

코드 생성 툴 사용법

최초 작성일 : 2003년 5월 9일 오후 2시 49분

최종 갱신일 : 2003년 5월 31일 오후 4시 14분

윤경구

CodeGen 사용법

About CodeGen

- 템플릿 방식 코드 생성 툴
- Zip 파일 형식 활용
- XML 기반
- Regular Expression 활용
- SQL 파서 내장
 - DDL 문의 테이블 이름, 필드 이름, 필드 유형, 필드 크기, PK 필드 이름 추출

CodeGen 주 기능

- 템플릿 기반 코드 생성 기능
 - Graphic User Interface 제공
 - codegen.bat
 - XML 문서를 활용한 배치 모드 지원
 - batchgen.bat
- 배치 입력 문서 편집 기능
 - XML 문서 생성해주는 GUI 제공
 - editor.bat
- SQL to XML 변환 기능
 - 오러클 SQL 문을 읽어 CREATE TABLE 문장에 파싱하여 XML 문서 생성
 - sql2xml.bat

코드 생성 기능 사용법

> java -jar codegen.jar -help

Code Generator Usage

Help(This screen) : -help

GUI Mode : default

Batch Mode 1 : -batch <input xml file.cgn>

Batch Mode 2 : -batch <directory in
which 'input xml file.cgn's reside>

Input File(*.cgn) Creation Mode : -edit

Options : -d <template root directory>

입력 XML 문서 생성

```
> java -cp codegen.jar  
    yoonforh.codegen.InputFileEditor -  
    help
```

Input File Editor Usage

Help(This screen) : -help

Options : -d <template root directory>

SQL to XML 변환

```
> java -cp codegen.jar  
    yoonforh.codegen.handler.DBInfoX  
    MLGenerator
```

```
Usage : java  
        yoonforh.codegen.handler.DBInfoX  
        MLGenerator <sql path> <output xml  
        path> <xml title>
```

CodeGen 템플릿 제작 방법

About CodeGen 템플릿

- codegen 프로그램은 템플릿 디렉토리 아래에 있는 템플릿 정보를 사용하여 코드를 생성.
- 하나의 템플릿은 템플릿 이름과 동일한 디렉토리 안에 “템플릿 이름.zip” 파일, 그리고 템플릿 구성 정보를 가지고 있는 template.xml 파일로 구성

템플릿 변수

- PATH와 텍스트 파일 안의 일부를 변수로 처리
- %%변수 이름[:대소문자 처리 명령]%% 구조
- 간단한 텍스트 대소문자 처리 명령 적용
 - csu : 대소문자 규칙 (대문자로 시작)
 - csl : 대소문자 규칙 (소문자로 시작)
 - upper : 모두 대문자로
 - lower : 모두 소문자로
 - csu_ : _로 구분된 복합명사를 대소문자 규칙 단어로 (대문자로 시작)
 - csl_ : _로 구분된 복합명사를 대소문자 규칙 단어로 (소문자로 시작)
 - upper_ : _을 없애고 모두 대문자로
 - lower_ : _을 없애고 모두 소문자로
 - csu_db : DB 테이블 헤더 부분을 없애고 _로 구분된 복합명사를 대소문자 규칙 단어로 (대문자로 시작)
 - csl_db : DB 테이블 헤더 부분을 없애고 _로 구분된 복합명사를 대소문자 규칙 단어로 (소문자로 시작)
 - upper_db : DB 테이블 헤더 부분을 없애고 _을 없애고 모두 대문자로
 - lower_db : DB 테이블 헤더 부분을 없애고 _을 없애고 모두 소문자로
 - db : DB 테이블 헤더 부분만 삭제

템플릿 문서 프로세서

- 파일 단위의 반복이 발생할 경우 처리할 수 있는 Processor 클래스 지정 가능
- 현재 DB 테이블 이름 단위의 반복 생성 지원

템플릿 핸들러

- 복잡한 처리를 위해 템플릿 정보 XML 문서에서 Handler 클래스를 지정하면 해당하는 클래스가 해당 변수 블록에 대해 실행되는 구조.
- 하나의 변수 블록에 대해 여러 개의 핸들러가 실행될 수 있다.
- 현재 테이블 이름들, 필드 이름들, 필드 유형들, PK 필드 이름들 등에 대해서 코드 블록의 반복 처리 지원
 - `yoonthor.codegen.handler.DBCodeHandler` 클래스

템플릿 핸들러 (계속)

- 코드 블록

- yoonforh.codegen.handler.DBCodeHandler 클래스에서 사용하는 방식
- <param> 태그의 name 속성이 code-block인 경우 해당하는 value 속성에 대해 각 테이블, 필드, PK 필드 등에 대해 반복해서 변수 포함된 코드 블록을 처리
- iteration-prefix, iteration-suffix를 사용하여 반복되는 prefix와 suffix를 지정할 수 있다.
 - iteration-prefix는 첫번째 반복 때에는 적용되지 않고 iteration-suffix는 마지막 반복 때에는 적용되지 않음
- 대체 코드 블록
 - 필드 타입의 반복일 경우, 해당하는 필드는 자바 자료형에 따라 다른 대체 코드 블록을 지정할 수 있다.
 - code-block-primitive
 - 기본 자바 자료형에 의해 표현된 필드 반복 블록
 - code-block-<Java Fully Qualified Class Name>
 - 해당하는 자바 자료형에 의해 표현될 필드 반복 블록