**개발가이드(UI)**

|  |  |
| --- | --- |
| 프로젝트명 | 차세대 스마트 원전연료 제조운영시스템 통합구축 |
| 단계 | 설계 |
| 활동 | 개발가이드 |
| 작업 | 개발가이드(UI) 작성 |
| 산출물 | 개발가이드(UI) |
| 문서명(파일명) | 넥사크로 개발가이드(UI)\_V0.1 |
| 버전 | V0.1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 소속 / R&R | 성명 | 서명 | 일자 |
| 작성자 | UI공통/ 담당자 | 정원식 |  |  |

개정 이력

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 버전 | 개정 일자 | 작성자 | 검토자 | 승인자 | 개정 내용 |
| 0.9 | 2025/05/20(수) | 정원식 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

목 차

[1. 개요 5](#_Toc198552995)

[1.1 목적 5](#_Toc198552996)

[1.2 범위 5](#_Toc198552997)

[1.3 문서의 관리 5](#_Toc198552998)

[2. 개발환경 6](#_Toc198552999)

[2.1 넥사크로 스튜디오 설치 6](#_Toc198553000)

[2.1.1 설치 6](#_Toc198553001)

[2.1.2 설치파일 7](#_Toc198553002)

[2.1.3버전확인 8](#_Toc198553003)

[2.1.4 실행환경 설정 8](#_Toc198553004)

[2.1.5 라이선스 설정 9](#_Toc198553005)

[2.1.6 넥사크로 스튜디오 주요기능 10](#_Toc198553006)

[2.3 프로젝트 구성요소 18](#_Toc198553007)

[2.1.1 디렉토리 구조 18](#_Toc198553008)

[2.1.2 Environment 변수 18](#_Toc198553009)

[2.1.3 Application 변수 18](#_Toc198553010)

[2.1.4 Global Dataset 19](#_Toc198553011)

[2.1.5 공통 라이브러리 19](#_Toc198553012)

[2.1.6 공통 팝업 20](#_Toc198553013)

[3. 개발가이드 20](#_Toc198553014)

[3.1 개발절차 20](#_Toc198553015)

[3.1.1 넥사크로 스튜디오 실행 20](#_Toc198553016)

[3.1.2 프로젝트 오픈 20](#_Toc198553017)

[3.1.3 코드 스니펫(Snippet) 20](#_Toc198553018)

[3.1.5 프로젝트 실행 26](#_Toc198553019)

[3.2 기능별 개발방법 27](#_Toc198553020)

[3.1.1 Transaction 27](#_Toc198553021)

[3.1.2 데이터 바인딩 27](#_Toc198553022)

[3.1.3 팝업 처리 30](#_Toc198553023)

[3.1.4 메시지 처리 31](#_Toc198553024)

[3.1.5 Calendar 32](#_Toc198553025)

[3.3 개발시 유의사항 32](#_Toc198553026)

[3.1.1 Column Type 32](#_Toc198553027)

[3.1.2 이벤트 32](#_Toc198553028)

[3.1.6 enableevent 33](#_Toc198553029)

[3.1.7 Dataset layout 33](#_Toc198553030)

[3.1.8 eval 함수 대체 34](#_Toc198553031)

[4. 개발 관련 참조 사이트 34](#_Toc198553032)

# 1. 개요

본 문서는 UI개발자를 위한 개발 가이드문서로 업무화면 개발에 필요한 템플릿과 패턴에 대한 가이드를 제공합니다.

## 1.1 목적

인천공항 인사급여시스템 구축 프로젝트를 수행하기 위한 개발 환경구성과

개발자의 편의성 및 구성원들의 동일한 표준 적용을 위해 프로젝트 개발 단계별 처리 항목과 디자인

영역, 공통 처리 부문, 환경 적용에 대한 순차적인 단계를 정의합니다.

## 1.2 범위

본 문서는 “인천공항 인사급여시스템 구축” 사업의 웹 어플리케이션 UI개발

단계 작업을 수행하는데 적용됩니다.

## 1.3 문서의 관리

본 문서의 작성 및 관리 책임은 KNFC 디지털융합부에 있으며 본 문서와 관련된 변경사항은 반드시 상기 부서의 승인 하에 이루어 져야 합니다.

# 2. 개발환경

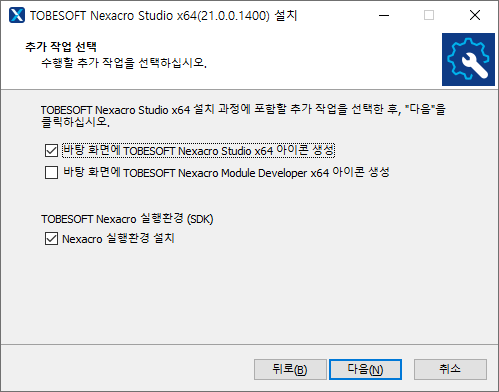
## 2.1 넥사크로 스튜디오 설치

### 2.1.1 설치

**설치 파일 :** c/EGOVFRAMEDEV-4.3.0-64BIT/NexacroN\_Setup\_x64.exe

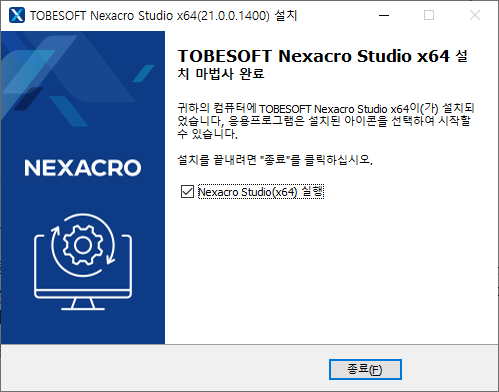
**라이선스 :** airport.hrpfe/src/main/NexaUI/licens/NexacroN\_tool\_license.XML

* 설치 파일(NexacroN\_SetupStudio\_x64.exe)을 실행하면 설치 과정이 진행됩니다.



바탕 화면에 단축 아이콘을 만들고자 한다면 '바탕 화면에 ~ 아이콘 생성' 항목을 체크합니다.

'Nexacro 실행환경 설치'는 최초 설치 시에는 체크되어야 하는 항목이며 기존 SDK를 유지하고 개발도구만 업데이트하는 경우에는 체크하지 않습니다.



[종료] 버튼을 클릭하여 프로그램 설치를 완료합니다. 'Nexacro Studio 실행' 항목이 체크된 경우에는 설치

완료 후 넥사크로 스튜디오를 실행합니다.

2**.1.2 설치파일**

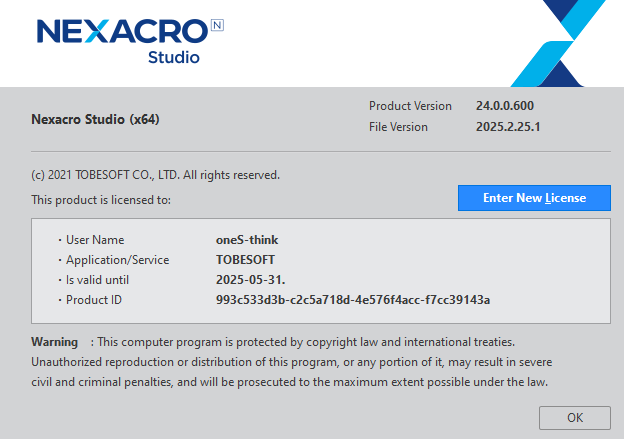
넥사크로 스튜디오를 설치한 폴더 내 생성되는 주요 파일과 폴더는 아래와 같습니다.

