ettool manual

Version: ettool0.7
사용설명서

김 종 태
1997.05.26
응용기술팀 평가기술GROUP

차 례

1.1 item

ettool에서 처리하는 data의 가장 작은 단위를 말한다. 측정data 자체를 말하는 것으로 raw data 파일에 하나의 줄로 표현된 data를 지칭한다. item는 또한, 두부분으로 나뉘여 진다. “이름”,“항목”으로 구분될 수 있다. 각각은 문자열이며 길이는 17자로 구성돼 있다.

◎ TR 25/0.5 “ ”Vth [V] “ 0.560

1.2 group

item들의 묶음을 말한다. 여기서는 좀 더 구체적으로 정의하면, 많은 raw data 파일내 item중 사용자가 원하는 item들만의 묶음를 말한다. batch file내에 Dstart,Dend 구문 사이에 group-data를 정의 한다.

◎ 내 모든 크기의 Vth들의 집합.

1.3 split

wafer list들의 집합으로 정의된다. 측정시 하나의 wafer는 하나의 파일로 저장되기 때문에 결국은 raw data 파일들의 집합으로 볼 수도 있다. 공정을 개발하기 위해서는 하나의 lot상에 다수의 조건으로 process를 진행하게 되는데, 이때 이 조건을 split이라 한다. 즉 split으로 정의된 wafer는 각각의 wafer로서의 의미보다는 전체적인 면으로 보아야 되기 때문에 etttool에서는 wafer란 개념보다는 split 단위로 모든 data를 분석하게 되어 있다.

group과 split의 관계를 살펴보면 split는 group에 종속적인 관계를 갖고 있음을 알게 될것이다. 왜냐하면, Fab진행시 공정개발자들은 흔히들 원하는 parameter를 얻기 위해 lot를 여러 조건으로 split하여 진행시킨다. 또한, Fabout된 lot를 분석할 때도 하나의 item를 놓고 split별로 분석한다. 그래서, ettool에서도 분석하고자 하는 item들의 집합이 group이므로 먼저 group을 정의하고, group내 split조건에 따른 raw data 파일을 지정하여 실행킨다.

◎ 190KeV BF2+ 1.4E12으로 진행된 wafer lists의 집합.

2. output styles:

2.1 file 출력 형식

ettool는 실행결과를 unix의 text형태의 파일로 만들어 낸다. 여타 응용프로그램에서 ettool의 결과를 이용하여 처리코자 할때 text형태만큼 data 호환성을 유지시켜줄 만한 형태가 없어 기본적인 출력 형태를 text를 선택하게 됐다. 또한, PostScript file 형태를 지원하고 있으나, 일부 출력형식만 지원한고 있다.

- ps: PostScript printer를 위한 출력형식으로 PostScript언어를 이용해 곧바로 출력할 수 있다.
- txt: 일반적인 text로 출력하며, 만일 내용을 print하고자 한다면, printer utility를 사용하면 된다. 보통 utility로는 unix의 ‘mpage'를 사용한다.
- 선택명령어: p.ffmt,p.opt.dat.ffmt

2.2 data 표현 방법

ettool는 실행결과를 표시할때 다양한 형태를 제공하고 있다. 기본적인 data 출력형태는 지수형(+1.00e+00)을 지원하고 있다. 사용자가 기본형 외에 다른 형태의 출력을 원할 경우는 p.format,p.opt.mode.format등의 명령어를 사용하여 직접 지정할 수 있다.

