3DEXPERIENCE®

개발 표준
(Customization Ground Rule)

DSIS Korea
June, 2019

목차

현 개발표준 소개

Customization 이슈 사례

개발표준 권장 및 주의사항

3DEXPERIENCE®

오브젝트 Naming 규칙

Naming 규칙

- Naming Rule 표준을 정의 및 기술하여 유지보수, 개발 소스 품질, 가독성 향상

- ENOVIA 기반 개발 시 시스템 전체에서 사용될 대표 Trigram 를 프로젝트 용어에 맞게 정의
  - **D**ongDaeMun **I**nfrastructure **G**lobal PDM = **DIG**

- New 오브젝트 생성 및 기존 오브젝트 확장 시 프로젝트 Trigram 를 Prefix 로 사용

Symbolic Naming 규칙

- Data Schema 에 해당하는 Administrative Object 는 반드시 Symbolic Name 를 생성하여 사용

- Administrative Object Type Name 과 Administrative Object Name 를 Underscore 로 구분 및 조합하여 사용
  - type_part, attribute_UnitOfMeasure

- Administrative Object Name 에 공백 혹은 Underscore 존재 시 삭제

Data Schema Object

오브젝트	정의	예시
Type Attribute	Project Trigram 과 Type 이름 조합	DIGPart, DIGRawMaterial DIGWeight
Policy	Project Trigram 과 Policy Type 이름 조합. 동일 Type 에 여러 Policy 적용시, Policy 뒤에 의미전달 단어조합 사용.	DIGECO DIGECOForOption
Relationship	Project Trigram 과 Reliationship 이름 조합. Relationship 의 경우 기본적으로 From / To 조합으로 정의	DIGPartToDocument

Dynamic UI Object

오브젝트	정의	예시
Web Form	Type 의 Symbolic Name 를 기준, Project Trigram 를 Prefix, Web Form 의 용도를 Suffix 로 조합하여 사용. Field 의 경우 Form 안에서 Unique 하게 정의	DIGtype_Part_CreateForm (생성) DIGtype_Part_EditForm (조회/수정) DIGtype_Part_SearchForm (검색)
Menu	Project Trigram 과 메뉴 용도를 조합	DIGPartMyDesk (My Desk 메뉴)
Command	Project Trigram 과 Command 용도 조합	DIGRemoveSelectedDocumentCmd DIGPartRelatedSpecLink
Portal Channel	Project Trigram 과 Portal 혹은 Channel 이름 조합	DIGProjectIssuePortal DIGProjectIssueChannel

기타

오브젝트	정의	예시
Java Package	Track Name 과 Domain (Module) Name 를 바탕으로 패키지 구조 기술 * Track : pdm (GPDM), cat (CATIA)... * Module : product, cad, bom...	com.dig.pdm.product.bean com.dig.pdm.product.servlet
JSP	OOTB 의 경우 Customer Filter (web.xml) 에 정의된 Prefix 를 추가. 신규인 경우 Project Trigram, 도메인 그리고 JSP 파일 용도의 조합으로 구성	DIG_emxpartCreatePart.jsp DIGCadCreateNewPartProcess.jsp (Business logic 처리용 jsp)
검색	검색를 위해 추가되는 ELEMENT 의 경우 Project Trigram 와 ELEMENT 를 Underscore 조합으로 config.xml 에 정의	DIG_FieldName