|  |  |
| --- | --- |
| SDK > | |
| 폴더명/파일명 | 설명 |
| embedded | - 넥사크로 스튜디오 내에서 사용하는 넥사크로 앱  그리드 콘텐츠 에디터나 에뮬레이터처럼 넥사크로를 사용해 만든 기능 관련 소스 코드가 저장된 폴더입니다. |
| help | - 넥사크로 도움말 폴더  컴파일된 HTML 도움말 파일(compiled html help, CHM)로 제공합니다.  넥사크로 스튜디오는 운영체제 언어 설정에 따라 해당하는 한국어, 영어 도움말 파일을 실행합니다. |
| nexacrolib | - 넥사크로 라이브러리 폴더  설치된 버전의 라이브러리 파일이 포함된 폴더입니다.  metainfo 파일을 포함하고 있습니다. metainfo 파일은 넥사크로 스튜디오에서만 사용하는 파일이며 배포 대상은 아닙니다. |
| SetupRuntime | 빌드 시 참조할 운영체제별 모듈 파일이 포함된 폴더입니다. |
| theme | - 테마 폴더  넥사크로 스튜디오 테마, 운영체제별 기본 테마 파일이 포함된 폴더입니다. |
| nexacro.exe | - NRE 실행 파일 |
| ResourceUpdater\_x64.exe | - NRE 배포 파일 생성 실행 파일 |
| ResourceUpdater\_x86.exe | - NRE 배포 파일 생성 실행 파일 |
| nexacromigrator.exe | - 이전 버전 프로젝트 마이그레이션 실행 파일 |
| Tool > | |
| license | - 라이선스 관련 문서 폴더  제품 라이선스, 오픈소스 라이선스 관련 문서 파일이 포함된 폴더입니다. |
| nexacrodeploy.exe | - 프로젝트 제너레이트, 난독화 실행 파일 |
| nexacroemulator.exe | - 에뮬레이터 실행 파일 |
| nexacrostudio.exe | - 넥사크로 스튜디오 실행 파일 |
| nexacromoduledeveloper.exe | - 넥사크로 모듈 디벨로퍼 실행 파일 |
| caddy.exe | - 로컬 웹 서버 실행 파일 |

2.1.3버전확인

넥사크로 스튜디오의 메뉴[Help > About Nexacro Studio]를 선택하면 설치된 넥사크로 스튜디오의 버전과 등

록된 라이선스 정보를 확인할 수 있습니다.



2.1.4 실행환경 설정

넥사크로 스튜디오를 처음 설치하면 60일 동안 사용할 수 있는 체험판으로 설정됩니다.

최초 실행 시 경고 팝업 창이 표시되는데 [OK] 버튼을 클릭하면 다음 단계로 진행할 수 있습니다. 제품을 구

매한 경우에는 발급받은 Product Key를 입력해 사용할 수 있습니다.



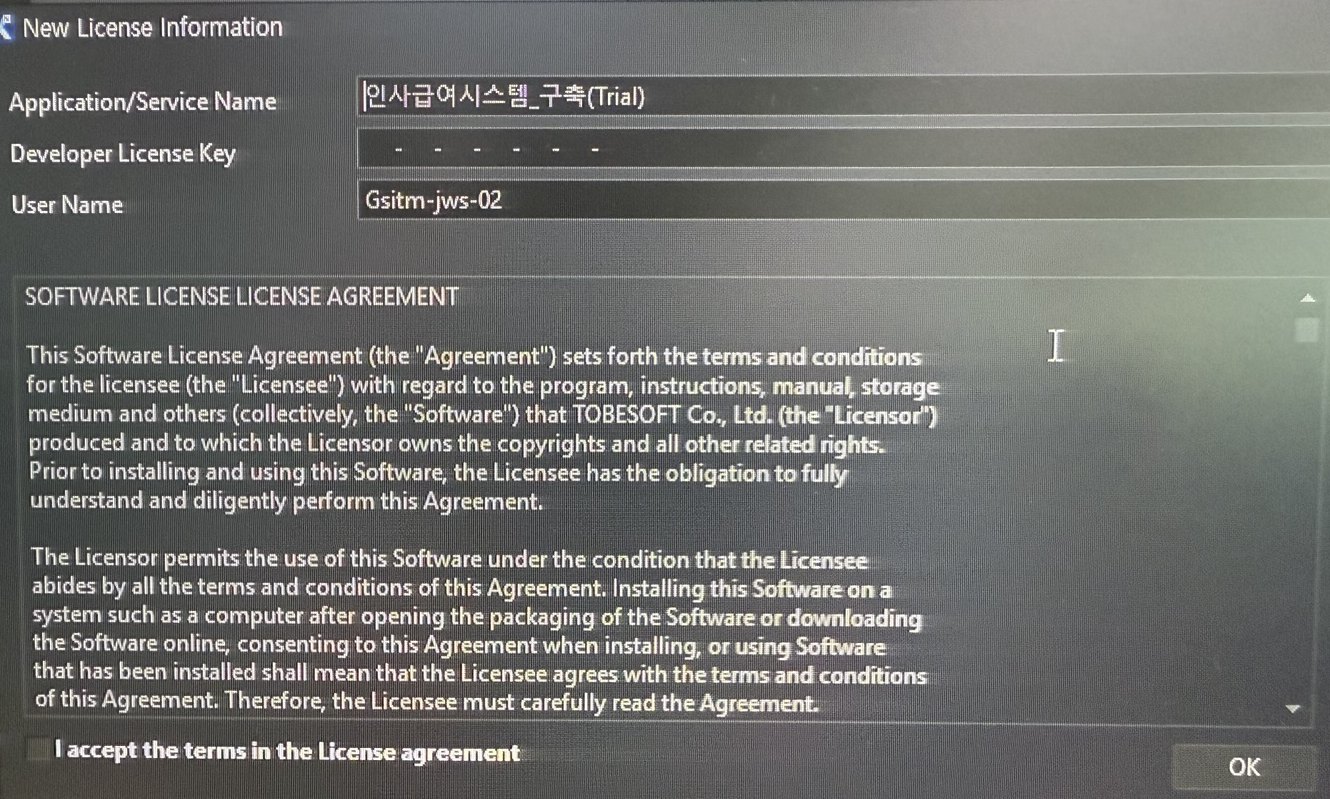
넥사크로 스튜디오 실행 환경을 설정합니다. Perspective, Command Type, Theme를 설정할 수 있습니다. 설

정한 항목은 메뉴[Options > Environment > General > Development Tools]에서 변경할 수 있습니다.



### 2.1.5 라이선스 설정

- NexaUI/llicense/NexacroN\_tool\_license.XML.



### 2.1.6 넥사크로 스튜디오 주요기능

(1) Start Page

넥사크로 스튜디오 실행 시 Start Page를 제공 합니다.

프로젝트 생성 또는 기존에 만들어진 프로젝트를 오픈 할 수 있으며, 새로운 매뉴얼, 팁 동영상이

링크되어 바로 확인 가능합니다.

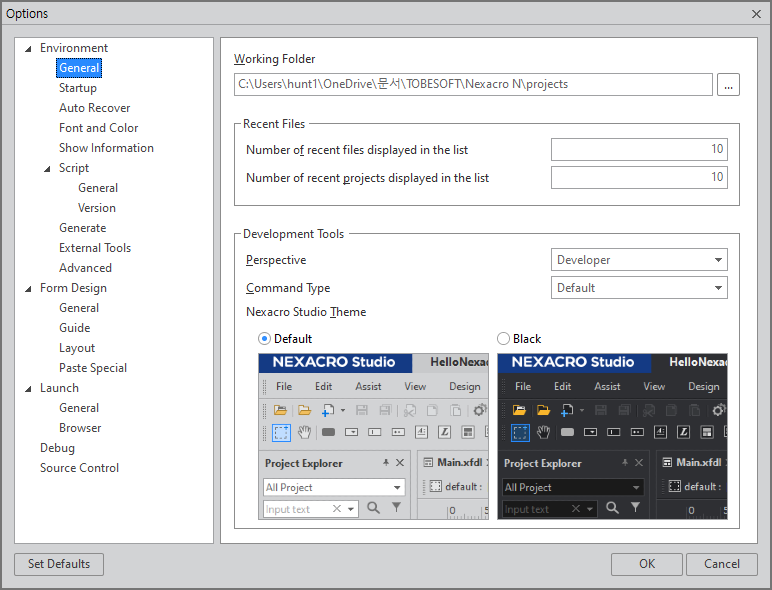


(2) White/Black 테마

넥사크로 스튜디오를 White 또는 Black 테마로 선택하여 실행 할 수 있습니다.

상단의 메뉴 구성도 Commandbar Type 메뉴와 Ribbon Type 메뉴를 선택 할 수 있습니다.

