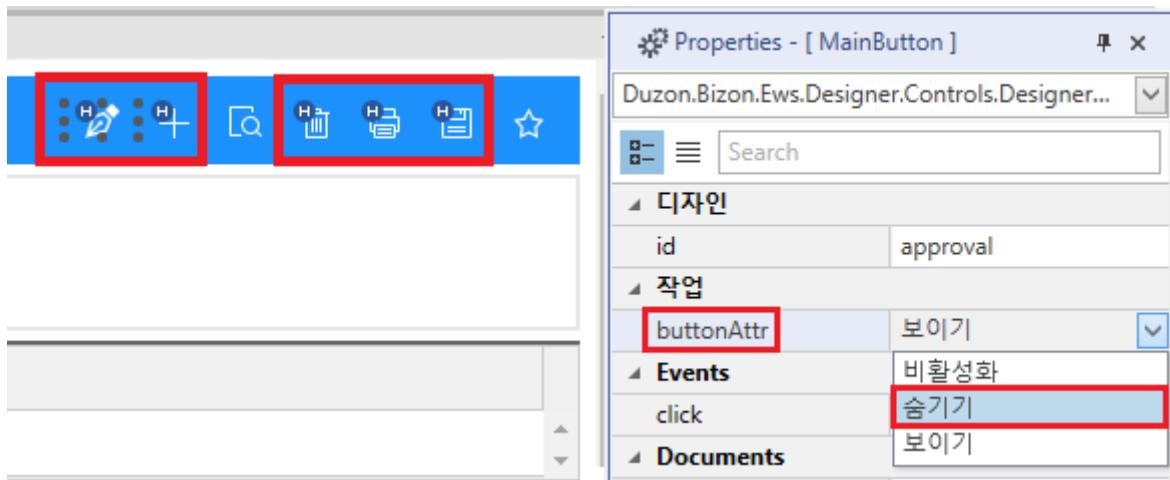
Grid7 [DesignerGrid]
change

1 function(e) {
2 self.acct_detail.read();
3 }

```

<그림 9-103. 계정 Header 그리드 change 스크립트 작성>

상단 우측 나머지 버튼들은 사용하지 않기 때문에 buttonAttr 속성을 숨기기로 설정한다.



<그림 9-104. 나머지 버튼 buttonAttr 속성 변경>

툴바에서 Build & Preview(Ctrl+B) 버튼을 눌러 미리보기를 실행하여 조회하면 정상적으로 데이터가 조회 되는 것을 확인할 수 있다.

## ■ 자금수지표

|       |      |     |       |                         |
|-------|------|-----|-------|-------------------------|
| 회계단위  | CE부문 | ▼ Q | 회계일   | 2018-10-17 ~ 2018-10-17 |
| 자금그룹1 |      | ▼ Q | 자금그룹2 |                         |

[금융상품](#) | [계정](#)

| No | 자금그룹     | 금융상품     | 기초(개시)잔액           |
|----|----------|----------|--------------------|
| >  | 현금및현금성자산 | 현금       | 보통예금 285,249,900.5 |
| 2  |          | 소계       | 285,249,900.5      |
| 3  |          | 합계       | 285,249,900.5      |
| 4  | 운용상품     | 매도가능금융자산 | 채무증권 10,500,000    |
| 5  |          | 소계       | 10,500,000         |
| 6  |          | 합계       | 10,500,000         |
| 7  |          | 총계       | 295,749,900.5      |

### 상세정보

| No | 금융기관    | 계좌명          | 계좌번호           | 기초(개시)잔액      |
|----|---------|--------------|----------------|---------------|
| >  | 국민은행    | 보통예금_국민      | 16308673513101 | 285,249,900.5 |
| 2  | (주)신한은행 | 보통예금- 신한_입출금 | 22222          | 10,500,000    |

계

28

<그림 9-105. 미리보기 후 조회 화면>

### 3. 운용상품조회

#### 1) 개발 화면

<그림 9-106>는 운용상품조회 예제 화면이다. 상단 조회조건의 경우 필수 조회 조건은 회계단위이고, 실행 시 초기값은 로그인한 정보이다. 회계단위, 구분, 금융기관은 필수 영역에 위치하며, 계정코드와 상품유형은 옵션 영역에 위치한다.

아래 데이터 부분은 왼쪽은 CardList, 오른쪽은 TapPanel로 되어 있으며, 그 밑에 Arcodien List가 있고 하위로 FormPanel이 위치하며 아래에 컨트롤들이 있다. 예제에서는 일반정보 탭의 기본정보 Arcodien List만 만들어 본다.

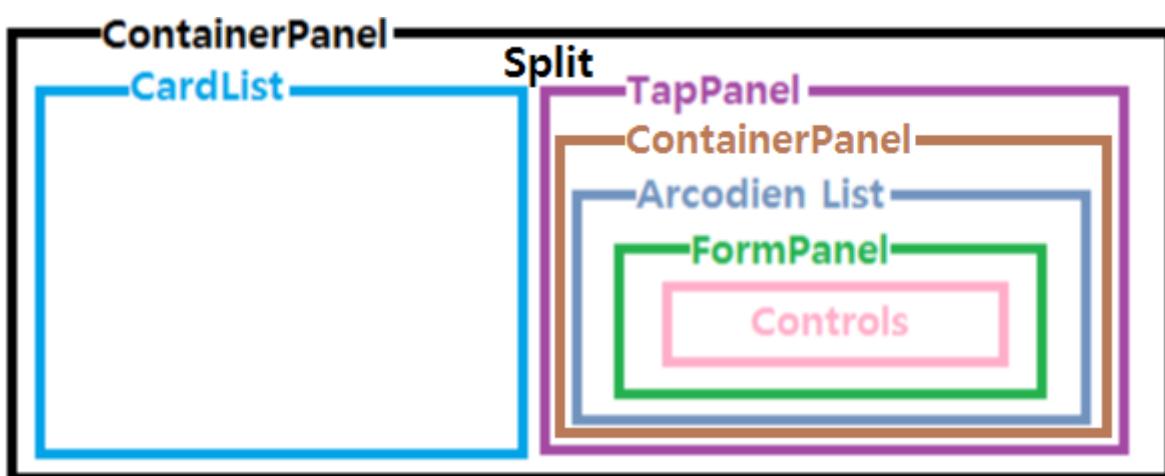
The screenshot shows the '운용상품조회' application interface. At the top, there are search filters for '회계단위' (CE부문), '구분' (필수 영역), '계정코드' (옵션 영역), and '상품유형'. Below the filters is a 'CardList' section containing two items:

| 취득일기준                      | 정렬     |
|----------------------------|--------|
| (액면발행) 투자사채 16-1차          | 2-1000 |
| 2016-05-15 / 3,600,000,000 |        |

| 취득일기준                      | 정렬     |
|----------------------------|--------|
| (할증발행) 투자사채 16-3차          | 2-1002 |
| 2016-05-15 / 3,800,000,000 |        |

To the right of the CardList is a 'TapPanel' with the '일반정보' tab selected. Below the TapPanel is a 'FormPanel' with fields for '계좌코드' (2-1000) and '금융기관' (신한은행). A search icon is also present.

<그림 9-106. 운용상품조회 개발 화면>



<그림 9-107. 데이터 부분 구조>

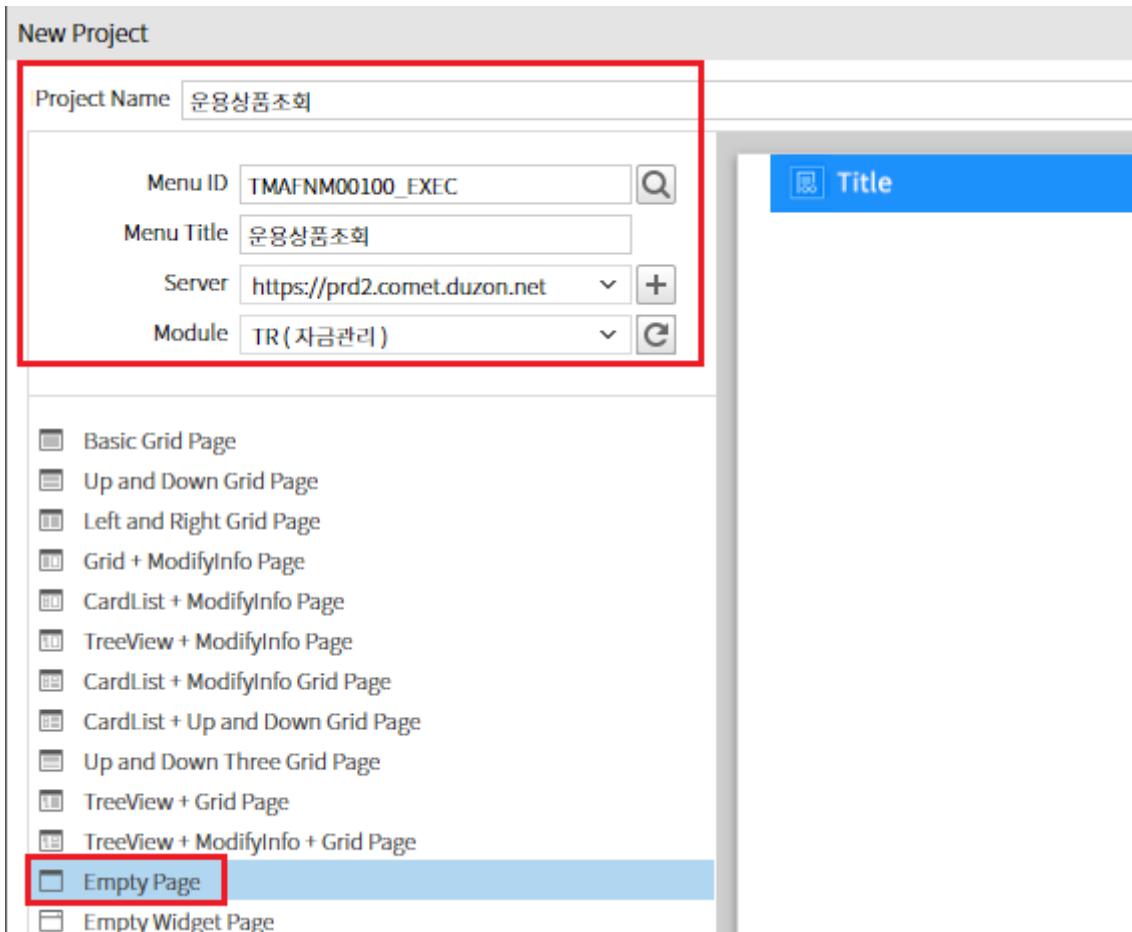
## 2) 화면 디자인

새 프로젝트를 생성한다. (File ▶ New ▶ Project... 또는 툴바의 

|                      |                                                                            |
|----------------------|----------------------------------------------------------------------------|
| Project Name (프로젝트명) | 운용상품조회 입력                                                                  |
| Menu ID (메뉴ID)       | TMAFNM00100_EXEC 입력                                                        |
| Menu Title (메뉴 제목)   | 운용상품조회 입력                                                                  |
| Server (서버 선택)       | <a href="https://prd2.comet.duzon.net">https://prd2.comet.duzon.net</a> 선택 |
| Module (모듈 선택)       | TR (자금관리) 선택                                                               |
| 템플릿(template)        | Empty Page 선택                                                              |

<표 9-13. 새 프로젝트 생성 입력>

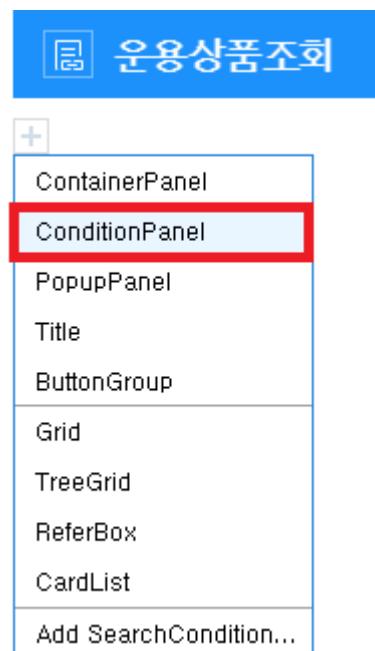
위 정보를 입력 후 OK 버튼을 클릭하여 프로젝트를 생성한다.



<그림 9-108. New Project 생성 화면 >

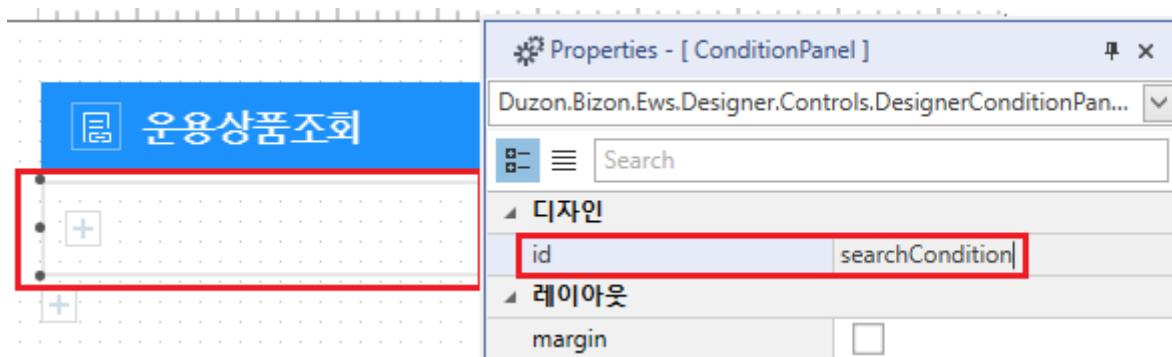
조회 조건에 해당하는 컨트롤들은 ConditionPanel, 데이터를 보여주는 그리드 컨트롤은 ContainerPanel에서 각각 생성한다. 먼저 조회 조건에 해당하는 컨트롤을 그린다.

 버튼을 누르고 ConditionPanel을 선택한다.



<그림 9-109. ConditionPanel 선택>

프로그램 실행 후 조회할 때 필수 조회 조건 체크를 하기 위해 ConditionPanel 선택 후 ID 속성을 searchCondition으로 입력한다.



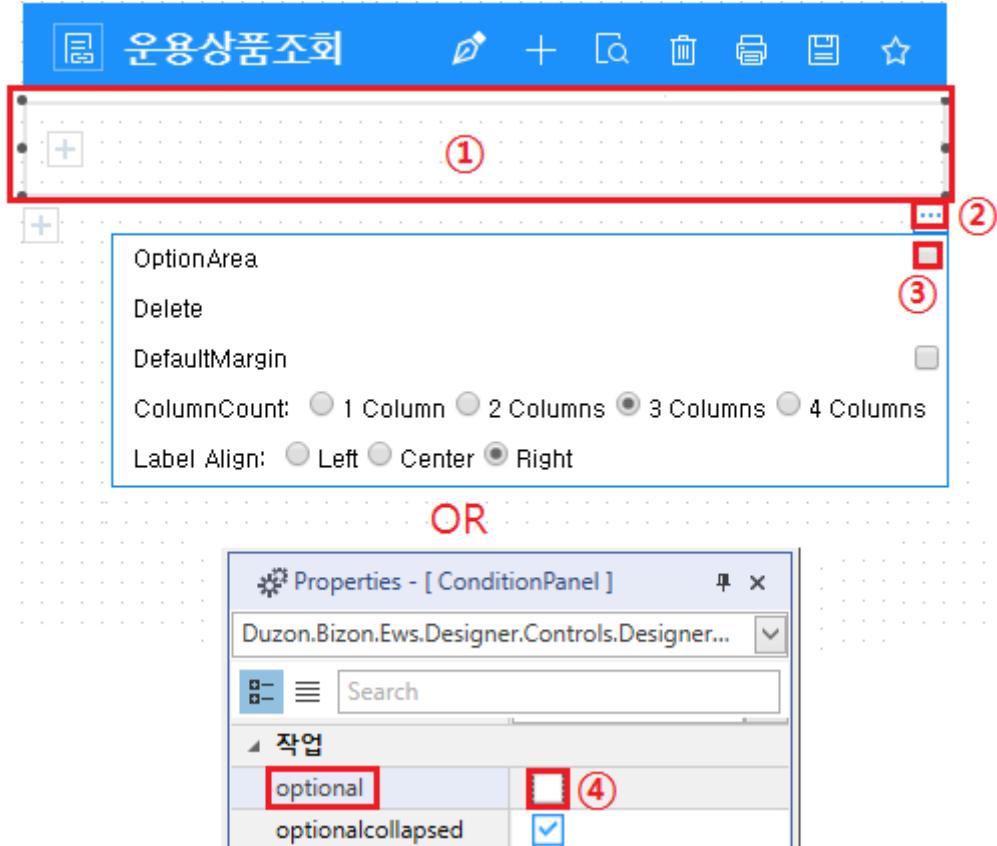
<그림 9-110. ConditionPanel 생성 및 ID 입력>

ConditionPanel 내의  를 눌러 <표 9-14>와 같이 필수 영역 3개 컨트롤을 추가하고 속성을 변경한다.

| 컨트롤 종류          | ConditionItem | 컨트롤 설정              |    |       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------|---------------|---------------------|----|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |               | Label               | ID | state | Help                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Multicodepicker | 회계단위          | pc_cd_search        | 필수 |       | helpCode 스크립트 버튼 </> 누른 후<br>Code Help 화면에서 회계단위 검색<br>Help Code : H_MA_PC_MST_S 선택<br>Code Field : PC_CD 선택<br>Text Field : PC_NM 선택 후 Apply<br>helpParmas:  (파라미터 매핑) 클릭 후<br>company_cd 파라미터명은 컨트롤/데이터<br>형은 number, 데이터값은<br>self.user.companyCode 입력<br>helpSize : 대                                                                                                                     |
| Multicodepicker | 구분            | dptse_fg_cd_searc h | 기본 |       | helpCode 스크립트 버튼 </> 누른 후<br>Code Help 화면에서 코드도움창 검색<br>Help Code : H_MA_CODEDTL_S 선택<br>Code Field : SYSDEF_CD 선택<br>Text Field : SYSDEF_NM 선택 후 Apply<br>helpParmas:  (파라미터 매핑) 클릭 후<br>company_cd 파라미터명은 컨트롤/데이터<br>형은 number, 데이터값은<br>self.user.companyCode 입력<br>module_cd 파라미터명은 컨트롤/데이터형<br>은 string, 데이터값은 FI<br>field_cd 파라미터명은 컨트롤/데이터형은<br>string, 데이터값은 P30090<br>helpSize : 중 |
| Multicodepicker | 금융기관          | partner_cd_s earch  | 기본 |       | helpCode 스크립트 버튼 </> 누른 후<br>Code Help 화면에서 금융기관 검색<br>Help Code : H_CI_BANK_MST_S 선택<br>Code Field : BANK_CD 선택<br>Text Field : FNLT_NM 선택 후 Apply<br>helpSize : 대                                                                                                                                                                                                                                                                                                              |

<표 9-14. 필수 영역 컨트롤 추가 설정>

옵션 영역에 위치한 계정코드과 상품유형 컨트롤을 추가하기 위해서 ① ConditionPanel 선택 후 ②  (더보기) 버튼을 눌러서 나오는 화면에서 ③ OptionArea 을 체크하거나 ConditionPanel 선택 후 작업의 ④ optional 속성을 체크한다.



<그림 9-111. 옵션 영역 선택>

다음으로 필수 영역 오른쪽  아래 방향을 눌러 옵션 영역을 표시한다



<그림 9-112. 옵션 영역 표시>

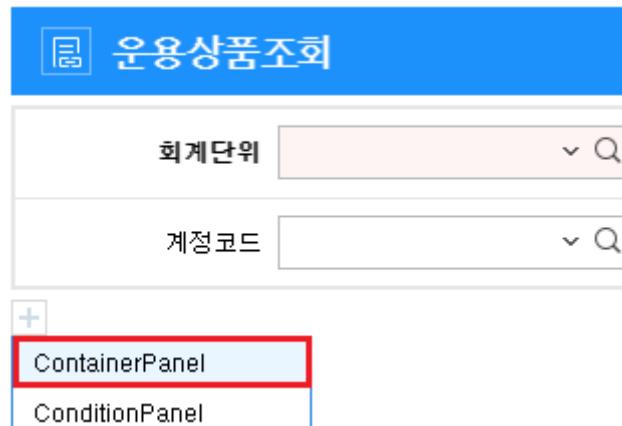
ConditionPanel 내의 를 눌러 <표 9-15>와 같이 옵션 영역에 2 개 컨트롤을 추가하고 속성을 변경한다.

| 컨트롤 종류          | ConditionItem | 컨트롤 설정             |    |       |                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|---------------|--------------------|----|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |               | Label              | ID | state | Help                                                                                                                                                                                                                                                                                                                                                                          |
| Multicodepicker | 계정코드          | acct_cd_search     |    | 기본    | helpCode 스크립트 버튼  누른 후<br>Code Help 화면에서 반제계정으로 검색<br>Help Code : H_MA_COA_MST_S02 선택<br>Code Field : ACCT_CD 선택<br>Text Field : ACCT_NM 선택 후 Apply<br>helpParmas:  (파라미터 매핑) 클릭<br>후 company_cd 파라미터명은 컨트롤/데이터형은 number, 데이터값은<br>self.user.companyCode 입력<br>helpSize : 대                                                                                                   |
| Multicodepicker | 상품유형          | item_tp_cd_searc_h |    | 기본    | helpCode 스크립트 버튼  누른 후<br>Code Help 화면에서 코드도움창 검색<br>Help Code : H_MA_CODEDTL_S 선택<br>Code Field : SYSDEF_CD 선택<br>Text Field : SYSDEF_NM 선택 후 Apply<br>helpParmas:  (파라미터 매핑) 클릭<br>후 company_cd 파라미터명은 컨트롤/데이터형은 number, 데이터값은<br>self.user.companyCode 입력<br>module_cd 파라미터명은 컨트롤/데이터형은 string, 데이터값은 FI<br>field_cd 파라미터명은 컨트롤/데이터형은 string, 데이터값은 P30091<br>helpSize : 중 |

<표 9-15. 옵션 영역 컨트롤 추가 설정>

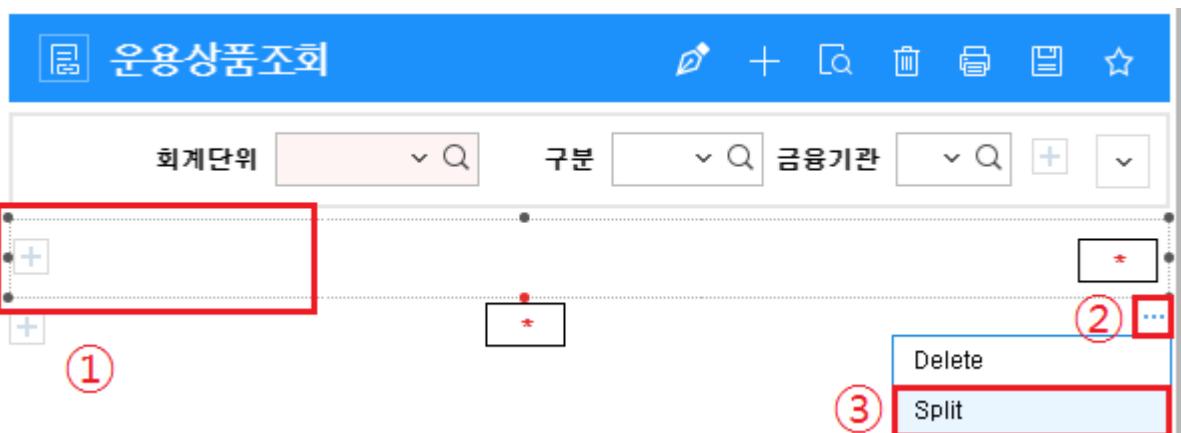
다음으로 조회 조건 아래 부분을 디자인한다. 전체 구조는 개발 화면에서 <그림 9-105>와 같이 되어 있다.

Canvas 을 눌러 ① ContainerPanel 을 추가한다.



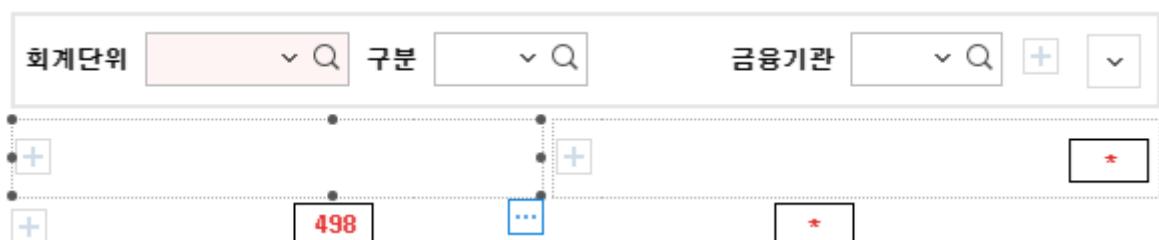
<그림 9-113. ContainerPanel 추가>

생성된 ContainerPanel 을 ① 선택하고 오른쪽 빠른 속성 ② 버튼을 눌러 ③ Split 를 눌러 화면을 2 개로 분할한다.



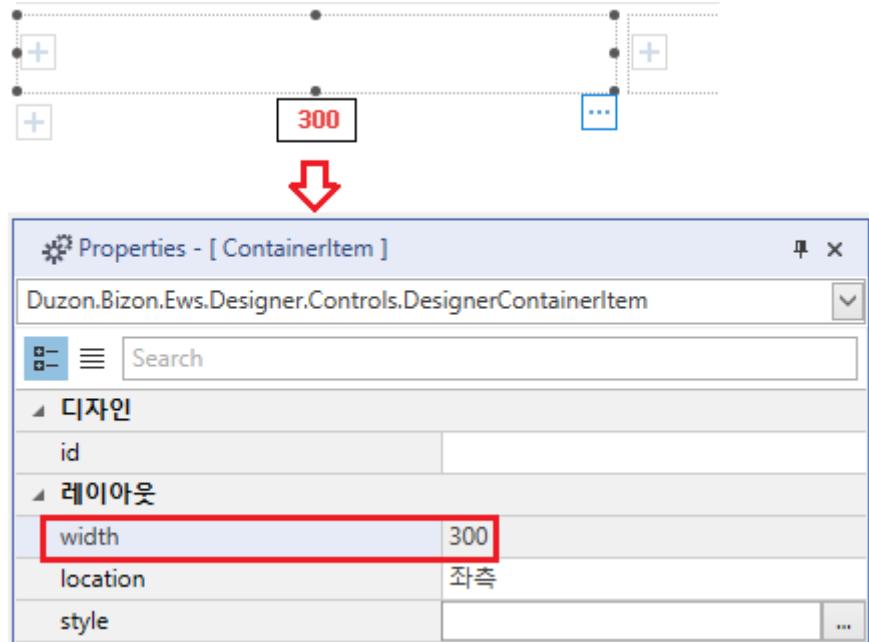
<그림 9-114. Split 로 화면 나누기>

2 개로 분할된 화면이다.



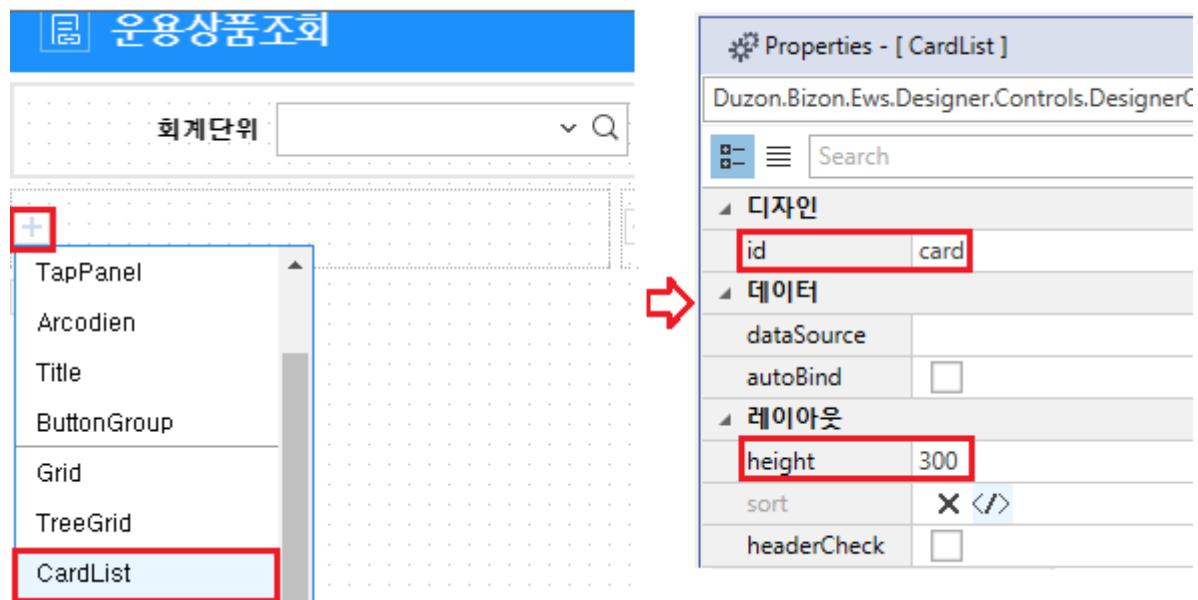
<그림 9-115. 분할된 화면>

좌측 ContainerItem 을 선택하고 width 를 300 을 입력한다.



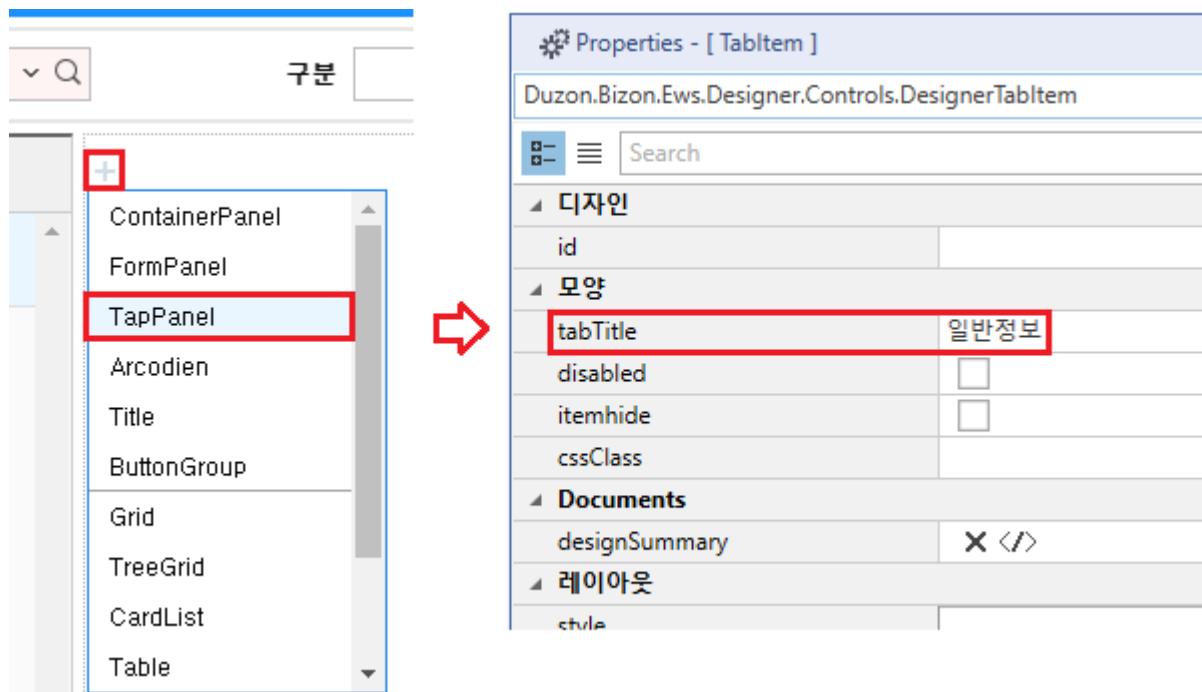
<그림 9-116. 좌측 ContainerItem width 변경>

왼쪽 를 눌러 CardList 를 추가하고 id 를 card, height 를 300 으로 입력한다.



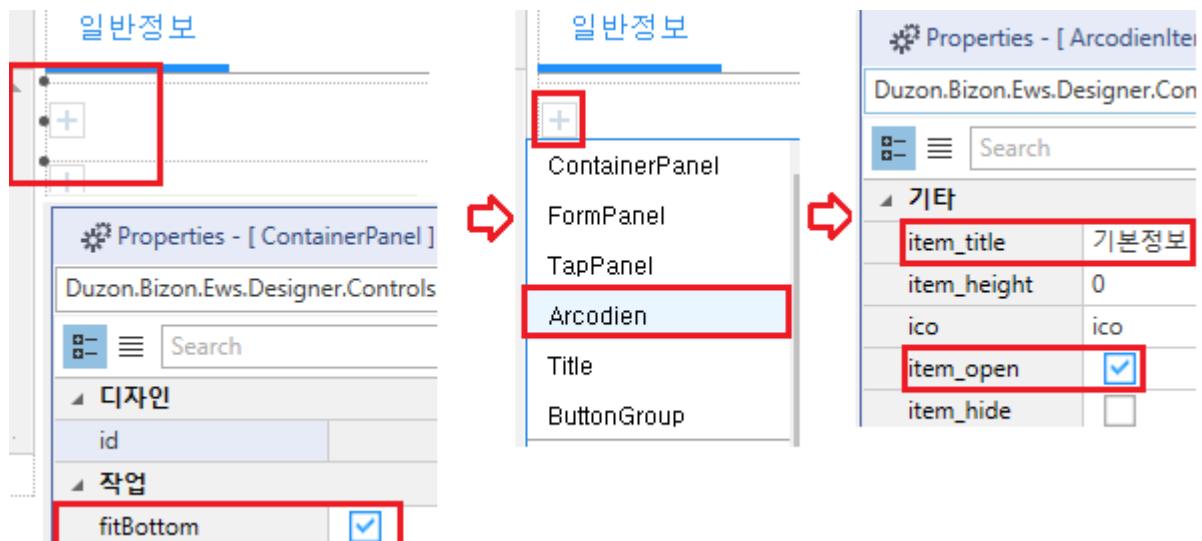
<그림 9-117. CardList 추가 및 속성 변경>

오른쪽 ContainerItem에서 을 선택하고 TapPanel을 추가하고 tabTitle을 일반정보로 변경한다.



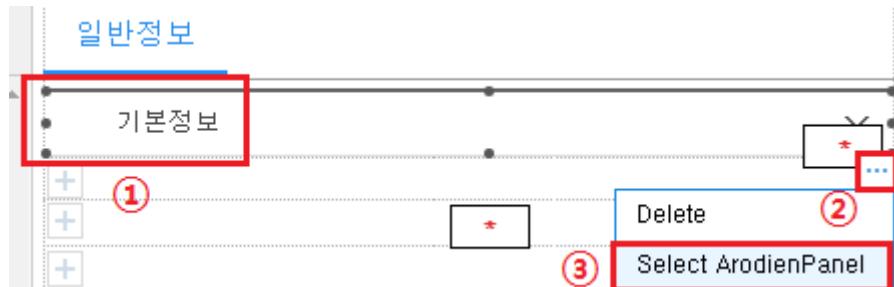
<그림 9-118. TapPanel 추가 및 tabTitle 변경>

탭패널에서 을 선택하고 ContainerPanel을 추가하고 선택하고 fitBottom을 체크 한다. 다음 ContainerPanel에서 을 눌러 Arcodien을 추가한다. ArcodienItem을 선택하고 item\_title을 기본정보로 변경하고 초기 열림 상태를 설정하기 위해 item\_open 속성을 체크한다.



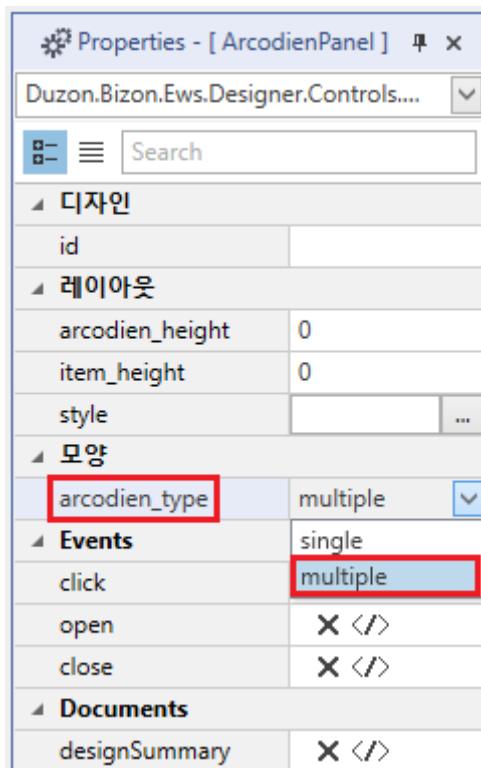
<그림 9-119. CotainerPanel 추가, Arcodien 추가 및 item\_title 변경>

기본정보, 이자정보 동시에 모두 열림 상태로 하기 위해 기본정보 ArcodienPanel 의 arcodien\_type 속성을 변경해야 한다.. ① 기본정보를 선택하고 오른쪽 ② ... 을 클릭하고 ③ Select ArcodienPanel 을 선택한다.



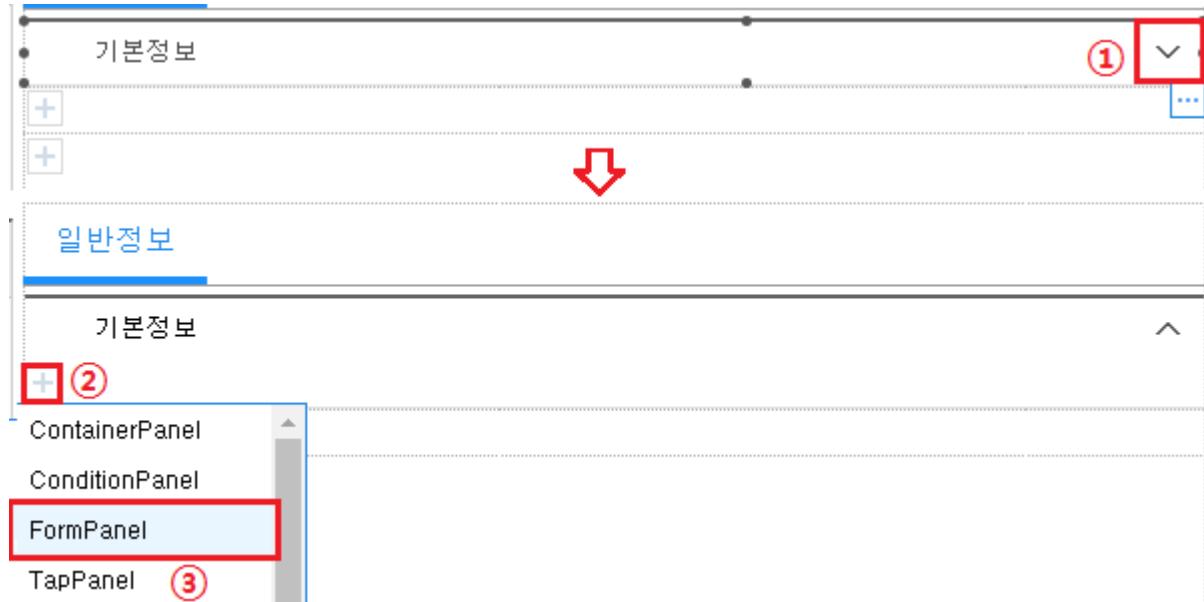
<그림 9-120. Select ArcodienPanel 선택>

ArcodienPanel 프로퍼티에서 모양의 arcodien\_type 을 multiple 를 선택한다. single 을 선택할 경우 기본정보나 이자정보 중 하나만 열린다.



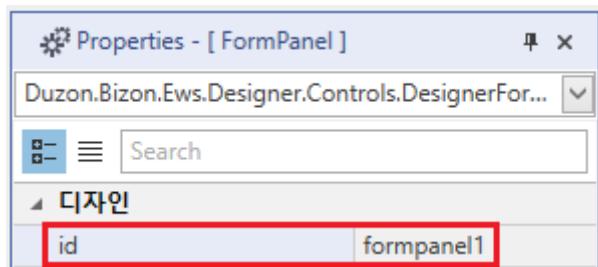
<그림 9-121. arcodien type 변경>

Arcodien에서 오른쪽 ▼를 눌러 확장하고 선택하고 FormPanel을 추가한다.



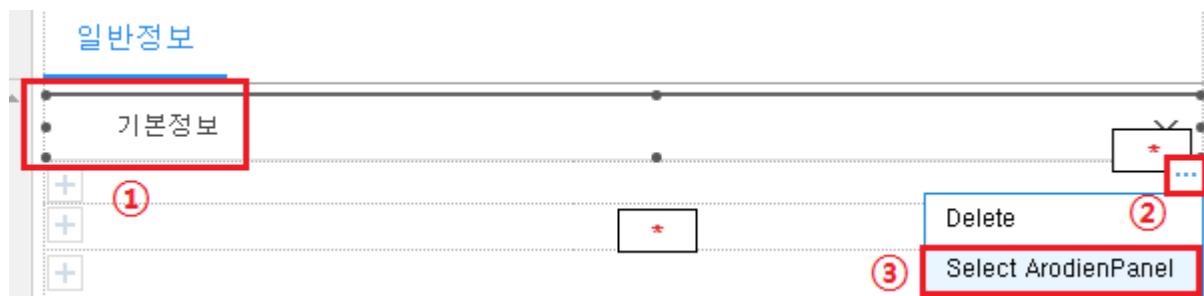
<그림 9-122. FormPanel 추가>

나중에 CardList와 바인딩하기 위하여 FormPanel의 id를 formpanel1로 입력한다.



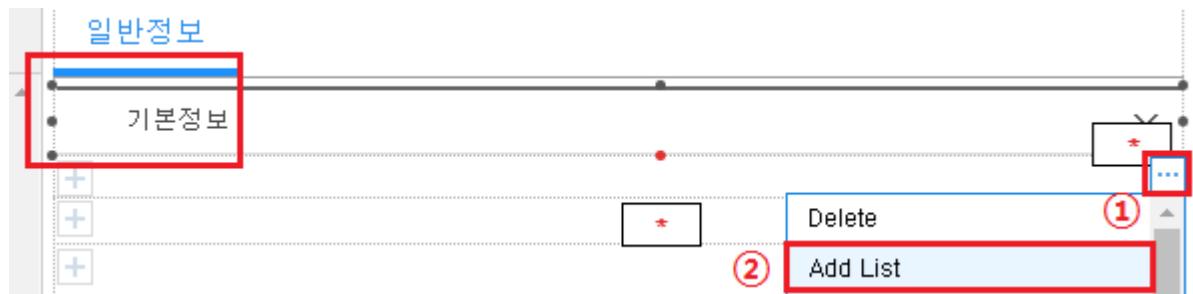
<그림 9-123. FormPanel id 변경>

이자정보를 나타내는 Arcodien을 추가하기 위하여 ① 기본정보를 선택하고 오른쪽 ② ...을 클릭하고 ③ Select ArcodienPanel을 선택한다.



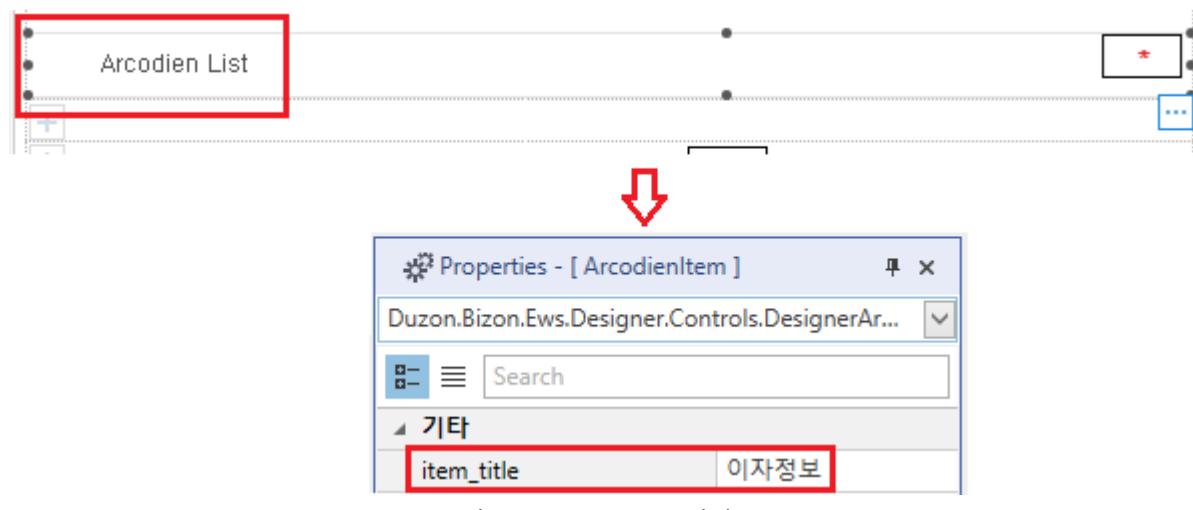