3DEXPERIENCE®

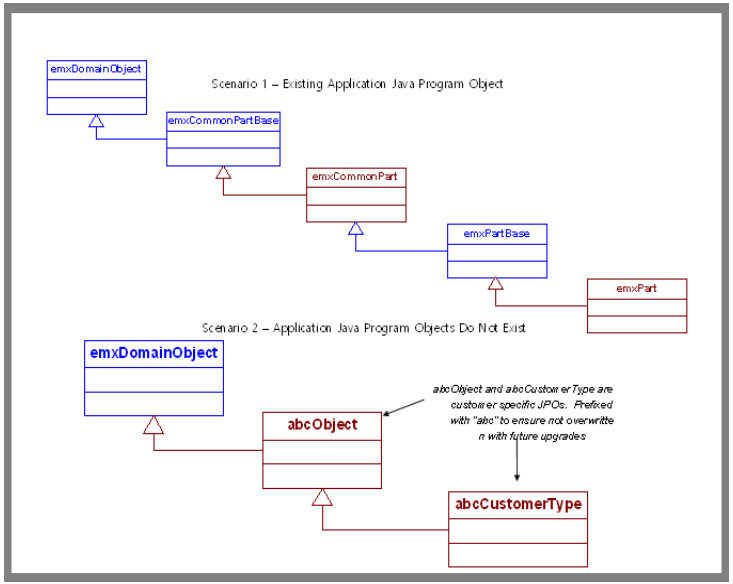
ENOVIA 확장 및 변경

ENOVIA 확장 및 변경 - 일반

오브젝트	확장 및 변경
Data Schema Object	기존 기능 확장의 경우 Type, Relationship 은 상속하여 수정하며, Policy 의 경우는 직접 수정. 새로운 기능의 경우 Naming Rule 에 의거 새롭게 생성.
Dynamic UI Object	OOTB 의 경우 원본 파일에 “_OOTB” suffix 를 추가 후 복사본 수정.
JSP	OOTB 파일 수정 시 OOTB 복사본에 Customer Filter (web.xml) 에 정의된 Prefix 추가하여 수정. 다만 “<Include>” 구문으로 포함된 JSP 파일들은 Filter 적용이 불가능하여 직접 수정하나, JSP 파일을 Include 하는 JSP 의 경우 수정이 없더라도 Customer Filter 를 적용하여 수정이 이루어졌는지는 식별할 수 있도록 함.

ENOVIA 확장 및 변경 - JPO

- 수정을 위해 파생된 JPO 함수에 OOTB 함수를 Override 하여 로직처리



Part 의 경우 “emxPartBase” 가 기본 클래스, “emxPart“ 가 파생된 클래스임.

OOTB 의 Business Logic 변경 시 파생 클래스를 Override 하여 처리.

ENOVIA 확장 및 변경 - Javascript

- 기존 수정 혹은 새롭게 추가시, 코드의 맨 마지막 부분에 주석 및 코드 추가
- WebFom Custom Validation 추가

항목	변경	예시
Common Custom JS	“emxSystem.properties” 파일의 “eServiceSuiteFramework.UIForm.ValidationFile” 에 파일 정의 후 변경사항 기술	eServiceSuiteFramework.UIForm.ValidationFile=scripts\CustomUIFormValidation.js
Application Specific Custom JS	“emxSystem.properties” 파일의 “eServiceSuite<Application-Name>.UIForm.ValidationFile” 에 파일 정의 후 변경사항 기술	eServiceSuiteEngineeringCentral.UIForm.ValidationFile=CustomAppsUIFormValidation.js

ENOVIA 확장 및 변경 – Property 파일

- “emxSystem.properties” 파일에 각 모듈에 사용하는 Property 파일 정의 및 수정.

예시
eServiceSuiteEngineeringCentral.ApplicationPropertyFile=emxEngineeringCentral.Properties eServiceSuiteEngineeringCentral.PropertyFileAlias=emxEngineeringCentral.Properties

ENOVIA 확장 및 변경 – String Resource

- OOTB 로 제공되는 Lable 혹은 Message 변경은 OOTB String Resource 파일에 변경할 String Resource Key 값을 주석 및 복사 후 수정
- 브라우저 언어설정에 기반 다국어 기능 지원

오브젝트	변경
Data Schema Object (Type, Attribute, Policy..)	emxFrameworkStringResource.properties 파일 수정.
Label 및 Message	emxSystem.properties 파일에 각 모듈에서 사용하는 StringResource 파일 정의 * Engineering Central 에서 사용하는 StringResource eServiceSuiteEngineeringCentral.StringResourceField=emxEngineeringCentralStringResource

목차

현 개발표준 소개

Customization 이슈 사례

개발표준 권장 및 주의사항

Customization 이슈 및 해결 사례 - 1

	이슈	원인 및 해결
1	“Field or Column XXX” does not exist	WebForm 혹은 Table 오브젝트의 Column 를 삭제하여 발생. 삭제항목을 OOTB 에 추가 후, “Access Exception” 설정을 통하여 Display 되지 않도록 처리.
2	“Item XXX” not in menu	Menu 오브젝트 Component 삭제하여 발생. 삭제된 Component 추가 및 Patch 이후 해당 Component 삭제.
3	“STATE_NAME” does not exist	Policy 에 정의된 Symbolic Name 과 매핑된 State 와 일치하지 않거나, State 가 존재하지 않음. 매핑되지 않은 State 이름을 추가 혹은 삭제된 State 이름을 추가 및 Patch 이후 삭제.