[Menu] Tools > Option > Environment > General



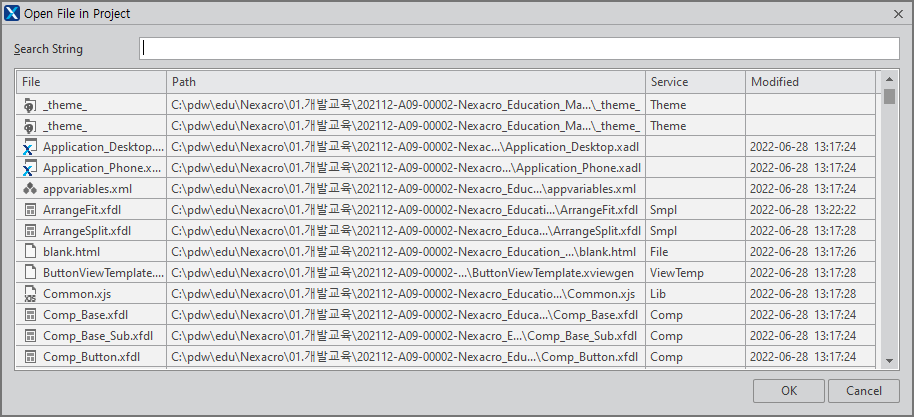
(3) Open File in Project

프로젝트에 생성되어 있는 파일을 쉽게 접근할 수 있는 기능입니다.

Open File in Project 창에서 파일명으로 검색을 하고, 검색 된 목록을 클릭하여 해당 파일로 이동 할 수

있습니다. 많은 양의 파일을 사용하는 대규모 프로젝트에서 파일을 찾는 시간을 단축 할 수 있습니다.

[Menu] Assist> Open File in Project

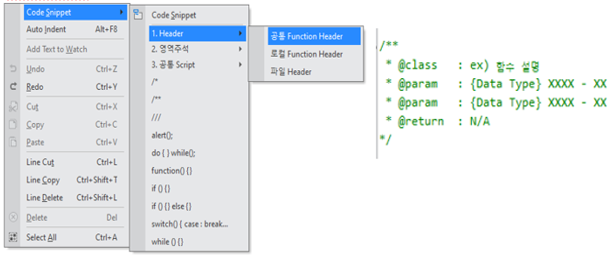


(4) Code Snippet

반복되는 코딩을 쉽게 적용하기 위한 기능 입니다. 자주 사용하는 주석, 스크립트를 미리 등록하여 Script Editor에서 Context Menu 또는 단축키를 통해 바로 적용 할 수 있습니다.

[Menu] Assist > Code Snippet

[ContextMenu] Code Snippet > Edit Code Snippet



추가한 Snippet 항목이 Inspector 목록으로 나타납니다.

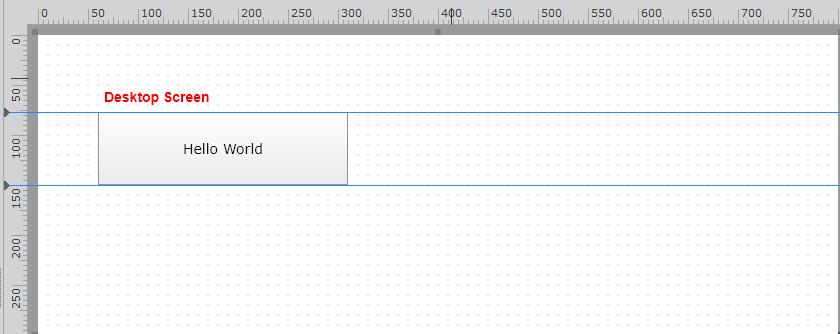
지정한 단축키 또는 마우스 우클릭을 이용하여 Snippet을 적용 합니다.

(5) 가이드 라인

화면의 룰러 부분을 클릭하거나 마우스 우클릭 메뉴를 이용하여 가이드 라인을 설정 할 수 있습니다.

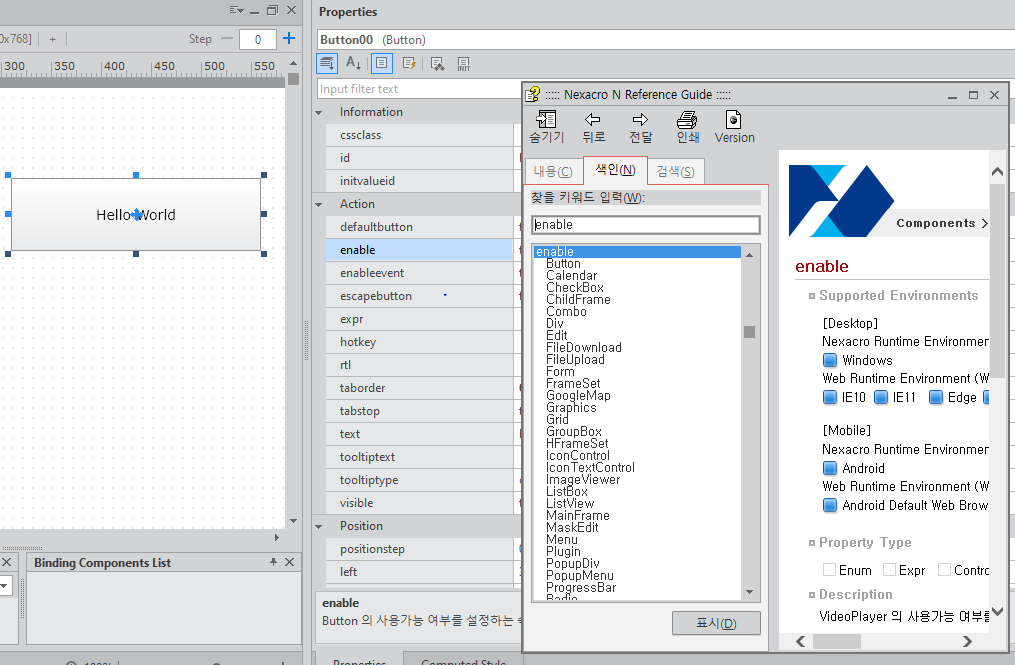
라인에 맞게 컴포넌트를 디자인 할 수 있으며, 가이드 라인을 이동시 라인에 따라 컴포넌트가 이동 또는

리사이즈 됩니다.



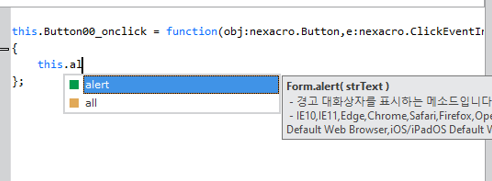
(6) 도움말

컴포넌트의 속성, 이벤트명을 선택하고 F1키를 누르면 선택한 항목의 도움말로 오픈 됩니다.



(7) Inspector & Code Assist

Script 편집 위치 기준으로 현재 Scope에서 사용할 수 있는 Object, Attribute, Method 등의 목록을 제공합니다.

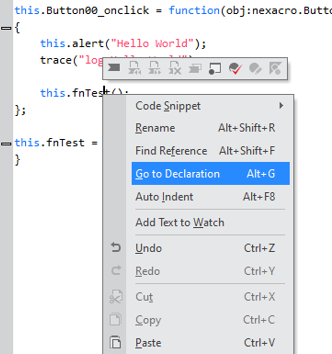


Function, Method를 선택한 경우 입력해야 하는 argument를 툴팁에 Bold로 표시됩니다.

(8) Go to Declaration

스크립트에서 사용한 함수의 선언 위치를 바로 찾아가는 기능 입니다.

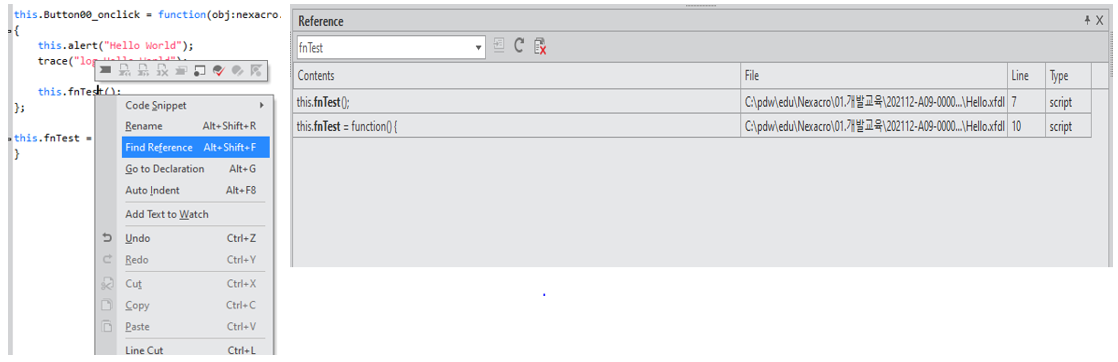
함수를 선택하고 마우스 우클릭 메뉴 또는 단축키(Alt+G)를 이용합니다.