<그림 9-124. ArcodienPanel 선택>

ArcodienPanel 이 선택되면 오른쪽 ② 을 눌러서 Add List 를 누르면 ArcodienItem 이 추가된다.



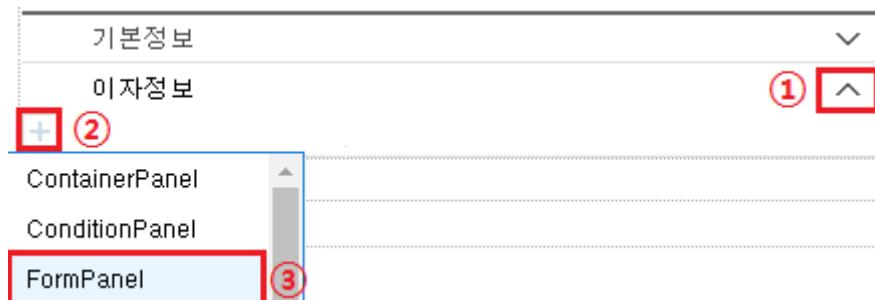
<그림 9-125. Arcodien List 추가>

추가된 ArcodienItem 을 선택하고 item\_title 을 이자정보로 변경한다.



<그림 9-126. item\_title 변경>

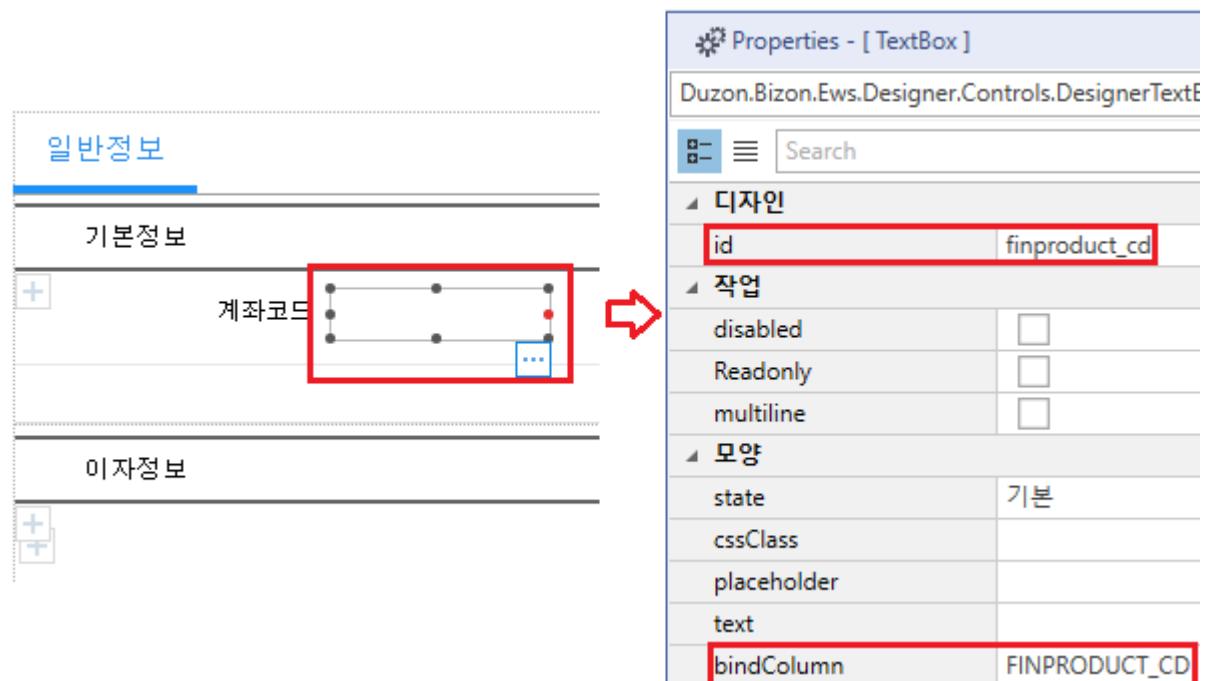
Arcodien 에서 오른쪽 를 눌러 확장하고 선택하고 FormPanel 을 추가하고, 마찬가지로 CardList 와 바인딩 하기 위하여 FormPanel 의 id 를 formpanel2 로 입력한다.



<그림 9-127. FormPanel 추가>

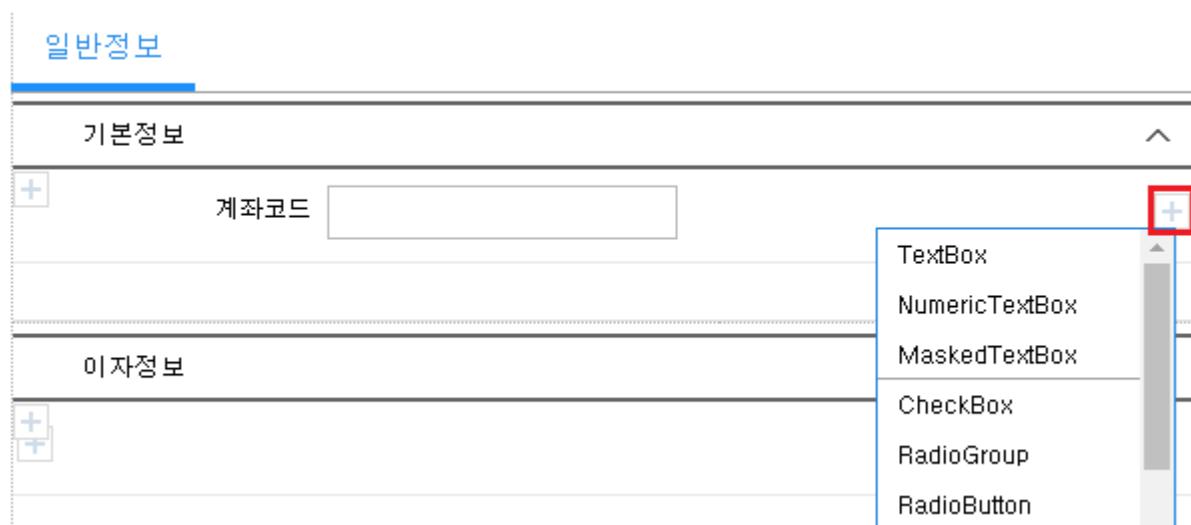
다음으로 기본정보 Arcodien 의 FormPanel 에 컨트롤들을 추가한다.

FormPanel 의 선택하고 TextBox 를 선택한다. Label 을 선택하면 FormItem 이 선택된다. 라벨을 계좌코드로 입력한다. TextBox 를 선택하고 id 를 finproduct\_cd を 입력하고 CardList 와 바인딩 될 컬럼을 설정하는 속성인 bindColumn 에 FINPRODUCT\_CD 를 입력한다.



<그림 9-128. TextBox 컨트롤 추가>

다음 컨트롤 추가는 오른쪽 을 선택하여 추가 한다



<그림 9-129. 다음 컨트롤 추가 방법>

이어서 모든 컨트롤을 추가하지 않고 <그림 9-130>과 같이 기본정보와 이자정보 Arcodien 에 다른 유형의 컨트롤들 몇 개만 추가한다.

<그림 9-130. 추가할 컨트롤들 화면>

먼저 기본정보 Arcodien 에 추가할 컨트롤들은 <표 9-16>을 참고하여 추가한다.

| 추가 컨트롤       | ConditionItem | 각 컨트롤 속성 설정 |                                                                                                                                                                           |
|--------------|---------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |               | 구분          | 설정 또는 입력값                                                                                                                                                                 |
| Dropdownlist | 라벨 : 상품유형     | 디자인         | id : item_tp_cd                                                                                                                                                           |
|              |               | 데이터         | dataSource: DataSource 생성 후 재설정<br>valueField: DataSource 생성 후 재설정<br>textField: DataSource 생성 후 재설정<br>bindValue : ITEM_TP_CD                                            |
| CodePicker   | 라벨 : 금융기관     | 디자인         | id : bank_cd                                                                                                                                                              |
|              |               | Help        | helpCode 스크립트 버튼 </> 누른 후<br>Code Help 화면에서 금융기관 검색<br>Help Code : H_CI_BANK_MST_S 선택<br>Code Field : BANK_CD 선택<br>Text Field : FNLT_NM 선택 후 Apply<br>bindCode : BANK_CD |

|            |                             |     |                                    |
|------------|-----------------------------|-----|------------------------------------|
|            |                             |     | bindText : FNLT_NM<br>helpSize : 대 |
| Datepicker | 라벨 : 가입일                    | 디자인 | id : join_dt                       |
|            |                             | 모양  | bindColumn : JOIN_DT               |
| TextBox    | 라벨 : 해약사유<br>columnSize : 2 | 디자인 | id : srnd_reason_dc                |
|            |                             | 모양  | bindColumn : SRND_REASON_DC        |

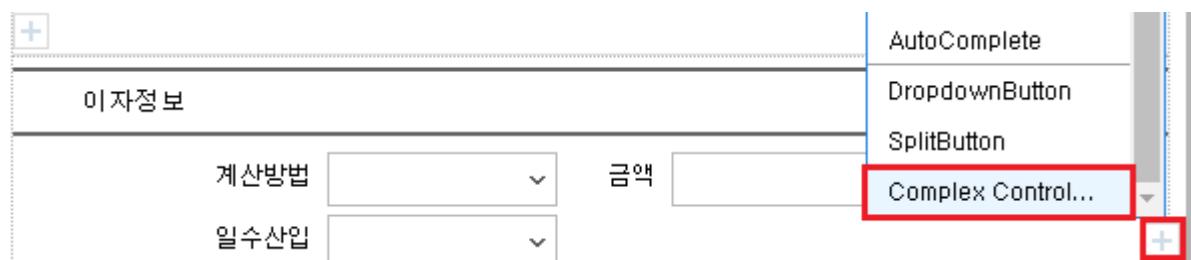
<표 9-16. 기본정보 Arcodien 추가 컨트롤들>

다음으로 <표 9-17>는 이자정보 Arcodien에 추가할 컨트롤들이다.

| 추가 컨트롤         | ConditionItem | 각 컨트롤 속성 설정 |                                                                                                                                      |
|----------------|---------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------|
|                |               | 구분          | 설정 또는 입력값                                                                                                                            |
| Dropdownlist   | 라벨 : 계산방법     | 디자인         | id : intr_cal_mthd_cd                                                                                                                |
|                |               | 데이터         | dataSource: DataSource 생성 후 재설정<br>valueField: DataSource 생성 후 재설정<br>textField: DataSource 생성 후 재설정<br>bindValue : INTR_CAL_MTHD_CD |
| NumericTextBox | 라벨 : 금액       | 디자인         | id : intr_amt                                                                                                                        |
|                |               | 모양          | format : #,##0.##<br>bindColumn : INTR_AMT                                                                                           |
| Dropdownlist   | 라벨 : 일수산입     | 디자인         | id : dy_cal_mthd_cd                                                                                                                  |
|                |               | 모양          | dataSource: DataSource 생성 후 재설정<br>valueField: DataSource 생성 후 재설정<br>textField: DataSource 생성 후 재설정<br>bindValue : DY_CAL_MTHD_CD   |

<표 9-17. 이자정보 Arcodien 추가 컨트롤들>

휴일처리 컨트롤의 경우 다른 컨트롤과 다르게 Complex Control..을 선택한다.



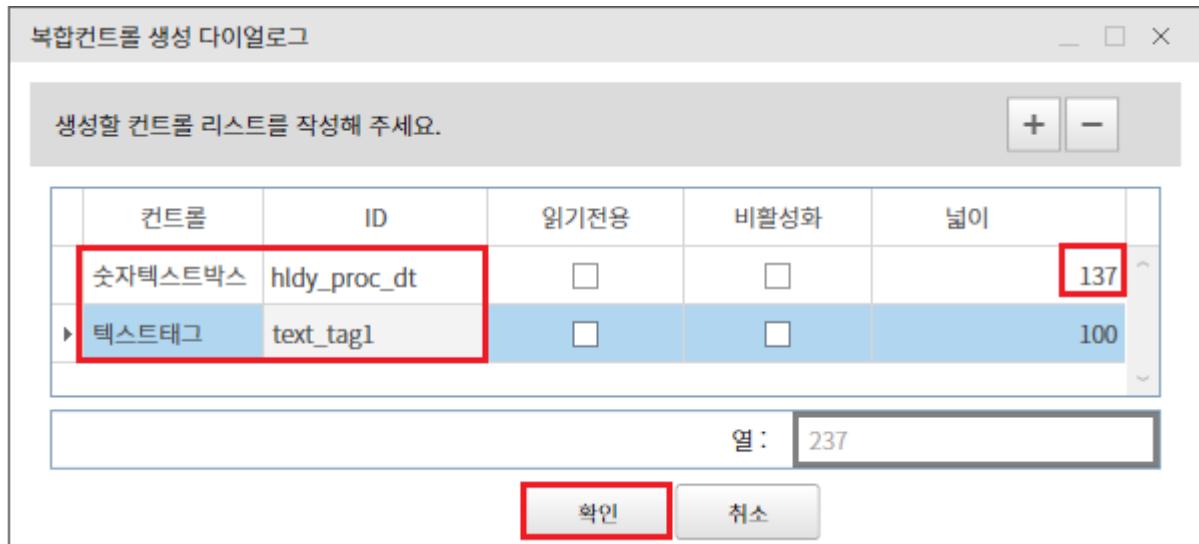
<그림 9-131. Complex Control 선택>

복합컨트롤 생성 다이얼로그 화면이 실행되며, + 표시를 누른다.



<그림 9-132. 복합컨트롤 컨트롤 추가>

컨트롤을 텍스트박스를 선택하고 ID를 hldy\_proc\_dt를 입력하고 넓이를 137로 입력하고 +을 눌러 컨트롤을 맨 아래 있는 텍스트태그를 선택하고 ID를 text\_tag1을 입력하고 확인을 누른다.



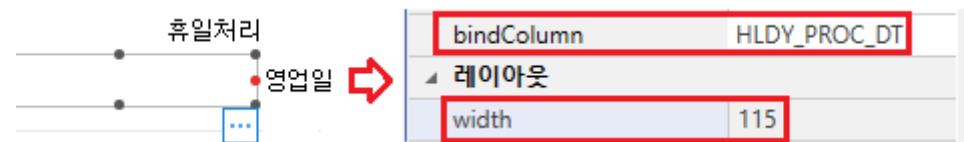
<그림 9-133. 숫자텍스트박스와 텍스트태그 추가>

라벨을 선택하고 프로퍼티의 라벨을 휴일처리로 입력하고, text를 선택한다. 프로퍼티창에서 text를 영업일로 수정한다.



<그림 9-134. 휴일처리 복합컨트롤 수정>

휴일처리가 아래로 내려가므로 TextBox 을 선택하고 크기를 115 로 줄이고 bindColumn 속성을 HLDY\_PROC\_DT 로 입력한다.



<그림 9-135. 휴일처리 TextBox 크기 변경 및 bindColumn 설정>

툴바에서 Build & Preview(Ctrl+B) 버튼을 눌러 미리보기를 실행하면 기본정보 Arcodien 아이템이 열려 있으며 이자정보 ▼ 펼침버튼을 눌렀을 때 기본정보도 같이 보여지는 멀티타입이다.

**<Single>**

**<Multiple>**

<그림 9-136. 미리보기 실행 후 이자정보 Arcodien 확인>

### 3) 데이터소스 추가 및 바인딩

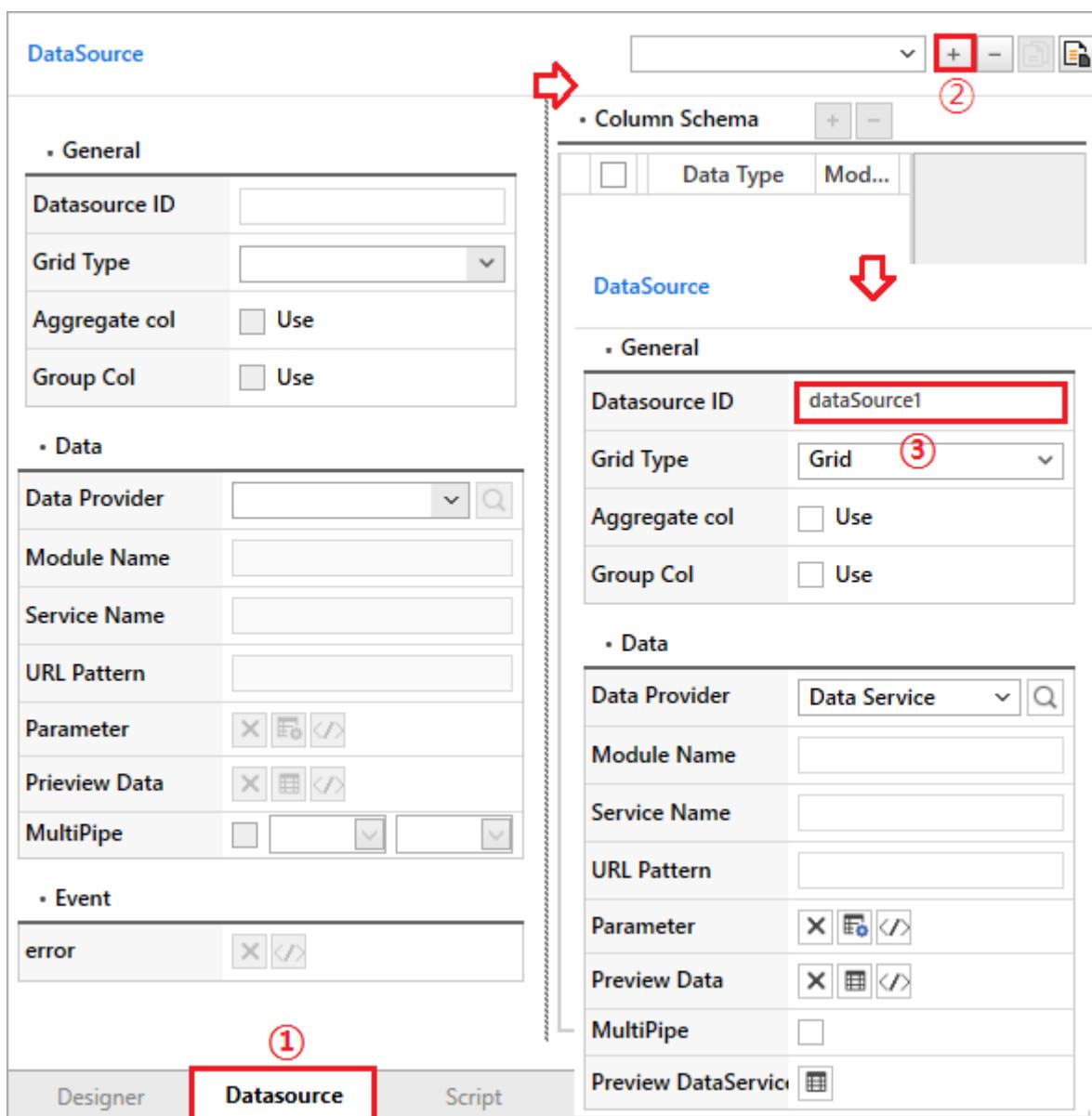
데이터소스는 CardList와 DropDownList 컨트롤인 상품유형, 계산방법, 일수산입 등 4개가 필요하므로 각각 생성한다.

먼저 CardList에 바인딩 할 데이터소스를 생성한다.

① Datasource 탭을 클릭한다.

② 우측 상단에서 을 눌러 Datasource를 추가한다

③ datasource1이 자동으로 추가되어 있다. ID 변경시 반드시 엔터키를 눌러야 반영된다.



<그림 9-137. 데이터소스 추가>

Datasource ID 를 cardlist\_datasource 으로 변경하고 엔터키를 누르고 Grid 에 사용하지 않으므로 Grid Type 은 None 을 선택한다.

이제 Data Provider(데이터 제공자)에서 데이터 서비스를 검색한다. Data Service가 기본값으로 선택되어 있으므로 검색 버튼을 클릭한다.

DataSource

• General

|               |                              |
|---------------|------------------------------|
| Datasource ID | cardlist_datasource          |
| Grid Type     | None                         |
| Aggregate col | <input type="checkbox"/> Use |
| Group Col     | <input type="checkbox"/> Use |

• Data

|               |                                  |                                   |
|---------------|----------------------------------|-----------------------------------|
| Data Provider | Data Service                     | <input type="button" value="🔍"/>  |
| Module Name   |                                  |                                   |
| Service Name  |                                  |                                   |
| URL Pattern   |                                  |                                   |
| Parameter     | <input type="button" value="X"/> | <input type="button" value="F0"/> |
| Prievue Data  | <input type="button" value="X"/> | <input type="button" value="F1"/> |
| MultiPipe     | <input type="checkbox"/>         |                                   |

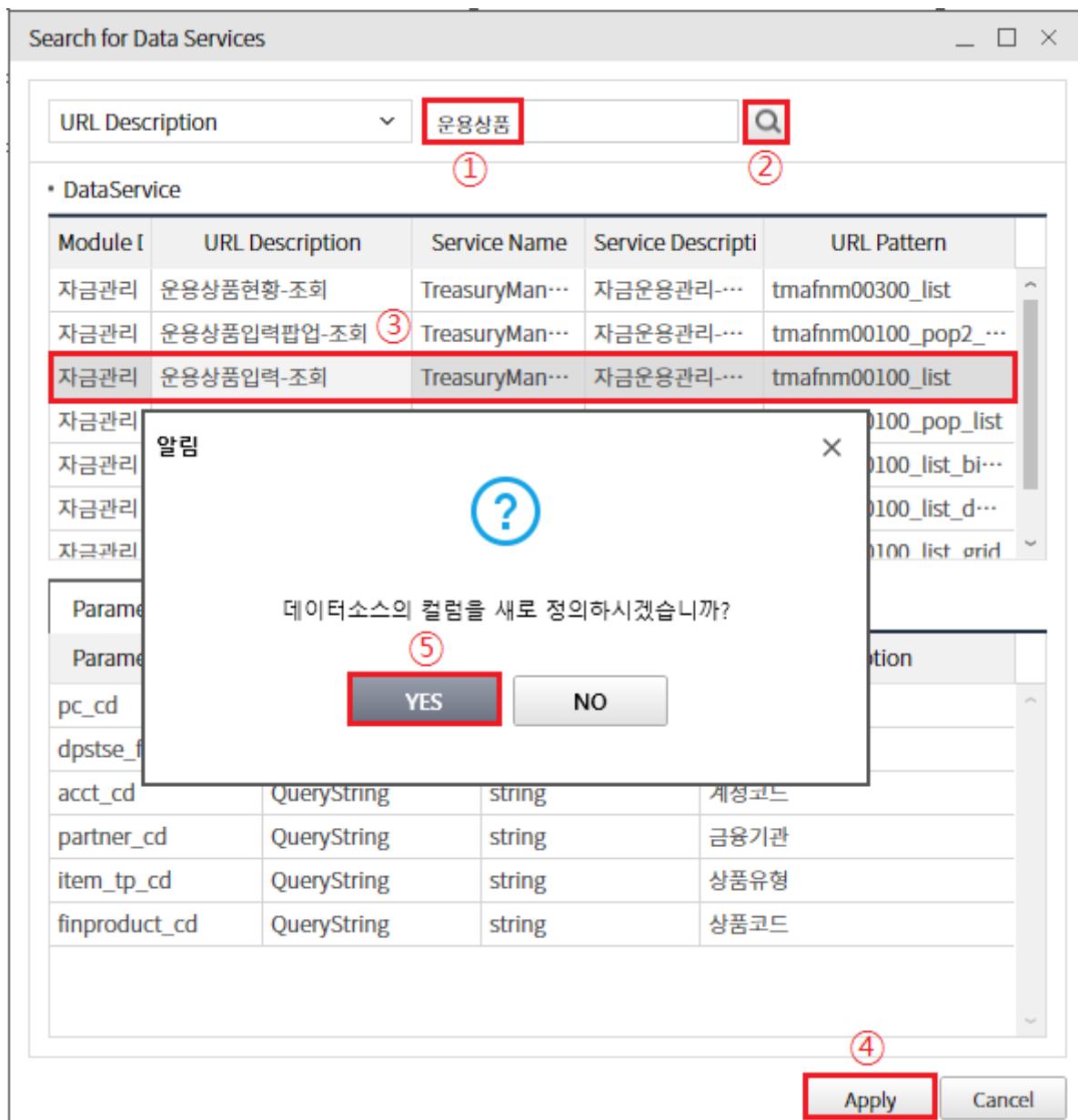
• Event

|       |                                  |                                   |
|-------|----------------------------------|-----------------------------------|
| error | <input type="button" value="X"/> | <input type="button" value="F1"/> |
|-------|----------------------------------|-----------------------------------|

<그림 9-138. Data Provider 화면 열기>

Search for Data Services 화면이 열리며, 검색 기준이 URL Description 으로 선택되어 있다.

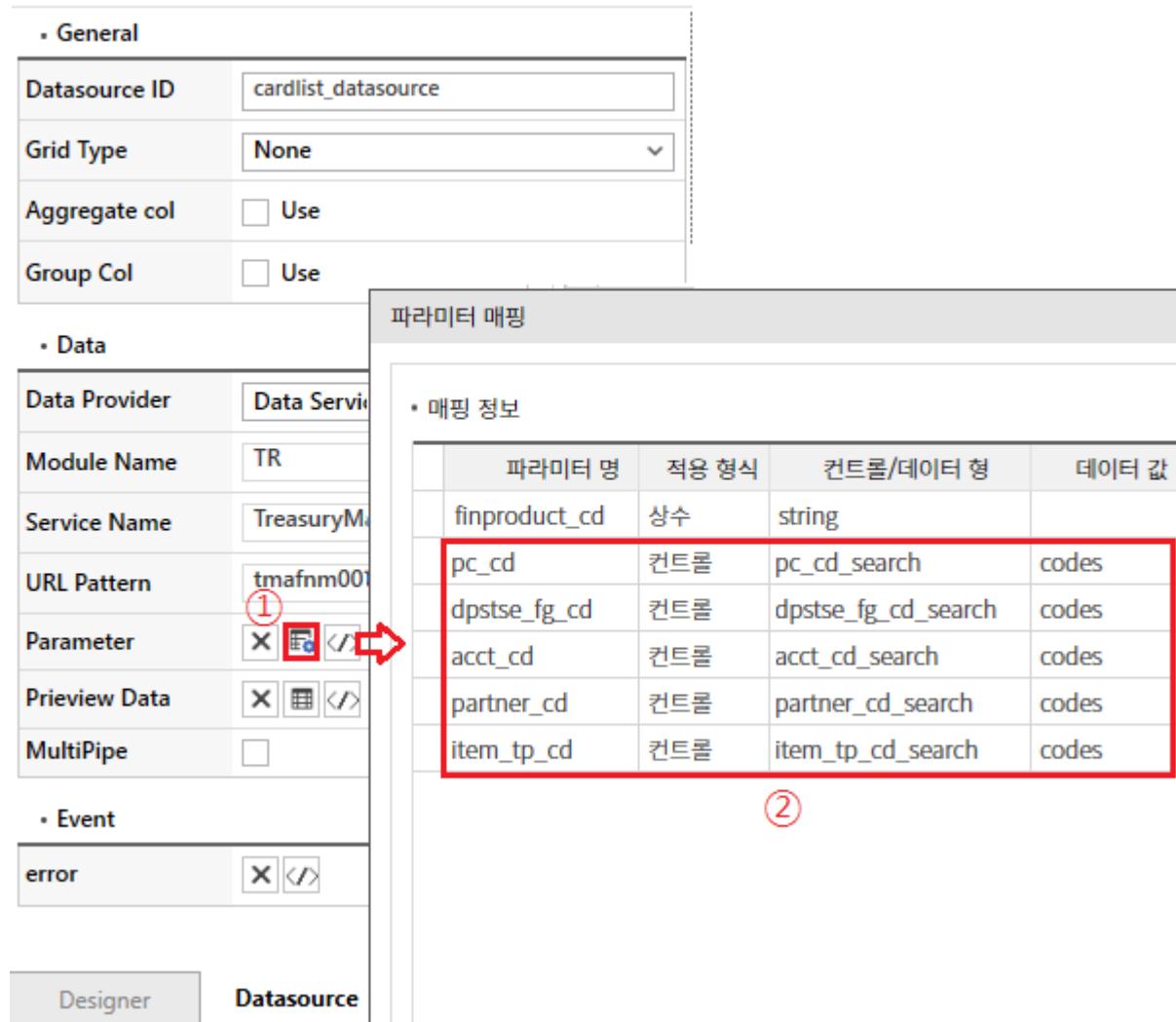
- ① 운용상품을 입력하고 엔터키를 누르거나 ② Search(검색)한다. ③ URL Description 컬럼에 운용상품입력 – 조회를 선택하고 ④ Apply 를 클릭한다. ⑤ 알림창이 뜨면 예를 선택한다.



<그림 9-139. URL Description 검색 조건 입력>

다음 Parameter 에서 ① Mapping... 버튼을 눌러 열리는 파라미터 매핑  화면에서 적용 형식, 컨트롤/데이터 형을 <그림 9-140> ②처럼 선택한다.

#### DataSource



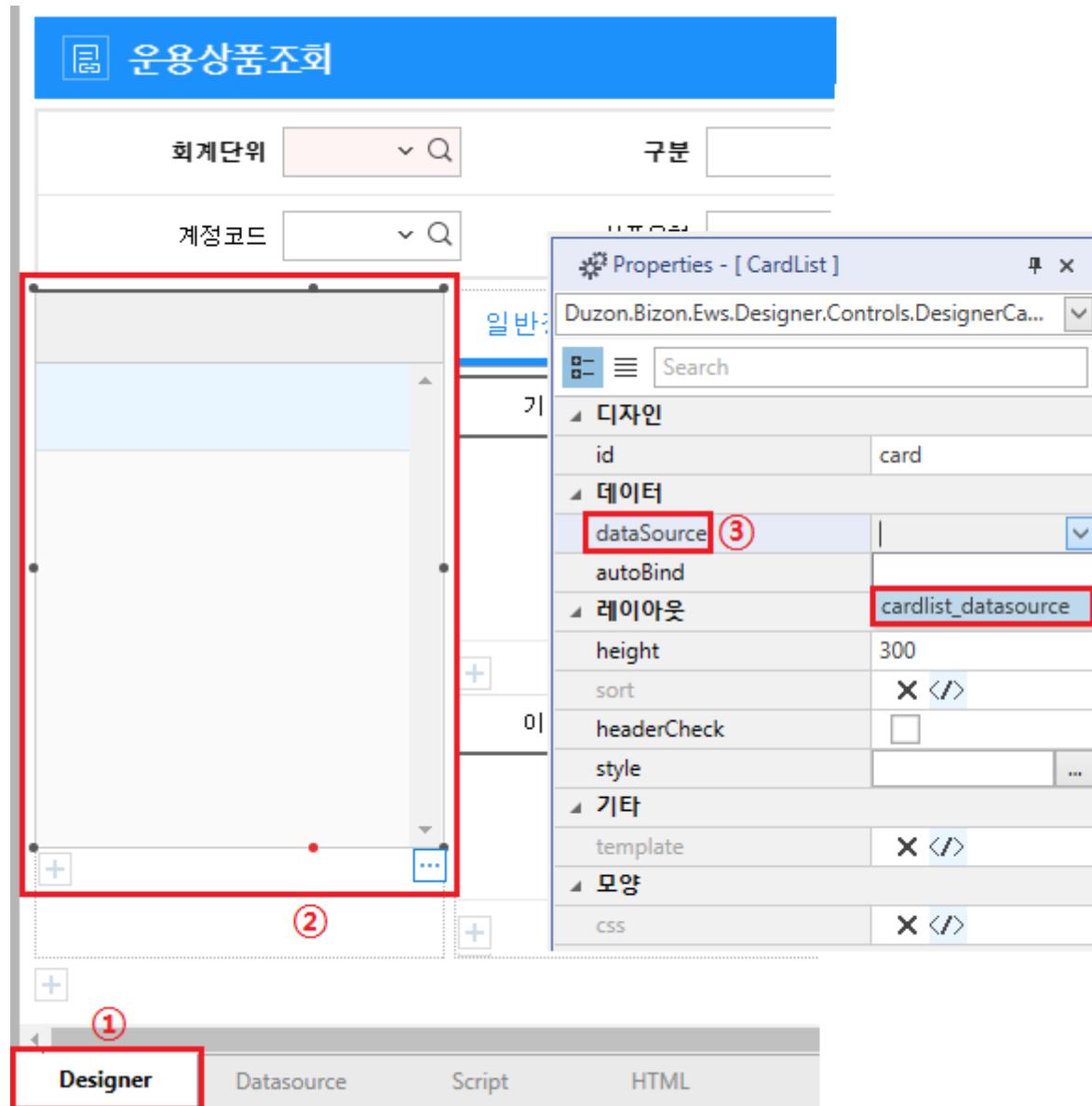
The screenshot shows the Oracle ADF Designer interface. On the left, there's a main configuration panel for a 'Datasource' with tabs for 'General', 'Data', and 'Event'. In the 'Data' tab, under 'Parameter', there's a 'Mapping...' button (circled in red with ①) which opens a separate window titled '파라미터 매핑' (Parameter Mapping). This window has a 'Mapping 정보' (Mapping Info) section with a table:

| 파라미터 명        | 적용 형식 | 컨트롤/데이터 형           | 데이터 값 |
|---------------|-------|---------------------|-------|
| finproduct_cd | 상수    | string              |       |
| pc_cd         | 컨트롤   | pc_cd_search        | codes |
| dpstse_fg_cd  | 컨트롤   | dpstse_fg_cd_search | codes |
| acct_cd       | 컨트롤   | acct_cd_search      | codes |
| partner_cd    | 컨트롤   | partner_cd_search   | codes |
| item_tp_cd    | 컨트롤   | item_tp_cd_search   | codes |

A red box highlights the last five rows of the table. A red arrow points from the 'Mapping...' button in the main panel to the table in the mapping window. A red circle with ② is placed at the bottom right of the mapping window.

<그림 9-140. 운용상품조회 CardList 데이터소스 파라미터 매핑>

다시 ① Designer 탭으로 가서 ② CardList를 선택한 후에, ③ DataSource 속성에서 cardlist\_datasource를 선택한다.



<그림 9-141. 금용 상품 Header Grid 의 dataSource 속성 설정>

CardList 의 경우 dataSource 를 지정했을 때 화면에 변화가 없고, 속성 중 template(템플릿을 설정)을 설정해야 한다.

CardList 을 선택하고 프로퍼티에서 template 의 스크립트 버튼 </>을 눌러 템플릿을 설정한다.

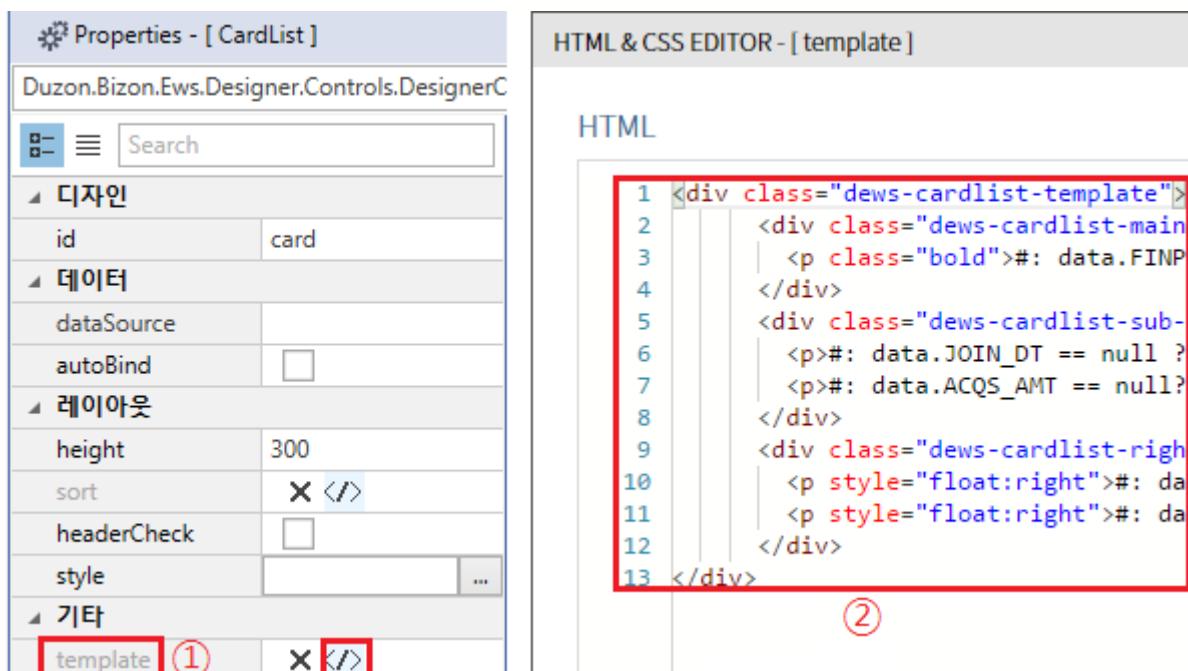
```
<div class="dews-cardlist-template">
<div class="dews-cardlist-main-text">
<p class="bold">#: data.FINPRODUCT_NM == null ? "" : data.FINPRODUCT_NM #</p>
</div>
<div class="dews-cardlist-sub-text">
<p>#: data.JOIN_DT == null ? "" : dews.date.format(data.JOIN_DT) #</p>
```