Customization 이슈 및 해결 사례 - 2

	이슈	원인 및 해결
4	“InstallXXXX.err” 파일 크기이상	에러가 없음에도 불구하고, Error 파일에 메세지가 출력되어 Rollback 되는 현상. 메세지 출력의 원인 사항 (쉘 스크립트의 echo 명령, log 레벨 등) 제거
5	호환성 이슈	Upgrade 이후 Administrative Object 객체 등이 기존 제품과 호환되지 않거나, 기존 Customization 기능이 작동되지 않음. 기능 오류부분의 수정이 요구되며, 빠른 원인파악 및 수정을 위한 Customization 변경사항 및 Impact 사항 정리 요구됨.

목차

현 개발표준 소개

Customization 이슈 사례

개발표준 권장 및 주의사항

개발표준 권장사항 - 1

	항목	설명
1	Data Model 변경	OOTB 에서 제공되는 Model 의 경우 Unified Typing Tool 를 사용하여 확장. New Model 인 경우 MQL 를 사용하여 확장
2	JSP Custom Filter 적용	OOTB JSP 수정시 web.xml 에 정의된 Custom Filter Prefix 를 통해 생성하도록 권장, 즉 기존 OOTB 소스를 유지하면서 커스터마이징 실행. “<Include>” 구문으로 포함된 JSP 파일들은 Filter 적용이 불가능. 직접 수정하나 JSP 파일을 Include 하는 JSP 의 경우 Customer Filter 를 적용하여 수정이 이루어졌는지는 식별할 수 있도록 함.

개발표준 권장사항 - 2

	항목	설명
3	JavaBean 사용	Rest 인터페이스 혹은 Batch 실행을 위한 JavaBean 를 사용하되 그외에는 사용하지 않고 JPO 사용함을 권장
4	JPO 확장	JPO 확장 시 파생 클래스에 변경될 로직을 Override 하여 기술하는게 원칙. 그외 커스텀으로 추가되는 함수가 존재시 파생 클래스를 상속받아 추가함을 권장. 특히 커스텀 함수가 많을 시 파생 클래스에 Override 명칭이 없을 시 실제 OOTB 를 Override 한것인지 판단이 어려우며, 해당 커스텀 함수들로 파생 클래스가 산재되어 있음.
5	Indented Table 사용	Indented Table 사용시 대량의 데이터의 Display 시 퍼포먼스 이슈 문제 존재. 데이터 Display 처리 로직설계 고려 요구됨.

개발표준 주의사항 - 1

	항목	설명
1	ENOVIA 예약어 사용	Administrative Object 정의시 예약어 (emx, eService, DMC ENC 등) 사용금지
2	WebForm 혹은 Table 오브젝트의 컬럼 삭제	해당 컬럼을 삭제하지 않고, 반드시 “Access Expression” 설정을 통하여 화면에서 보이지 않도록 설정
3	테이블 컬럼값 로직처리	테이블 컬럼값을 Program 를 통하여 Display 시, 한번에 값을 처리할 수 있도록 API 를 통한 처리 요구됨. 간혹 매번 Loop 를 돌면서 처리하는 경우가 존재

개발표준 주의사항 - 2

	항목	설명
4	JPO 함수호출	JPO 메소드에서 타 JPO 함수 호출시 Invoke 를 사용하여 호출하지 않음. 일반 Class 처럼 Instantiate 후 타 JPO 함수 호출.
5	3rd Party 라이브러리 사용	jQuery, POI Library 등 3rd Party 라이브러리 사용은 원칙적으로 금지. 기능구현에 반드시 필요한 경우 Domain 혹은 Tech Leader 에게 사용여부 확인 후 진행
6	Deprecated 사용	Deprecated 함수는 경우 반드시 사용하지 않음
7	Javascript 함수	특정 브라우저에서만 지원하는 함수를 반드시 사용하지 않음