(9) Find Reference

스크립트에서 선택한 문자열을 프로젝트 파일에서 검색하여 사용하고 있는 위치를 제공합니다.

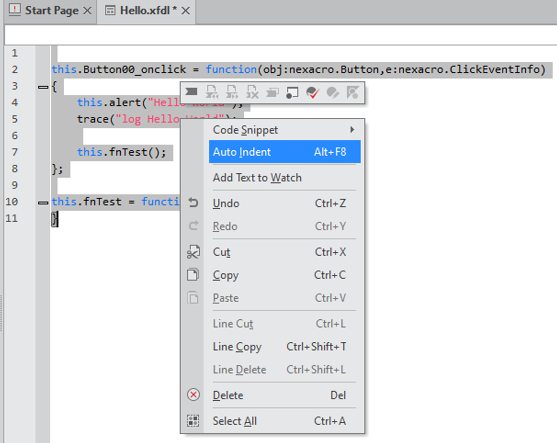
함수를 선택하고 마우스 우클릭 메뉴 또는 단축키(Alt+Shift+F)를 이용합니다.



(10) 자동 들여쓰기

스크립트에서 블록 지정한 영역을 자동으로 들여쓰기를 적용합니다

블록을 지정하고 마우스 우클릭 메뉴 또는 단축키(Alt+F8)를 이용합니다.



(11) 주요 단축키

주요 기능에 대해서 단축키를 제공하여 애플리케이션 개발 시 생산성 향상에 도움을 줍니다.

1) 공통

|  |  |  |
| --- | --- | --- |
| 기능명 | 단축키 | 기능 |
| New Project | Ctrl + Shift + N | 새로운 프로젝트를 생성한다 |
| New | Ctrl + N | 새로운 파일을 생성한다. |
| Open Project | Ctrl + Shift + O | 이전에 생성한 프로젝트를 연다. |
| Open | Ctrl + O | 이전에 생성한 파일을 연다. |
| Save | Ctrl + S | 파일에서 수정한 내용을 저장한다. |
| Save All | Ctrl + Shift + S | 모든 파일에 수정한 내용을 저장합니다. |
| Copy | Ctrl + C, Ctrl + Ins | 선택된 내용을 Clipboard에 복사한다. |
| Cut | Ctrl + X, Shift + Del | 선택된 내용을 잘라내어 Clipboard에 복사한다. |
| Paste | Ctrl + V, Shift + Ins | Clipboard의 내용을 붙여 넣는다. |
| Delete | Del | 선택된 내용을 삭제한다. |
| Select All | Ctrl + A | 모든 항목을 선택한다. |
| Undo | Ctrl + Z, Alt + Back | 바로 이전 작업상태로 되돌린다. |
| Redo | Ctrl + Y | 바로 전에 작업한 내용을 반복한다. |
| Find in Files | Ctrl + Shift + F | Find in Files 창을 띄운다. |
| Open File in Project | Alt + Shift + O | Open File in Project 창을 띄운다. |
| Project Explorer | Ctrl + Alt + L | Project Explorer 창을 띄운다. |
| Resource Explorer | Ctrl + Shift + E | Resource Explorer 창을 띄운다. |
| Properties | Alt + Enter | Properties 창을 띄운다. |
| Computed Style | Ctrl + Shift + U | Computed Style 창을 띄운다. |
| Bookmark | Ctrl + W | Bookmark 창을 띄운다. |
| Dataset Editor | Ctrl + D | Dataset Editor 창을 띄운다. |
| Output | Alt + 2 | Output 출력 창을 띄운다. |
| Error List | Ctrl + \ | Error List 창을 띄운다. |
| Help | F1 | Nexacro 도움말(레퍼런스 가이드)이 나타난다 |
| Launch (Screen) | Ctrl + F5 | 현재 열려있는 프로젝트를 지정한 Screen 정보로 실행한다. |
| Start Debugging | F5 | 디버깅을 시작한다. |
| Stop Debugging | Shift + F5 | 디버깅을 종료한다. |
| Restart | Ctrl + Shift + F5 | 디버깅 모드를 재 시작한다. |
| Step Into | F11 | 한 스텝 디버그 작업 진행한다.  (다른 함수 호출 시 해당 함수로 이동) |
| Step Over | F10 | 현재 함수의 나머지 부분을 실행하고,  함수 호출이 이루어진 다음 문장에서 멈춘다. |
| Step Out | Shift + F11 | 한 스텝 디버그 작업 진행한다.  (현재 스텝을 모두 실행하고, 다음 스텝으로 이동) |
| Run to Cursor | Shift + F10 | 현재 커서가 위치한 줄까지 실행 후 멈춘다. |
| Exit | Alt + F4 | 넥사크로스튜디오를 종료합니다. |

2) Window

|  |  |  |
| --- | --- | --- |
| 기능명 | 단축키 | 기능 |
| Next Window | Ctrl + Tab | 열려있는 다음 파일을 활성화합니다. |
| Previous Window | Ctrl + Shift + Tab | 열려있는 이전 파일을 활성화합니다. |
| Expand Max View | Ctrl + Shift + Enter | 모든 도킹창을 최소화하고 현재 윈도우를 최대화합니다. |
| Change Tab Next | Ctrl + Page Down | 해당 탭 바로 다음 탭으로 이동합니다.  (Design/Source/Script) |
| Change Tab Previous | Ctrl + Page Up | 해당 탭 바로 이전 탭으로 이동합니다.  (Design/Source/Script) |

2) Project Explorer

|  |  |  |
| --- | --- | --- |
| 기능명 | 단축키 | 기능 |
| Rename | F2 | Project Explorer에서 선택된 Item의 이름을 변경합니다. |
| Search | F3 | 입력된 Filter Keyword를 포함한 Item을 선택합니다. |

3) Form Design

|  |  |  |
| --- | --- | --- |
| 기능명 | 단축키 | 기능 |
| Escape | Esc | 현재 작업을 취소합니다. (현재 작업이 없다면 선택된 컴포넌트의 상위 컴포넌트를 선택합니다) |
| Next | Tab | 선택된 컴포넌트의 다음Tab Order 컴포넌트를 선택합니다. |
| Previous | Shift + Tab | 선택된 컴포넌트의 이전 Tab Order컴포넌트를 선택합니다. |
| Move | 방향키 ↑,↓,←,→ | 선택된 컴포넌트를 상,하,좌,우로 이동합니다. |
| Move(minutely) | Ctrl + 방향키  ↑,↓,←,→ | 선택된 컴포넌트를 상,하,좌,우로 미세하게 이동합니다. |
| Resize | Shift + 방향키  ↑,↓,←,→ | 선택된 컴포넌트의 사이즈를 변경합니다. |
| Resize(minutely) | Ctrl + Shift + 방향키 ↑,↓,  ←,→ | 선택된 컴포넌트의 사이즈를 미세하게 변경합니다. |
| InplaceEdit | F2 | 선택된 컴포넌트의 Text를 변경할 수 있는 InplaceEdit를 활성화합니다. |
| Select All | Ctrl + A | 전체 컴포넌트들을 선택합니다. |
| Hotkey Editor | Ctrl + H | 전체 컴포넌트의 Hotkey 편집기를 띄웁니다. |
| Tab Order Editor (View  Type) | Ctrl + T | View 타입의 Tab Order 편집 모드를 활성화합니다. |
| Tab Order Editor (List Ty  pe) | Ctrl + Shift + D | Tab Order 편집기를 띄웁니다. |
| Bind State View | Ctrl + B | 컴포넌트들의 Bind 정보를 표시합니다. |
| Quick View | Ctrl + F6 | 현재 Form을 실행합니다 |
| Copy ID (Include Parent) | Ctrl + Shift + C | 선택된 컴포넌트의 ID 텍스트를 복사합니다. |
| Paste Special | Ctrl + Shift + V | Paste Special을 실행합니다. |

