개요

- SummaryCore 는 독립프로세스로 검증이 가능하나 실행프로세스가 있어야 실질적인 사용이 가능함.
  - 다른 SFI 라이브러리와 달리 stdin 으로 파일이름이 아닌 라인단위 데이터를 put 명령어로 받음.
  - 서머리시점에는 svf 명령어로 conf 내용에 따라 summary 결과를 파일에 저장
- SummaryCore 객체를 생성한 서머리실행프로세스 개발
  - LQMS 는 LQMS_Meta_Summary.py 를 개발해서 사용
  - 파일이름이나 서머리결과는 데이터에 따라 유동적이므로 개별 프로젝트에 맞는 실행프로세스가 있어야 함
  - SummaryCore 를 import 하여 실제 서머리로직은 구현이 필요없음.
  - csv 로 된 데이터를 라인단위로 put한 후 서머리가 필요한 시점에 svf 명령어로 서머리결과를 파일로 저장.
  - svf 명령어를 부른 후에는 다시 SummaryCore 객체를 생성해야 서머리결과가 지워진다.
- 서머리되는 key 는 conf 파일에서 정의
- conf 파일 :
  - csv 형식의 put 데이터의 인덱스(0 부터 시작)로 key column, value column 을 정의
  - put 되는 데이터가 여러개의 table로 서머리되는 경우 1개의 conf 파일에 해당 table 을 모두 구성한다.
  - 아래는 1개의 데이터를 읽어서 META_KPI, META_KPI_IF 2개의 테이블로 서머리되는 경우임.

```
[General]
summary dir = /home/mobigen/nforyou/META_KPI/Summary_DIR

[META_KPI]
key column index = 0,1,2,3,4,5,6,7,8,9,10,11,12
value column index = 13,14,15,16,17,18
#functions = count, additional
#top rank = 4,100,5,100

[META_KPI_IF]
key column index = 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
value column index = 16,17,18,19,20,21,22
```

사용 방법 / 예제(LQMS)

- 실행 프로그램 실행 : LQMS 사용 예제 프로그램 LQMS_Meta_Summary.py 의 예시
  - 로그 파일이름 지정이나 데이터 특정 상 conffile 1개에 tableName 이 1개인 구조.
  - conffile 1개에 여러 개 tableName 이 있는 경우 이에 맞게 수정/개발.

```
./LQMS_Meta_Summary.py confFileName tableName
```

- 실행 프로그램 및 설명

```
from SummaryCore.SummaryCore import SummaryCore as SummaryCore
...

def Main() :
  while True :
    inMsg = sys.stdin.readline().strip()  # ==> stdin 으로 put 할 데이터파일이름을 받는다.
    fileName = inMsg[7:]                  # ==> SIOEventFlow 사용시 file://파일이름. 따라서 'file://' 는 제외

    sc = None                             # ==> stdin 으로 fileName 을 받으면 SummaryCore 객체 생성
    if os.path.exists(fileName) :
      sc = SummaryCore( confFileName )    # ==> SummaryCore 객체 생성
      fh = open( fileName )
      for line in fh :
        cmd = "PUT,%s" % line
        sc.sfio( '%s₩n' % cmd )           # ==> SummaryCore 의 sfio 함수 call(SFI 의 라이브러리 인터페이스 규약)
      fh.close()                          # ==> 여기서는 file 1개씩 서머리가 돌아가는 구조


      cmd = 'SVF,%s,%s'  % (filePfx,dataPfx)        # => output file name = file_pfx_테이블이름.csv
```

```
                                                # => dataPfx 가 정의되어 있으면 output file 에 dataPfx가 제일 앞에 붙어서 생성됨
      retStr = sc.sfio( '%sWn' % cmd ).strip()
.....
      sys.stdout.write( "file://%sWn" % outFileName ) # => SIOEventFlow 에 적용 시  stdout 출력포맷
      sys.stdout.flush()
.....
      sys.stderr.write("%sWn" % inMsg)                # => SIOEventFlow 적용시 stderr 로 stdin 으로 받은 내용 출력
      sys.stderr.flush()
.....
```

sfi_v2_summarycore/summarycore.txt · 마지막 수정: 2011/10/25 10:38 작성자 정혜정