

PAP 기능 정의 (2차)

Process DB ver1.0 및 UI

추가: 제어 데이터 비교 체크 개념으로 (연계값 vs FEED 값)
즉 FEED에 제어용 데이터가 있는 경우 MDBS로 같이
인식해서 비교함
데이터 소스가 중요함

공정설계팀/화공기술센터

2020.04.17

0. Table 종류 및 설명

이름	설명	테이블 특징	데이터 특징
Component	사용되는 물질들의 총 목록	- 고정	전 사용자 공통 (서버)
Dictionary	자동 매핑을 위한 유의어사전	- 별도 UI로 레코드 Insert/Update - Default 데이터 제공 (업데이트 중)	전 사용자 공통 (서버)
HMB_Composition	표준 HMB 구조 (물질)	- 외부 파일 (xlsx, csv, sql 등) 에서 Insert	PJT별
HMB_Condition	표준 HMB 구조 (공정조건)	- 외부 파일 (xlsx, csv, sql 등) 에서 Insert	PJT별
Instrument	제어 계기 리스트 및 데이터	- 외부 파일 (xlsx, csv, sql 등) 에서 레코드Insert - 나머지 필드는 연계 (TBD)	PJT별
Line	배관 라인 리스트 및 데이터	- 외부 파일 (xlsx, csv, sql 등) 에서 레코드Insert - 나머지 필드는 연계 (TBD)	PJT별
Material	사용되는 재질들의 총 목록	- 레코드 데이터 계속 축적 중 - PMC 필드는 는 별도 UI 로 Update	전 사용자 공통 (서버)
Special Service	특수 Line 의 조건 정의	- 별도 UI로 레코드 전체 Insert/Update	PJT별
Symbol Type	작화 모듈에 사용되는 심볼들	- 고정 - Connection point Table 은 새로 생성 필요	전 사용자 공통 (로컬?)
Unit	항목별 단위 모음	- 고정 (업데이트 중)	전 사용자 공통 (로컬?)
eq_///	기기별 데이터	- 외부 파일 (xlsx, csv, sql 등) 에서 Insert	PJT별
usersetting_///	PJT별 단위/물질 선택 목록	- 행 개수 고정 - 별도 UI로 행 데이터 전체 Insert/Update	PJT별

1.0 업로드 기능 - General

- (1) 업로드 대상 파일 형식은 xls, csv, sql **혹은 서버 데이터 연동 (TBD)**
- (2) 업로드한 데이터를 임시 table 로 생성해서 standard DB (process DB) 에 일정한 logic 으로 연동해서 데이터를 옮기는 방식
- (3) 업로드한 날짜/사용자 이름/IP등 으로 데이터 추적관리 필요
- (4) 업로드 후의 process DB 들은 모두 user 가 별도 UI로 개별로 값 수정 가능해야 함
(값 수정 가능/불가능 필드들은 추후 안내, 지금은 모두 가능하게 유지)
→ HYTOS 의 하단부분 stream data 수정하듯이

1.1 업로드 기능 - HMB

(1) 파일 선택 (xls, csv, sql) **혹은 서버 데이터 연동 (TBD)**

- 대상: MDBS로 인식해서 추출된 HMB



HMB 예시

(2) 임시 Table 로 import 후 행/열 구조 검사

- 행: stream no (숫자 / 숫자+문자 / 문자+숫자 / 공백)

- 열: attribute (문자 / 공백)

Stream List 업로드

(3) 행 개수만큼 레코드 생성

(4) 열에 있는 attribute 단어를 취득 후 → Dictionary 를 거쳐서 → 표준 attribute 단어로 변환

(5) 변환된 표준 attribute 단어가 위치한 HMB_Condition 에 → 각 레코드 별 데이터 입력

“Auto Mapper”

(6) 변환된 표준 attribute 중 물질로 분류되는 것들은 (IsComponent=1)

→ usersetting_component table 을 거쳐서 component 번호를 알아낸 뒤 (1~20)

→ HMB_Composition table 에 데이터 입력

이름	설명	테이블 특징	데이터 특징
HMB_Composition	표준 HMB 구조 (물질)	- 외부 파일 (xlsx, csv, sql 등) 에서 Insert	PJT별
HMB_Condition	표준 HMB 구조 (공정조건)	- 외부 파일 (xlsx, csv, sql 등) 에서 Insert	PJT별

1.2 업로드 기능 - Line

(1) 파일 선택 (xls, csv, sql) **혹은 서버 데이터 연동 (TBD)**
 - 대상: SPPID 의 piperun attribute report

(2) 임시 Table 로 import 후 행/열 구조 검사
 - 행: line no (숫자 / 문자)
 - 열: attribute (문자 / 공백)

} Line List 업로드

(3) 행 개수만큼 레코드 생성

(4) 열에 있는 attribute 단어를 취득 후 → Dictionary 를 거쳐서 → 표준 attribute 단어로 변환

(5) 변환된 표준 attribute 단어가 위치한 필드에 → 각 레코드 별 데이터 입력

} "Auto Mapper"

이름	설명	테이블 특징	데이터 특징
Line	배관 라인 리스트 및 데이터	- 외부 파일 (xlsx, csv, sql 등) 에서 레코드Insert - 나머지 필드는 연계 (TBD)	PJT별

(6) Line 의 Related Stream no1 --- HMB_condition table 의 Stream Number 관계 형성 (같은 값)

(7) Line 의 Attribute 들에 HMB_condition 의 값들을 읽어서 그대로 씀 } 사전 Mapping

(8) 특정 열들은 내부 calculation module 으로 채워짐 (TBD)