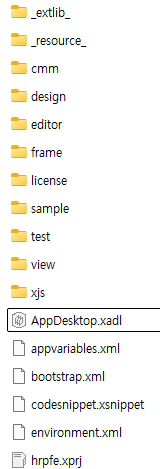
4) Debug

|  |  |  |
| --- | --- | --- |
| 기능명 | 단축키 | 기능 |
| Start Debugging | F5 | 디버깅을 시작합니다. |
| Stop Debugging | Shift + F5 | 디버깅을 중지합니다. |
| Step Into | F11 | 한 스텝 디버그 작업 진행합니다.  (다른 함수 호출 시 해당 함수로 이동) |
| Step Over | F10 | 현재 함수의 나머지 부분을 실행하고, 함수 호출이 이루어진 다음 문장에서 멈춥니다. |
| Step Out | Shift + F11 | 한 스텝 디버그 작업 진행합니다.  (현재 스텝을 모두 실행하고, 다음 스텝으로 이동) |
| Run to Cursor | Ctrl + F10 | 현재 커서가 위치한 줄까지 실행 후 멈춥니다. |
| Start Form Debugging | F6 | Form을 디버깅합니다 |

## 2.3 프로젝트 구성요소

### 2.1.1 디렉토리 구조

|  |  |
| --- | --- |
| 폴더명 | 설명 |
| \_resource\_ | 애플리케이션에서 고정으로 사용하는 Resource 폴더로 공통라이브러리,Theme, Image, UserFont 등의 정보를 관리 합니다. |
| cmm | 공통 폼(alert,confiorm,월,기간,페이징,팝업 외) |
| design | 디자인 가이드 및 업무 기본 템플릿 |
| frame | 프로젝트 구성 관련 폴더 |
| license | Nexacro client license 파일 폴더 |
| xjs | 업무 폼 공통 라이브러리 |
| sample | 개발자 샘플 폴더 |
| test | 테스트 폴더 |
| View | 업무폴더 |



### 2.1.2 Environment 변수

애플리케이션과 화면에서 공통으로 사용할 변수를 지정합니다.

|  |  |  |
| --- | --- | --- |
| Id | Initval | 설명 |
| evRunMode | 0 | 로컬/개발/운영 구분 |
| evTraceMode | true | 운영시 trace 로그 관리용 |

스크립트는 아래와 같은 방식으로 사용합니다.

*//Variable 값 반환*

*var sValue = nexacro.getEnvironmentVariable("evRunMode");*

### 2.1.3 Application 변수

애플리케이션과 전역적으로 사용하는 변수를 관리합니다.

|  |  |  |
| --- | --- | --- |
| Id | Initval | 설명 |
| app.v.login | false | 로그인 여부(sso) |
| app.v.mode | “S” | 실행모드 S:”studio” L: “로컬” D : 개발” |
| app.v.max | 15 | 열린 메뉴 최대 갯수 |

스크립트는 아래와 같은 방식으로 사용합니다.

*//Application Object 를 이용하여 접근*

*app.v.login;*

### 2.1.4 Global Dataset

애플리케이션에서 전역적으로 사용하는 2차원 데이터를 관리합니다.

|  |  |
| --- | --- |
| Id | 설명 |
| gdsMenu | 메뉴정보 |
| gdsMessage | 메시지 정보 |
| gdsComCode | 공통코드 |
| gdsOpenMenu | 열린 메뉴정보 |
| gdsGridPopupMenu | 그리드 팝업 메뉴 |
| gdsFavor | My메뉴 정보 |
| gdsUserInfo | 사용자 정보 |

### 2.1.5 공통 라이브러리

애플리케이션에서 공통으로 적용되는 라이브러리를 관리합니다.

* 위치 : nexacrolib > component > Libcommon

|  |  |
| --- | --- |
| 파일명 | 설명 |
| Constant.js | 공통 변수 선언 관련 공통함수 |
| Excel.js | 엑셀 export/import 관련 공통함수 |
| File.js | File 업로드/다운로드 관련 공통함수 |
| Frame.js | 프레임/다국어 관련 공통함수 |
| Grid.js | 공통/기준정보 코드 조회 공통함수 |
| Log.js | 로그 관련 공통 함수 |
| Message.js | 메시지 처리 공통함수 |
| Override.js | Nexacrolib override js |
| Work.js | 업무공통 라이브러리 |
| Popup.js | 팝업관련 공통함수 |
| Transaction.js | 데이터 서비스(transaction) 관련 공통함수 |
| Util.js | Util 관련 공통함수 |
| Validation.js | 정합성체크 관련 공통함수 |

### 2.1.6 공통 팝업

애플리케이션에서 공통으로 적용되는 공통팝업 화면을 관리합니다.

|  |  |
| --- | --- |
| 화면id | 설명 |
| cmmAlert.xfdl | 메시지 창 공통(alert) |
| cmmCalFromTo.xfdl | 기간 달력 공통 폼 |
| cmmCalMM.xfdl | 월력 공통 폼 |
| cmmCalMMFromTo.xfdl | 월력(기간) 공통 폼 |
| cmmColumnHide.xfdl | 그리드 공통(컬럼 숨김) |
| cmmConfirm.xfdl | 메시지 창 공통(confirm) |
| cmmDebug.xfdl | 개발자 디버깅 공통 폼 |
| cmmFileDownload.xfdl | 파일업로드(멀티) 공통 |
| cmmGridFind.xfdl | 그리드 찾기 기능 |
| cmmGridTotal.xfdl | 그리드 합계 공통 폼 |
| cmmPaging.xfdl | 페이징 공통 폼 |
| cmmFileUpDown.xfdl | 파일업로드(멀티) 공통 |
| cmmGridTotal.xfdl | 그리드 합계 공통 |
| cmmPaging.xfdl | 페이징 공통 함수 |
| cmmPdfViwer.xfdl | Pdf viwer 공통 팝업 |
| cmmReportViewer.xfdl | 리포트 팝업 |
| cmmWebEditor.xfdl | 웹 에디터 팝업 |

# 3. 개발가이드

## 3.1 개발절차

### 3.1.1 넥사크로 스튜디오 실행

바탕화면 및 프로그램의 메뉴의 [TOBESOFT Nexacro Studio(x64)] 아이콘을 더블클릭 하여 넥사크로 스튜디오를 실행합니다.

### 3.1.2 프로젝트 오픈

메뉴[ File> Open> Prject ] 또는Start Page의 Project Open 버튼을 클릭하여 프로젝트 파일(hrpfe.xprj)

을 선택하여 프로젝트를 오픈 합니다.

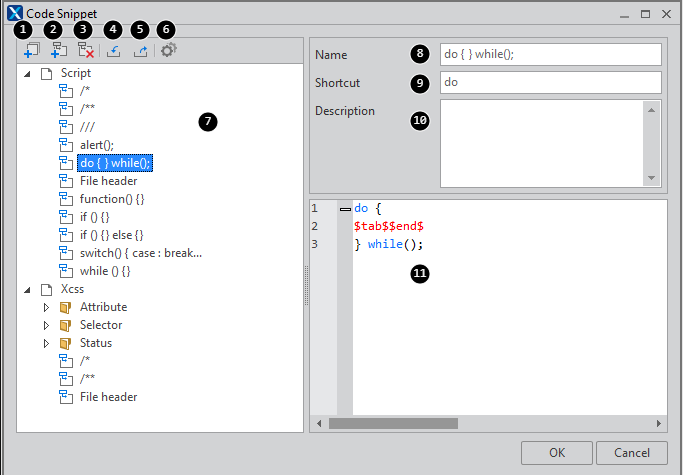
### 3.1.3 코드 스니펫(Snippet)

코드 스니펫은 사용자가 미리 정의한 코드 조각을 의미합니다. 자주 사용하는 코드를 미리 정의하고 파일 편집 시 가져다 사용할 수 있는 기능입니다..

1. 코드 스니펫 정의

리본 메뉴 [EDIT - Assist - Code Snippet] 또는 메뉴 [Assist > Code Snippet] 항목을 선택한 후 호출된 창에서 스니펫을 정의할 수 있습니다.