```

<p>#: data.ACQS_AMT == null? "" : dews.number.format(data.ACQS_AMT) #</p>
</div>
<div class="dews-cardlist-right-text">
 <p style="float:right">#: data.FINPRODUCT_CD == null ? "" : data.FINPRODUCT_CD
#</p>

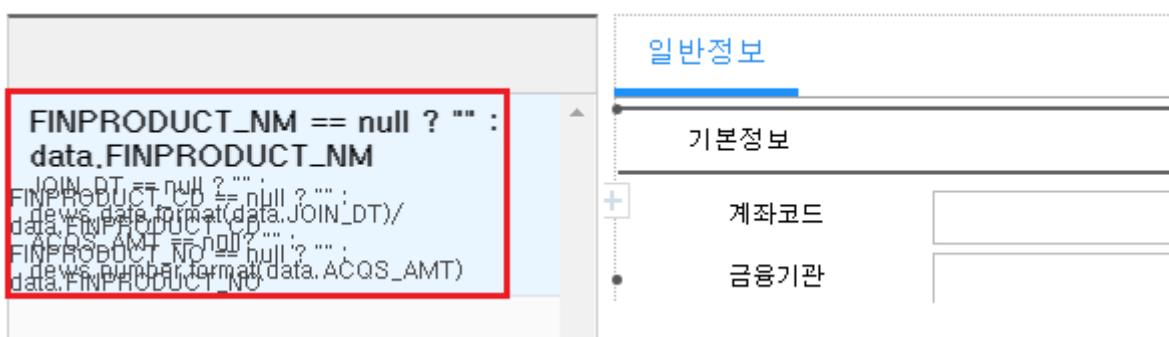
 <p style="float:right">#: data.FINPRODUCT_NO == null ? "" : data.FINPRODUCT_NO #</p>
</div>
</div>

```



<그림 9-142. CardList 컨트롤의 template 스크립트 작성>

template 속성에 스크립트를 작성하면 CardList 화면에 코드가 표시됩니다.



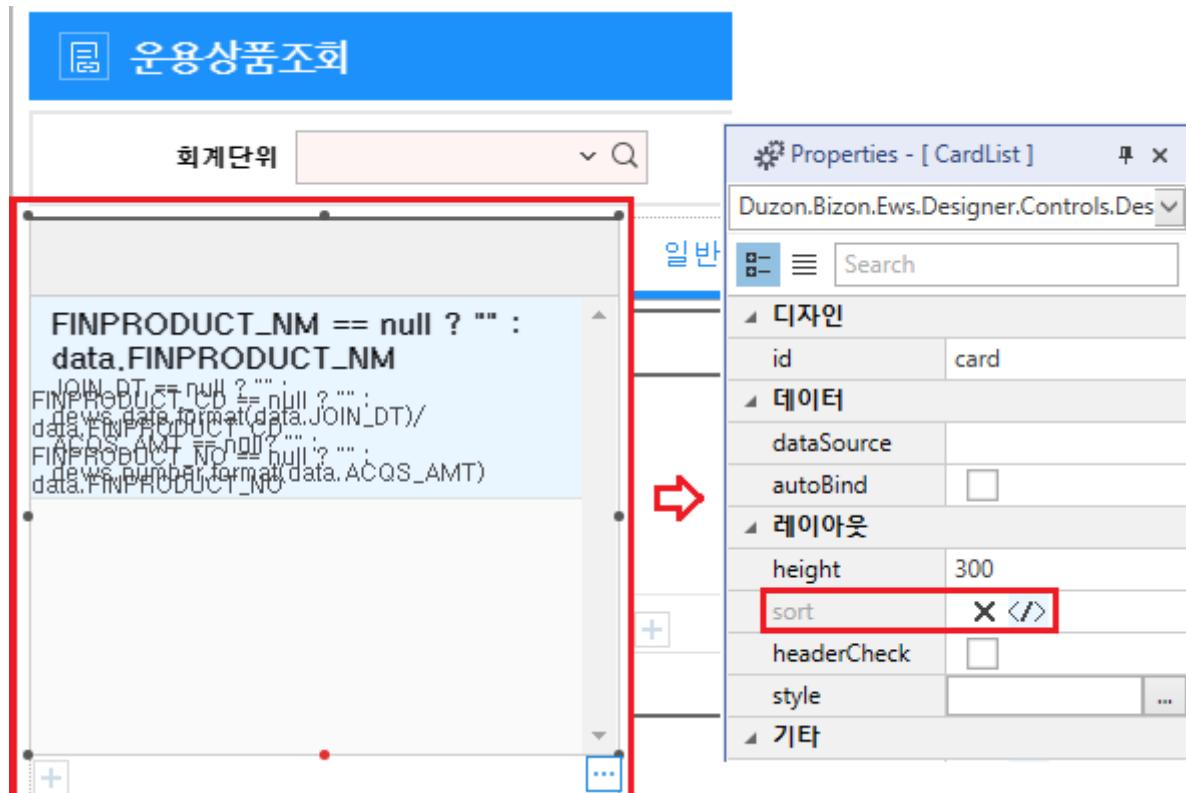
<그림 9-143. template 스크립트 작성 후 CardList 화면>

이제 CardList 의 정렬을 하기 위해 정렬 속성을 설정한다.



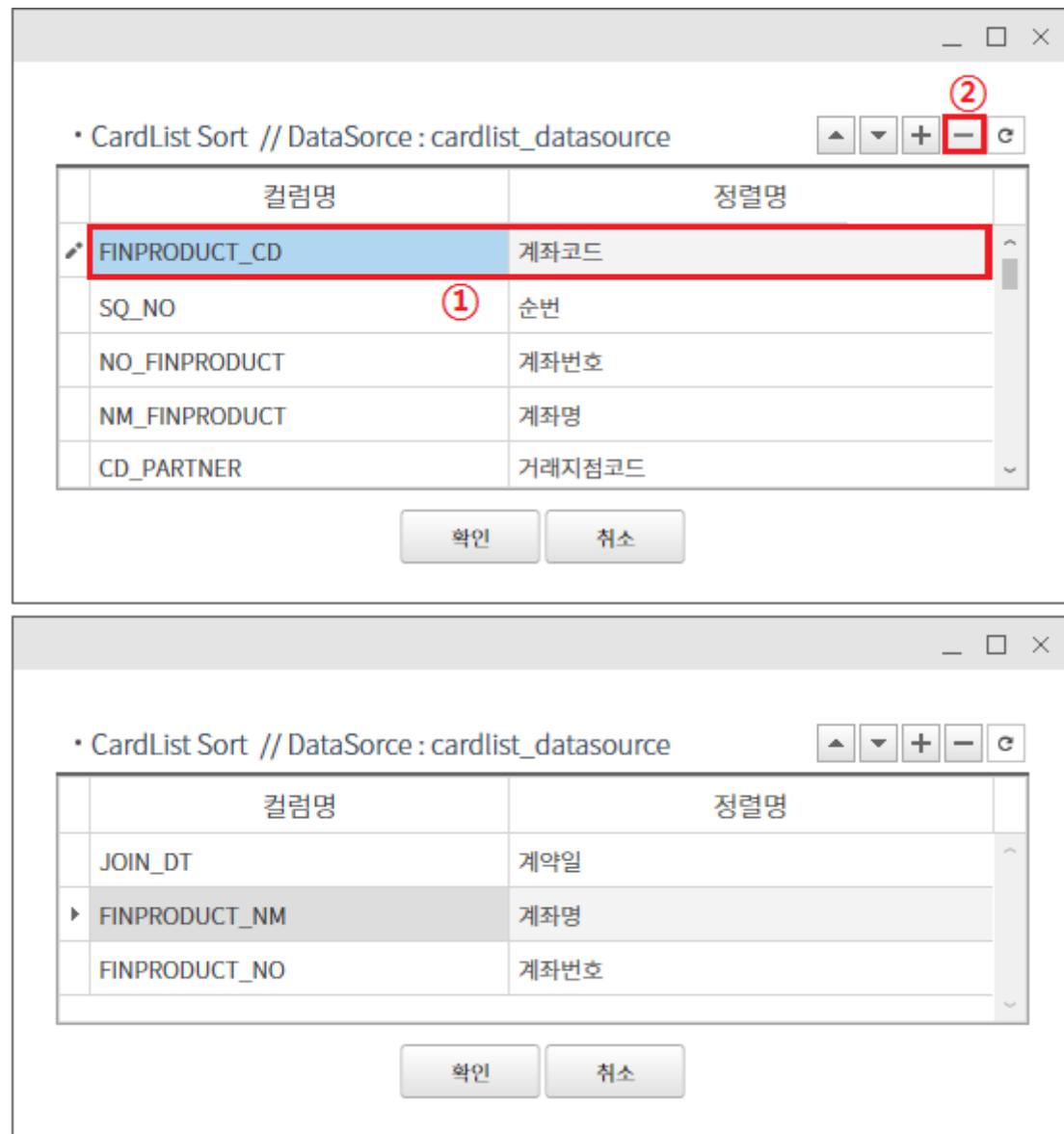
<그림 9-144. CardList 정렬>

<그림 9-144> 처럼 3 개의 정렬 순서로 설정하기 위하여 CardList 를 선택 후 프로퍼티 레이아웃에 sort 스크립트를 클릭한다.



<그림 9-145. CardList 정렬 스크립트 실행>

데이터소스(cardlist\_datasource)의 컬럼 정보가 보인다. ① 컬럼을 선택하고 ② -버튼을 누르거나 위아래 화살표를 눌러 JOIN\_DT, FINPRODUCT\_NM, FINPRODUCT\_NO 컬럼만 남긴다.



<그림 9-146. CardList 정렬 컬럼 설정>

위 아래 화살표 버튼을 눌러 정렬하고, 정렬명을 JOIN\_DT 는 '취득일기준 정렬', FINPRODUCT\_NM 은 '상품명으로 정렬', FINPRODUCT\_NO 는 '계좌번호로 정렬'로 각각 수정하고 확인을 누른다.

• CardList Sort // DataSorce : cardlist\_datasource

컬럼명	정렬명
JOIN_DT	취득일기준 정렬
FINPRODUCT_NM	상품명로 정렬
FINPRODUCT_NO	계좌번호로 정렬

<그림 9-147. 정렬명 설정>

CardList에 정렬이 표시된다.

The screenshot shows the SAP Fiori designer interface with the title "운용상품조회". Below the title, there are two dropdown menus: "회계단위" and "계정코드". A dropdown menu for sorting is open at the bottom, with the option "취득일기준 정렬 Sort" highlighted and surrounded by a red box. The code for the sorting logic is visible in the background:

```

FINPRODUCT_NM == null ? "" :
data.FINPRODUCT_NM
JOIN_DT == null ? "" :
new Date().format(data.JOIN_DT) /
ACQS_AMT == null ? "" :
new Number().format(data.ACQS_AMT)

```

<그림 9-148. 정렬 표시>

다음으로 DropDownList에서 사용될 데이터소스를 추가한다.

상품유형, 계산방법, 일수산입은 DEWS 메뉴 중 통합코드관리에서 금융상품유형과 이자계산방법, 일수계산방법 코드에서 가져온다.

No	관리코드	관리코드명
>	12	단기금융상품
2	13	장기금융상품
3	14	단기매매증권(채권)

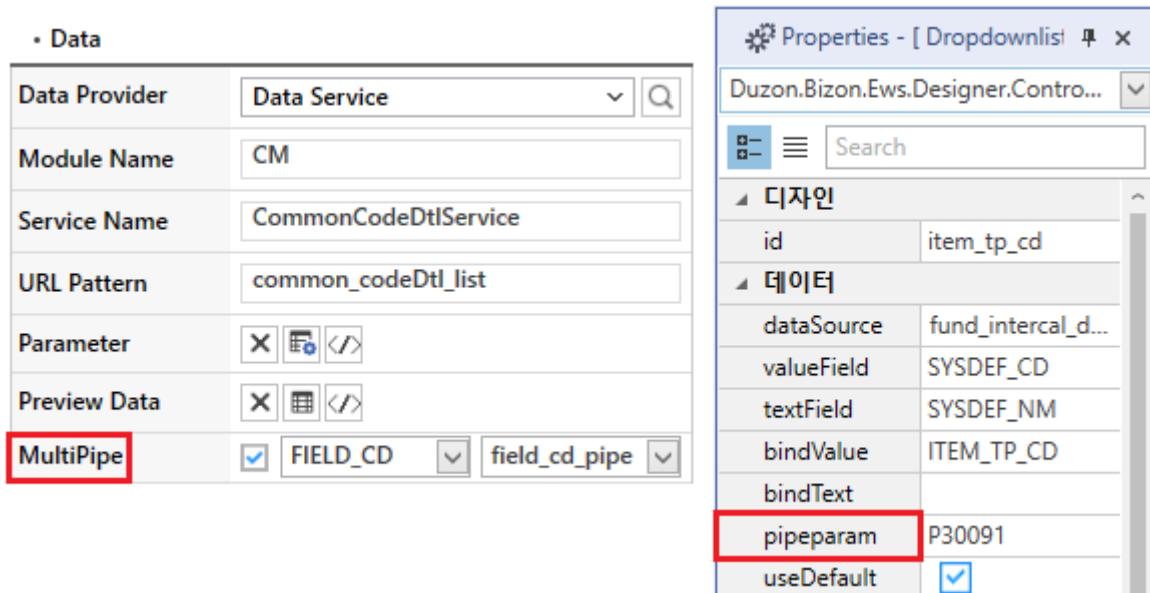
No	관리코드	관리코드명
>	1	정액법
2	2	일할계산
3	3	균등계산

No	관리코드	관리코드명
>	1	초일산입
2	2	말일산입
3	3	양편널기

<그림 9-149. 통합코드관리에서 DropDownList 코드 확인>

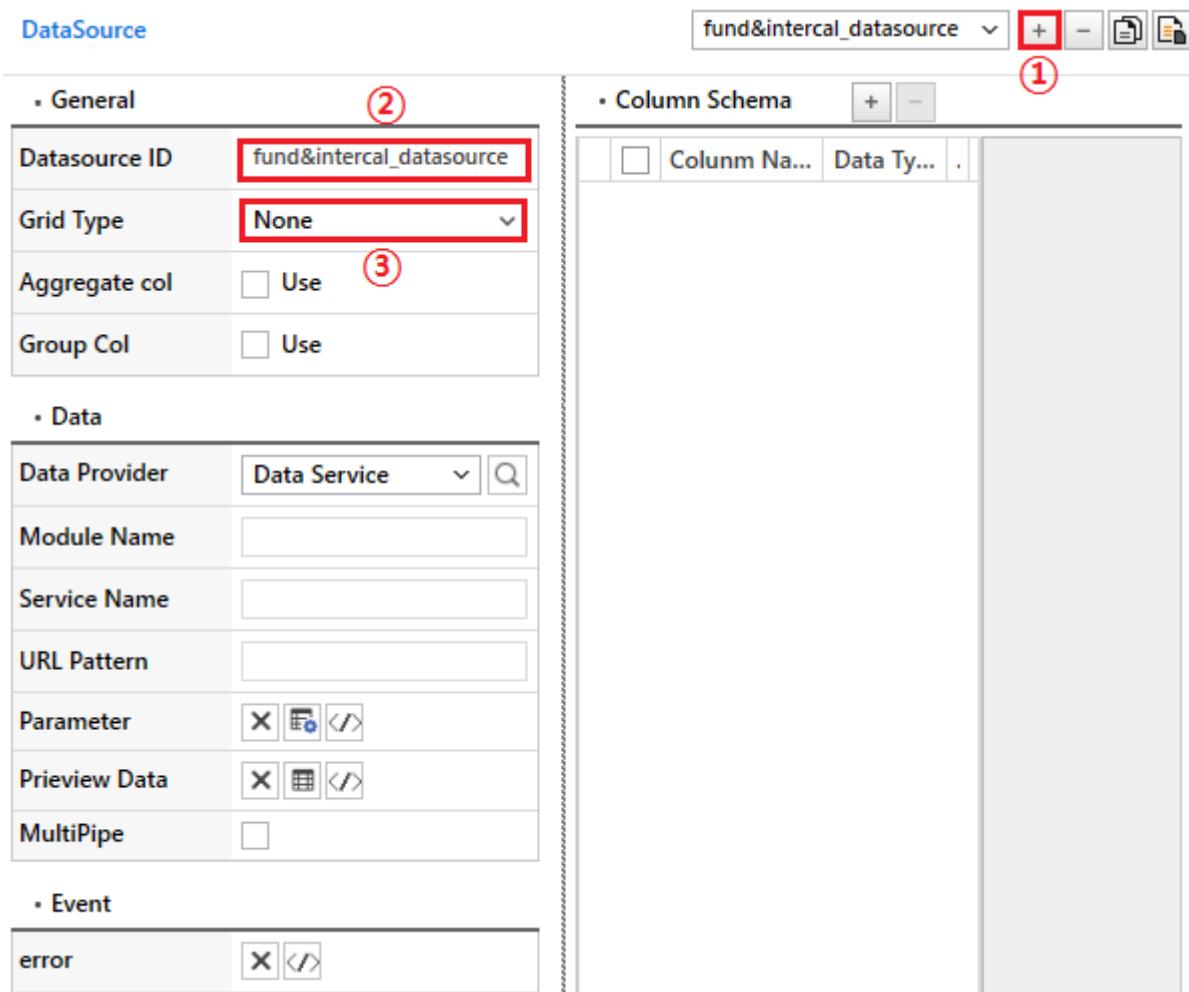
이처럼 통합코드관리에 있는 코드를 가져올 때는 코드디테일데이터 가져오는 서비스를 검색하여 데이터소스를 만들며, 상품유형과 계산방법의 경우 같은 Module 즉, FI(재무회계)이므로 DataSource에서 Data의 MultiPipe 기능과 컨트롤의 pipeparam 속성을 활용하여 하나의 데이터소스를 여러 컨트롤에서 사용할 수 있다.



<그림 9-150. DataSource의 Data > MultiPipe 와 컨트롤의 pipeparam>

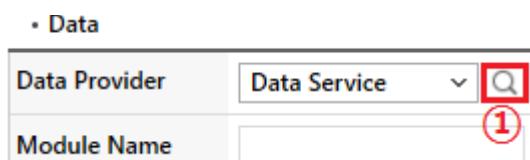
먼저 상품유형과 계산방법에 사용할 Datasource를 생성한다.

- ① 을 눌러 데이터소스를 추가한다.
- ② Datasource ID 를 fund&intercal\_datasource 를 입력한다.,
- ③ Grid 에 사용하지 않으므로 Grid Type 을 None 으로 선택한다.



<그림 9-151. 상품유형과 계산방법에 바인딩할 데이터소스 추가>

Data Provider(데이터 제공자)에서 데이터 서비스를 검색한다. Data Service가 기본값으로 선택되어 있으므로 검색 버튼을 클릭한다.



<그림 9-152. Data Provider 화면 열기>

Search for Data Services 화면이 열리며, 검색 기준이 URL Description으로 선택되어 있다.

- ① 코드디테일을 입력하고 엔터키를 누르거나 ② Search(검색)한다. ③ URL Description 필드에 코드디테일-데이터 조회(URL Pattern: common\_codeDtl\_list)를 선택하고 ④ Apply를 클릭한다. ⑤ 알림창이 뜨면 예를 선택한다.

Search for Data Services

Module	URL Description	Service N.	Service I.
구매관리	SCM공통코드디테일 조회	Purcha...	pubqua00100_MA_SCMCODE_DTL
관리회계	손익센터전표조회 코드디테일 조회 API	ProfitC...	정보분석 pcapia00100_list_codedtl
생산관리	공통코드디테일-조회	ShopFl...	제조... MA_SCMCODE_DTL_list
시스템...	코드디테일 - 데이터 조회(RETURN : ...)	Comm...	common_codeDtl_list
시스템...	CI 코드디테일		
시스템...	코드디테일 - 대		
시스템...	CI 코드디테일 -		
그룹정...	기준정보설정 -		

알림

데이터소스의 컬럼을 새로 정의하시겠습니까?

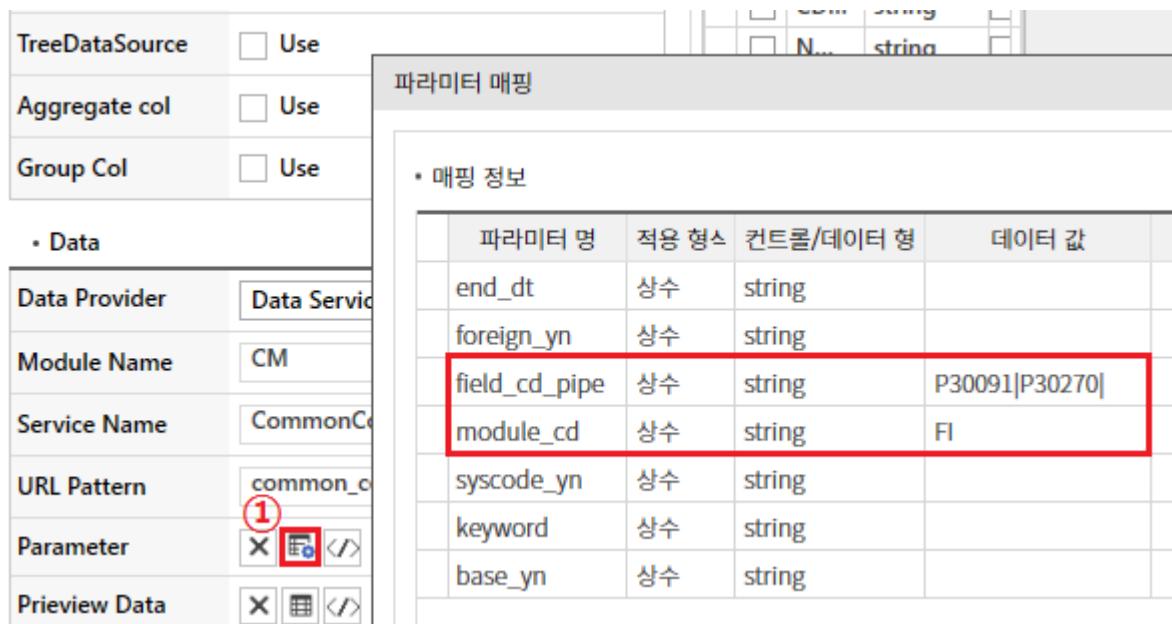
(5)
NO
YES

Parameter	Data Schema		
Parameter Name	Transfer Type	Data Type	
end_dt	QueryString	string	종료일-종료일이 있는 경우 종료일 이전 데이터 제외
foreign_yn	QueryString	string	외국언어적용 유무(Y,N)-Y : NM_SYSDEF2 사용
field_cd_pipe	QueryString	string	코드디테일 코드(PIPE 사용)
module_cd	QueryString	string	모듈 코드
syscode_yn	QueryString	string	시스템코드 유무(Y,N)
keyword	QueryString	string	검색할 코드 또는 명
base_yn	QueryString	string	디폴트 코드구분(Y,N)

(4)
Apply
Cancel

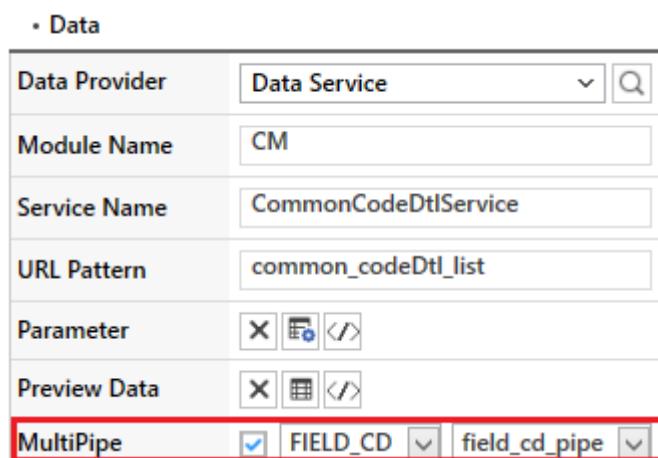
<그림 9-153. URL Description 검색 조건 입력>

다음 Parameter 에서 ① Mapping... 버튼을 눌러 나오는 파라미터 매핑 화면에서 module\_cd 파라미터의 데이터값은 FI, field\_cd\_pipe 는 통합코드관리에서 금융상품유형과 이자계산방법 코드인 P30091,P30270 와 구분자인 파이프()을 붙여 P30091|P30270이를 입력하고 확인을 누른다.



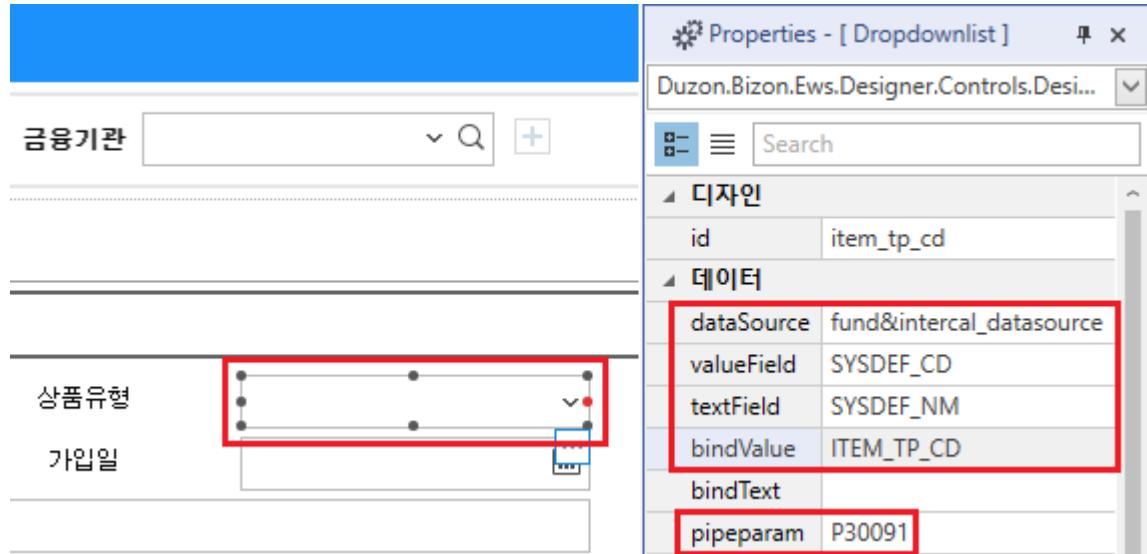
<그림 9-154. 파라미터 매핑 및 데이터값 입력>

MultiPipe을 체크하면 2개의 선택하는 콤보박스가 활성화 된다. 앞은 멀티값을 설정할 컬럼이며 뒤에는 멀티값을 넘기는 파라미터를 지정한다. 디테일코드인 FIELD\_CD가 멀티로 지정할 컬럼이며, 데이터값으로 코드를 입력하는 파라미터는 field\_cd\_pipe이므로 각각 선택한다.



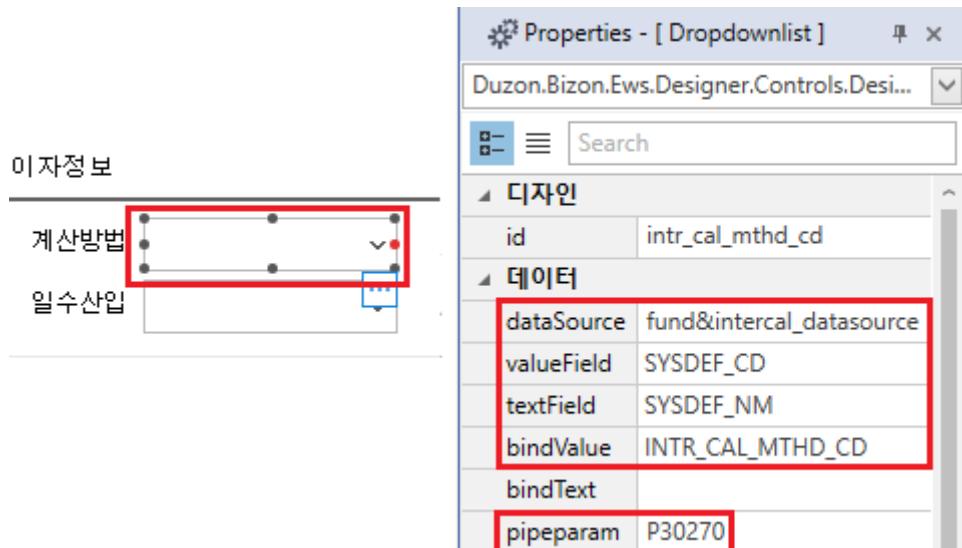
<그림 9-155. MultiPipe 체크 >

다시 Designer 탭으로 가서 상품유형을 선택하고 DataSource를 fund&intercal\_datasource를 선택하고 valueField는 SYSDEF\_CD, textField는 SYSDEF\_NM를 선택한다. pipeparam은 상품유형 코드인 P30091을 입력한다.



<그림 9-156. 상품유형 DropDownList dataSource 속성 설정>

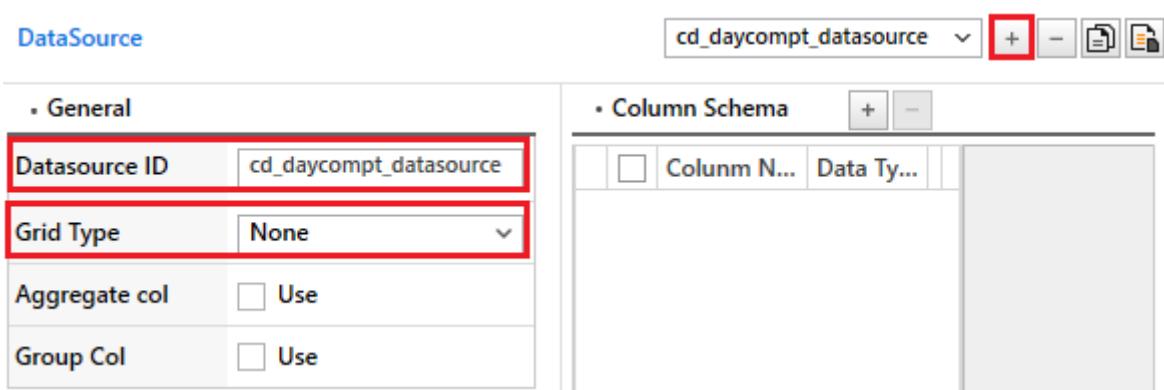
이번에는 계산방법을 선택하고 DataSource 를 상품유형과 같은 fund&intercal\_datasource 를 선택하고 valueField 는 SYSDEF\_CD, textField 는 SYSDEF\_NM 를 선택한다. pipeparam 는 계산방법 코드인 P30270 을 입력한다.



<그림 9-157. 계산방법 DropDownList DataSource 속성 설정>

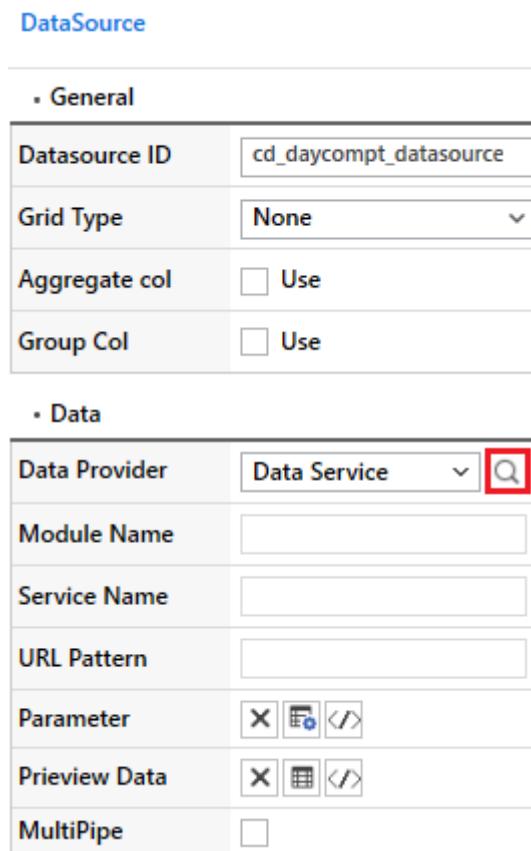
이제 일수산입 DropDownList에 바인딩할 데이터소스를 생성한다.

[+] 을 눌러 데이터소스를 추가한다. Datasource ID 를 cd\_daycompt\_datasource 를 입력한다., Grid 에 사용하지 않으므로 Grid Type 을 None 으로 선택한다.



<그림 9-158. 일수산입에 바인딩할 데이터소스 추가>

Data Provider(데이터 제공자)에서 데이터 서비스를 검색한다. 검색 버튼을 클릭한다.



<그림 9-159. Data Provider 화면 열기>

Search for Data Services 화면이 열리며, 검색 기준이 URL Description으로 선택되어 있다.

- ① 코드디테일을 입력하고 엔터키를 누르거나 ② Search(검색)한다. ③ URL Description 필드에 코드디테일-데이터 조회(URL Pattern: common\_codeDtl\_list)를 선택하고 ④ Apply를 클릭한다. ⑤ 알림창이 뜨면 예를 선택한다.

Search for Data Services

Module I	URL Description	Service N.	Service I
구매관리	SCM공통코드디테일 조회	Purcha...	pubqua00100_MA_SCMCODE_DTL
관리회계	손익센터전표조회 코드디테일 조회 API	ProfitC...	pcapia00100_list_codedtl
생산관리	공통코드디테일-조회	ShopFl...	제조...
시스템...	<b>코드디테일 - 데이터 조회(RETURN : ...)</b>	Comm...	MA_SCMCODE_DTL_list
시스템...	CI 코드디테일 -		
시스템...	코드디테일 -		
그룹정...	기준정보설정 -		

알림

데이터소스의 컬럼을 새로 정의하시겠습니까?

?

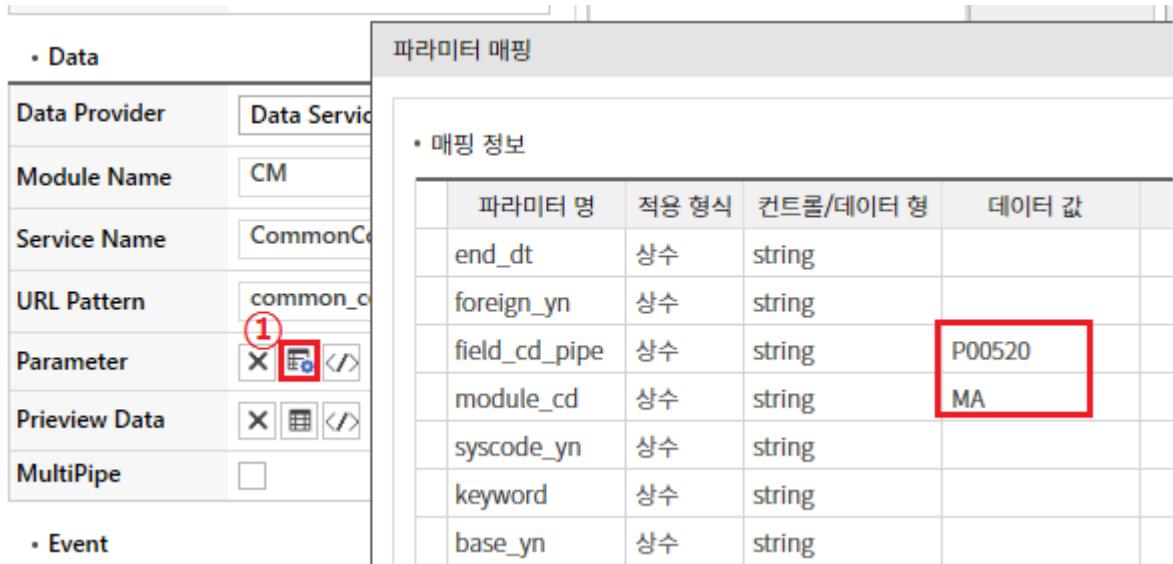
YES
NO

Parameter	Data Schema		
Parameter Name	Transfer Type	Data Type	설명
end_dt	QueryString	string	종료일-종료일이 있는 경우 종료일 이전 데이터 제외
foreign_yn	QueryString	string	외국언어적용 유무(Y,N)- Y : NM_SYSDEF2 사용
field_cd_pipe	QueryString	string	코드디테일 코드(PIPE 사용)
module_cd	QueryString	string	모듈 코드
syscode_yn	QueryString	string	시스템코드 유무(Y,N)
keyword	QueryString	string	검색할 코드 또는 명
base_yn	QueryString	string	디폴트 코드구분(Y,N)

4 Apply Cancel

<그림 9-160. URL Description 검색 조건 입력>

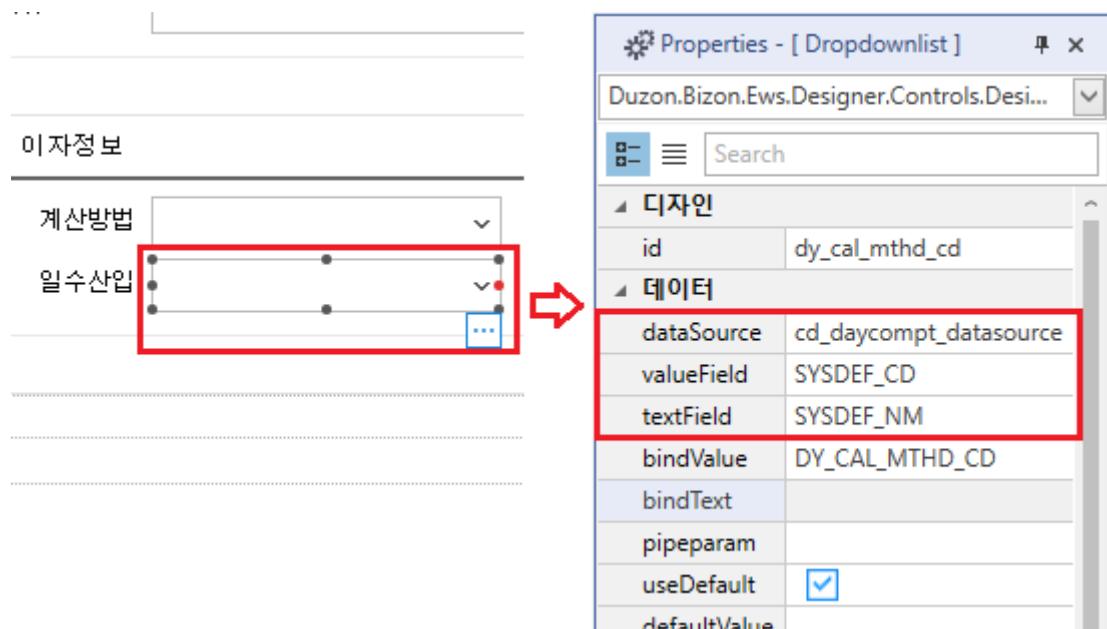
다음 Parameter에서 ① Mapping... 버튼을 눌러 나오는 파라미터 매핑  화면에서 module\_cd 파라미터의 데이터값은 MA, field\_cd\_pipe는 통합코드관리에서 일수산입 코드인 P00520을 입력하고 확인을 누른다.



파라미터 명	적용 형식	컨트롤/데이터 형	데이터 값
end_dt	상수	string	
foreign_yn	상수	string	
field_cd_pipe	상수	string	P00520
module_cd	상수	string	MA
syscode_yn	상수	string	
keyword	상수	string	
base_yn	상수	string	

<그림 9-161. 파라미터 매핑 및 데이터값 입력>

다시 Designer 탭으로 가서 일수산입을 선택하고 DataSource를 cd\_daycompt\_datasource로 선택하고 valueField는 SYSDEF\_CD, textField는 SYSDEF\_NM를 선택한다.



dataSource	cd_daycompt_datasource
valueField	SYSDEF_CD
textField	SYSDEF_NM

<그림 9-162. 상품유형 DropDownList dataSource 속성 설정>

미리보기를 실행하여 이자정보 카드리스트도 펼쳐보면 상품유형, 계산방법, 일수산입에 데이터를 확인할 수 있다.



<그림 9-163. 미리보기에서 DropDownList 컨트롤들 데이터 확인>

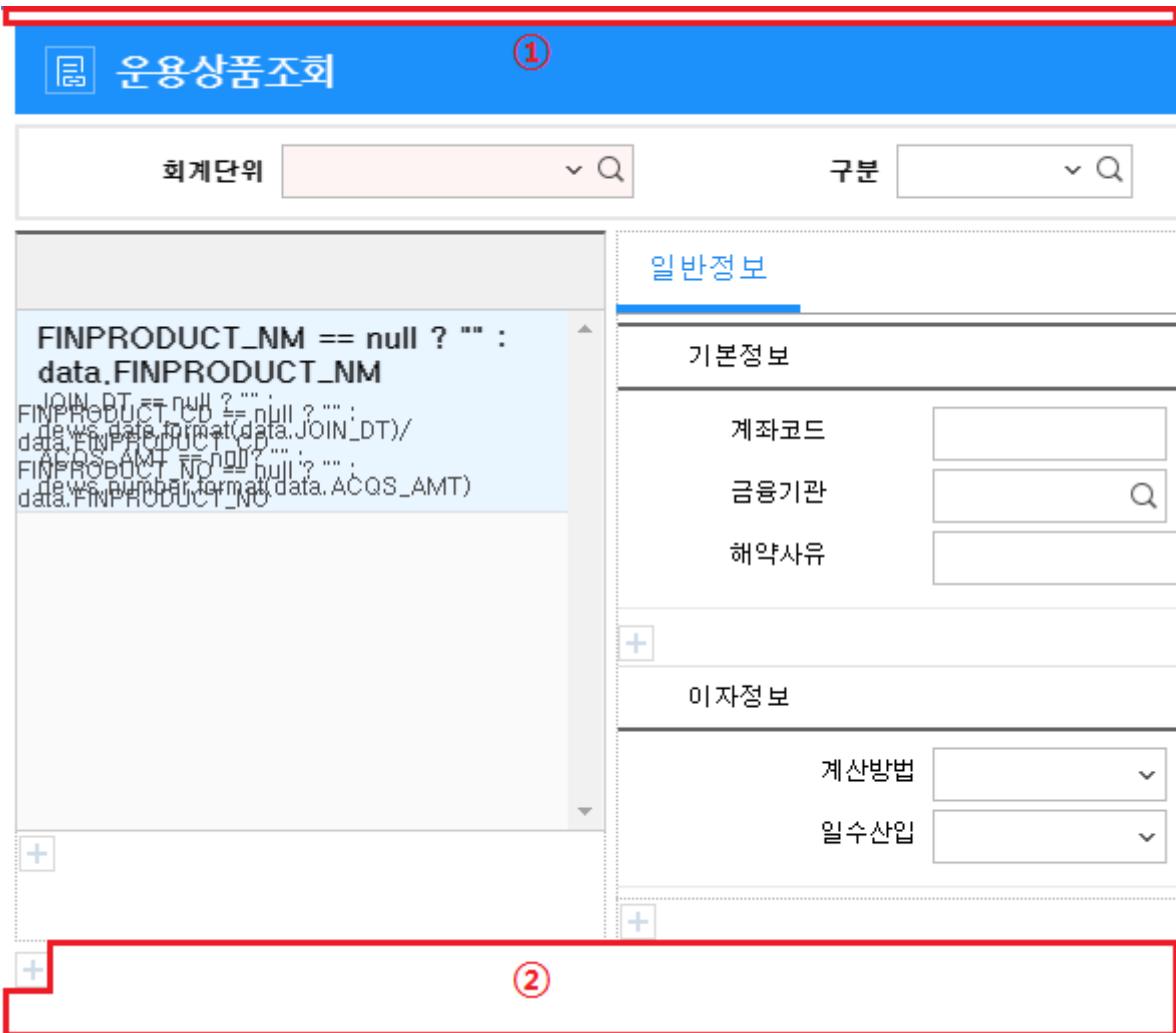
#### 4) 이벤트 스크립트 설정

운용상품조회에서 사용되는 이벤트 스크립트 리스트이다.

컨트롤(객체)	이벤트	설명
Cavas	preReady	페이지가 로드될 때 초기값 설정
MainButton - search	click	조회 버튼 눌렀을 때 필수 조회값 체크 및 조회
CardList	change	CardList 값이 변경되었을 때 FormPanel과 바인딩 되는 스크립트 작성
FormPanel	change	FormPanel 내부 컨트롤들의 값이 변경되었을 때 CardList와 바인딩 되는 스크립트 작성

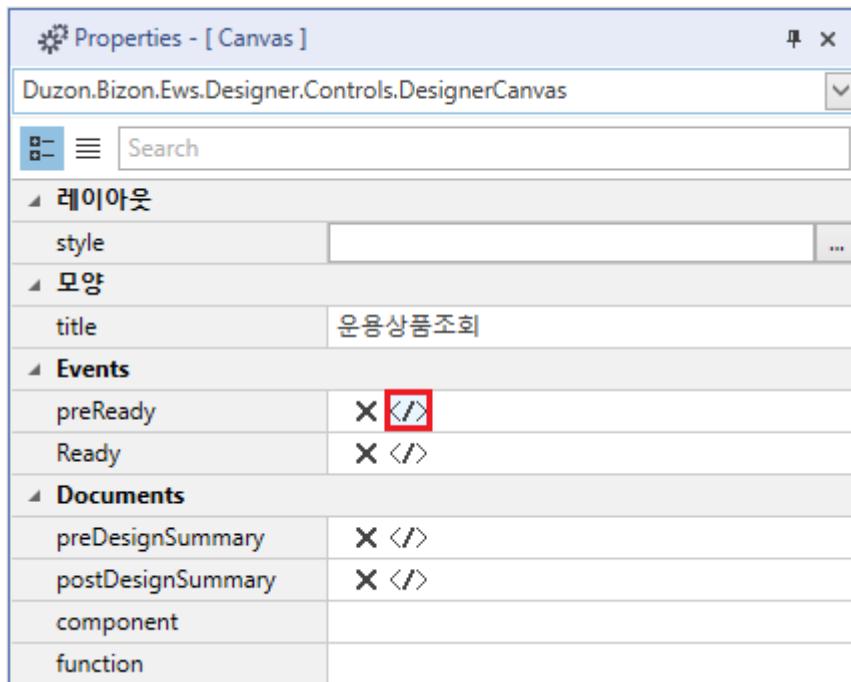
<표 9-18. 운용상품조회 이벤트>

먼저 초기값을 설정하는 스크립트를 작성한다. 회계단위에 초기값을 설정하며, 최상위 Cavas 의 Events 에서 구현한다. Cavas 는 문서 제목 위에 ① 상단 또는 아래 ② 빈 하단 부분을 클릭하여 선택한다.



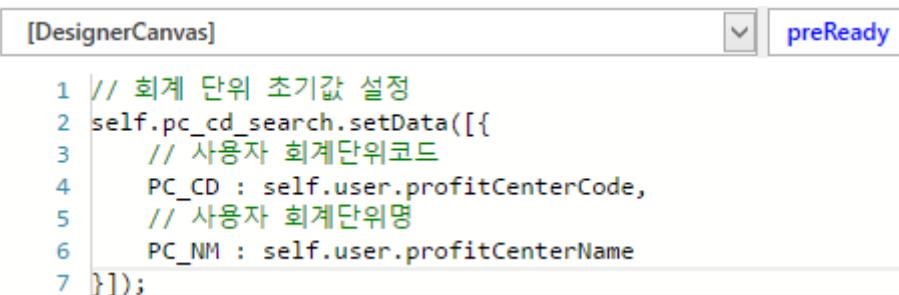
<그림 9-164. Cava 선택>

이제 이벤트 preReady 스크립트 버튼 </> 을 눌러 CodeEditor 화면에서 회계단위의 초기값을 설정하는 스크립트를 작성한다.



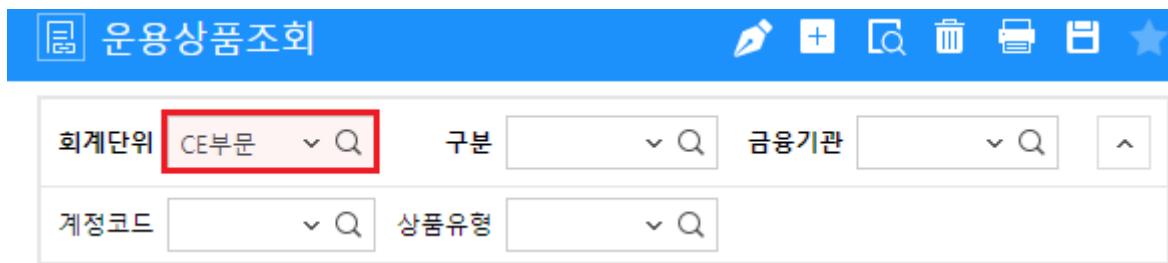
<그림 9-165. preReady 이벤트 스크립트 버튼 실행>