1.3 업로드 기능 - Instrument

(1) 파일 선택 (xls, csv, sql) **혹은 서버 데이터 연동 (TBD)**
 - 대상: SPPID 의 instrument report

(2) 임시 Table 로 import 후 행/열 구조 검사
 - 행: line no (숫자 / 문자)
 - 열: attribute (문자 / 공백)

} Instrument List 업로드

(3) 행 개수만큼 레코드 생성

(4) 열에 있는 attribute 단어를 취득 후 → Dictionary 를 거쳐서 → 표준 attribute 단어로 변환

} "Auto Mapper"

(5) 변환된 표준 attribute 단어가 위치한 필드에 → 각 레코드 별 데이터 입력

이름	설명	테이블 특징	데이터 특징
Instrument	제어 계기 리스트 및 데이터	- 외부 파일 (xlsx, csv, sql 등) 에서 레코드 Insert - 나머지 필드는 연계 (TBD)	PJT별

(6) 외부 PSN 구동 결과물로 Instrument Line size, Material class, Seq no, Location 을 입력 } 사전 Mapping

(7) 상기 Location --- Line table의 line no 와 관계형성

(8) Instrument 의 Attribute 들에 Line 의 값들을 읽어서 그대로 씀 } 사전 Mapping

1.4 업로드 기능 - Equipment

(1) 파일 선택 (xls, csv, sql) **혹은 서버 데이터 연동 (TBD)**

- 대상: MDBS로 인식해서 추출된 HMB

(2) 임시 Table 로 import 후 행/열 구조 검사

- 행: stream no (숫자 / 숫자+문자 / 문자+숫자 / 공백)

- 열: attribute (문자 / 공백)

} Equipment List 업로드

(3) DB내 대상 Table 을 결정짓는 Type 을 UI로 결정함

(4) 열에 있는 attribute 단어를 취득 후 → Dictionary 를 거쳐서 → 표준 attribute 단어로 변환

} "Auto Mapper"

(5) 변환된 표준 attribute 단어가 위치한 eq_/// table 에 → 각 레코드 별 데이터 입력

이름	설명	테이블 특징	데이터 특징
eq_///	기기별 데이터	- 외부 파일 (xlsx, csv, sql 등) 에서 Insert	PJT별

2.0 유저 세팅 - General

- (1) 유저 세팅 값은 별도 UI를 통한 project 설정을 의미하며, 각 table 들의 개별 data value 값 수정을 의미하지 않음
- (2) 즉, HMB, Line, Instrument, eq_/// 는 UI 없이 HYTOS 의 stream data 수정하듯이 sheet 처럼 수정

2.1 유저 세팅 - Dictionary

이름	설명	테이블 특징	데이터 특징
Dictionary	자동 매핑을 위한 유의어사전	- 별도 UI로 레코드 Insert/Update - Default 데이터 제공 (업데이트 중)	전 사용자 공통 (서버)

현재 감지된 단어

전체 단어 리스트
(하단 텍스트 박스의 글자에 따라 filtering)

텍스트 박스 (감지된 단어의 첫글자 자동완성)

전체 단어 리스트에 없으면 신규 단어 추가

OK Cancel

← Display Word

← Standard Word

← User Input

← User Input

2.2 유저 세팅 - Material

이름	설명	테이블 특징	데이터 특징
Material	사용되는 재질들의 총 목록	- 레코드 데이터 계속 축적 중 - PMC 필드는 는 별도 UI 로 Update	전 사용자 공통 (서버)

Material + CA 리스트

사용자 추가한 Material + CA 리스트

입력한 PMC 채워짐

>

<

사용자 추가한 Material + CA 리스트 중 특정 아이템 선택 시, PMC를 입력받을 수 있는 공간

OK Cancel

2.3 유저 세팅 - Special Service

이름	설명	테이블 특징	데이터 특징
Special Service	Line+HMB 의 조건문에 이용	- 별도 UI로 레코드 전체 Insert/Update	PJT별

현재 추가된 서비스 이름들

좌측 이름 클릭 시 조건 나열

Reference 값

추가하고 싶은 서비스 이름 입력

← Service Name

+

항목 (attribute)

조건 (등호 부등호)

값 (숫자)

+

항목 (attribute)

조건 (등호 부등호)

값 (숫자)

OK

Cancel

}

최대 10개

2.4 유저 세팅 – Component

이름	설명	테이블 특징	데이터 특징
Component	사용되는 물질들의 총 목록	- 고정	전 사용자 공통 (서버)

물질 찾는 텍스트박스

총 물질 이름	총 물질 포물라	사용자 추가한 물질 이름	사용자 추가한 포물라	타입	Pseudo
리스트					
...					
		1			
		2			
		3			
		20			

ADD PSEUDO

← 이 버튼 클릭 시에는 좌측 물질 리스트가 아닌 user input 에 따른 pseudo 가 생성되어 추가됨 (group 개념)

Reference 값

OK

Cancel

2.5 유저 세팅 – Unit

이름	설명	테이블 특징	데이터 특징
Unit	항목별 단위 모음	- 고정 (업데이트 중)	전 사용자 공통 (로컬?)

HYTOS
처럼

		항목셋 (Key)	단위셋 (Value) ▼
항목 (Key)	단위 (Value) ▼	항목 (Key)	단위 (Value) ▼
항목 (Key)	단위 (Value) ▼	항목 (Key)	단위 (Value) ▼
항목 (Key)	단위 (Value) ▼	항목 (Key)	단위 (Value) ▼
항목 (Key)	단위 (Value) ▼	항목 (Key)	단위 (Value) ▼
...		...	
		OK	Cancel

← 단위 세트 개념