기본적인 코드 스니펫은 넥사크로 스튜디오에서 제공하며, 원하는 항목을 추가할 수 있습니다.



|  | **이름** | **설명** |
| --- | --- | --- |
| 1 | Add Group | 코드 스니펫 그룹을 추가 |
| 2 | Add Code Snippet | 새로운 코드스니펫을 추가 |
| 3 | Delete Code Snippet | 선택된 코드스니펫을 삭제 |
| 4 | Import Code Snippet | XML 파일로 생성된 코드 스니펫을 가져오거나 xpackage 파일을  선택하고 Import Wizard를 실행합니다. |
| 5 | Export Code Snippet | 코드 스니펫을 XML 파일로 저장합니다 |
| 6 | Options | 코드 스니펫 파일 위치를 설정합니다. |
| 7 | Code Snippet List | 코드 스니펫 목록 |
| 8 | Name | 코드 스니펫의 이름 정의 |
| 9 | Shortcut | 코드 스니펫을 호출할때 사용할 바로가기 문구 정의  (XCSS 또는 Script 편집창에서 Shortcut으로 지정된 문구 입력 시  코드 스니펫 반영) |
| 10 | Description | 코드 스니펫에 대한 설명 정의 |
| 11 | Code Snippet Source | 적용될 코드 스니펫 정의 |

2) 코드 스니펫 예약어

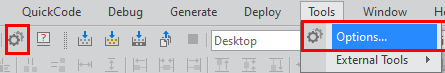
스니펫 코드 작성 시 아래와 같은 예약어를 사용할 수 있습니다. 예를 들어 $DATE$를 입력하면 오늘 날짜를 표기하는 문자열로 변환해 입력합니다.

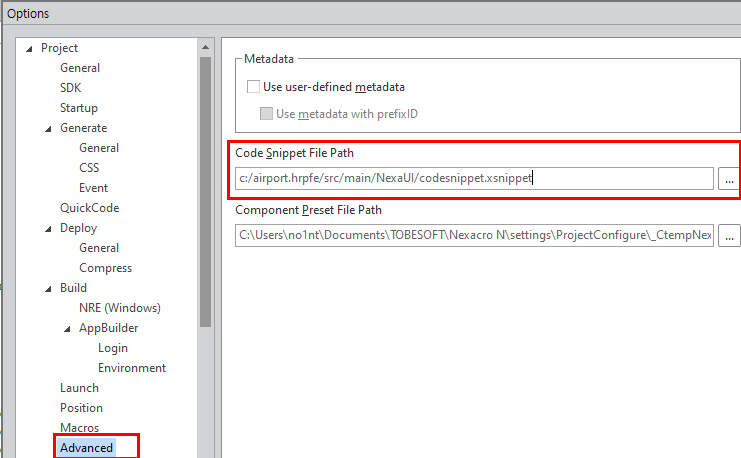
| **reserved string** | | **설명** | | |
| --- | --- | --- | --- | --- |
| General | $end$ | | | 코드 스니펫을 적용한 후 커서의 위치 |
|  | $selected$ | | 현재 선택된 문자열. 선택된 항목이 없는 경우에는 표시하지 않습니다. | |
| $clipboard$ | | 클립보드에 복사된 문자열 | |
| $tab$ | | 탭 문자 | |
| File | $FILE$ | | | 파일명을 포함한 파일 경로 |
|  | $FILE\_EXT$ | | 파일 확장자 | |
| $FILE\_BASE$ | | 확장자를 뺀 파일명 | |
| $FILE\_PATH$ | | 파일 경로 | |
| $FILE\_EXT\_UPPER$ | | 파일 확장자 (대문자) | |
| $FILE\_BASE\_UPPER$ | | 확장자를 뺀 파일명 (대문자) | |
| $FILE\_PATH\_UPPER$ | | 파일 경로 (대문자) | |
| Date | $DATE$ | | | 년/월/일 형식 (yyyy/MM/dd) |
|  | $DATE\_REVERSE$ | | 로케일 형식에 따른 현재 일자를 표기 | |
| $DAY$ | | 현재 일자를 숫자로 표기 | |
| $DAY\_02$ | | 현재 일자를 "dd" 형식으로 표기 | |
| $DAYNAME$ | | 로케일 형식에 따른 현재 요일을 짧게 표기 (Sun, Mon..) | |
| $DAYLONGNAME$ | | 로케일 형식에 따른 현재 요일을 표기 (Sunday, Monday..) | |
| $MONTH$ | | 현재 월을 숫자로 표기 | |
| $MONTH\_02$ | | 현재 월을 "MM" 형식으로 표기 | |
| $MONTHNAME$ | | 로케일 형식에 따른 현재 월을 짧게 표기 (Jan,Feb..) | |
| $MONTHLONGNAME$ | | 로케일 형식에 따른 현재 월을 표기 (January,February..) | |
| $YEAR$ | | 현재 연도를 숫자로 표기 | |
| $YEAR\_02$ | | 현재 연도를 "yy" 형식으로 표기 | |
| Time | $HOUR$ | | | 현재 시간을 숫자로 표기 |
|  | $HOUR\_02$ | | 현재 시간을 "HH" 형식으로 표기 | |
| $MINUTE$ | | 현재 시간을 "mm" 형식으로 표기 | |
| $SECOND$ | | 현재 시간을 "ss" 형식으로 표기 | |

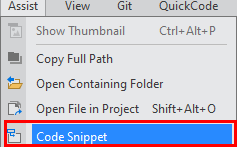
※ 참고 : [http://docs.tobesoft.com/development\_tools\_guide\_nexacro\_n\_ko/006f4682fdb92150#d147d1117e1eb53b](http://docs.tobesoft.com/development_tools_guide_nexacro_n_ko/006f4682fdb92150%23d147d1117e1eb53b)

3) 프로젝트 등록

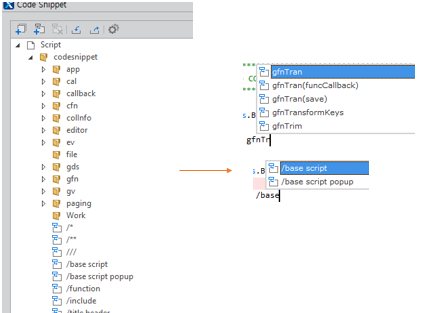
[ 메뉴 탭 ] [ Script탭 ]



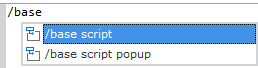
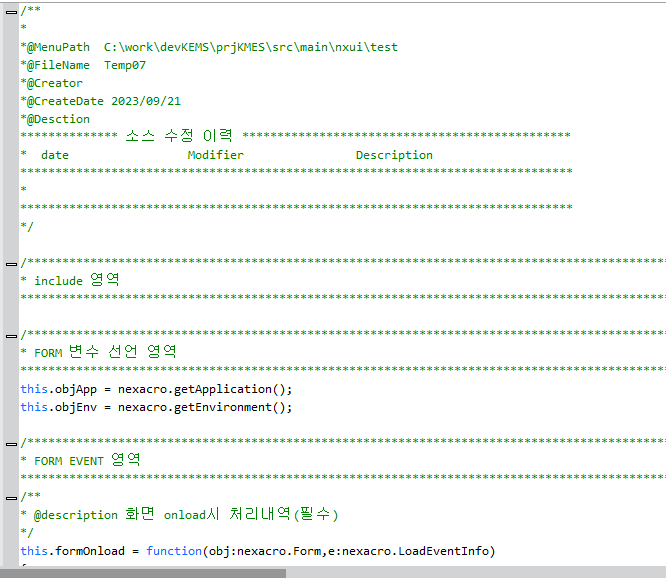




* 실행은 등록된 폴더 이름으로 실행하여 사용합니다.

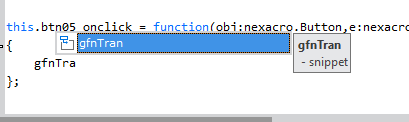


4) 사용방법

1. 기본 스크립트 생성개발자 개발하기 필요한 기본 fucntion을 생성 해줍니다.  
    명령어 : /base script   
     
   
2. 주석관련  
   주석관련 snippet는 /로 시작하면 아래 표와 같습니다.

|  |  |
| --- | --- |
| 단축어 | 설명 |
| /base script | 업무화면 기본scrpt |
| /base script popup | 팝업화면 기본script |
| /title header | 프로젝트 타이틀 |
| /function | 공통 function |

1. 공통함수(LibCommon)  
   공통함수 snippet는 gfn 으로 입력하여 해당 라이브러리를 사용 합니다.



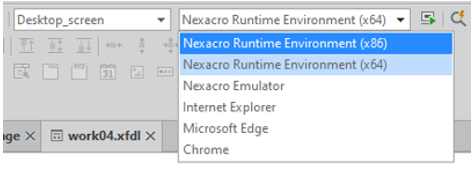
### 3.1.5 프로젝트 실행

넥사크로 스튜디오에서 편집한 결과물을 넥사크로에서 실행할 수 있습니다. 프로젝트 실행은 앱을 실행하는

Launch기능과 Form만 실행하는 QuickView 기능으로 나뉩니다.