```
// 회계 단위 초기값 설정
self.pc_cd_search.setData([
 // 사용자 회계단위코드
 PC_CD : self.user.profitCenterCode,
 // 사용자 회계단위명
 PC_NM : self.user.profitCenterName
]);
```



<그림 9-166. Cavas preReady Event 스크립트 작성>

미리보기를 실행하면 회계단위에 초기값이 설정되어 있다.



<그림 9-167. 미리보기에서 초기값 확인>

다음으로 필수 조회 조건을 체크하고 조회하는 이벤트 스크립트를 추가한다.

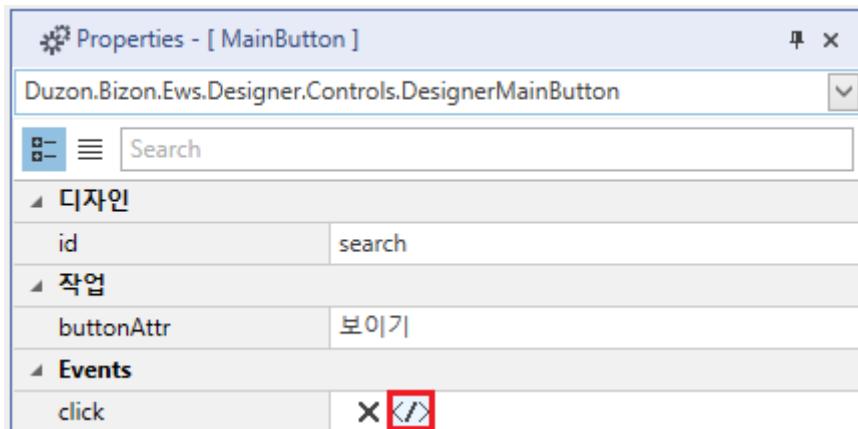
필수 조회 조건 체크는 조회 컨트롤들을 포함하고 있는 ConditionPanel에서 체크하며, 조회는 그리드와 바인딩되는 Datasource의 read 함수를 호출한다.

조회버튼을 선택한다.



<그림 9-168. 조회 버튼 선택>

프로퍼티 창에서 click 이벤트를 클릭한다.



<그림 9-169. Click 이벤트 선택>

CodeEditor 화면에서 다음과 같이 입력하고 OK를 누른다.

```
// 필수 조회 조건 체크(컨디션 패널 ID)
function() {
 if (self.searchCondition.validate({tooltip:true,message:'필수 항목을 입력해 주십시오.'}))
 {
 // CardList 데이터소스 읽기
 self.cardlist_datasource.read();
 }
}
```

The screenshot shows a script editor window with the following details:

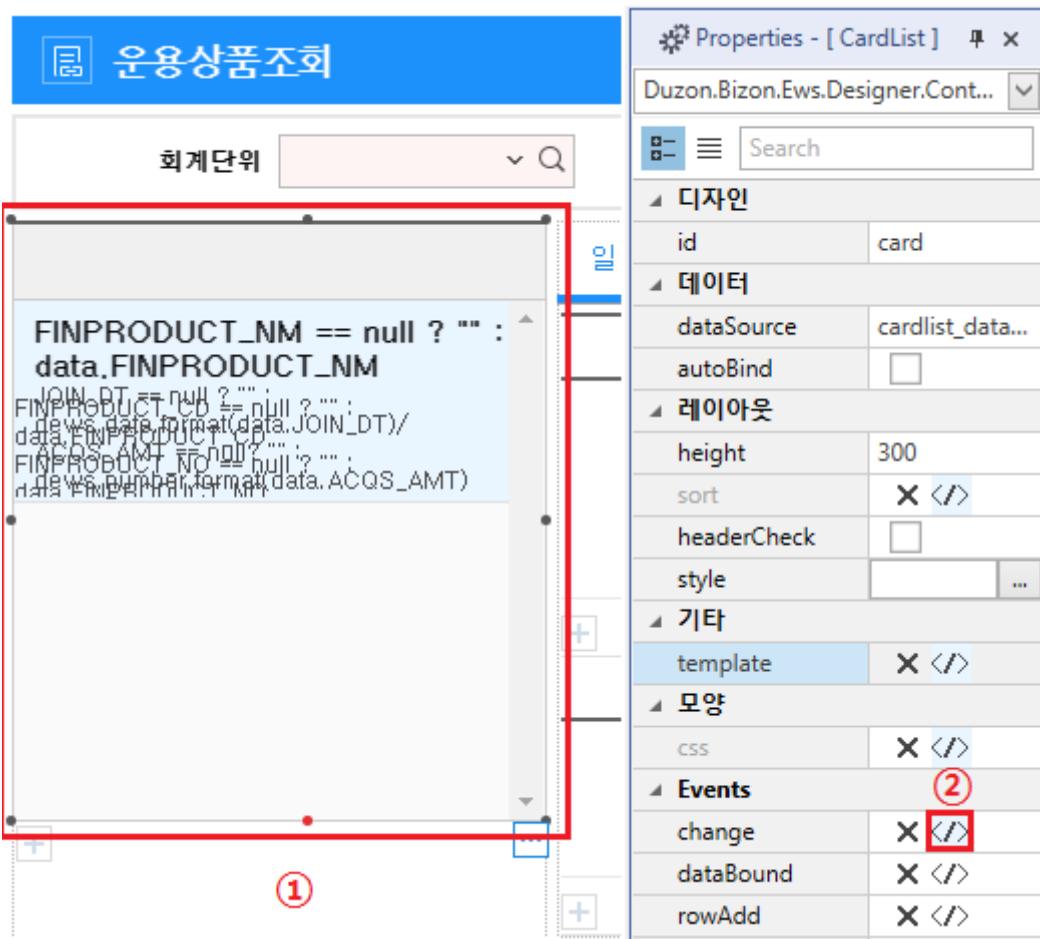
- Top bar: "search [DesignerMainButton]" and a dropdown menu.
- Tab: "click" (highlighted in blue).
- Script content:

```
1 function() {
2 if (self.searchCondition.validate({ tooltip: true, message: '필수 항목을
3 // CardList 데이터소스 읽기
4 self.cardlist_datasource.read();
5 }
6 }
7
```

<그림 9-170. 조회 버튼 Click 이벤트에 스크립트 입력 화면>

이제 CardList 값이 변경되었을 때 FormPanel 과 바인딩 되는 스크립트를 작성하기 위하여 CardList 의 Change 이벤트에 스크립트를 작성한다. 현재 FormPanel 은 기본정보 및 이자정보 ArcodienPanel 에 formpanel1 과 formpanel2 ID 로 각각 생성되어 있다.

CardList 를 선택하고 Events 에 Change 이벤트 스크립트를 클릭한다.



<그림 9-171. CardList change 이벤트 스크립트 실행>

change 스크립트 화면에서 다음과 같이 입력한다.

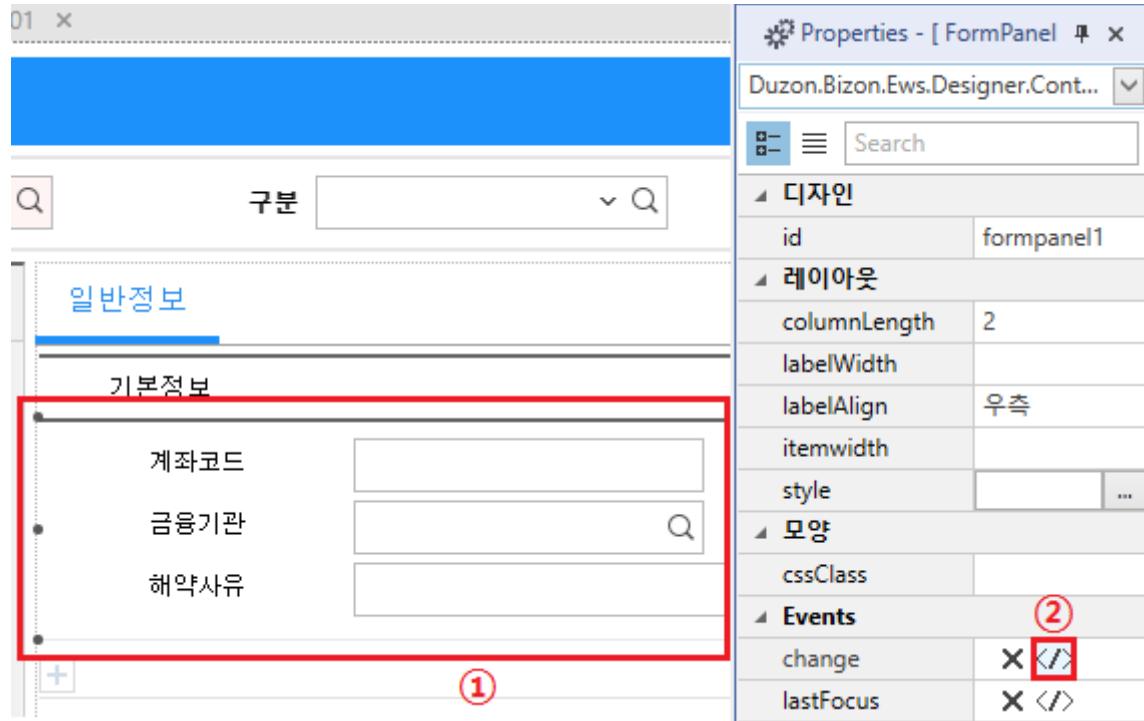
```
function(e) {
 //bindPanel(panel) : 카드리스트의 해당 row에 맞는 데이터를 폼패널에 바인딩
 self.card.bindPanel(self.formpanel1);
 self.card.bindPanel(self.formpanel2);
}
```

card [DesignerCardList]	▼	change
<pre>1 function(e) { 2     //bindPanel(panel) : 카드리스트의 해당 row에 맞는 데이터를 폼패널에 바인딩 3     self.card.bindPanel(self.formpanel1); 4     self.card.bindPanel(self.formpanel2); 5 }</pre>		

<그림 9-172. CardList change 이벤트 스크립트 입력>

이번에는 FormPanel 내의 컨트롤들의 값이 변경 되었을 때 CardList 와 바인딩 되는 스크립트를 작성하기 위하여 FormPanel 의 Change 이벤트에 스크립트를 작성한다.

기본정보 ArcodienPanel 의 ① formpanel1 을 선택하고 ② change 스크립트를 클릭한다.



<그림 9-173. FormPanel change 이벤트 스크립트 실행>

change 스크립트 화면에서 다음과 같이 입력한다.

```
function(e) {
 //bindTo(component) : 현재 품패널에 입력된 데이터를 특정 데이터 컴포넌트로 바인딩
 self.formpanel1.bindTo(self.card);
}

formpanel1 [DesignerFormPanel] change
1 function(e) {
2 //bindTo(component) : 현재 품패널에 입력된 데이터를 특정 데이터 컴포넌트로
3 self.formpanel1.bindTo(self.card);
4 }
```

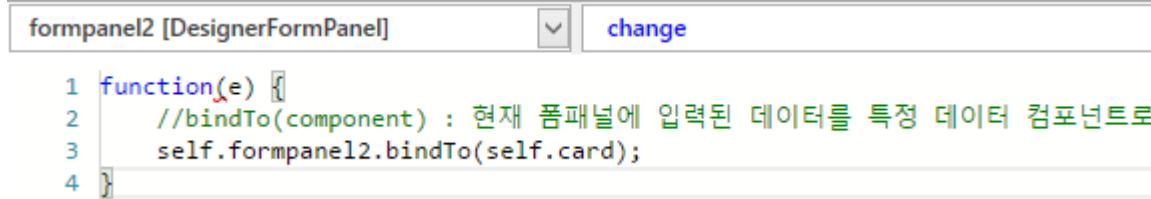
<그림 9-174. formpanel1 change 이벤트 스크립트 입력 >

마찬가지로 이자정보 ArcodienPanel 의 formpanel2 의 change 스크립트도 같이 작성한다.

```

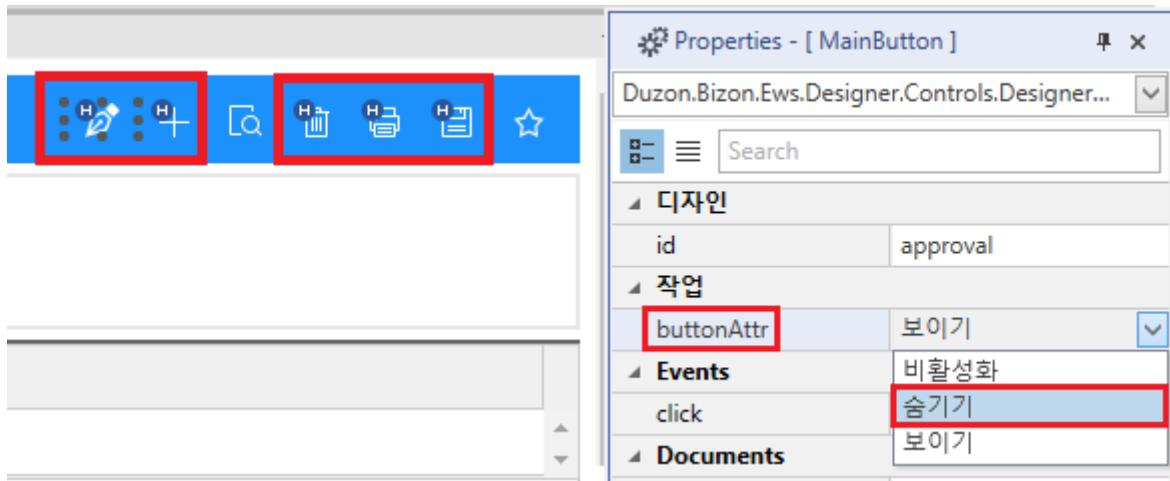
function(e) {
 //bindTo(component) : 현재 폼패널에 입력된 데이터를 특정 데이터 컴포넌트로 바인딩
 self.formpanel2.bindTo(self.card);
}

```



<그림 9-175. formpanel2 change 이벤트 스크립트 입력 >

상단 우측 나머지 버튼들은 사용하지 않기 때문에 buttonAttr 속성을 숨기기로 설정한다.



<그림 9-176. 나머지 버튼 buttonAttr 속성 변경>

미리보기를 하고, 조회를 했을 때 데이터가 표시되고 왼쪽 카드리스트 포커스를 이동하면 오른쪽 폼패널안의 컨트롤 값이 바뀌는 것을 확인할 수 있다.

운용상품조회

회계단위	CE부문	구분
계정코드		상품유형
취득일기준 정렬		
<b>(액면발행) 투자사채 16-1차</b> 2016-05-15/ 3,600,000,000 2-1000		
<b>(할증발행) 투자사채 16-3차</b> 2016-05-15/ 3,800,000,000 2-1002		
<b>(할인발행) 투자사채 16-2차</b> 2016-05-15/ 3,400,000,000 2-1001		
<b>(액면발행) 투자사채 16-4차</b> 2016-08-15/ 400,000,000 2-1003		
<b>보통예금_국민</b> 2017-01-01/ 1002		
<b>보통예금- 신한_입출금</b> 2017-01-01/ 1001		

**일반정보**

---

계좌코드 2-1000

금융기관 신한은행 Q

해약사유

**이자정보**

---

계산방법 균등계산

일수산입 초일산입

<그림 9-177. 미리보기 후 데이터 확인>

만약 오른쪽 폼패널에서 값을 변경할 경우 왼쪽 카드리스트에 반영되기 위해서는 카드리스트와 연결된 데이터소스의 컬럼 속성이 수정 가능해야 한다.

Column Name	Data Ty...	Modify
FINPRODUCT_CD	string	<input checked="" type="checkbox"/>
NO_SQ	string	<input type="checkbox"/>
FINPRODUCT_NO	string	<input type="checkbox"/>
<input checked="" type="checkbox"/> FINPRODUCT_NM	string	<input type="checkbox"/>
FINPRODUCT_NM2	string	<input type="checkbox"/>

<그림 9-178. CardList DataSource 컬럼 수정 설정>

이처럼 해당 컬럼의 Modify 를 체크 해야 CardList 에 반영된다.

**일반정보**

계좌코드: 2-1001000

금융기관: 신한은행

해약사유:

**[수정 체크 전]**

**일반정보**

계좌코드: 2-1001000

금융기관: 신한은행

해약사유:

**[수정 체크 후]**

<그림 9-179. 폼 패널 컨트롤 값 변경 할 때 CardList 에 반영>

## 4. 빠른거래처 등록

### 1) 개발 화면

<그림 9-194>는 빠른거래처 등록 화면이다. 상단 필수 입력 조회조건은 거래처구분과 거래처분류이며 컨디션 패널은 필수 영역과 옵션 영역으로 되어 있다. 아래 그리드의 필수 입력 항목은 거래처코드, 거래처명, 사업자등록번호이고 번호 옆에 체크 박스가 있으며 삭제 시 체크된 항목만 삭제 된다. 등록 시에 국가코드, 거래처구분코드, 거래처분류코드는 자동입력되며, 사업자 등록번호의 경우 에디터에 마스크텍스트로 입력한다.

No	거래처코드	거래처명	대표자명	사업자등록번호	업태
1	2000001009	GS글로벌	김성곤	123-89-70650	유통업
2	<input checked="" type="checkbox"/> 2000001007	SKC Haas	이태화	126-75-70210	정유
3	<input checked="" type="checkbox"/> 23232323232	aaaaaaa2222	aaaaa22222	130-85-17232	aaaa
4	<input type="checkbox"/> 2000001005	기아자동차	박한우	124-84-70340	정유

No	거래처코드	거래처명	국가코드	거래처구분코드	거래처분류코드
1	2000001002	현대파워텍	대한민국	주요	매입
>			대한민국	주요	매입

대표자명	사업자등록번호	업태	업종	국가코드
김성곤	123-89-70650	유통업	상품 종합 도매업	대한민국
김해진	129-83-74770	미션	제조	대한민국
				대한민국

<그림 9-180. 빠른거래처 개발화면>

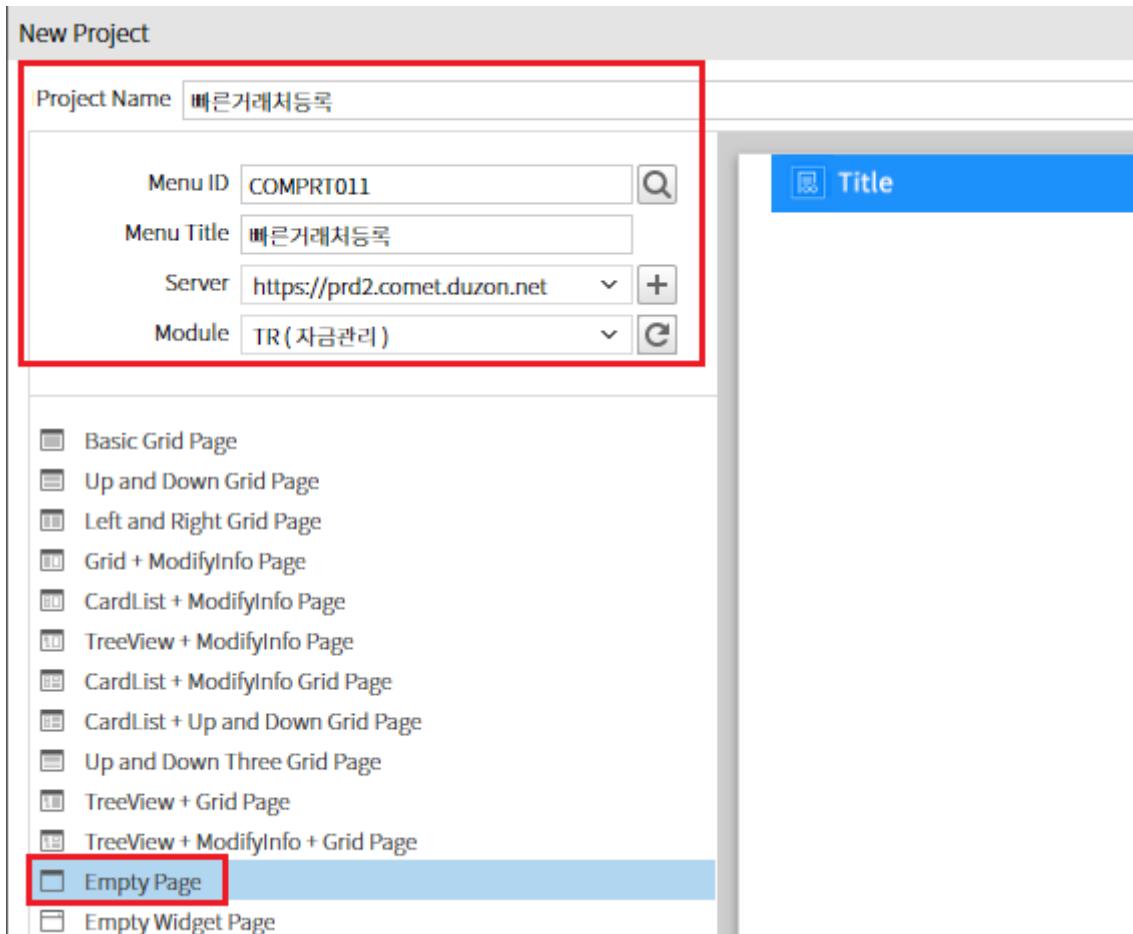
## 2) 화면 디자인

새 프로젝트를 생성한다. (File ▶ New ▶ Project... 또는 툴바의 

Project Name (프로젝트 명)	빠른거래처등록 입력
Menu ID (메뉴ID)	COMPRT011 입력
Menu Title (메뉴 제목)	빠른거래처등록 입력
Server (서버 선택)	https://prd2.comet.duzon.net 선택
Module (모듈 선택)	TR (자금관리) 선택
템플릿(template)	Empty Page 선택

<표 9-19. 새 프로젝트 생성 입력>

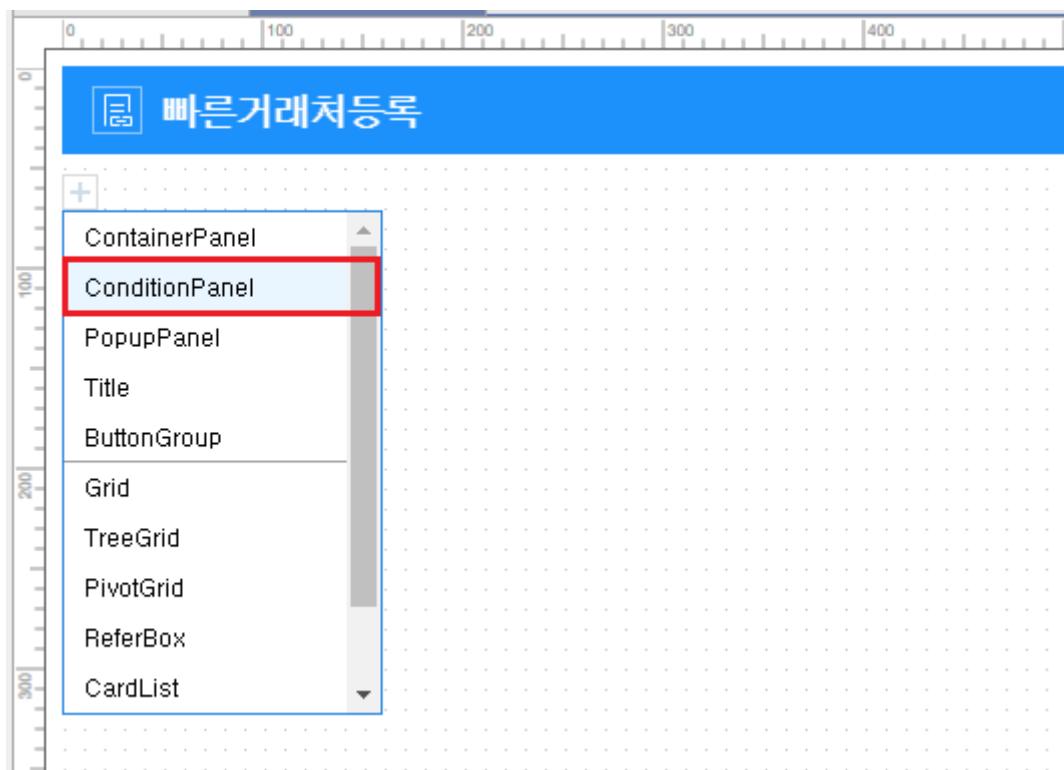
위 정보를 입력 후 OK 버튼을 클릭하여 프로젝트를 생성한다.



<그림 9-181. New Project 생성 화면 >

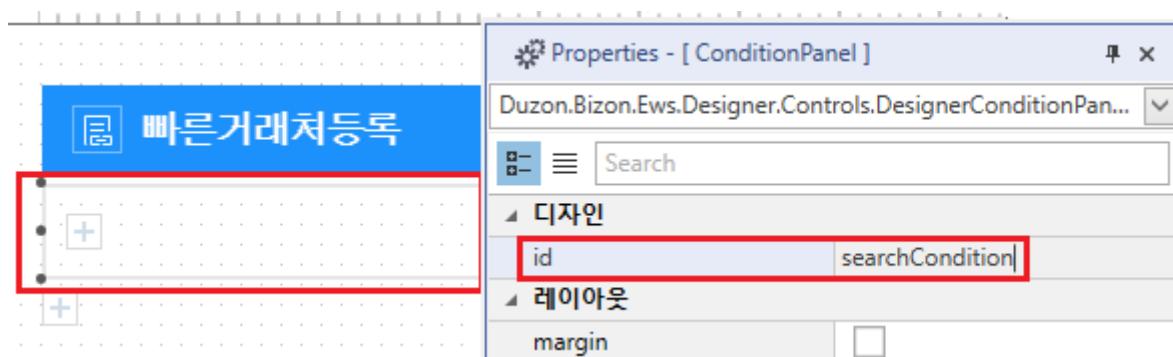
조회 조건에 해당하는 컨트롤들은 ConditionPanel, 데이터를 보여주는 그리드 컨트롤은 ContainerPanel에서 각각 생성한다. 먼저 조회 조건에 해당하는 컨트롤을 그린다.

- ① 버튼을 누르고 ConditionPanel을 선택한다.



<그림 9-182. ConditionPanel 선택>

프로그램 실행 후 조회할 때 필수 조회 조건 체크를 하기 위해 ConditionPanel 선택 후 ID 속성을 searchCondition으로 입력한다.



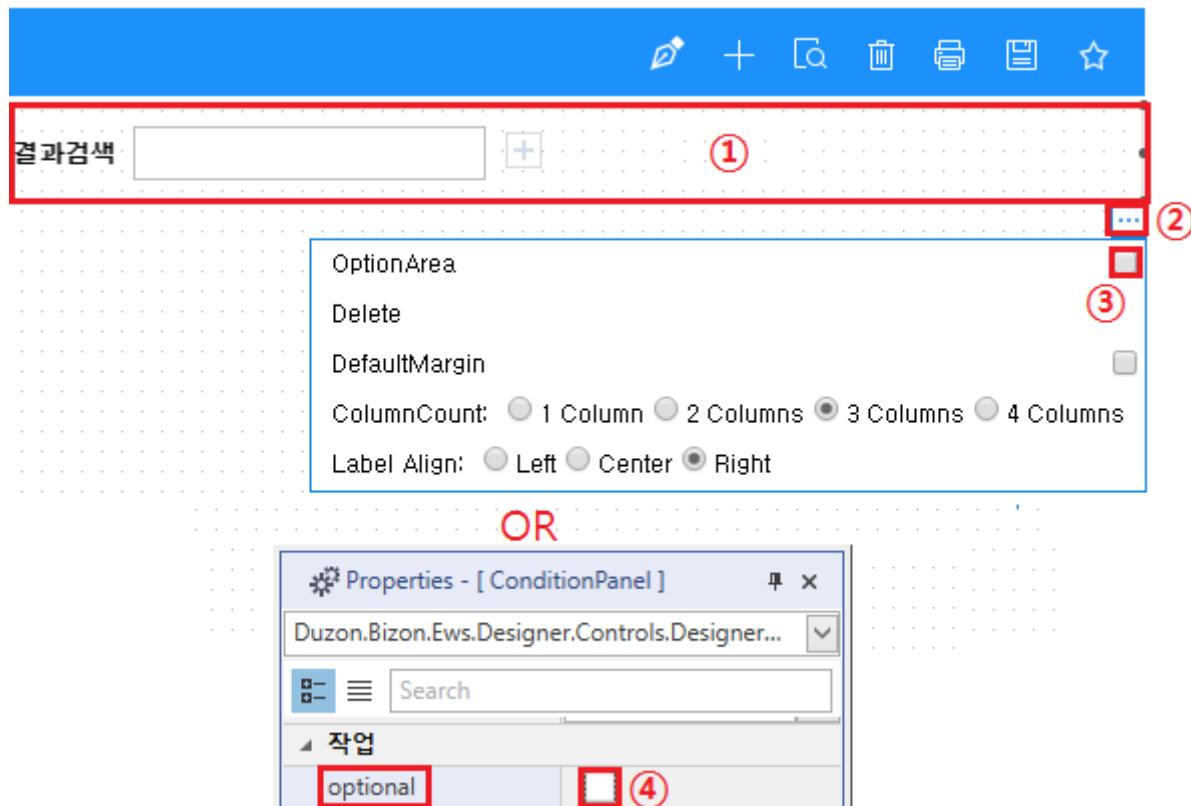
<그림 9-183. ConditionPanel 생성 및 ID 입력>

ConditionPanel 내의 를 눌러 <표 9-25>와 같이 필수 영역 3 개 컨트롤을 추가하고 속성을 변경한다.

컨트롤	ConditionItem	각 컨트롤 속성 설정	
		구분	설정 또는 입력값
Dropdownlist	라벨: 거래처구분	디자인	id : partner_fg_cd
		데이터	DataSource 생성 후 재설정
		모양	필수
Dropdownlist	라벨: 거래처분류	디자인	id : partner_fg_cd
		데이터	DataSource 생성 후 재설정
		모양	필수
TextBox	라벨: 결과검색	디자인	id : search_tx

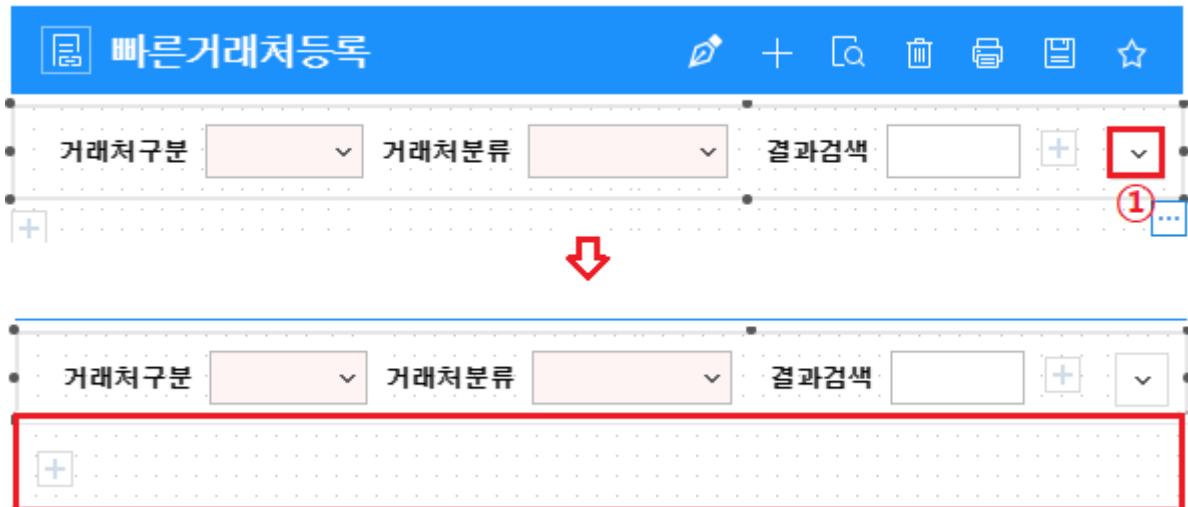
<표 9-20. 필수 영역 컨트롤 추가 설정>

옵션 영역에 추가하기 위해서 ConditionPanel 선택 후 작업의 ④ optional 속성을 체크하거나  
 ① ConditionPanel 선택 후 ② (더보기) 버튼을 눌러서 나오는 화면에서 ③ OptionArea 을 체크한다.



<그림 9-184 옵션 영역 선택>

다음으로 ① 아래 방향을 눌러 옵션 영역을 표시한다.



<그림 9-185. 옵션 영역 표시>

ConditionPanel 내의 를 눌러 <표 9-26>와 같이 옵션 영역에 2 개 컨트롤을 추가하고 속성을 변경한다

컨트롤	ConditionItem	각 컨트롤 속성 설정	
		구분	설정 또는 입력값
Dropdownlist	라벨: 조회구분	디자인	id : regma_yn
		데이터	DataSource 생성 후 재설정
Dropdownlist	라벨: 사용여부	디자인	id : use_yn
		데이터	DataSource 생성 후 재설정

<표 9-21. 옵션 영역 컨트롤 추가 설정>

### 3) 데이터소스 추가 및 바인딩

빠른거래처에서 필요한 데이터소스는 조회 조건과 그리드 내의 컬럼인 거래처구분(코드)과 거래처분류(코드) 2개, 그리드 내의 컬럼인 국가코드 1개, 조회 조건의 조회구분과 사용여부 2개, 전체 그리드 1개 등 총 6개가 필요하다. 하지만 국가코드, 거래처구분, 거래처분류는 멀티파이프(MultiPipe)를 사용하여 1개로, 조회구분과 사용여부는 같은 데이터를 사용하므로 1개로 총 3개만 생성한다.

General	Data	Column Schema	적용 컨트롤
<ul style="list-style-type: none"> <li>- Datasource ID : com_partner_ds</li> <li>- Grid Type: None</li> </ul>	<ul style="list-style-type: none"> <li>- Data Provider : Data Service ⇒ 코드디테일검색 URL Pattern - common_codeDtl_list 선택</li> <li>- Parameter : Mapping... 버튼 클릭 후 field_cd_pipe ⇒ 상수/string/ P00110 P00080 P00170  module ⇒ 상수/string/MA</li> <li>- MultiPipe : 체크/ FIELD_CD/field_cd_pipe</li> </ul>	자동생성	<조회조건> 거래처구분, 거래처분류  <그리드> 국가코드, 거래처구분 코드, 거래처분류 코드
<ul style="list-style-type: none"> <li>- Datasource ID : yn_ds</li> <li>- Grid Type: None</li> </ul>	<ul style="list-style-type: none"> <li>- Data Provider : Local Data</li> <li>- Data : 스크립트 버튼  또는 JsonConvert 눌러서 아래 입력 [ {value: "Y", text: "Yes"}, {value: "N", text: "No"} ]</li> </ul>	+ 눌러 value text 추가 (둘다 String)	<조회조건> 조회구분, 사용여부
<ul style="list-style-type: none"> <li>- Datasource ID : grid_ds</li> <li>- Grid Type: Grid</li> </ul>	<ul style="list-style-type: none"> <li>- Data Provider : Data Service ⇒ URL Pattern 선택 comprt00100_list 조회 및 선택</li> <li>- Parameter : Mapping... 버튼 클릭 후 regma_yn ⇒ 컨트롤/regma_yn partner_csf_cd ⇒ 컨트롤/ partner_csf_cd search_tx ⇒ 컨트롤/search_tx use_yn ⇒ 컨트롤/use_yn partner_fg_cd ⇒ 컨트롤/partner_fg_cd</li> </ul>	자동생성	그리드

<표 9-22. 데이터소스 추가>

Designer 탭으로 가서 조회 조건의 거래처구분을 선택한다. 프로퍼티에서 DataSource 는 com\_partner\_ds 을 선택, autoBind 체크, valueField 는 SYSDEF\_CD, textField 는 SYSDEF\_NM 를 선택한다. 마지막으로 pipeparam 은 P00080 을 입력한다.

같은 방법으로 거래처분류의 pipeparam 은 P00110 을 입력한다.

The figure consists of two side-by-side examples. Each example has a selection bar on the left and a properties panel on the right. A blue arrow points from each selection bar to its respective properties panel.

**Example 1: 거래처구분 (Customer Type)**

디자인	
id	partner_fg_cd
데이터	
autoBind	<input checked="" type="checkbox"/>
dataSource	com_partner_ds
valueField	SYSDEF_CD
textField	SYSDEF_NM
bindValue	
bindText	
pipeparam	P00080

**Example 2: 거래처분류 (Customer Category)**

디자인	
id	partner_csf_cd
데이터	
autoBind	<input checked="" type="checkbox"/>
dataSource	com_partner_ds
valueField	SYSDEF_CD
textField	SYSDEF_NM
bindValue	
bindText	
pipeparam	P00110

<그림 9-186. 조회 조건 거래처구분, 거래처분류 설정>

조회구분, 사용여부의 경우 autoBind 체크, dataSource 는 yn\_ds, valueField 는 value, textField 는 text 를 선택한다.

The figure shows two sections on the left and their corresponding properties panels on the right. A red box highlights the '사용여부' section on the left.

**조회구분 (Search Condition)**

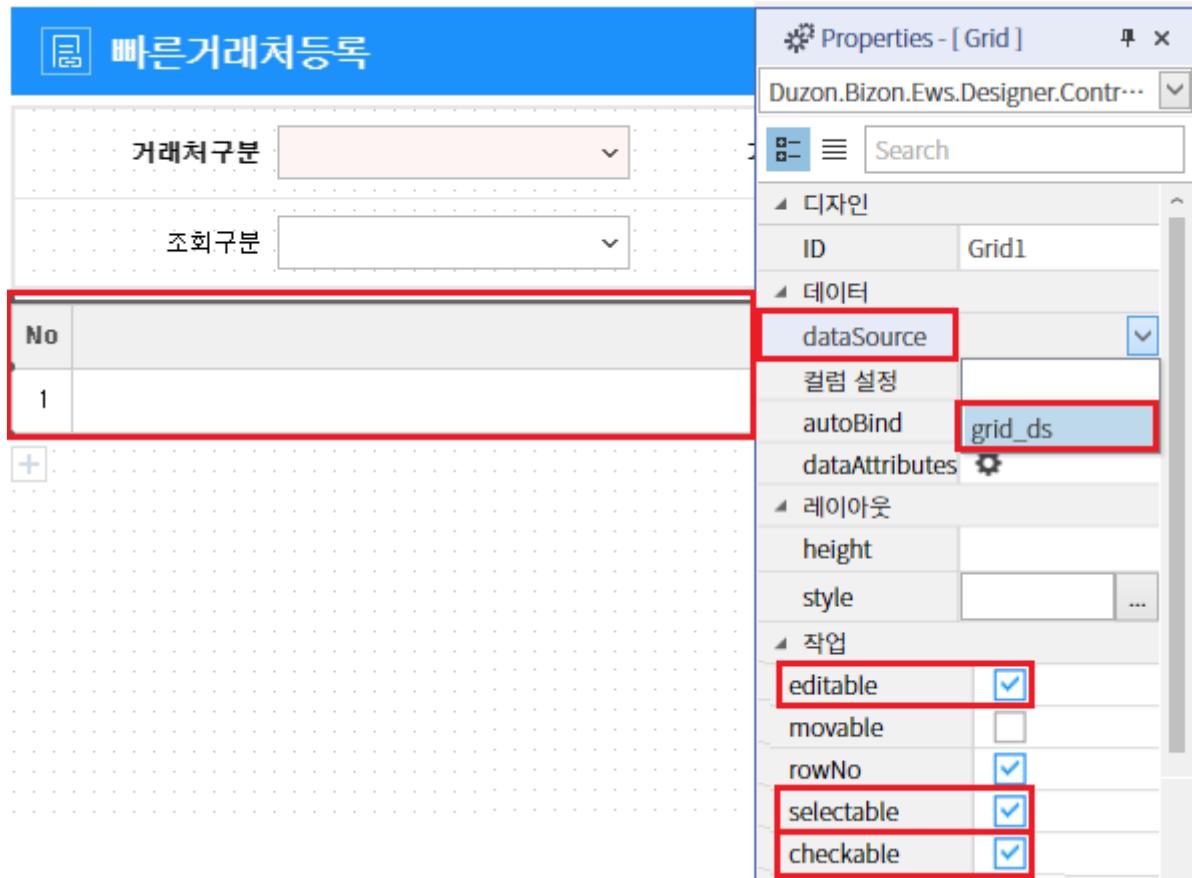
데이터	
autoBind	<input checked="" type="checkbox"/>
dataSource	yn_ds
valueField	value
textField	text

**사용여부 (Usage Status)**

데이터	
autoBind	<input checked="" type="checkbox"/>
dataSource	yn_ds
valueField	value
textField	text

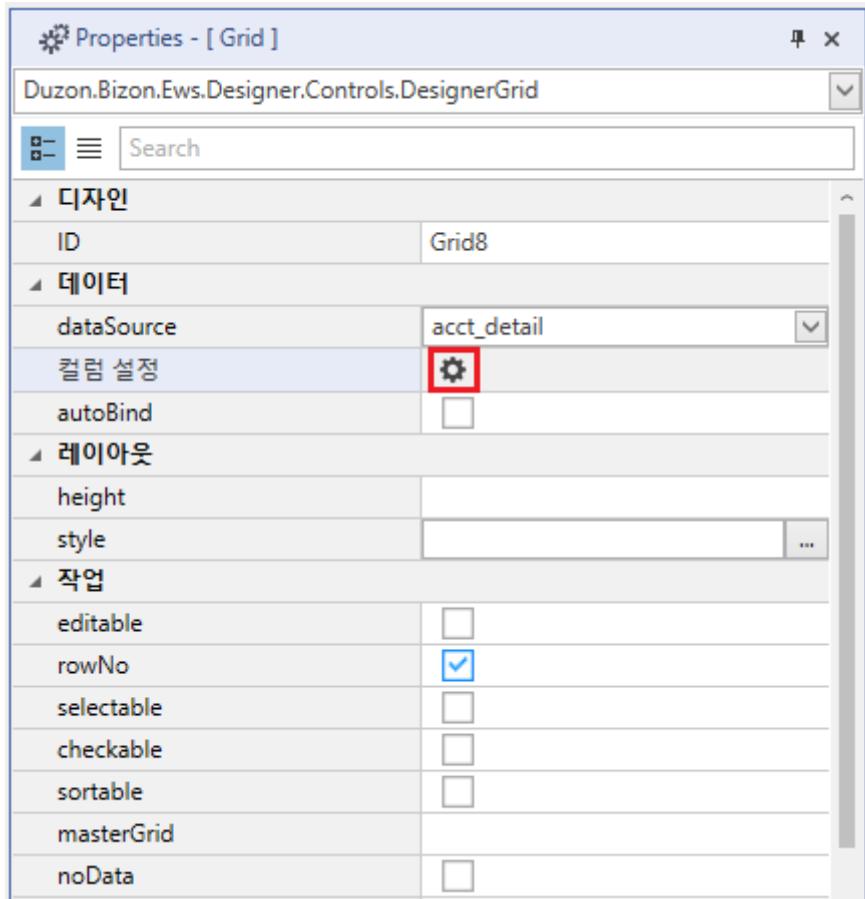
<그림 9-187. 조회 조건 조회구분, 사용여부 설정>

이제 Grid 를 선택한 후에, ③ DataSource 속성에서 acct\_detail 을 선택한다. 수정 편집이 가능해야 하므로 프로퍼티에서 editable 체크, 체크 박스가 있으므로 checkable 에 체크한다.



<그림 9-188. 계정 Detail Grid 의 dataSource 속성 설정>

컬럼별 속성을 정해주기 위해 컬럼설정 을 클릭한다.



<그림 9-189. 계정 Header Grid 컬럼 설정 열기>

그리드-컬럼설정 화면에서 컬럼별 속성을 <표 9-11>과 같이 설정한다(출력 컬럼만 설정)

컬럼명	설정	
	헤더타이틀	속성
PARTNER_CD	거래처코드	<p>attributes : required 에디터 ►텍스트박스 maxLength : 20 editable :</p> <pre>if(e.grid.getRowState(e.row.index) === 'added') {     return true; } else {     return false; }</pre>
PARTNER_NM	거래처명	attributes : required
CEO_NM	대표자명	

BIZR_NO	사업자등록 번호	attributes : required 출력 포맷 ► 마스크 마스크 문자열 : 000-00-00000 에디터 ► 마스크텍스트박스 format : 000-00-00000 editable : <pre>if(e.grid.getRowState(e.row.index) === 'added') {     return true; } else {     return false; }</pre>
BIZC_NM	업태	attributes : none
BIZTP_NM	업종	attributes : none
NATION_CD	국가코드	attributes : required 에디터 ► 드롭다운리스트 dataSource : com_partner_ds, dataValueField : SYSDEF_CD, dataTextField : SYSDEF_NM, dataPipe : P00170
PARTNER_FG_CD	거래처구분 코드	attributes : none 에디터 ► 드롭다운리스트 dataSource : com_partner_ds, dataValueField : SYSDEF_CD, dataTextField : SYSDEF_NM, dataPipe : P00080
PARTNER_CSF_CD	거래처분류 코드	attributes : none 에디터 ► 드롭다운리스트 dataSource : com_partner_ds, dataValueField : SYSDEF_CD, dataTextField : SYSDEF_NM, dataPipe : P00110

<표 9-23. 컬럼 상세 설정>

설정을 완료한 계정 탭 화면이다.

#### 4) 이벤트 스크립트 설정

빠른거래처 등록에서 사용되는 이벤트 스크립트 리스트이다.

컨트롤(객체)	이벤트	설명
Grid	rowAdd	<pre>// 그리드 행이 추가된 이후에 발생하는 이벤트 // 추가 후 데이터 자동 입력 (메인버튼의 // add에서도 입력 가능) // e.grid 그리드 객체, e.row.index 그리드 행 // 인덱스 e.grid.setCellValue(e.row.index, 'NATION_CD', 'KR'); e.grid.setCellValue(e.row.index, 'USE_YN', 'Y'); e.grid.setCellValue(e.row.index, 'USE_YN_M', 'Y'); e.grid.setCellValue(e.row.index, 'PARTNER_FG_CD', self.partner_fg_cd.value()); e.grid.setCellValue(e.row.index, 'PARTNER_CSF_CD', self.partner_csf_cd.value()); e.grid.setCellValue(e.row.index, 'OUT_PARTNER_YN', 'Y');  e.grid.setCellValue(e.row.index, 'CAPL_AMT', 0); e.grid.setCellValue(e.row.index, 'PR_DRCT_NOPL_CNT', 0); e.grid.setCellValue(e.row.index, 'MNG_EMP_CNT', 0); e.grid.setCellValue(e.row.index, 'CMNY_PRARNDE_CNT',0); e.grid.setCellValue(e.row.index, 'PAY_PRRG_CNT',0); e.grid.setCellValue(e.row.index, 'REGMA_YN', 'Y'); e.grid.setCellValue(e.row.index, 'BIZ_COND_FG', '1'); e.grid.setCellValue(e.row.index, 'PAY_DEF_R_YN', 'N');</pre>

TextBox (결과검색)	keyDown	<pre> if (event.keyCode == 13) {     var options = {         fields:         ['PARTNER_CD','PARTNER_NM','CEO_NM','BIZR_N O'],         value: self.search_tx.text(),         startIndex: self.Grid1.select() + 1     };     self.Grid1.searchCell(options); } </pre>
MainButton - add	click	<p>// TODO: 그리드에 데이터를 추가합니다.</p> <pre>self.Grid1.addRow();</pre>
MainButton - search	click	<p>// TODO: 데이터소스를 조회합니다.</p> <p>// 컨디션패널 내 폼컨트를 필수 항목 체크</p> <pre> if (self.searchCondition.validate({tooltip: true, message: "필수 항목을 입력해 주십시오."})) {     // 그리드 데이터소스 읽기     self.grid_ds.read(); } </pre>
MainButton - delete	click	<p>// TODO: 삭제시 강제 팝업에 대한 알림창을 끕니다.</p> <pre>self.mainButtons.delete.useDefaultConfirm = false;</pre> <p>// TODO: 그리드에서 체크된 항목을 삭제합니다.</p> <pre> if (self.Grid1.getCheckedIndex().length == 0) {     dews.alert('선택된 데이터가 없습니다.', 'warning');     return false; } self.Grid1.removeRow(self.Grid1.getCheckedIndex ()); </pre>

		<pre> // 변경된 데이터 유무 체크 if (self.grid_ds.getDirtyDataCount() == 0) {     dews.alert('변경된 데이터가 없습니다.');     return false; }  // 그리드 필수 입력 항목 체크 // mode : 변경된 행만 , showToolTip : 필수 // 입력 셀로 이동하고 필수 입력 툴팁 띄움 if (self.Grid1.validate({ mode: false, showTooltip: true }).result == false ) {     return false; }  // 저장 API 호출 (서비스 파라미터명과 // 데이터소스명이 다른경우) dews.confirm('데이터를 저장하시겠습니까 ?', 'question').yes(function () {     dews.api.post(dews.url.getApiUrl('MA', 'CommonInformationConfigurationPRTService', 'comprt00100_save'), {         async: false,         data: {             //ServiceParameterName :             ServiceParameterValue             datasource_1:             JSON.stringify(self.grid_ds.getDirtyData()),         }     }).done(function (data) {         dews.ui.snackbar.ok('자료가 정상적으로 저장되었습니다');         self.grid_ds.read();     }).fail(function (xhr, status, error) {// xhr :         XmlHttpRequest , status 는 HTTP 상태 텍스트         부분, error : 에러 내용         dews.ui.snackbar.error(error);     }); } </pre>
MainButton - save	click	

		<pre>}).no(function () {     dews.ui.snackbar.warning('저장이     취소되었습니다.'); });</pre>
--	--	--------------------------------------------------------------------------------------

<표 9-24. 빠른거래처 등록 이벤트 스크립트 정리>

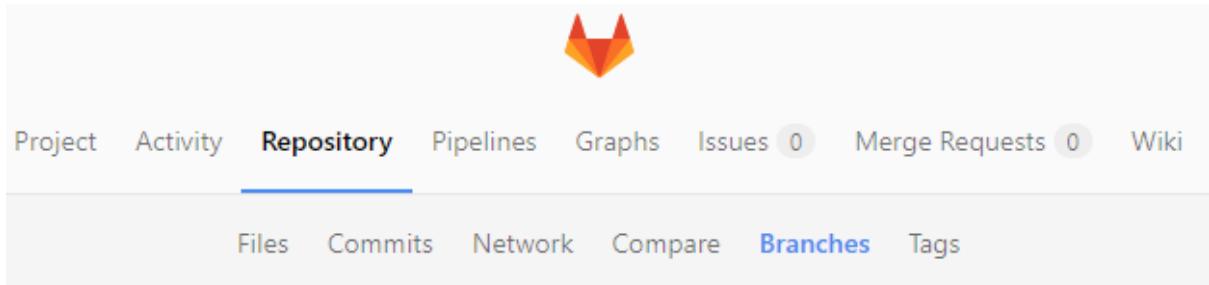
## [제10장] Git

Git 이란

Git 사용 예제

## 1. Git 이란

### 1) Git 개요



<그림 10-1. Git Web Server 화면>

Git이란 소스코드를 효과적으로 관리하기 위해 개발된 분산 버전 관리 시스템(DVCS)이며, 크게 백업기능과 협업기능을 한다. Git을 사용하면 소스 코드가 변경된 이력 확인하고 특정 시점에 저장된 버전을 비교할 수 있고, 특정한 시점으로 돌아갈 수 있다. DEWS Front Designer는 Git과의 연동이나 소스뱅크를 이용하여 통하여 소스 코드를 관리한다.

## 2. Git 사용 예제

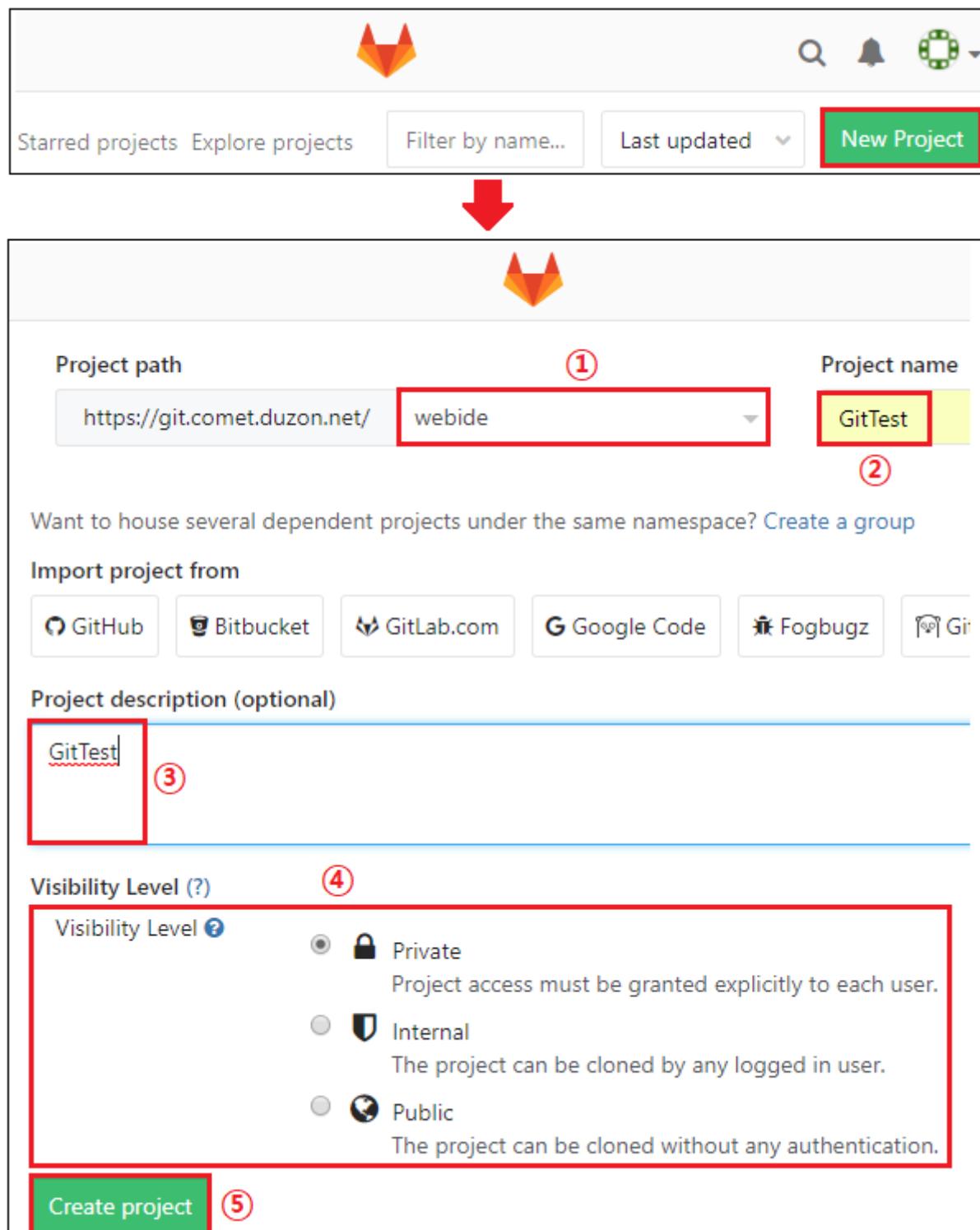
### 1) Git Web Server 프로젝트 생성

DEWS Front Designer은 DEWS에서 사용하는 원격 저장소(remote repository<sup>3</sup>)를 이용하며, 예제를 위해서 먼저 Git Web Server(<https://git.comet.duzon.net/>)에서 프로젝트를 생성한다.

홈에서 New Project 버튼을 누르면 New Project 생성 화면으로 넘어간다.

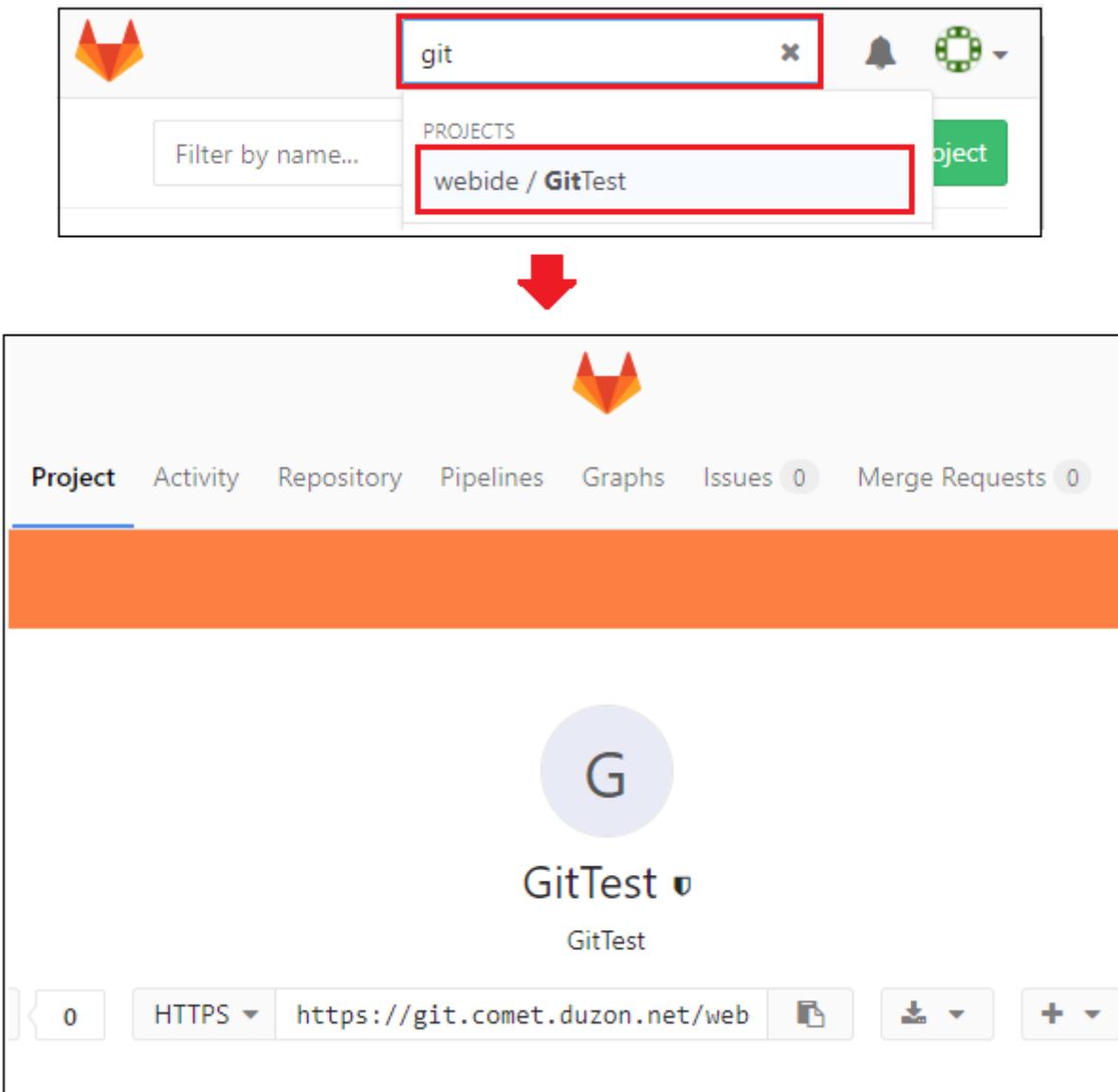
- ① Project path는 소속된 그룹 중에 Master 권한이 있거나 또는 사용자 계정을 선택
- ② Project Name 프로젝트 이름 입력
- ③ Project Description 프로젝트 상세 입력
- ④ 접근권한을 설정 ( Private : 명시적으로 지정된 사용들만 접근 가능, Internal : git에 로그인한 사용자들만 접근 가능, Public은 로그인하지 않아도 프로젝트 복사 가능)
- ⑤ Create Project를 눌러 생성한다.

<sup>3</sup> 파일이나 폴더를 저장해 두는 곳. 파일이 변경 이력 별로 구분되어 저장된다. 비슷한 파일이라도 실제 내용 일부 문구가 서로 다르면 다른 파일로 인식한다.



<그림 10-2. Git Web Server 프로젝트 생성>

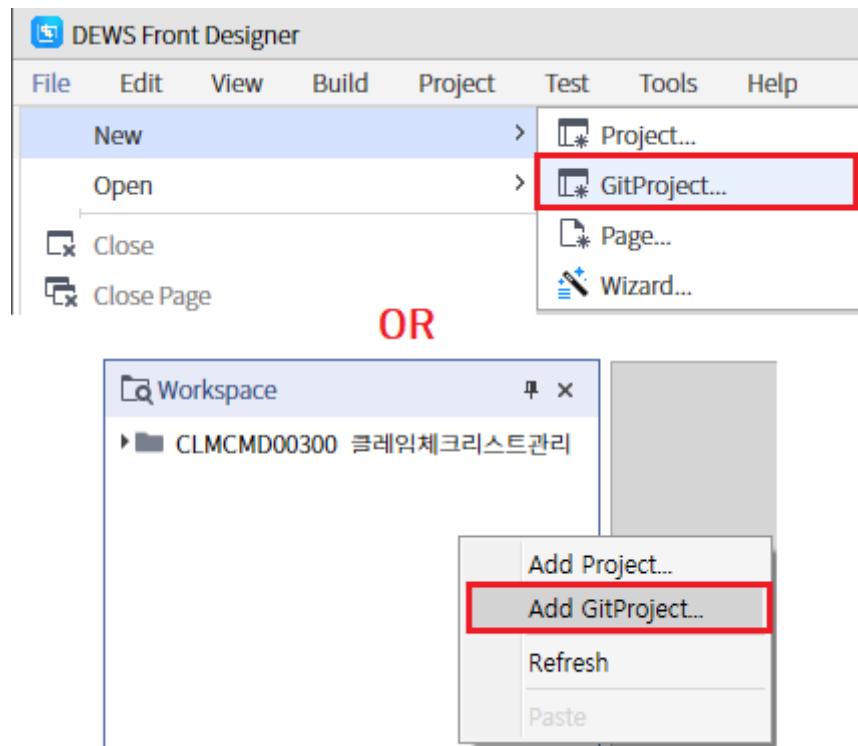
생성 후에 오른쪽 검색 입력란에 git 입력하면 자동으로 Project를 찾아주고 선택하면 프로젝트로 이동된다.



<그림 10-3. Git Web Server 프로젝트 검색>

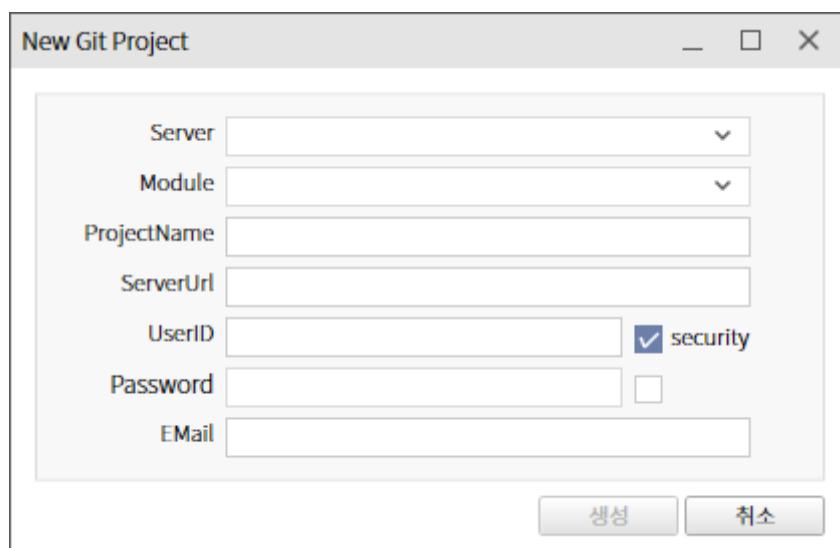
## 2) DEWS FD 프로젝트 생성

메뉴 Tools 의 Option > System Settings > IDE Options 옵션에서 Git 기능 체크 후 FD 를 재시작하고 메뉴의 File – New 에서 GitProject... 또는 workspace 의 빈 공간에서 우클릭하여 Add GitProject...을 선택한다.



<그림 10-4. New Git Project 추가>

New Git Project 화면이 열리며 항목들을 입력한다.



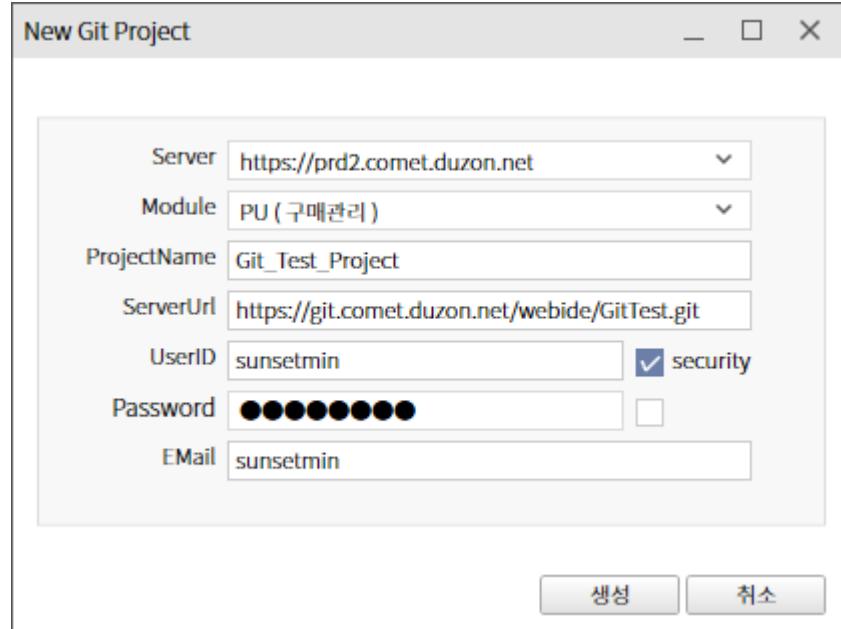
<그림 10-5. New Git Project 입력 항목들>

ServerUrl 항목은 Git Web Server 의 Project 탭에서 주소를 복사해서 입력한다.



<그림 10-6. Git Web Server URL 복사>

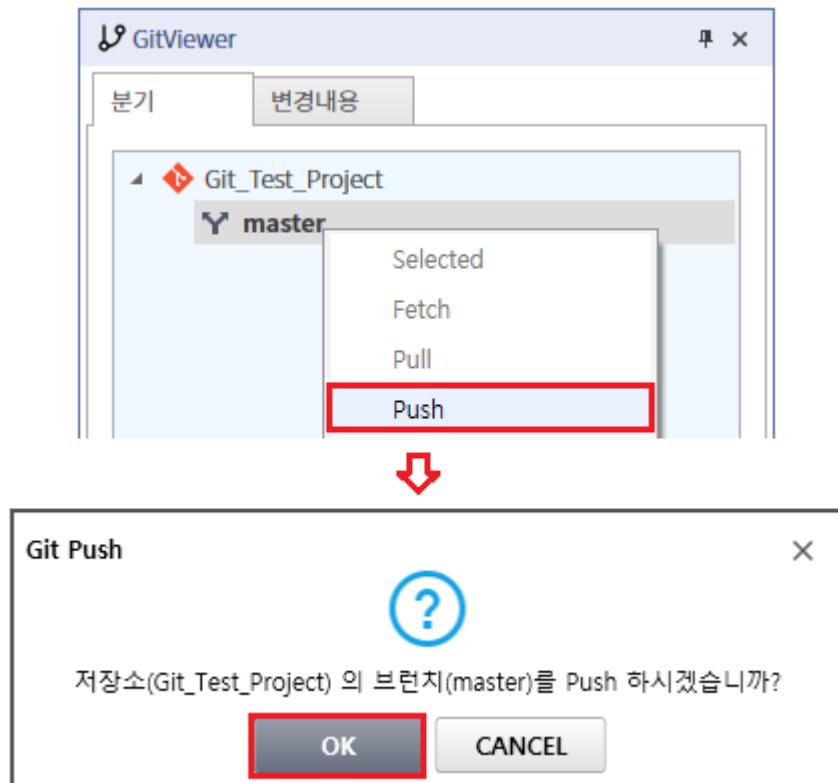
UserID 와 Password 는 Git 로그인 계정과 암호를 입력한다 나머지 항목들도 입력한다.



<그림 10-7. New Git Project 항목 입력>

<그림 10-8 과 같이>Workspace 에 일반 프로젝트와 다르게 파란색 열쇠 모양의 프로젝트 생성되고 프로퍼티에 GitViewer 탭을 클릭하면 master 가 보인다. 처음 Git Project 를 만들었을 경우 기본적으로 master 브랜치<sup>4</sup>가 생성되어 있다.

우선 통합브랜치<sup>5</sup> (Integration Branch)로 master 를 사용하기 위하여 <그림 10-8>과 같이 Push 를 한다.



<그림 10-8. master 브랜치 Push 화면>

<sup>4</sup> 분기 또는 가지라고 하며, 독립적으로 어떤 작업을 진행하기 위한 개념, 즉, 여러 개발자들이 동시에 다양한 작업을 할 수 있게 만들어 주는 기능이다. 필요에 의해 만들어지는 각각의 브랜치는 다른 브랜치의 영향을 받지 않기 때문에, 여러 작업을 동시에 진행할 수 있다. 기본적으로 Git은 master 브랜치를 만든다

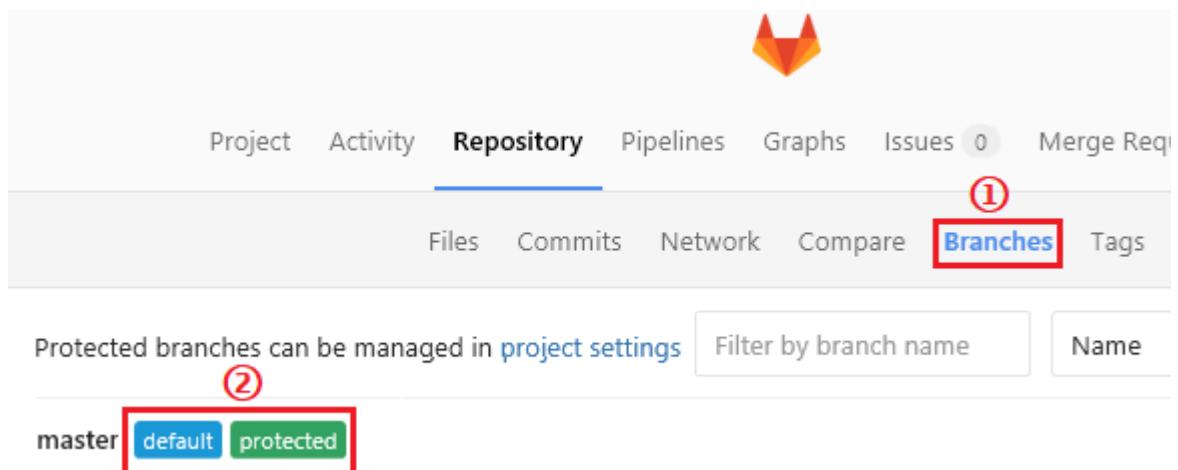
<sup>5</sup> 언제든지 배포할 수 있는 버전을 만들 수 있어야 하는 브랜치이며, 모든 기능이 정상적으로 동작하고 있는 상태가 유지해야 한다. 일반적으로 처음 저장소를 만들었을 때는 생성되는 master 브랜치를 통합 브랜치로 사용한다. 통합 브랜치는 Git 서버에서 default와 protected로 설정되어 있다.

<그림 10-9>와 같이 push 후 Git 프로젝트의 브랜치가 ① 접혀져 있으며 프로젝트를 펼쳐 보면 ② origin/master 브랜치도 같이 생성되어 있다. origin/master 는 원격(remote) 저장소의 master 원본 이력을 가져오며 로컬 저장소에서만 가지고 있는 정보이다. 삭제하더라도 Fetch(가져오기), Pull(끌어오기)이나 push(밀어넣기)할 경우 다시 생성된다.



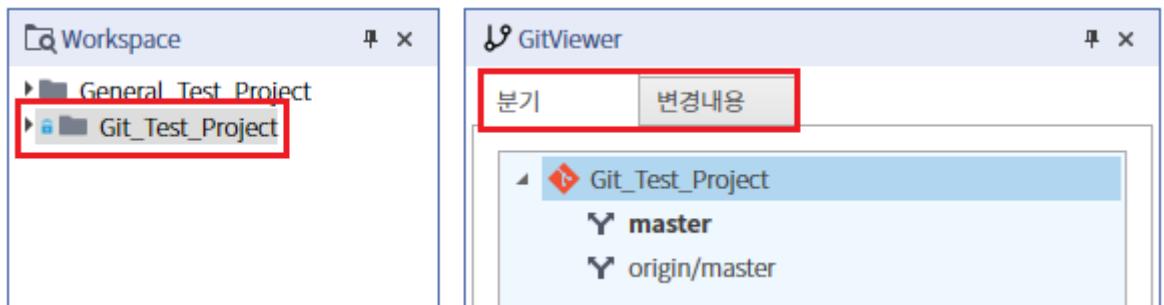
<그림 10-9. push 후 GitViewer 화면>

또한 Git Web Server 접속 후 <그림 10-10>과 같이 ① Branches 를 클릭하면 master 브랜치가 생성되어 있으며 ② default, protected(보호) 설정이 되어 있다



<그림 10-10. push 후 Git Web Server master 브랜치 화면>

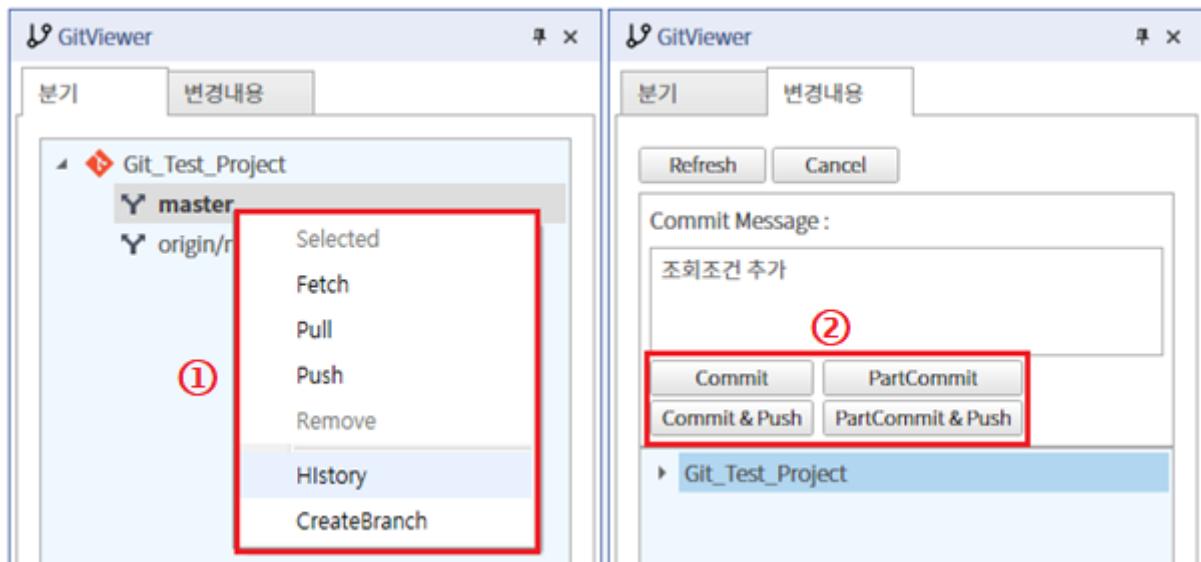
GitViewer 탭은 <그림 10-11>와 같이 분기와 변경내용 2개의 탭으로 구성되어 있다.



<그림 10-11. Workspace 및 GitViewer 분기 탭 화면>

분기 탭은 <그림 10-12>와 같이 브랜치 선택 후 우클릭 후 ① 표시되는 메뉴에서 브랜치 선택(Selected), 가져오기(Fetch), 끌어오기(Pull), 밀어넣기(Push), 삭제(Remove), 이력조회(History), 브랜치생성(CreateBranch) 등 다양한 작업을 할 수 있다.

변경내용 탭은 주로 작업한 내용을 입력하고 ② Commit<sup>6</sup>, PartCommit<sup>7</sup>, Commit&Push, PartCommit & Push을 하거나 새로고침(Refresh), 작업한 내용을 취소(Cancel) 한다.

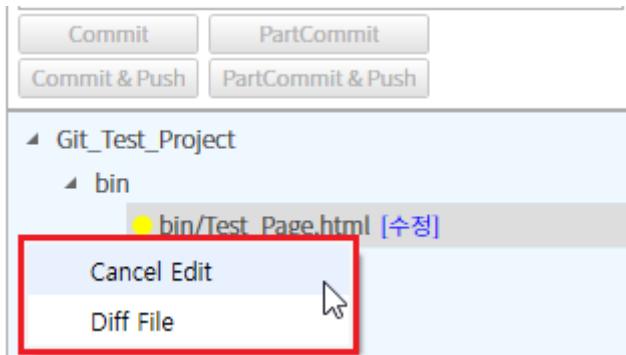


<그림 10-12. GitViewer 변경내용 화면>

<sup>6</sup> 파일 및 폴더의 추가/변경 사항을 저장소에 기록

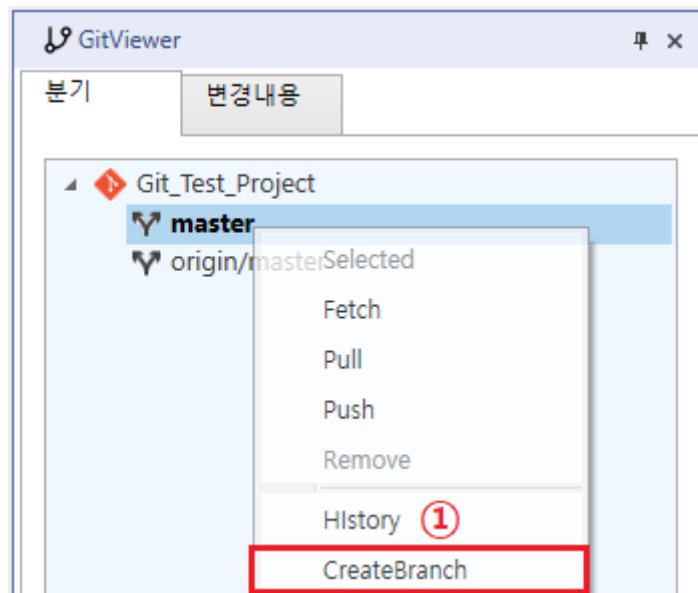
<sup>7</sup> ewp 파일(FD 개발 파일)과 html, ewd(설계 문서 파일) 중 선택 commit

프로젝트에서도 <그림 10-13>처럼 ewp, ewd, html 파일을 선택하고 우클릭 후 메뉴에서 작업한 내용을 취소(Cancel Edit)하거나 버전을 비교(Diff File)할 수 있다



<그림 10-13. 파일 선택 후 우클릭 메뉴 화면>

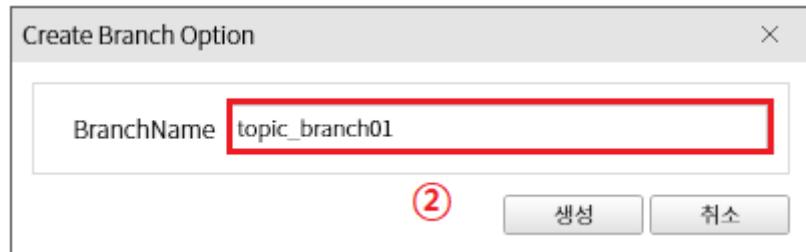
기본적으로 생성된 master 브랜치는 통합브랜치 (Integration Branch)로 사용하고 단위작업을 위한 토픽 브랜치<sup>8</sup>(Topic Branch)를 만든다. <그림 10-14>와 같이 GitViewer에서 master 우클릭 후 나오는 메뉴에서 CreateBranch 을 선택한다.



<그림 10-14. Create Branch>

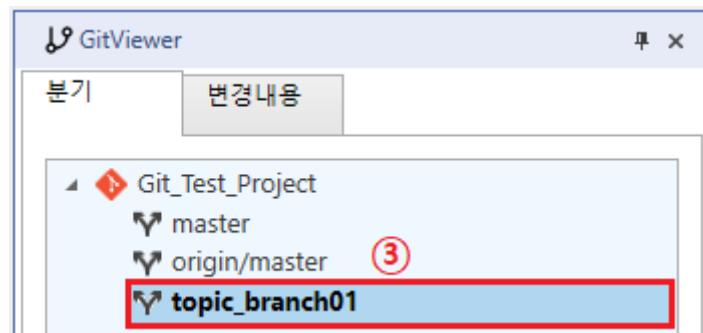
<sup>8</sup> 기능 추가나 버그 수정과 같은 단위 작업을 위한 브랜치. 보통 통합 브랜치로부터 만들어 내며, 토픽 브랜치에서 특정 작업이 완료되면 다시 통합 브랜치에 병합하는 방식으로 진행한다.

브랜치 이름을 topic\_branch01 를 입력하고 생성 버튼을 누른다.



<그림 10-15. Branch Name 입력>

topic\_branch01 가 생성되며, 자동으로 선택(체크아웃<sup>9</sup>)된다. 만약 생성한 브랜치로 자동 선택이 되지 않은 경우 master 에 변경된 내용이 있어 Commit 을 하거나 작업을 취소(Cancel Edit)하면 Selected 가 활성화 된다.

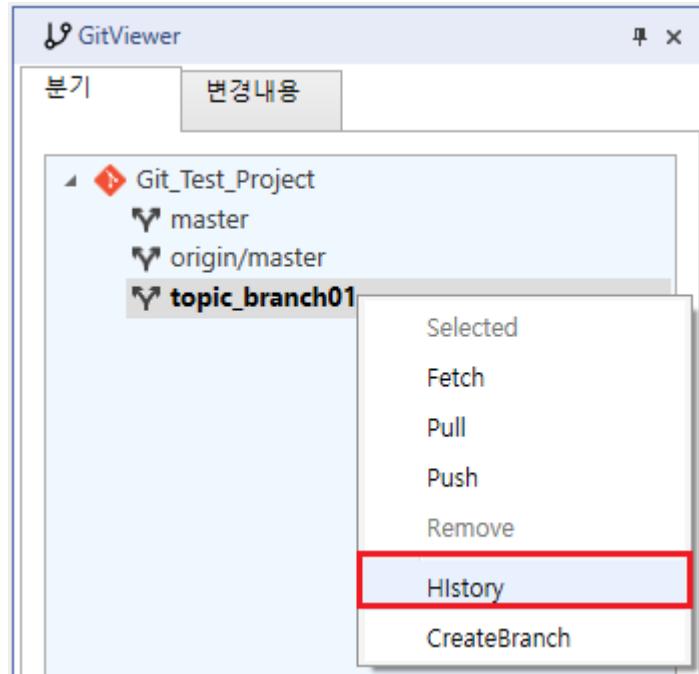


<그림 10-16. topic\_branch01 생성>

---

<sup>9</sup> 체크아웃(checkout)은 현재 선택된 브랜치가 아닌 다른 브랜치에서 작업하고 싶을 때 사용

<그림 10-17>처럼 생성된 topic\_branch01 우클릭 메뉴에서 History를 실행한다.



<그림 10-17. History 메뉴 실행>

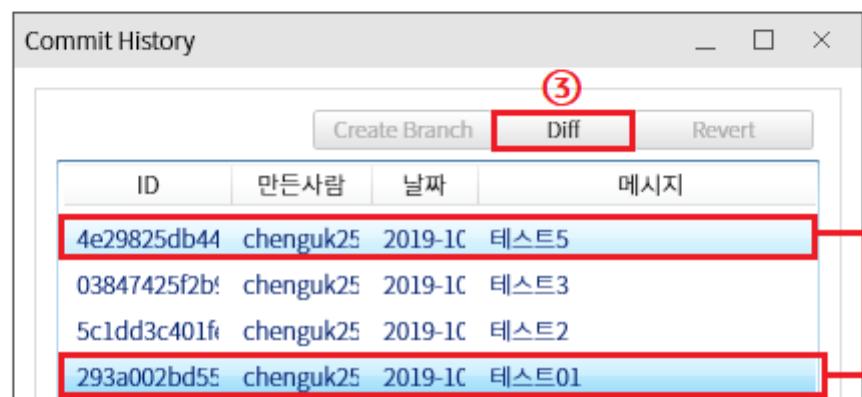
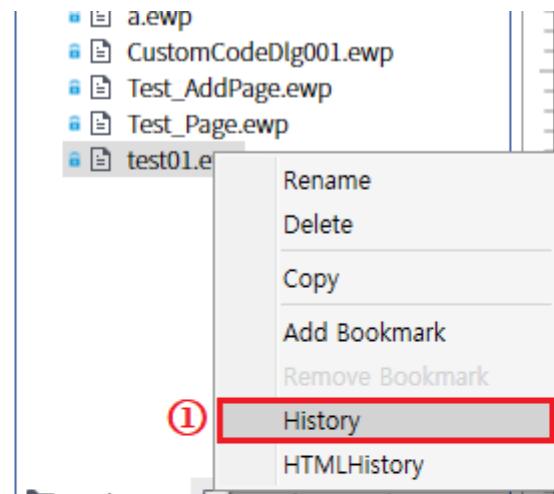
<그림 10-18>와 같이 Commit History 화면이 열리며 작업 변경 이력(버전)과 내용을 확인할 수 있다. master에서 생성된 브랜치이므로 master와 같은 내역을 가진다.

Create Branch는 왼쪽에서 선택한 버전으로 새 브랜치를 생성한다. Diff는 버전 간 변경된 내용을 비교하며, 분기탭은 여러 파일(ewp, ewd, html)의 History가 동시에 보여주므로 비활성화 되며, Revert 되돌리기이며 이전 버전의 commit을 취소한 내용(Revert으로 새 버전으로 만든다.

버전 비교는 <그림 6-19>와 같이 Workspace 구성요소에서 프로젝트의 페이지(ewp 파일) 선택하고 우클릭 후 History나 HTMLHistory에서 반드시 2개 버전을 선택해야 Diff 버튼이 활성화된다. 버튼을 눌러서 버전을 비교한다.

Commit History				
				Create Branch
ID	만든사람	날짜	메시지	Diff
038b8c711c813	sunsetmin	2019-02-2	555	
db926a994e674	강상민	2019-02-2	Merge branch 'revert-eb5afe58' into 'master'	
400a6afa9c532	강상민	2019-02-2	Revert "Merge branch 'topic_branch01' into '	Revert

<그림 10-18. Commit History 화면>