(1) 실행모드

리본 메뉴[HOME - Launch] 또는 툴바에서 Browser Combo 항목을 선택하여 실행 모드를 설정할 수 있습니다.



1) NRE(Nexacro Runtime Environment)

32비트 또는 64비트 넥사크로 런타임 실행환경(NRE)으로 실행합니다. nexacro.exe 파일을 통해 실행되며 웹서버 없이 생성된 파일에 직접 접근하여 앱을 실행할 수 있습니다.

2) 웹브라우저

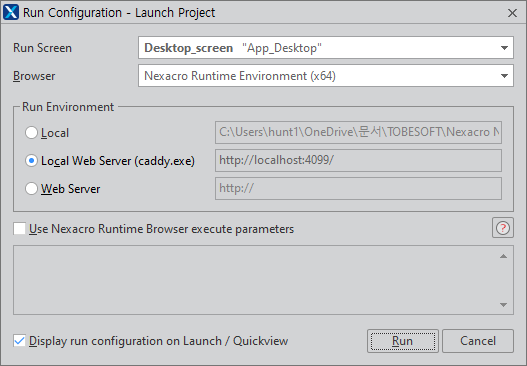
웹브라우저에서 앱을 실행할 수 있습니다. 사용자 디바이스에 설치된 웹브라우저에서 앱을 실행합니다. 모바일 디바이스를 사용하는 경우에는 URL을 직접 입력해 실행할 수 있습니다.

(2) Launch

넥사크로 앱을 실행하는 기능입니다. ScreenDefinition을 먼저 설정하고 실행모드를 선택합니다.

메뉴[Generate > Launch] 또는 단축키 (Ctrl + F5)로 실행할 수 있습니다. 웹 브라우저로 실행 할 경우

URL을 입력하는 창이 팝업되고 입력한 URL을 선택한 웹 브라우저에서 실행합니다. 로컬 웹 서버를 사용하는 경우에는 메뉴 [Tools > Options > Project > Launch] 항목에서 지정한 "Server Host Url"이 URL로 지정됩니다.



(3) Quick View

활성화된 뷰가 Form 편집 뷰일 경우 QuickView를 실행할 수 있습니다. 또는, Project Explorer가 활성화된 상태에서 선택된 트리 아이템이 Form 파일일 경우에도 QuickView를 실행할 수 있습니다.

메뉴[Generate > QuickView]로 실행할 수 있습니다.

## 3.2 기능별 개발방법

### 3.1.1 Transaction

서버와 데이터를 송/수신 시 사용하는 함수로 서비스 호출 URL, input/out Dataset, 콜백함수, 통신방식 등을 설정할 수 있습니다. 공통함수로 , 형태로 데이타셋을 처리 할 수 있습니다.  
snippet : gfnTran

*/\*\**

*\* @class 서비스 호출 공통함수 <br>*

*\* Dataset의 값을 갱신하기 위한 서비스를 호출하고, 트랜젝션이 완료되면 콜백함수을 수행하는 함수*

*\* @param {string} id : 서비스 ID*

*\* @param {string} url - 서비스 URL*

*\* @param {string} inputdataset*

*\* @param {string} output dataset(생략가능)*

*\* @param {string} arguments (생략가능)*

*\* @param {function}{String} [callBackFnc] - 콜백 함수명*

*\* @param {Boolean} [isAsync] - 비동기통신 여부 (생락가능 기본:async)*

*\* @return*

*\* @example*

*\*/*

*var id = "svcSearch";*

*var url = "/login.do";*

*var inds = "inSys1=dsUser";*

*var outds = "dsOut=sysOut";*

*var callback = "fnCallback"; // 생략가능*

*this.gfnTran(id, url, inds, outds, callback);*

- Application 성능 향상을 위해 통신방식은 Async(비동기) 통신을 사용하고 응답 후 처리는 콜백함수에

기술해야 합니다. 여러개의 통신을 여러번 호출하는 것보다 1개의 통신에 여러개의 input/out Dataset을 보낸 후 서버에서 일괄 처리하면 Application 성능이 향상될 수 있다.

### 3.1.2 데이터 바인딩

바인딩(Binding)이란 데이터 컴포넌트와 표현(Presentation) 컴포넌트를 스크립트 코딩 없이 연결하여 데이

터를 입출력 하거나 컴포넌트를 동작시키는 방법 입니다.

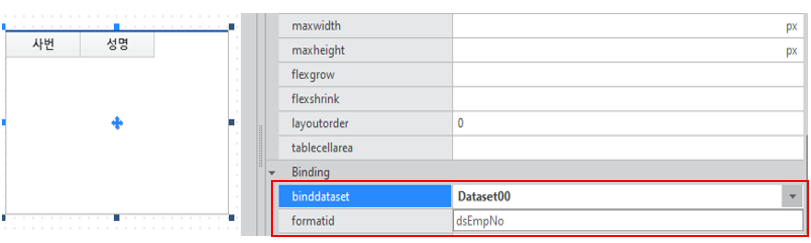
화면의 스크립트 양을 상당부분 줄여주고, 출력 속도를 높일 수 있으며, 임의의 데이터를 넣어서 동작을 확인하

면서UI/UX 설계도 할 수 있는 기능입니다.

(1) 그리드 바인딩

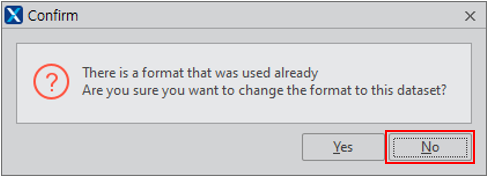
Dataset과 Grid간의 바인딩 처리를 합니다.

1) 그리드 포맷을 유지한 데이터 바인딩



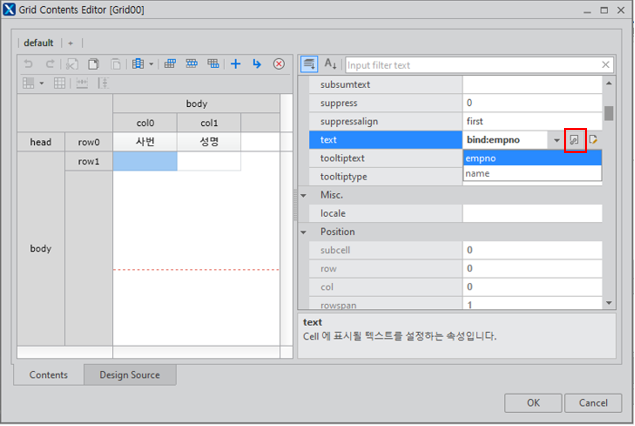
그리드 컴포넌트 Properties창의 binddataset 속성에 Dataset을 지정합니다.

아래와 같은 확인 창에서 No를 선택하여, 기존에 작성한 Grid 포맷을 그대로 사용합니다.



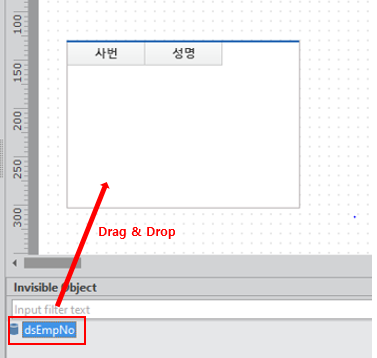
그리드를 더블 크릭하여 그리드 편집창을 오픈 합니다.

body의 컬럼 정보를 선택 후 text속성을 데이터셋 컬럼과 연결합니다. (모든 컬럼 동일 적용)

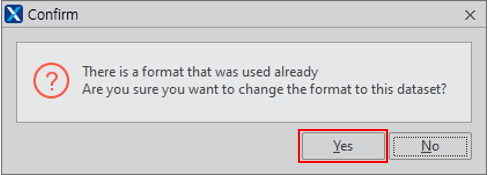


2) 그리드 포맷을 새로 생성하여 데이터 바인딩

작업중인 화면의 Invisible Object 데이터셋을 선택하여 그리드에 Drag & Drop하여 바인딩 합니다.



아래와 같은 확인 창에서 Yes를 선택하여, 데이터셋 컬럼 기준으로 그리드 포맷을 다시 생성합니다.