```

293a002bd5536e00d4bfb694890f8bf3f7f7da: 4e29825db4473bba5475e2753640c024df4b

<?xml version="1.0" encoding="utf-8"?> <?xml version="0" encoding="utf-8"?>
<page title="test01" name="test01" type="page" <page title="t@1" na@="test01" type="page">
<canvas uid="canvas-1485142915100" type="<canvas uid='nvas-85142915100' type="page">
</conditionItem> </conditioem>

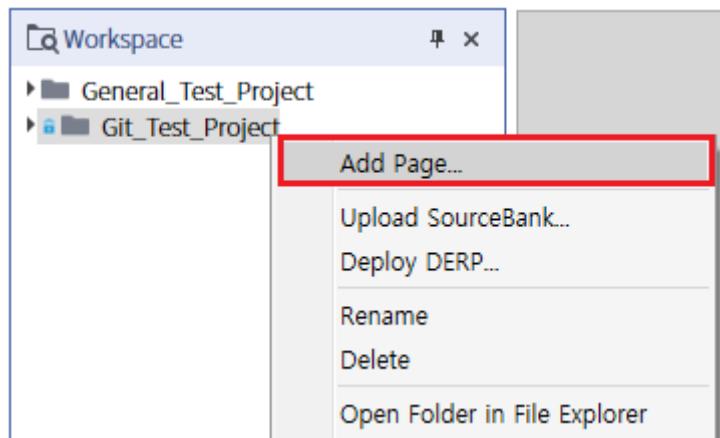
<conditionItem uid="conditionItem-15711157" <conditiom uicconditionItem-15711157
<button uid="button-156766920046" <button uid='button-156766920046'
</conditionItem> </conditiom uicconditionItem-15711157

<conditionItem uid="conditionItem-15711157" <conditiom uicconditionItem-15711157
<codepicker uid="codepicker-157111578" <codepicid uid='codepicker-157111578666'>
</conditionItem> </conditiom uicconditionItem-15711157

```

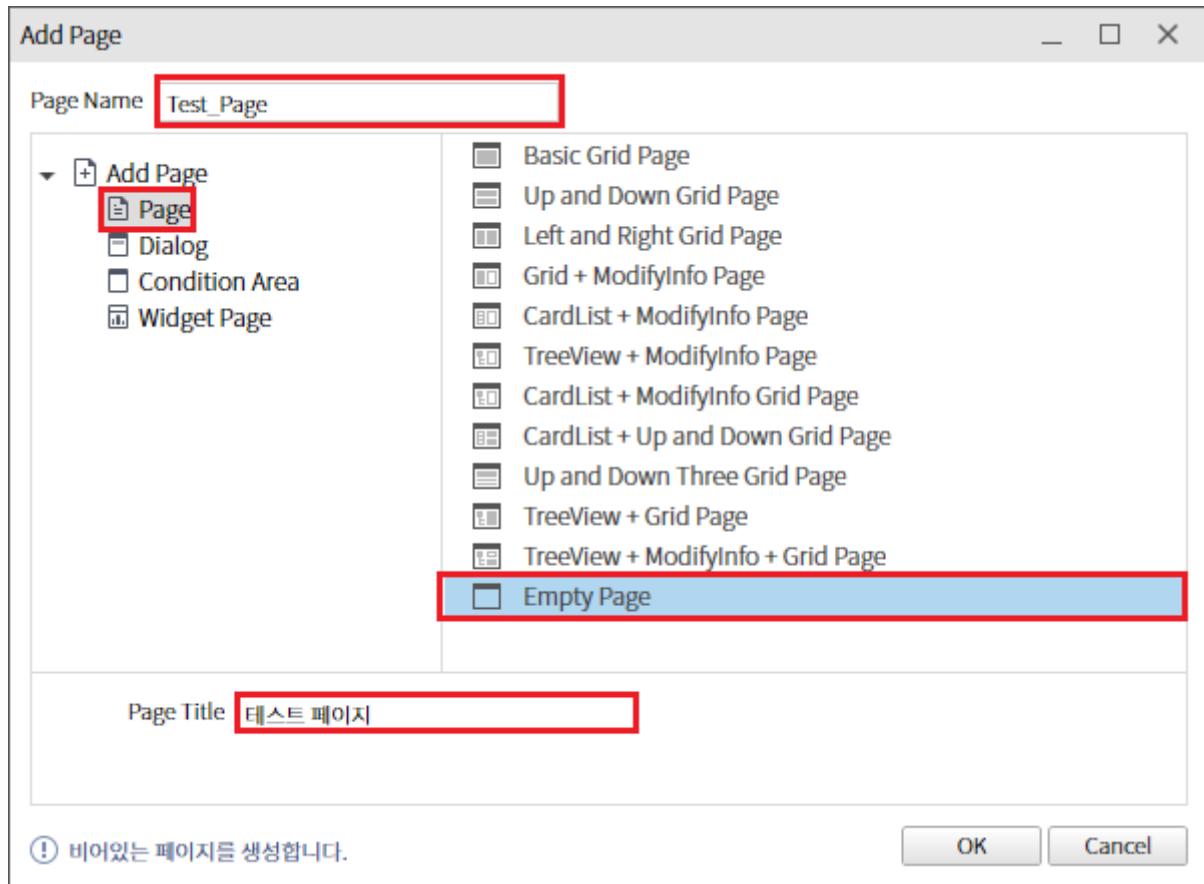
<그림 10-19. Workspace에서 Diff 실행 화면>

지금까지 Git 메뉴에 대해 살펴보았고 새 페이지를 추가하기 위하여 프로젝트를 선택하고 우클릭하여 Page 를 추가한다.



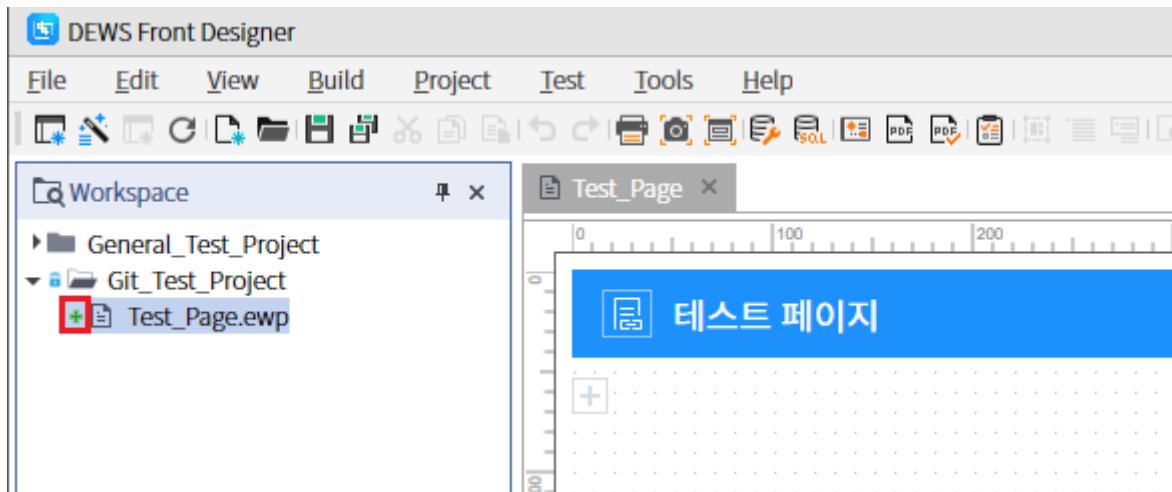
<그림 10-20. Add Page 실행>

Page Name 은 Test\_Page 를 입력하고 아래 왼쪽은 Page, 오른쪽 레이아웃은 Empty Page 를 선택하고 Page Title 은 테스트 페이지를 입력한다.



<그림 10-21. Add Page 항목 입력>

Workspace 영역에서 보면 Test\_Page.ewp 페이지 파일과 왼쪽에 + 아이콘이 생성된다.



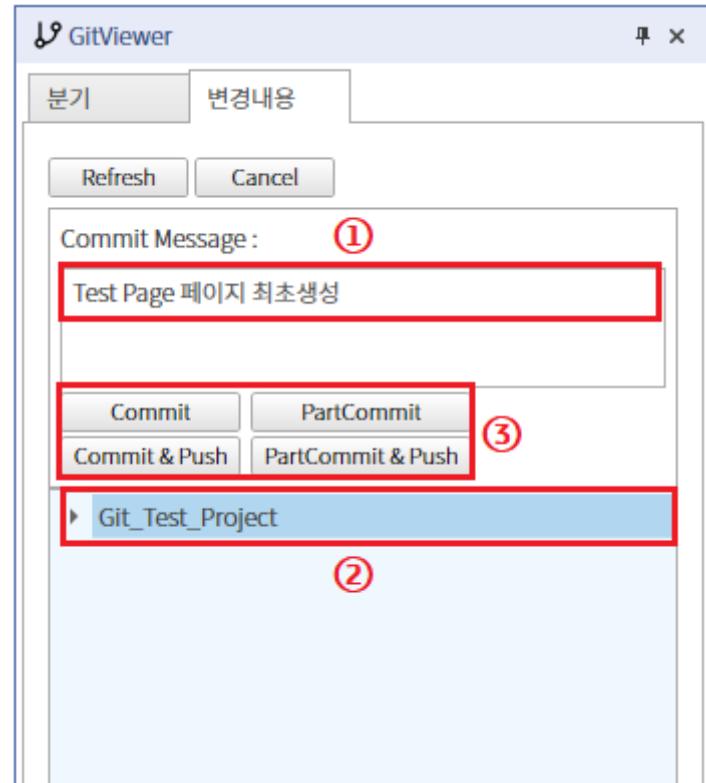
<그림 10-22. Test Page 생성>

페이지에서 +은 파일 추가, ✓은 수정사항이 있을 경우, 🔍은 수정사항이 없을 경우 표시되는 아이콘이다.

GitViewer의 변경내용 탭으로 가서 변경 사항을 기록하고 Commit 한다.

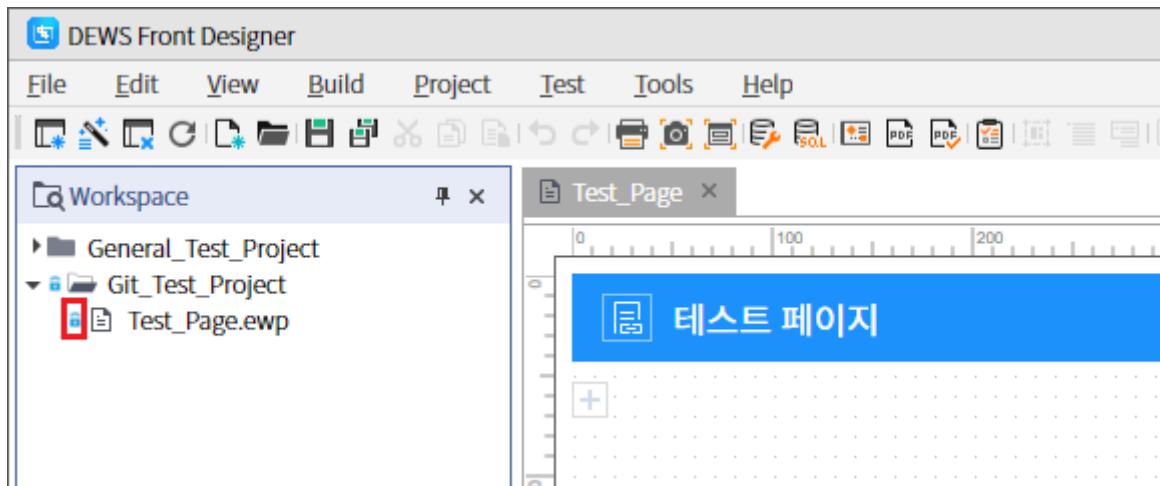
- ① Commit Message에서 추가/변경 사항을 입력
- ② 프로젝트를 선택(선택하지 않으면 Commit 버튼이 비활성화)
- ③ Commit 버튼을 눌러 전체 변경 사항을 기록하거나 PartCommit을 눌러 선택해서 변경사항을 기록하거나, Commit & Push를 눌러 Commit 동시에 Push(밀어넣기)하거나, PartCommit & Push를 눌러 선택 변경 사항 기록 및 Push(밀어넣기) 등의 작업을 할 수 있다

화면을 다시 불러올 때는 Refresh 버튼을 누르고, 진행한 작업을 취소할려면 Cancel을 누른다



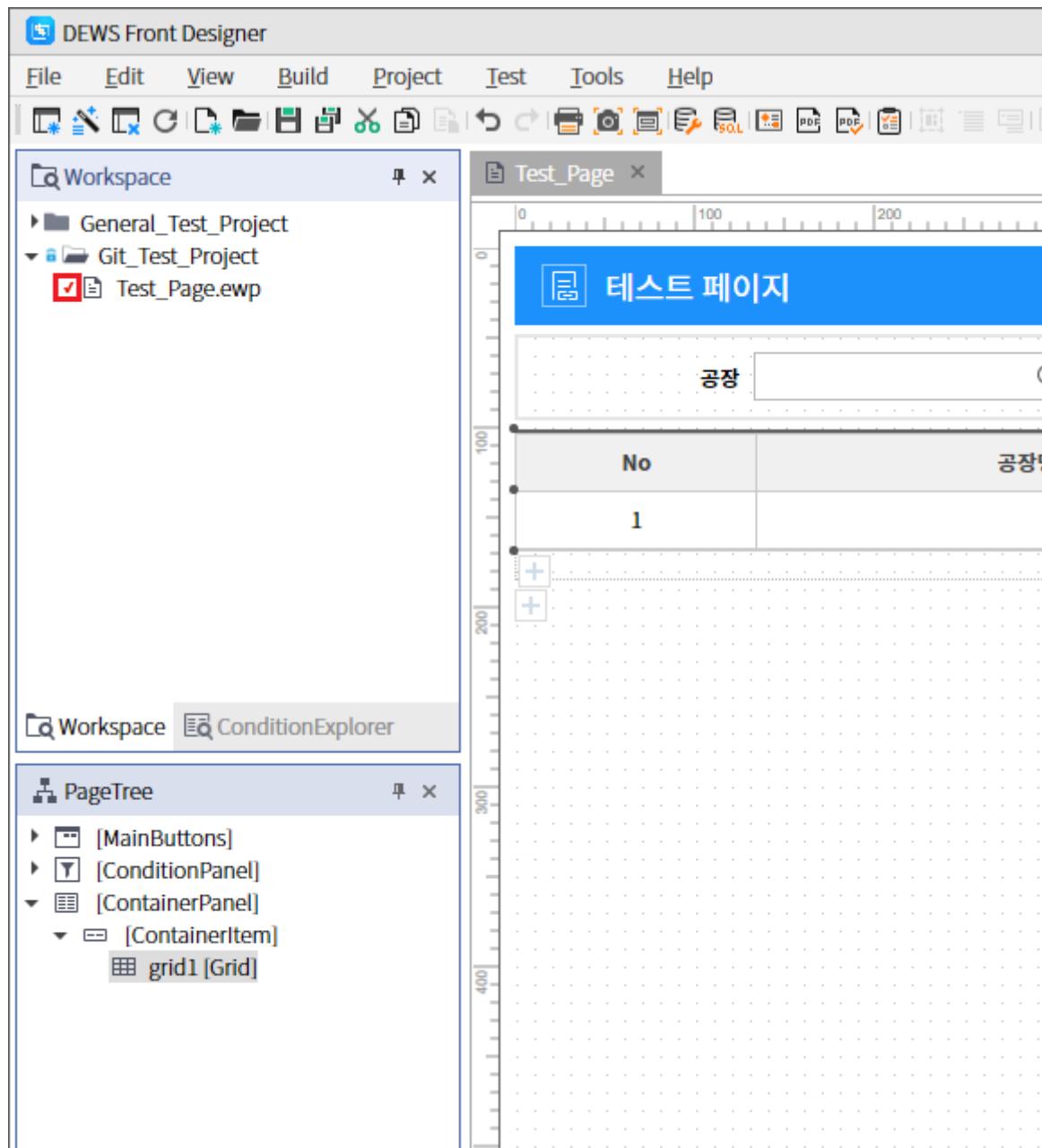
<그림 10-23 Commit 순서>

페이지 앞의 아이콘 모양이 으로 변경되었다.



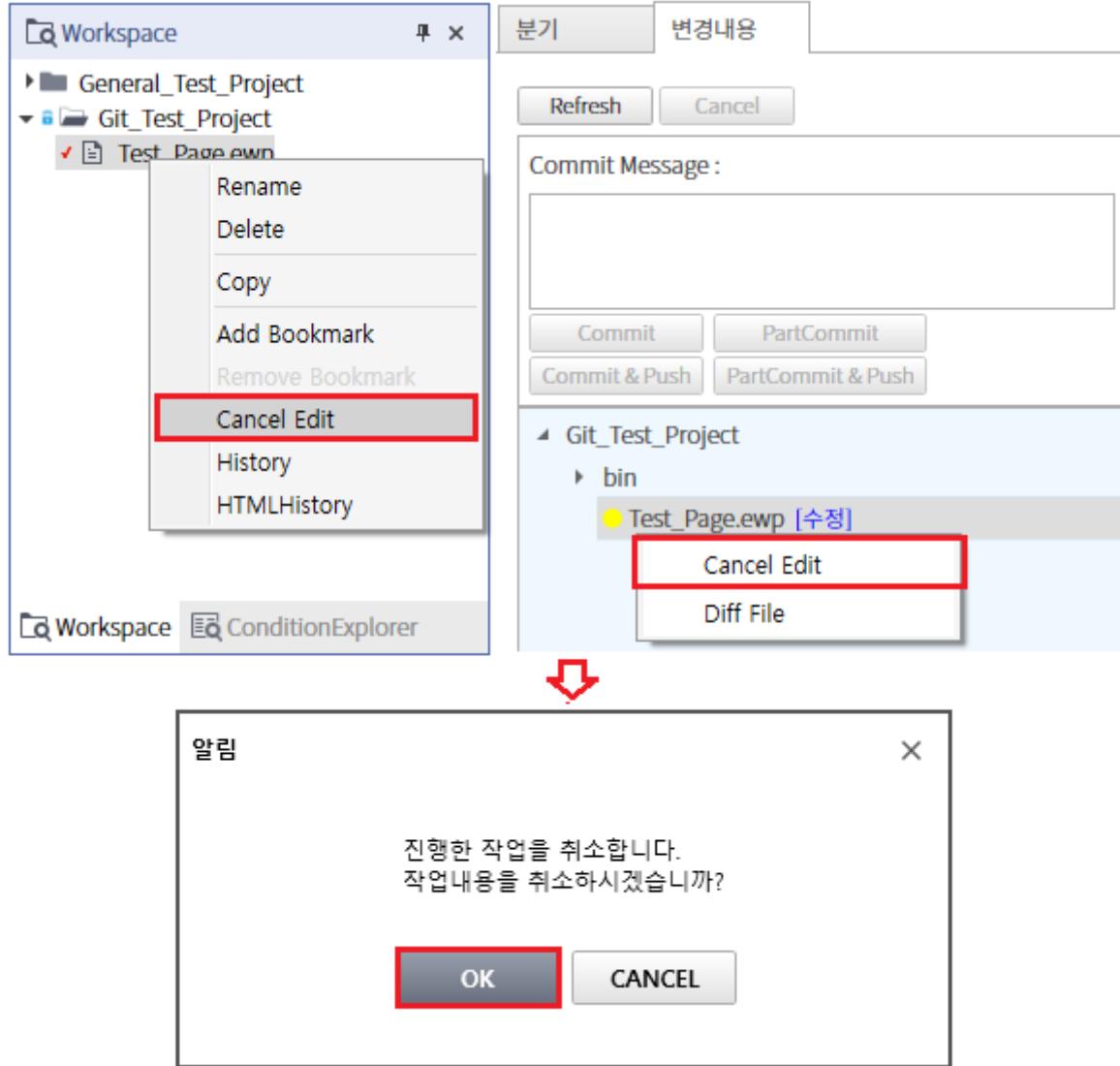
<그림 10-24. 수정사항 없음 아이콘 모양>

화면과 데이터소스를 만들고 Build 및 미리보기를 진행하여 메뉴를 완성한다. <그림 10-25>와 같이 페이지 앞에 아이콘 모양이 ↗으로 수정사항이 발생된 것을 확인할 수 있다.



<그림 10-25. 수정사항 있음 아이콘 모양>

만약 작업한 내용을 취소(되돌리기)할 경우 Workspace에서 페이지를 선택하고 우클릭 후 Cancel Edit 를 눌러 취소하거나, GitViewer의 변경내용 탭에서 파일(epw, ewd, html)을 선택하고 우클릭 후 Cancel Edit 를 눌러 취소한다. Workspace는 epw 파일에 작업한 내용을 취소한다.

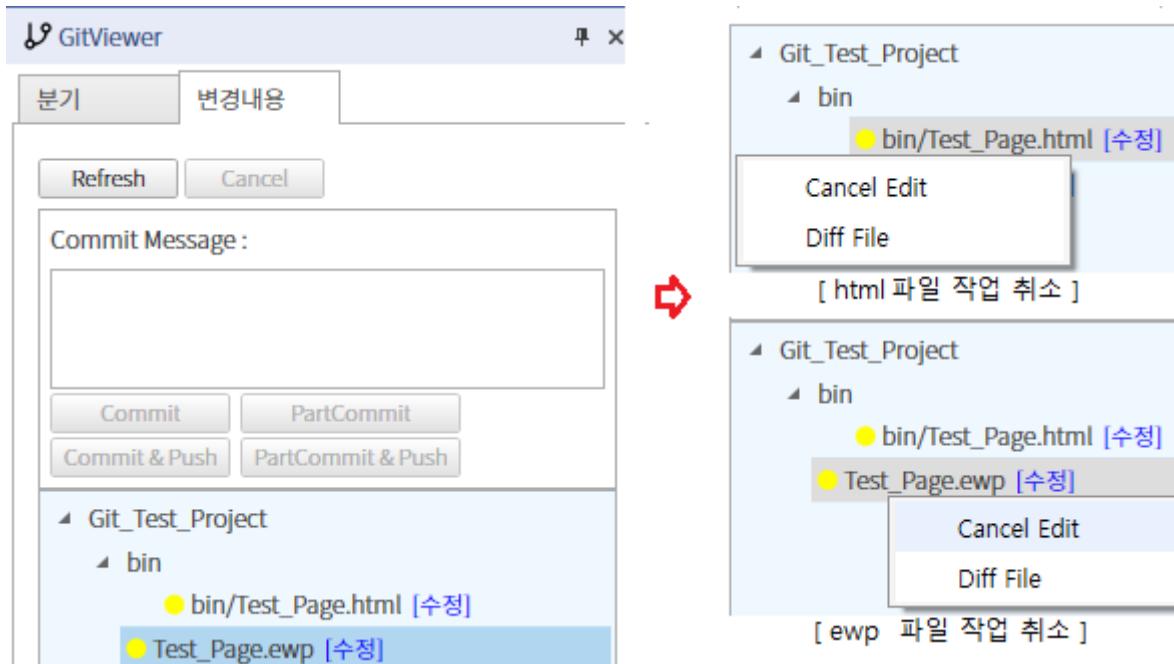


<그림 10-26. Cancel Edit 메뉴>

또한 우클릭 나오는 메뉴 중에 History 는 epw 파일 변경 이력을, HTMLHistory 는 html 파일 변경 이력을 확인할 수 있다.

Commit History에서 버전을 선택하고 Create Branch하거나 2개 버전을 선택하고 Diff를 눌러 버전 간 변경된 내용을 비교할 수 있다.

GitViewer 탭의 변경내용을 펼쳐보면 bin 안에 html 파일과 epw 및 ewd 파일이 있으며 선택 후 우클릭 메뉴에서 Cancel Edit를 실행하여 개별적으로 작업을 취소할 수 있다.



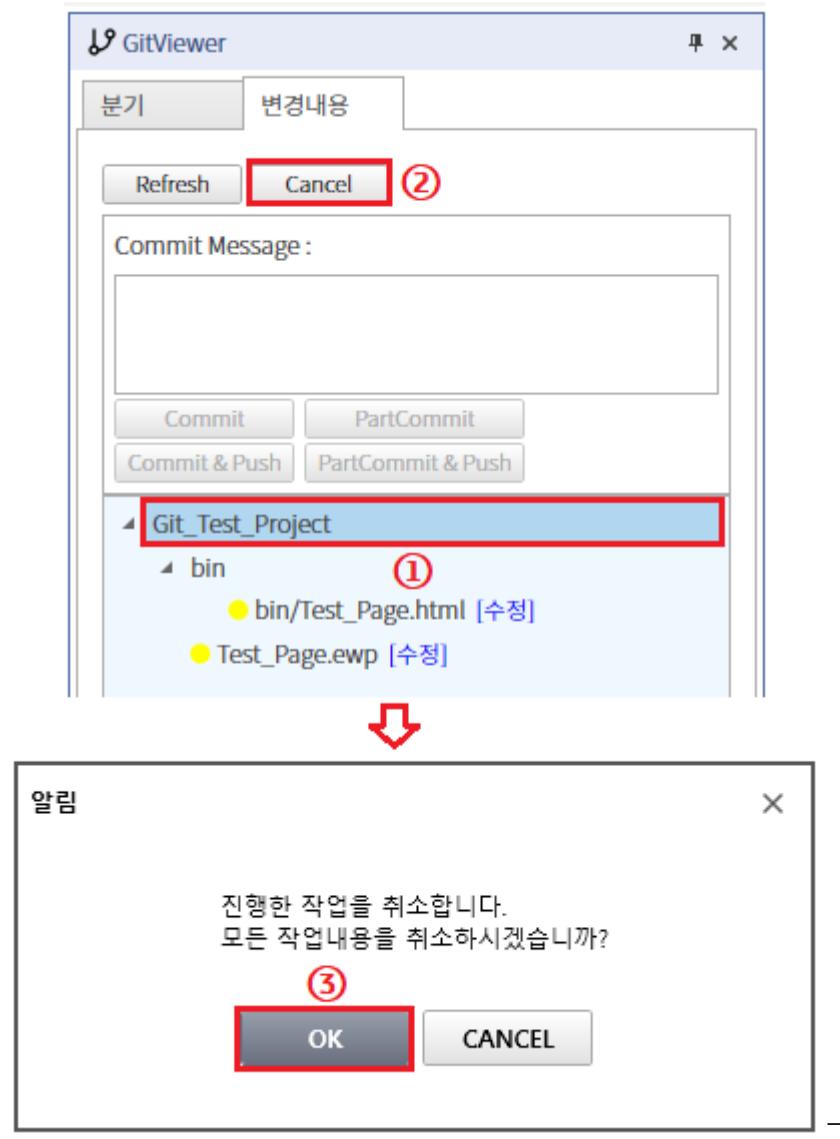
<그림 10-27. 개별 파일 작업 취소>

메뉴 중 Diff File 을 선택하면 변경된 사항들을 색상으로 확인할 수 있으며, ◀ 또는 ▶ 버튼을 눌러 이전 변경 사항이나 다음 변경 사항을 찾아간다.



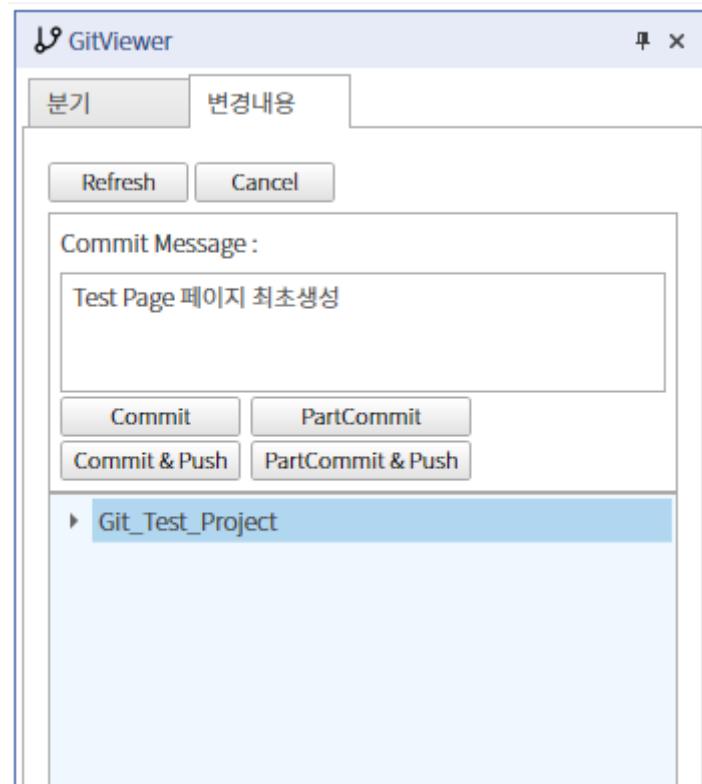
<그림 10-28. Diff 실행>

ewp 와 ewd, html 모두 작업 취소(전체 취소) 할려면 ① 프로젝트를 선택 후 ② Cancel 버튼을 누르고 ③ OK 를 누른다.



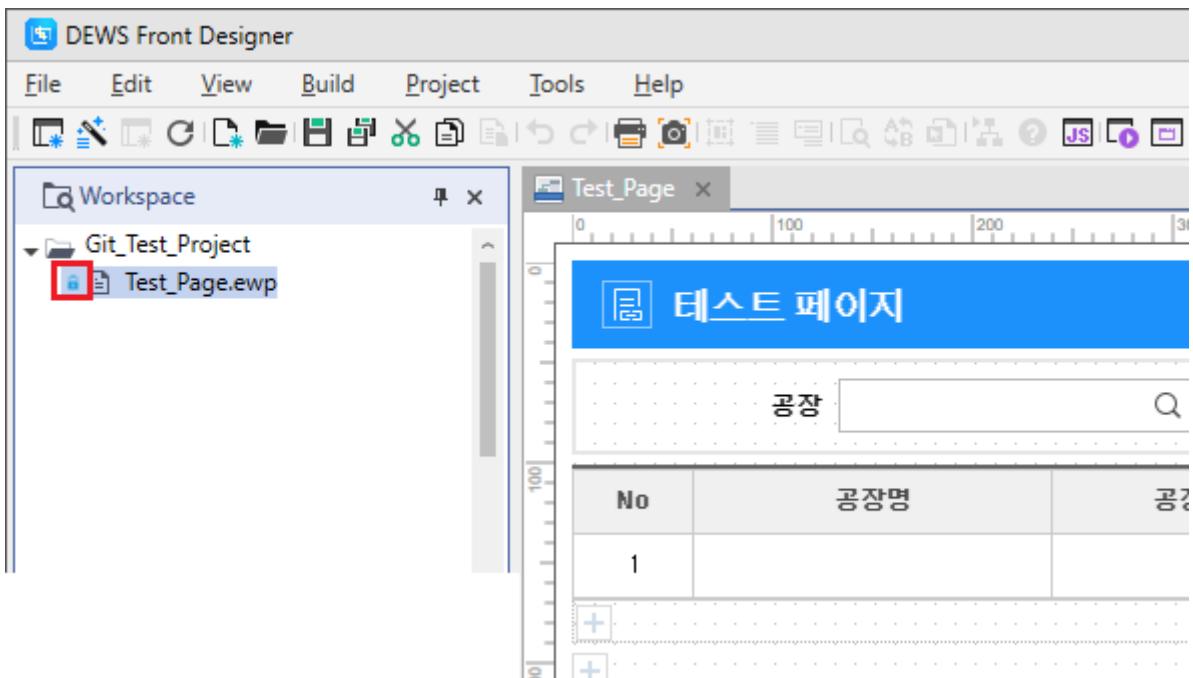
<그림 10-29. 전체 작업 취소>

이제 최종적으로 GitViewer 의 변경내용 탭으로 가서 변경 사항을 기록하고 Commit 한다.



<그림 10-30. 최종 commit 실행>

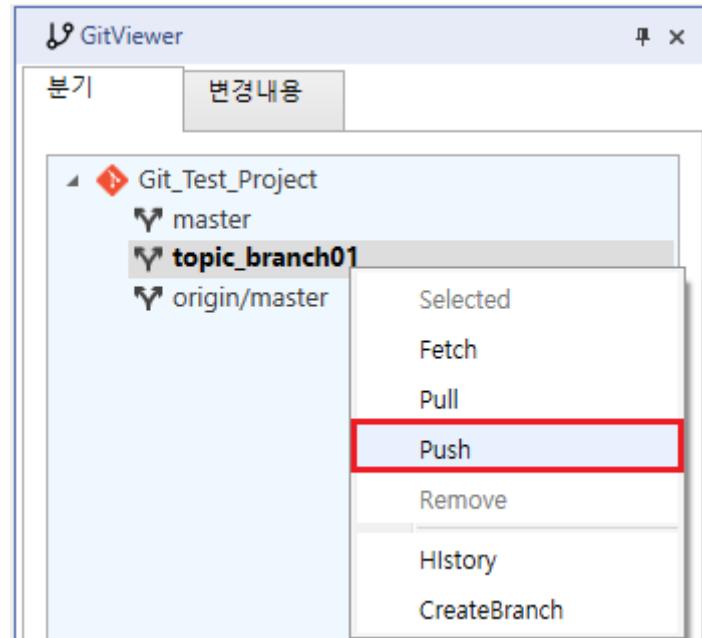
페이지 아이콘 모양이 에서 으로 변경된다.



<그림 10-31. 수정사항 없음 아이콘 모양으로 변경 >

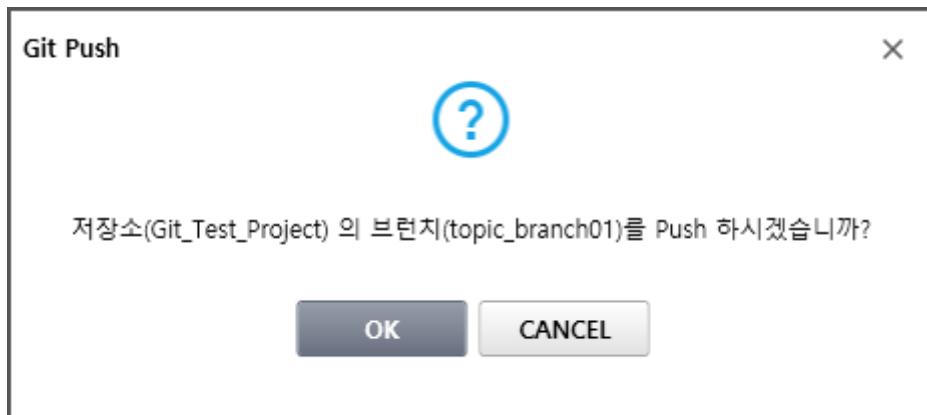
이제 생성한 topic\_branch01 을 원격 저장소에 밀어넣기(Push)을 한다. master 브랜치에 병합은 로컬 저장소에서 가능하나 master 의 권한이 없는 경우 병합 요청(Merge Request)을 진행해야 하므로 Git Web Server 에서 진행한다.

topic\_branch01 을 선택하고 우클릭 메뉴에서 Push 를 선택한다.



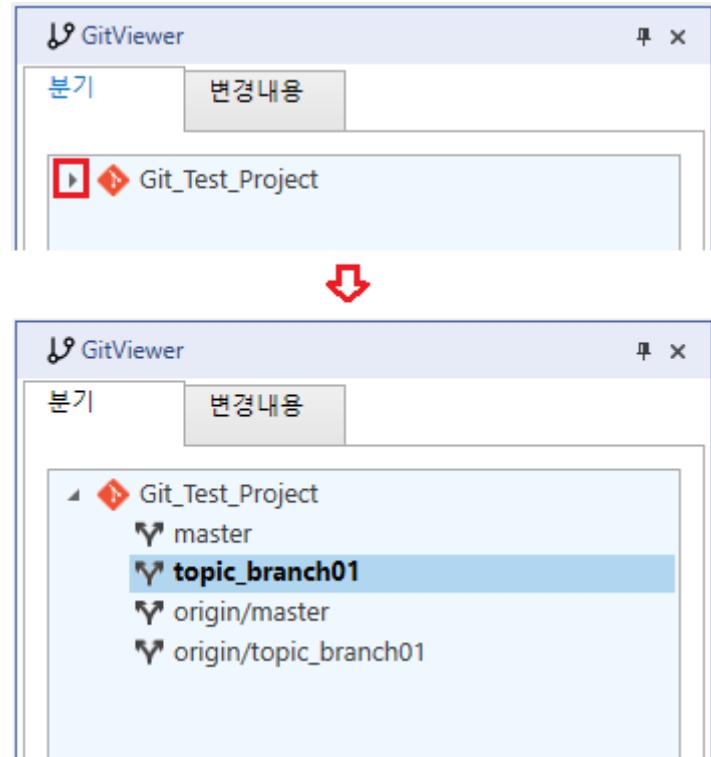
<그림 10-32. Push 메뉴 실행>

Push 확인 메세지창이 열리면 OK 를 누른다.



<그림 10-33. Push 메세지>

GitView 분기가 닫혀있으면, ▶을 눌러 펼쳐보면 4 개 브랜치가 보이며, origin/topic\_branch01 은 origin/master 와 마찬가지로 원격 저장소(remote repository) 의 topic\_branch01 원본 이력을 가져오며 로컬 저장소에서만 가지고 있는 정보이다.



<그림 10-34. GitView 브랜치 확인>

Git Web Server 를 확인해보면 저장소(Repository)의 브랜치(Branches)에 master 와 topic\_branch01 등 2 개 브랜치가 있다.

The screenshot shows the GitLab interface for the project 'webide / GitTest'. The 'Repository' tab is selected. Below it, the 'Branches' tab is highlighted with a red border. The page displays two branches: 'master' and 'topic\_branch01'. The 'master' branch is marked as 'protected' and has a green button next to it. The 'topic\_branch01' branch has a red box around it. A red circle labeled '①' points to the 'protected' button on the master branch, and another red circle labeled '②' points to the 'topic\_branch01' branch name. A message at the top says 'Protected branches can be managed in project settings'. On the right side, there are buttons for 'Delete merged branch', 'Merge Request', and 'Compare'.

<그림 10-35. Git Web Server Branch 확인>

### 3) Git Web Server 병합 요청

DEWS FD 에서 push(밀어넣기)한 프로젝트는 Git Web Server 에서 메세지로 확인할 수 있다.

The screenshot shows the GitLab interface for the project 'webide / GitTest'. The 'Project' tab is selected. A prominent orange message box at the top states: 'You won't be able to pull or push project code via SSH until you [add an SSH key](#) to your profile'. Below this, there are two links: 'Don't show again' and 'Remind later'. At the bottom of the screen, there is a message box containing the text 'You pushed to topic\_branch01 about an hour ago' and a blue button labeled 'Create Merge Request'.

<그림 10-36. Git Web Server push 메세지 확인 >

Git Web Server 화면 오른쪽에 Create Merge Request 버튼이 있으며 이 버튼을 눌러 통합 브랜치인 master에 병합요청을 한다.

New Merge Request 페이지에서 ① Title 입력, ② 상세 설명, ③ 담당자, ④ 프로젝트가 도달해야 하는 목표 지점, ⑤ 이슈의 종류 태그, ⑥ 병합할 브랜치, ⑦ 병합 대상 브랜치, ⑧ 병합 요청이 승인 후 원본 브랜치 삭제 여부 등의 정보들을 입력 또는 선택하고 마지막으로 ⑨ 병합 요청 버튼을 누른다.

### New Merge Request

From `topic_branch01` into `master`

Title ① Topic branch01  
Start the title with `WIP:` to prevent a **Work In Progress** merge request if you're still working on it.  
Add [description templates](#) to help your contributors communicate effectively.

Description ② Merge Topic\_branch into the Master branch.  
Styling with [Markdown](#) and slash commands are supported.

Assignee ③ 강상민

Milestone ④ Milestone

Labels ⑤ TestLabel

Source branch ⑥ topic\_branch01

Target branch ⑦ master [Change branches](#)

⑧  Remove source branch when merge request is accepted.

⑨ [Submit merge request](#)

<그림 10-37. 병합 요청 항목 입력>

메인 메뉴 Merge Requests 를 보면 1 개가 있고, 하위 Merge Requests 메뉴 밑에는 Open, Merged, Closed, All 등 4 개로 분류되어 있다. 처음 병합 요청은 Open 에 위치한다. Merged 는 병합된 내역, Closed 는 요청했으나 취소된(닫은) 내역, All 은 모든 내역을 볼 수 있다.

The screenshot shows the GitLab interface for the project 'GitTest'. At the top, there's a navigation bar with links for Project, Activity, Repository, Pipelines, Graphs, Issues (0), Merge Requests (1), Wiki, and a gear icon. The 'Merge Requests' link is highlighted with a red box. Below the navigation is a secondary navigation bar with tabs for Merge Requests, Labels, and Milestones, where 'Merge Requests' is also highlighted with a red box. Underneath these bars, there are four buttons for filtering: Open (1), Merged (7), Closed (1), and All (9), with 'Open' being the active filter. To the right of these buttons are 'Filter by name ...' and a green 'New Merge Request' button. Further down, there are dropdown menus for 'Author' and 'Assignee', and a sorting dropdown set to 'Last created'. A specific merge request is highlighted with a red box: 'Topic branch01' (ID 13), opened about an hour ago by '강상민' (TestLabel), and updated about an hour ago. The GitLab logo is visible at the top center.

<그림 10-38. Merge Requests 메뉴>

master 또는 Owner 권한이 있을 경우 Title 을 클릭하면 상세 내용을 볼 수 있으며 master 권한이나 Owner 인 경우 병합 요청을 ① 승인(Accept)을 하거나, ② 또는 ②'을 눌러 취소(Close)를 할 수 있다. Close 한 경우라도 다시 열어(ReOpen) 승인할 수 있다.

The screenshot shows a GitHub merge request interface. At the top, there is a summary card for a merge request titled 'Topic branch01'. The card includes the title, a timestamp ('!13 · opened about an hour ago by 강상민'), a label ('TestLabel'), and an update timestamp ('updated about an hour ago'). A red arrow points down from this card to the main merge request page.

The main page displays the merge request details:

- Topic branch01**: The title of the merge request.
- Merge Topic\_branch into the Master branch.**: The description of the merge request.
- Edited about 4 hours ago**: The last edit timestamp.
- Request to merge topic\_branch01 into master**: The main action button.
- Check out branch** and **Download**: Buttons for managing the branch.
- Accept Merge Request**: A green button with a red border and a circled '1' above it, indicating the first step of accepting the request.
- Remove source branch**: An unchecked checkbox.
- Modify commit message**: A checked checkbox.
- You can also accept this merge request manually using the command line.**: A note for manual acceptance.
- Discussion 0**, **Commits 3**, **Changes 2**: Navigation links for the merge request.
- Write** and **Preview**: Buttons for writing comments.
- B I " </> [list icons]**: Styling tools for Markdown and slash commands.
- Write a comment or drag your files here...**: The comment input field.
- Styling with Markdown and slash commands are supported**: A note about supported styling.
- Attach**: A button for attaching files.
- Comment** and **Close merge request**: Buttons at the bottom of the comment section.

<그림 10-39. 병합 요청 승인>

승인(Accept Merge Request) 버튼을 누르게 되면 병합되며, Remove Source Branch 를 눌러 병합한 원본 브랜치(Topic\_branch)를 삭제하거나, Revert<sup>10</sup>나 Cherry-pick<sup>11</sup> 작업을 할 수 있다.

## Topic branch01

Merge Topic\_branch into the Master branch.

Edited less than a minute ago

Request to merge [topic\\_branch01](#) into [master](#)

Merged by 강상민 less than a minute ago

The changes were merged into [master](#). You can remove the source branch now.

Remove Source Branch

Revert

Cherry-pick



0



0



<그림 10-40. 병합 결과 화면>

메인 메뉴 Repository 의 Branches 를 보면 master 에 병합된 메세지를 확인할 수 있다.

Protected branches can be managed in project settings

Filter by branch name Name Delete merged branches

master default protected

-o eb5afe58 · Merge branch 'topic\_branch01' into 'master' · about an hour ago

<그림 10-41. 브랜치 master 메세지 확인>

<sup>10</sup> 되돌리기이며, 이전 버전의 commit을 취소한 내용을 새로운 버전으로 만드는 명령