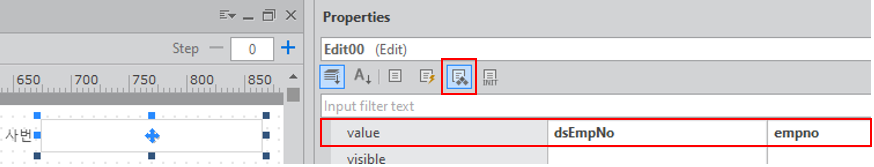


작업중인 화면의 Objects 항목에 데이터셋 목록이 표현 됩니다.

데이터셋을 선택하여 그리드에 드래그앤드롭하여 적용합니다.

(2) 컴포넌트 바인딩

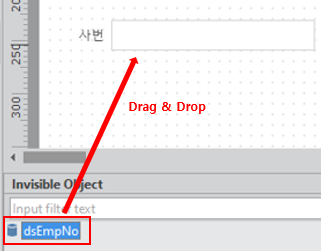
1) 바인딩 속성을 이용한 바인딩



선택한 컴포넌트의 속성창의 Bindable 아이콘을 선택합니다.

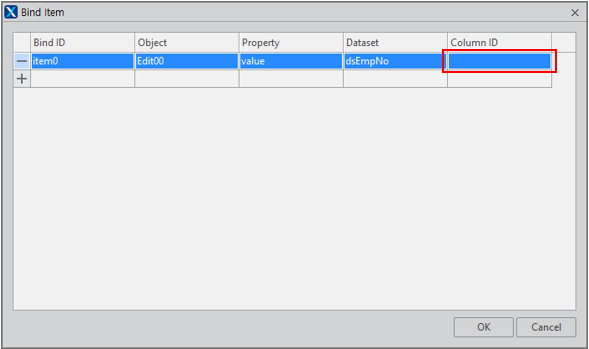
value속성에 데이터셋을 연결 후 데이터셋 컬럼을 지정합니다.

2) Drag & Drop을 이용한 바인딩



화면의 Invisible Object 데이터셋을 선택하여 컴포넌트에 Drag & Drop하여 바인딩 합니다.

컴포넌트와 데이터셋 컬럼과의 바인딩 정보창이 오픈 됩니다.



컴포넌트와 연결할 데이터셋 컬럼을 선택 연결 후 OK버튼으로 적용합니다.

### 3.1.3 팝업 처리

팝업 종류별 (모달, 모달리스)로 공통함수(gfnOpenPopup)를 사용하여 팝업을 호출 합니다

(1) 팝업 종류별 차이점

| 팝업 종류 | 처리방식 | 콜백함수 | 프레임 밖 이동 |
| --- | --- | --- | --- |
| modal | aSync | 유 | 불가능 |
| Modeless | aSync | 유 | 가능 |

(2) 팝업 옵션

| 팝업 옵션 | 설명 | 팝업 옵션 | 설명 |
| --- | --- | --- | --- |
| oOption.top | 상단좌표 | oOption.left | 좌측좌표 |
| oOption.width | 넓이 | oOption.popuptype | 팝업종류(modal, modeless) |
| oOption.title | 팝업 타이틀 | oOption.layered | 투명 윈도우 |
| oOption.opacity | 투명도 | oOption.autosize | autosize |

(3) 팝업   
 - snippet : /gfnShowModal(modal)

/gfnOpen(modeless)  
 - sample : sample::sample003Popup.xfdl

*//파업호출*

*/\*\**  Modal

*var args = {};*

*args.id = "popId"; //필수*

*args.url = "sample::sample0031PopupP.xfdl"; //필수*

*args.titletext = '샘플팝업'; //필수*

*args.paramCode = "abcd"; // paremeter*

*this.gfnShowModal(args,*

*(resVar)=>{ //callback*

*});*

*//modeless*

*var args = {};*

*args.id = "popModeless"; //필수*

*args.url = "sample::sample0031PopupP.xfdl"; //필수*

*args.titletext = '샘플팝업'; //필수*

*args.paramCode = "abcd"; // paremeter*

*this.gfnOpen(args);*

### 3.1.4 메시지 처리

공통 메시지 팝업을 통해 사용자에게 알림, 확인창을 호출할 수 있습니다.

사용할 메시지는 메시지 DB에 등록되어 있어야 합니다.

(1) 알림 팝업 메시지

메시지 정보의 TYPE이 ‘A’ (알람)인 메시지는 알림 메시지 팝업으로 처리 됩니다.

(2) 다이얼로그 팝업 메시지

메시지 정보의 TYPE이 ‘C’ (Confirm)인 메시지는 다이얼로그 메시지 팝업으로 처리 됩니다.

팝업 메시지는 공통함수 *gfnAlert함수를* 이용하여 아래와 같이 작성 합니다.

*/\*\**

*\* @class : 공통 메세지팝업오픈*

*this.gfnAlert("code",[param]);*

*return;*

*또는*

*this.gfnAlert("code",(yes)=>{*

*});*

### 3.1.5 Calendar

(1) 기본 Calendar

기본으로 생성되는 Calendar 컴포넌트 입니다.



(2) Calendar Type

Calendar가 화면에 표시되는 형식을 변경합니다.

| Property | Value | 설명 |
| --- | --- | --- |
| Type | normal | 날짜 입력영역과 팝업달력 드롭다운 버튼이 있는 Calendar 가 표시 |
| spin | 날짜 입력영역과 날짜값을 변경하는 스핀버튼이 있는 Calendar 가 표시 |
| monthonly | 일반적인 월 단위 달력형태의 Calendar 가 표시 |

(3) DateForamt





## 3.3 개발시 유의사항

3.1.1 Column Type

Dataset의 Column을 이용한 수식 계산 시 Type을 사용합니다.

*//그리드 합계*

*getSum(“SALARY”) => getSum("nexacro.toNumber(SALARY)");*

*//Sort 기능*

*Sort는 Column Type이 기준이 되며, 형 변환은 불가 합니다.*

3.1.2 이벤트

onload, onrowsetchanged, onrowposchanged 등 Event객체에 e.reason이 있는 경우 반드시 분기 처리

합니다.

*this. ds\_emp\_onrowposchanged = function(obj:nexacro.NormalDataset,e:nexacro...)*

*{*

*if(e.reason == 12) { // append*

*}*

*else if(e.reason == 20) { // delete*

*}*

*else if(e.reason == 30) { // sort*

*}*

*};*

3.1.6 enableevent

for문 등을 이용하여 여러 건의 컴럼 값을 변경하고자 할 때 enableevent 속성을 이용하여 Dataset의 이벤트를 멈추고 변경해야 합니다.

set\_enableevent(false); 를 통해 Dataset 이벤트를 멈추었다면 스크립트 수행 후에는 반드시

set\_enableevent(true); 를 이용하여 Event를 다시 실행해 주어야 합니다.

*Dataset명.set\_enableevent(false);*

*for(var i=0;i<100;i++)*

*{*

*Dataset명.setColumn(i, ”컬럼”, ”값“);*

*}*

*Dataset명.set\_enableevent(true);*

for 문을 수행하면서 Dataset의 컬럼 값을 변경하게 되면 Nexacro 엔진단에서는 해당 Dataset과 연결되어 있는 컴포넌트를 대상으로 repaint하는 기능을 수행하게 됩니다.

따라서 Dataset의 컬럼이 변경되면 시각적으로 표현되지 않지만 내부 엔진을 여러가지 기능을 수행하게 되며, 성능에 영향을 줄 수 있습니다.

3.1.7 Dataset layout

서비스 호출 시 데이터가 없는 경우에도 반드시 Layout이 필요합니다.

서비스 호출 시 서버에서 받아온 Dataset으로 치환됩니다.

3.1.8 eval 함수 대체

eval은 어떤 코드라도 무조건 실행하기 때문에 보안 문제도 있다. 따라서 신뢰하지 못하거나 모르는 코드가 포함되어 있을 경우 절대로 사용해서는 안됩니다.

*var arrObj = eval("this.Tab00.Tabpage1.form").all;*

*=> var o = new Function("return this.Tab00.Tabpage1.form").call(this);*

*var arrObj = o.all*

# 4. 개발 관련 참조 사이트

투비소프트 기술지원 : http://support.tobesoft.co.kr

공식가이드 : https://docs.tobesoft.com/nexacro\_n\_v24\_ko

넥사크로 커뮤니트 사이트(play nexacro) : <http://www.playnexacro.com>