<sup>11</sup> 다른 브랜치의 전체 commit 내역을 복사해오지 않고 특정 commit 복사하여 작업중인 브랜치에 붙임

Repository 의 서브 메뉴는 <표 10-1>에서 정리하였다

서브 메뉴	설명
Files	브랜치 파일 관리
Commits	브랜치 커밋 이력 표시
Network	브랜치 변경 이력을 년월 그래프 형태로 표시
Compare	브랜치간 변경사항 비교
Branches	브랜치 관리
Tags	중요한 지점을 표시할 수 있는 태그(꼬리표)를 관리

<표 10-1. Repository 서브 메뉴>

Activity 메뉴는 작업한 모든 이벤트를 확인할 수 있다.

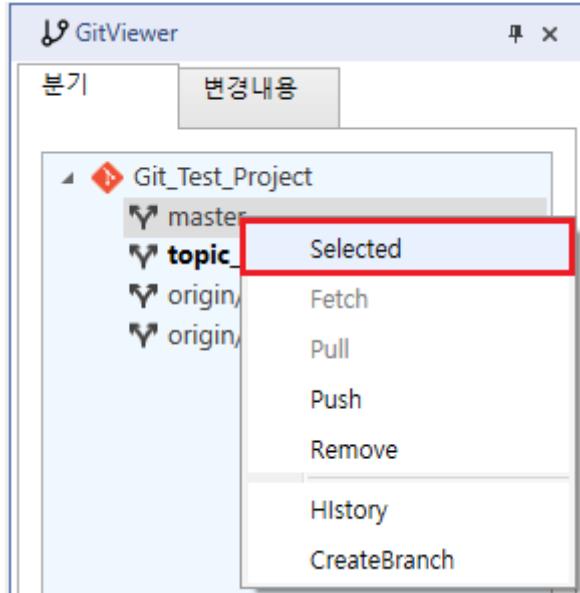
The screenshot shows the GitLab interface with the 'Activity' tab selected. At the top, there's a navigation bar with 'Project' (selected), 'Activity' (underlined), 'Repository', 'Pipelines', 'Graphs', 'Issues 1', 'Merge Requests 0', and 'Wiki'. Below the navigation bar, there are filters: 'All', 'Push events', 'Merge events', 'Issue events', 'Comments', and 'Team'. The main area displays five activity items:

- 강상민 accepted merge request !13  
Topic branch01 abou
- 강상민 pushed to branch master  
eb5afe58 · Merge branch 'topic\_branch01' into 'master'  
1d60e23b · 공장명 텍스트박스 추가  
... and 2 more commits. Compare 12bce12e...eb5afe58 abou
- 강상민 opened merge request !13  
Topic branch01 abou
- 강상민 closed merge request !13  
Topic branch01 abou
- 강상민 opened merge request !13  
Topic branch01 abou

<그림 10-42. Activity 메뉴>

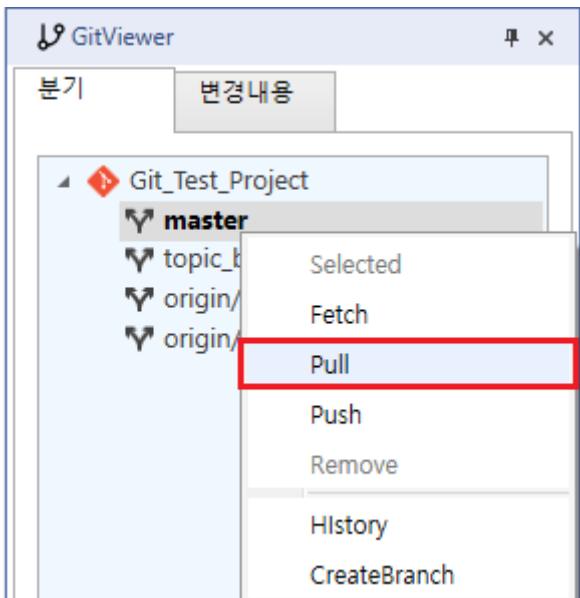
#### 4) DEWS FD Pull & Merge

이제 Git Web Server에서 병합한 master의 정보를 가져오기 위해 DEWS FD 화면에서 끌어오기(Pull)을 실행한다. GitViewer에서 master를 우클릭 후 선택한다.



<그림 10-43. 브랜치 선택>

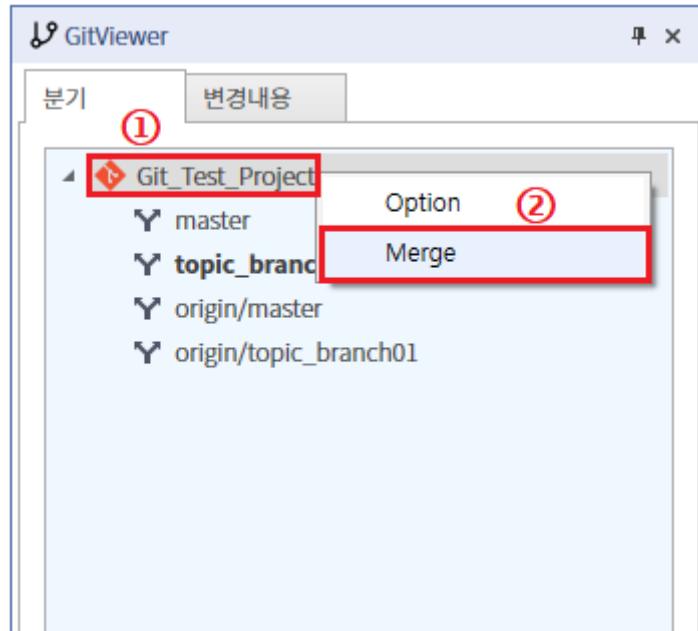
만약 Selected 메뉴가 비활성화 되어 있다면 topic\_branch01에 변경사항이 있는 것이며, Commit을 먼저 진행해야 한다. master가 선택되면 우클릭 후 Pull 메뉴를 실행하여 끌어오기를 한다.



<그림 10-44. Pull 실행>

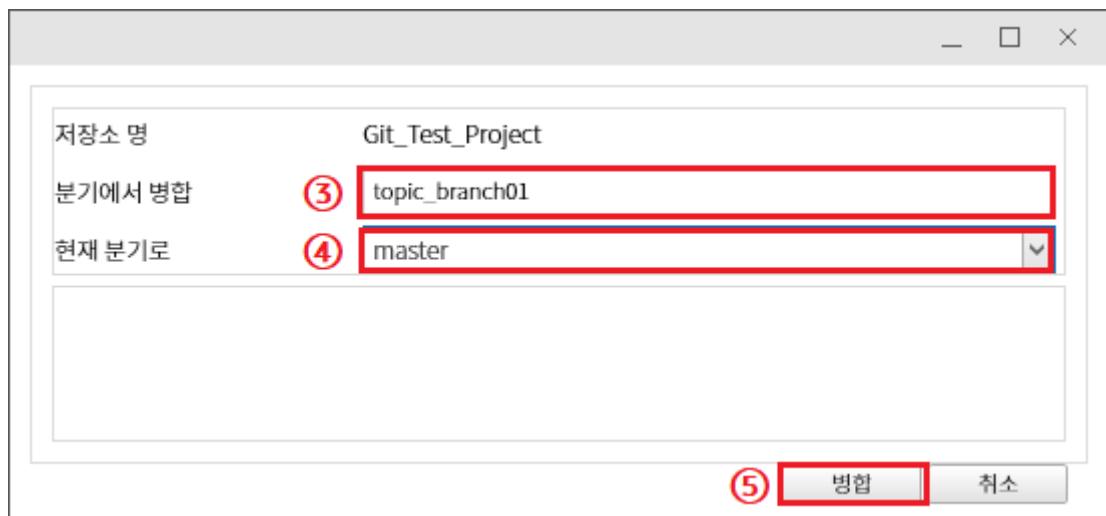
다른 사용자가 master 변경 했을 때 최신 master 정보를 가져와서 현재 작업하는 브랜치에 반영해야 되는 경우 master에서 pull한 뒤에 우클릭 후 현재 작업 중인 브랜치를 선택(Selected) 한다.

- ① 상단 프로젝트를 선택하고 우클릭하여 ② Merge 메뉴를 실행한다.



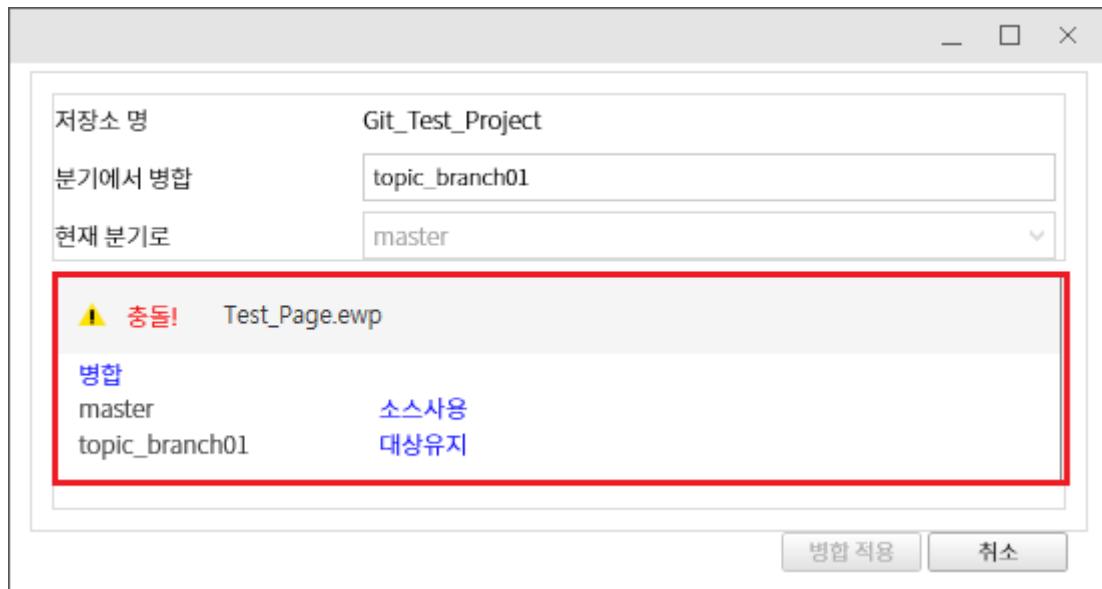
<그림 10-45. 병합 실행>

병합될 대상 브랜치, 즉 선택된 브랜치인 ③ topic\_branch01 가 자동으로 선택되어 있으며(변경 불가), 병합할 원본 브랜치인 ④ master를 선택하고 ⑤ 병합 버튼을 눌러 병합을 실행한다. 즉, master에 있는 내용들이 topic\_branch01에 반영된다.



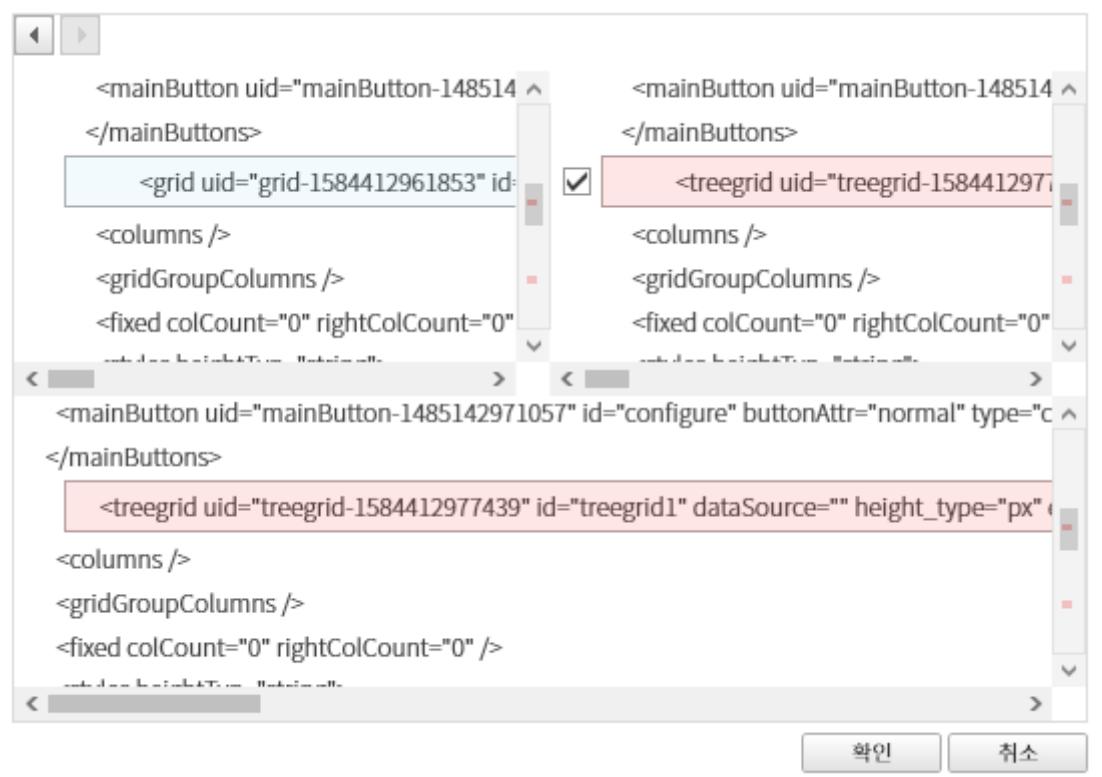
<그림 10-46. 병합 화면>

병합 시에 충돌이 발생할 경우 메세지를 출력한다.



<그림 10-47. 충돌 화면>

병합을 눌러 <그림 10-48> 처럼 둘 중 선택을 하고 확인버튼을 누르거나(병합), master 의 소스를 사용하거나(소스사용), topic\_branch01 의 내용을 유지할 수 있다(대상유지).



<그림 10-48. 병합 선택>

병합 충돌을 줄이기 위해서는 토픽 브랜치(Topic Branch)에 통합 브랜치(Integration Branch : master) 정보를 주기적으로 병합하여 반영해야 한다.

## [제11장] JIRA 연동

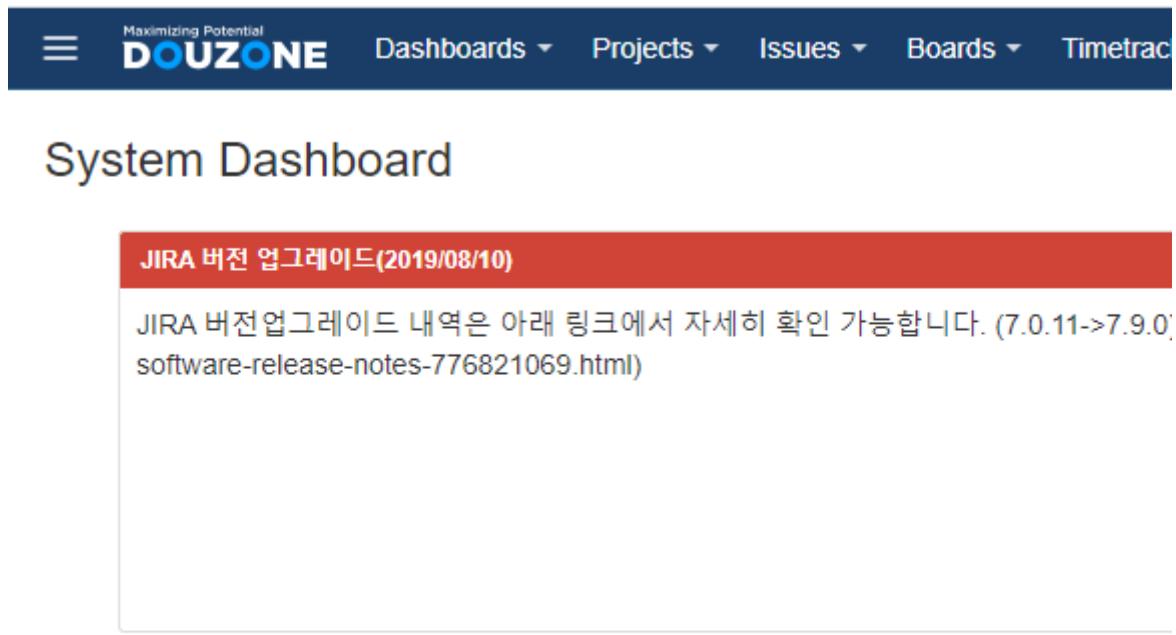
소개

환경 설정 및 실행 방법

## 1. 소개

JIRA는 애틀라시안(ATLASSIAN) 사에서 개발한 이슈 트래킹 시스템(ITS, ISSUE TRACKING SYSTEM<sup>12</sup>)이다. 버그나 결함, 추가 기능을 추적하는 이슈 추적 도구 또는 프로젝트 참여자의 업무 진행과정을 기록하는 프로젝트 관리 시스템을 의미하며, 각 프로젝트에 이슈를 작성하면서 담당자를 지정하고, 버그나 개선사항을 등록할 수 있는 도구이다.

DEWS FD에서 설계자가 완성된 메뉴 UI와 설계문서에 대하여 개발자에게 개발 요청을 위한 이슈 등록을 하거나 개발자가 개발 완료된 메뉴에 대해 설계자에게 검토/테스트 요청을 위한 이슈를 등록할 수 있다.



### Introduction



Welcome to JIRA

New to JIRA? Check out the [JIRA User's Guide](#).

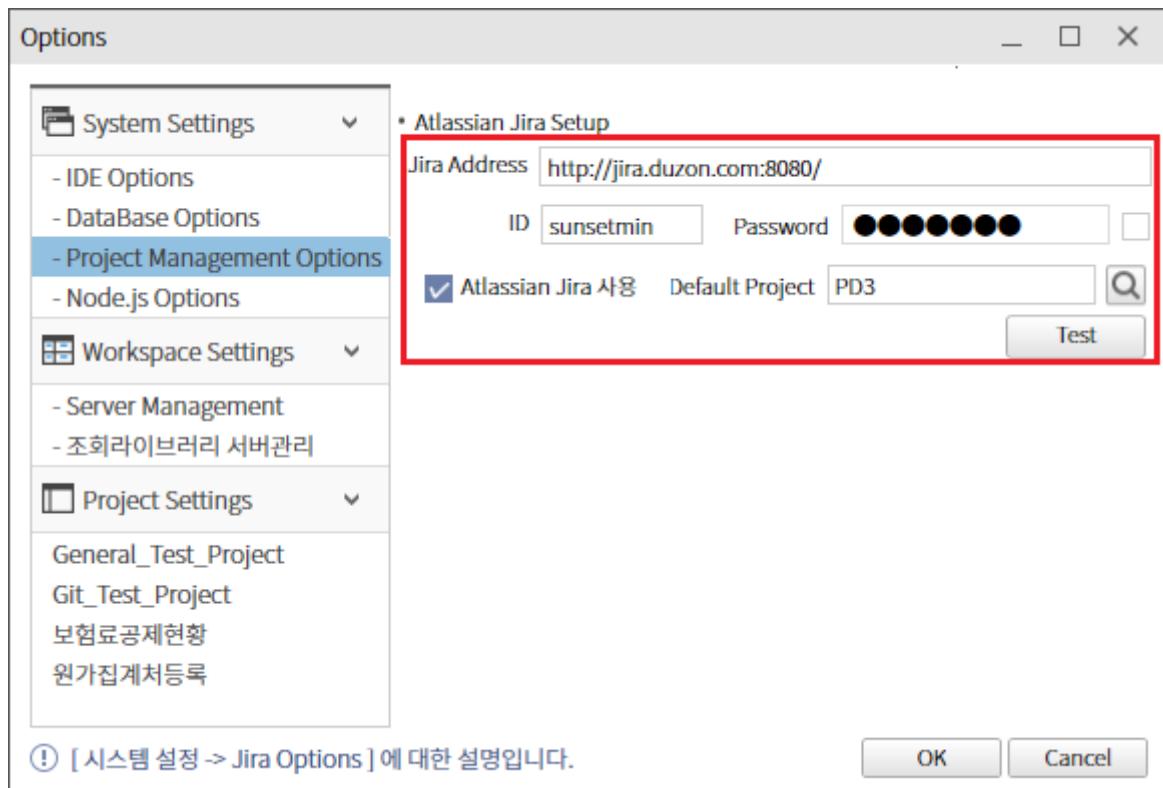
<그림 11-1. JIRA 화면>

<sup>12</sup> 이슈 트래킹 시스템이란, 프로젝트에서 예상 되거나 또는 이미 발생한 '이슈'들을 관리하는 시스템을 말한다.

## 2. JIRA 설정 및 실행 방법

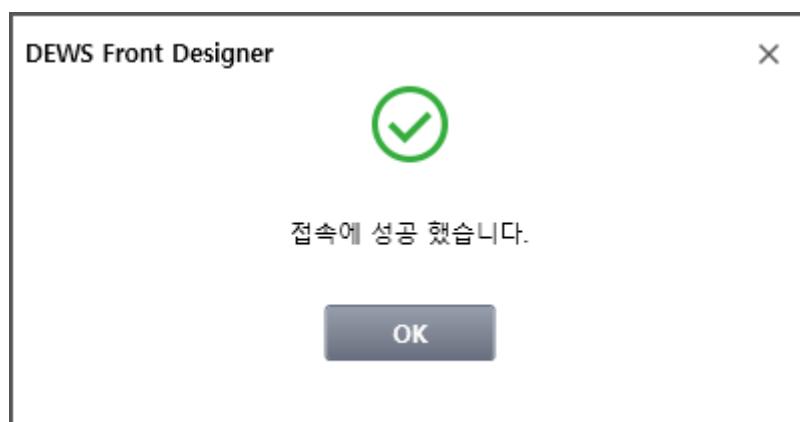
### 1) JIRA 설정

개발도구에서 JIRA를 사용하기 위해 메뉴 Tools > Options 실행하고 Project Management Options에서 JIRA Address, ID, Password, Atlassian Jira 사용 체크, 기본 프로젝트 설정 후 TEST 버튼을 통해 접속 확인을 한다.



<그림 11-2. JIRA 옵션 설정 화면>

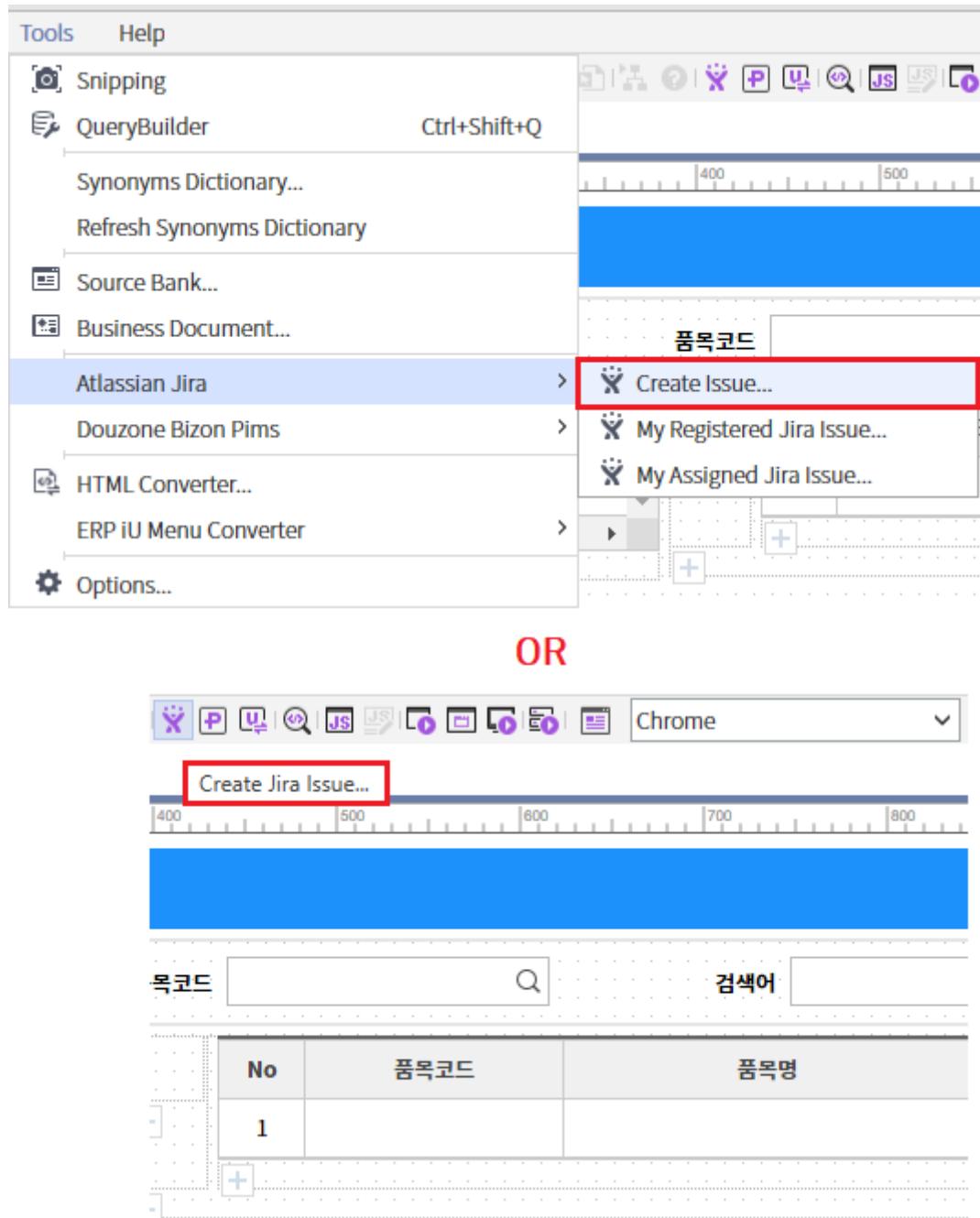
Test 버튼을 눌렀을 때 “접속에 성공했습니다”라는 메세지 박스가 나타난다. OK를 누른다



<그림 11-3. 접속 성공 메세지 박스 화면>

## 2) JIRA 실행

설계 도구 실행 방법은 <그림 11-4>와 같이 메뉴 Tools > Atlassian Jira > Create Issue을 실행하거나 툴바에서 Create Jira Issue를 클릭해서 실행한다.

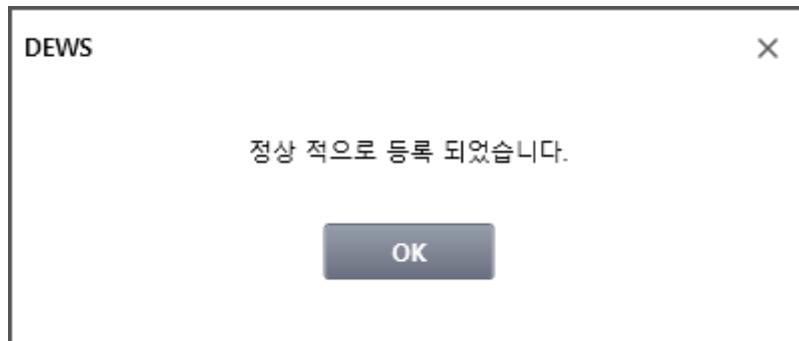


Create Issue 화면이 열리며 항목을 선택하거나 입력한다. 마지막 항목 Attachment(첨부 파일)인 경우 Business document는 설계 문서(pdf)가 자동으로 첨부되며, ewz는 압축파일로 ewp, ewd, ewi 파일 등이 압축되어 있다.

The screenshot shows the 'Create Issue' dialog box. It includes fields for Project (selected: [플랫폼개발센터] 플랫폼개발3팀), Issue Type (selected: Task), Summary (PCOCOP00100), Reporter (강상민(sunsetmin)), Assignee (강상민(sunsetmin)), and Component/s (DFD). A note in the Description field says '설계 완료 하였으니 개발 진행 요청 드립니다.' (Design completed, requesting development progress). In the Attachment section, there is a cloud icon with the text 'Drop files to attach, or browse.' Below it are two checked checkboxes: 'Business document' and 'Source Files(ewz)'. At the bottom right are buttons for '(1)', 'Create' (highlighted with a red border), and 'Cancel'.

<그림 11-5. Create Issue 화면>

Create 버튼을 누르면 “정상적으로 등록되었습니다”라는 메세지 박스가 열린다.



<그림 11-6. JIRA 등록 안내 메세지 화면>

Jira 사이트를 접속 하면 등록되어 있는 것을 확인할 수 있다.

#### Kanban board

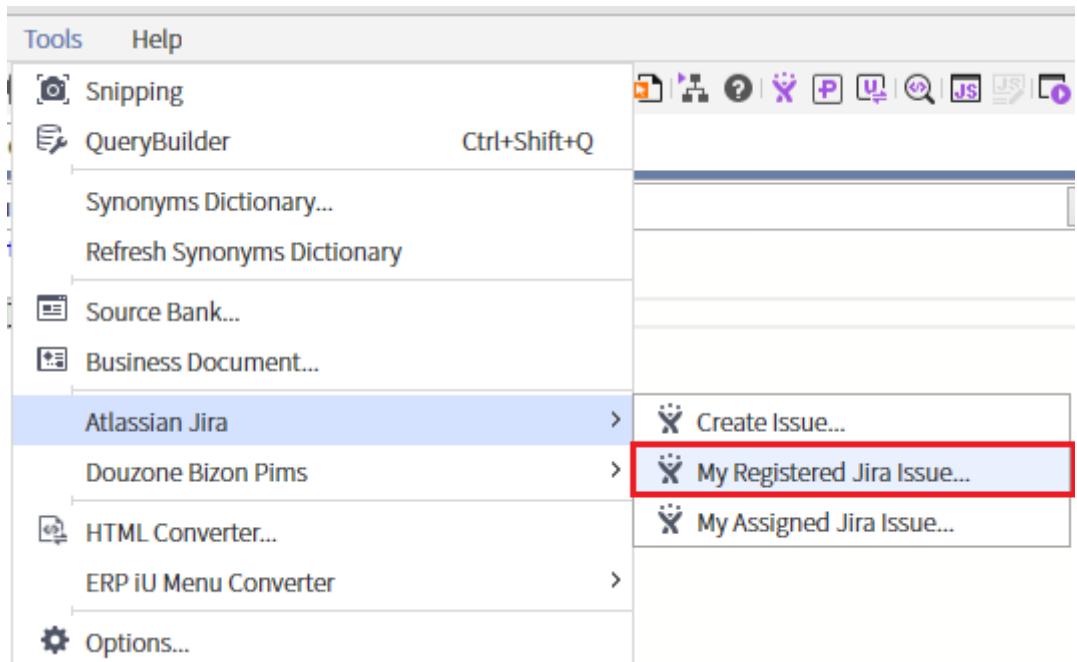
The screenshot shows a Jira Kanban board with the following details:

- QUICK FILTERS:** Epic, 우선순위낮음, 우선순위높음, 일주일전, 1일전, **Recently Updated**, 1주간 작업목록, 이번주, 다음주, 다음달, 중, 장기.
- Backlog:**
  - To Do:** 1 of 2 To Do items. One item is selected: PD3-2597 (checked), PCOCOP00100 (orange arrow), DFD, DFDSTUDI..., None. This row is highlighted with a red box.
  - Ready:** 0 Ready items.
  - In Progress:** 0 of 3 In Progress items.
  - Hold:** 0 of 1 Hold items.
  - Complete:** 0 of 1 Complete items.
  - Release...:** 0 items.
  - Done:** 0 items.
- Issue Detail View (PD3-2597):**
  - Description:** 설정 완료 하였으니 개발 진행 요청 드립니다.
  - Comments:** There are no comments yet on this issue.
  - Attachments:**
    - Drop files to attach, or browse.
    - PCOCOP00100\_20200330.pdf (2020/03/30 1:35 PM, 153 kB)
    - PCOCOP00100.ewz (2020/03/30 1:35 PM, 46 kB)

<그림 11-7. Jira에 등록된 화면>

또한 사용자가 등록한 리스트와 사용자에게 담당자로 지정된 Jira 리스트를 확인할 수 있다.

먼저 사용자가 등록한 Jira 리스트는 메뉴 Tools > Atlassian Jira > My Registered Jira Issue을 실행한다.



<그림 11-8. 사용자가 등록한 Jira 리스트 실행 화면>

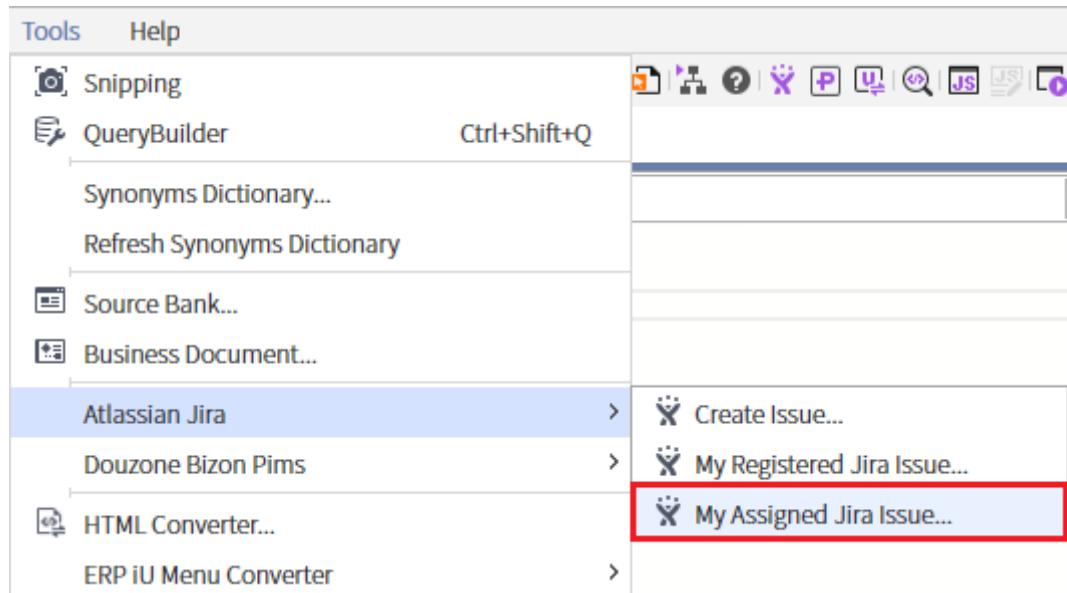
My Registered Jira Issue 화면이 열리며 등록한 Jira 리스트를 확인할 수 있다. 이슈 번호(Issue No), 생성 일자(Create), 요약(Summary), 담당자(Assignee), 상태(Status), 파일(File Name) 등의 정보를 확인할 수 있다. FD에서 등록한 리스트만 보여준다.

The screenshot shows the 'My Registered Jira Issue' dialog box. It has a search bar at the top with 'From: 2020-02-29' and 'To: 2020-03-30'. Below the search bar is a tree view under '원가집계처등록' with one item 'PCOCOP00100' selected. To the right is a grid table with columns: Issue NO., Created, Summary, Assignee, and Status. One row is expanded to show 'File Name', 'Size(KB)', and 'Created' details for two files: 'PCOCOP00100\_20200330.pdf' and 'PCOCOP00100.ewz'. A message at the bottom left says '(정상적으로 조회 되었습니다.)' and a 'Close' button is at the bottom right.

	Issue NO.	Created	Summary	Assignee	Status
PCOCOP00100	PD3-2597	2020-03-30	PCOCOP00100	강상민(sunsetmin)	To Do / Open
File Name				Size(KB)	Created
PCOCOP00100_20200330.pdf				156	2020-03-30
PCOCOP00100.ewz				47	2020-03-30

<그림 11-9. 사용자가 등록한 Jira 리스트 화면>

다음으로 사용자에게 지정(할당)된 Jira 리스트는 메뉴 Tools > Atlassian Jira > My Assigned Jira Issue을 실행한다.



<그림 11-10. 사용자에게 지정된 Jira 리스트 실행 화면>

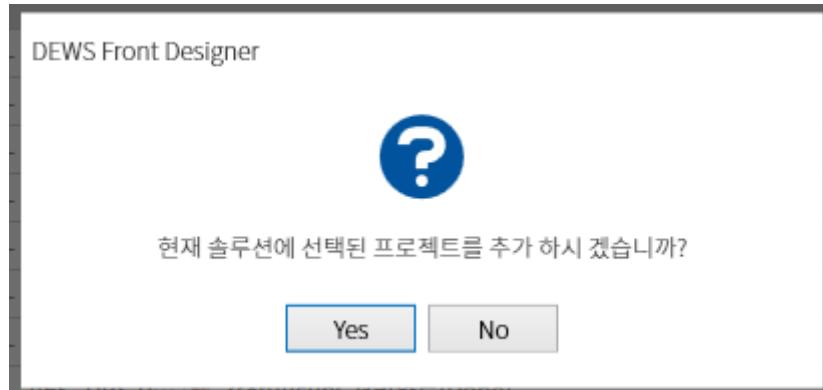
My Assigned Jira Issue 화면이 열리며 사용자에게 할당된 Jira 리스트를 확인할 수 있다. 이슈 번호(Issue No), 프로젝트(Project), 생성 일자(Create), 마감일(DueDate), 요약(Summary), 보고자(Reporter), 상태(Status)를 보여주며 +가 있는 항목인 경우 펼치면 파일(File Name) 등의 정보를 확인할 수 있다. 사용자가 등록한 Jira 리포트와 다르게 사용자에게 할당된 모든 리스트를 보여준다.

The screenshot shows the 'My Assigned Jira Issue' dialog box. It has a search bar at the top with 'From: 2020-02-29' and 'To: 2020-03-30'. Below is a table with columns: Issue NO., Project, Created, DueDate, Summary, Reporter, and Status. One row is expanded to show attachments:

Issue NO.	Project	Created	DueDate	Summary	Reporter	Status
PD3-2598	PD3	2020-03-30		개발도구 - 교육 : 플랫폼...	강상민(sun)	To Do / Open
PD3-2597	PD3	2020-03-30		PCOCOP00100	강상민(sun)	To Do / Open
				File Name	Size(KB)	Created
				PCOCOP00100_20200330.pdf	156	2020-03-30
				PCOCOP00100.ewz	47	2020-03-30
PD3-2595	PD3	2020-03-27	2020-03-27	개발도구 - 고객지원 : D...	강상민(sun)	처리완료(Done)
PD3-2594	PD3	2020-03-27	2020-03-27	개발도구 - 기타 : 공통도...	강상민(sun)	처리완료(Done)
PD3-2589	PD3	2020-03-26	2020-03-30	개발도구 - 기타 : 메뉴얼...	강상민(sun)	진행중(In Progress)
PD3-2588	PD3	2020-03-17	2020-03-17	개발도구 - 그...	강상민(sun)	진행중(In Progress)

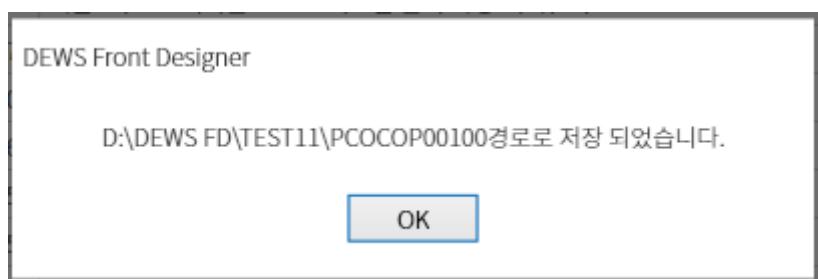
<그림 11-11. 사용자가 등록한 Jira 리스트 화면>

여기에서 압축파일인 ewz 파일을 클릭 하면 현재 작업하고 있는 솔루션에 추가할 것인지를 물어보는 메세지 박스가 열린다. Yes를 누르면 현재 Workspace에 프로젝트로 추가되며 NO를 누르면 다운로드를 진행한다.



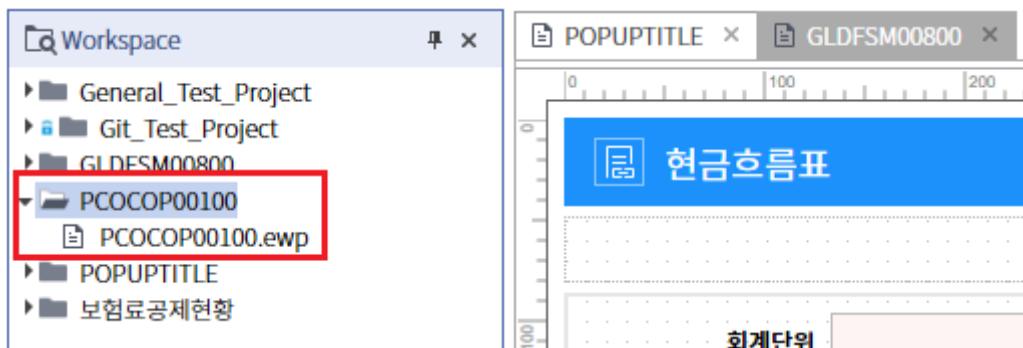
<그림 11-12. ewz 파일 더블클릭 시 프로젝트 추가 여부 확인 메세지 화면>

Yes를 누르면 경로에 저장되었다는 메세지박스가 열린다. ewz 파일이름이 프로젝트 이름으로 저장된다.



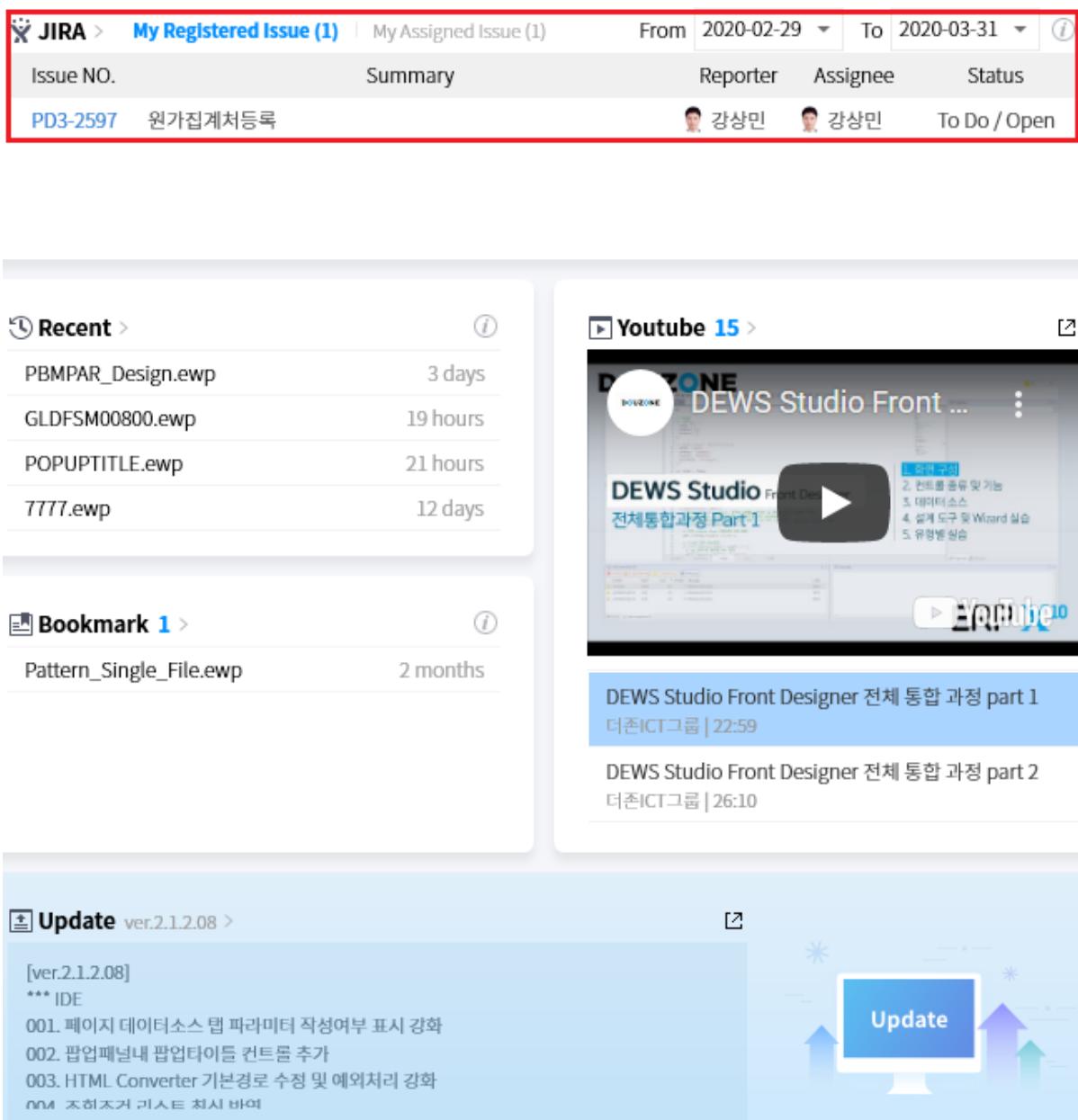
<그림 11-13. 경로 저장 메세지 화면>

Workspace에 보면 프로젝트가 추가되어 있음을 확인할 수 있다.



<그림 11-14. Workspace 프로젝트에 추가된 화면>

다시 DFD를 실행하거나 오른쪽 상단  Tip을 눌러서 Welcome 페이지를 열면 상단에 JIRA 리스트가 보이는 것을 확인할 수 있다.



The screenshot shows the JIRA Welcome page with several sections:

- Recent:** A list of recently opened files:
  - PBMPAR\_Design.ewp (3 days ago)
  - GLDFSM00800.ewp (19 hours ago)
  - POPUPTITLE.ewp (21 hours ago)
  - 7777.ewp (12 days ago)
- Bookmark:** A list of bookmarked files:
  - Pattern\_Single\_File.ewp (2 months ago)
- Update:** Information about version 2.1.2.08:
  - [ver.2.1.2.08]
  - \*\*\* IDE
    - 001. 페이지 데이터소스 템 파라미터 작성여부 표시 강화
    - 002. 팝업페널내 팝업타이틀 컨트롤 추가
    - 003. HTML Converter 기본경로 수정 및 예외처리 강화
    - 004. 조회조작 리스트 최신 버전
- Youtube:** A list of YouTube videos:
  - DEWS Studio Front ... (Thumbnail shows a video player with a play button and a list of 5 points: 1. 환경설정, 2. 컨트롤 종류 및 기능, 3. 데이터소스, 4. 설계 도구 및 Wizard 활용, 5. 유형별 설정)
  - DEWS Studio Front Designer 전체 통합 과정 part 1 (더존ICT그룹 | 22:59)
  - DEWS Studio Front Designer 전체 통합 과정 part 2 (더존ICT그룹 | 26:10)

<그림 11-15. Welcome 페이지 JIRA 화면>