- %로 시작되어 변환문자로 끝냈다. (p.format="%1.1U")
- 변환문자(도표)

변환문자	의 미	예
e	floating point number with an exponent	-7.001100e+03
f	floating point number without an exponent	-7001.100000
g	the shorter of e or f is selected	-7001.1
U	the shorter using SI unit (a,p,u,m,K,M,G,T)	-7.11K

- 변환 예 (x=33.123456789012345678 90, y=-555.111111111e-9)

변환문자	식	출 력 내 용	의 미
%f	x	"333.123444"	precision 6 (D)
%.1f	x	"333.1"	precision 1
%.9f	x	"333.123443603"	precision 9
%20.3f	x	" 333.123"	right adjusted
%-20.3f	x	"333.123 "	left adjusted
%e	y	"-5.551111e-07"	precision 6 (D)
%.3e	y	"-5.551e-07"	precision 3
%.1U	y	"-555.1n"	precision 1

2.3 출력형식(type)

ettool에서는 실행후 실행 결과들을 여러가지 형태의 형식으로 만들어 준다. 현재까지 개발된 것으로는 dat,dat2,lot,lot2,map,map1,plt,rpt,sum,sum1등이 있다. 출력형식들을 정의하는 명령어는 p.mode이며, 지정시 작게는 한개부터 많겠는 전부 지정할 수 있는데, 각 출력물 형식을 '|'를 분리자로 사용하여 공백문자 없이 기술하면 된다. 즉, p.mode=dat|dat2|lot|lot|rpt등으로 사용한다.

▷ 출력형식별 특징

출력형식	주 요 특 징
dat	raw data를 출력해 주는 출력양식. x축으로 item, y축으로 wafer-lists을 열거
dat2	raw data를 출력해 주는 출력양식, x축으로 wafer-lists, y축으로 item을 열거
lot	통계처리값들을 출력해 주는 양식, split당 item-lists 형태.
lot2	통계처리값들을 출력해 주는 양식, item당 split-lists 형태.
map	wafer의 all-mapping용 양식으로, 하나의 wafer 기준.
map2	wafer의 all-mapping용 양식으로, 한 split기준.
plt	출력결과를 2차원 plot으로 그리기 위해 data를 2차원 table로 둔 형태.
rpt	report 양식 출력. (Text,PostScript 동시 지원)
sum,sum1	split당 median값만을 이어서 보여주는 양식.

3. batch file

ettool을 실행코자 할 경우는 반드시 파일에 group-data,ettool program을 갖게 작성하여 실행시킨다. 여기서의 이 파일을 batch file이라 부른다. batch file에는 group-data의 시작과 끝, ettool program의 시작과 끝을 표시하는 예약어가 있으며 각각 Dstart,Dend, Pstart,Pend이다. 예약어를 무시하고 작성한 batch file는 실행시 error를 발생시키며 실행할 수 없게 된다. 그리고, group-data,ettool program 부분들은 batch file내 어느 위치에 있었도 문제가 되지 않는다.

batch file는 사용자들의 편의를 위해 macro 언어를 내장하고 있다. 일반 unix에서 지원하는 m4 macro 언어를 import하여 사용하고 있으므로 사용자는 필요시 batch file내에 macro 언어를 사용할 수 있다. 대표적인 macro 명령어로는 define,include등이 있으며 자세한 사용법을 알고자 할 경우 unix에서 man 명령어를 사용하면 자세한 사용법을 얻을 수 있다.

3.1 특수문자(special character)

batch file를 작성하기 쉽게 하기 위해 ettool는 batch file에서 사용하는 특수문자를 지원한다. 다음은 특수문자와 그 기능을 간략히 설명한 것이다.

▷ 주석(remark) 표시 문자

◎ ! : 위치한 곳부터 그 줄의 마지막까지 기록된 내용을 무시한다.

◎ # : 줄의 첫번째 칼럼에 표시해야 되며, 해당 줄 전체 내용을 무시한다.

▷ 줄 연결 표시 문자

◎ + : 표시는 줄의 마지막 칼럼에 표시해야 되며, 표시된 줄과 다음 줄을 연력해 하나의 줄로 만든다. 앞줄과 뒷줄의 연결시 공백문자를 두어 분리시켜 연결시킨다.

◎ ＼: 표시는 줄의 마지막 칼럼에 표시해야 되며, 표시된 줄과 다음 줄을 연력해 하나의 줄로 만든다. 앞줄과 뒷줄의 연결시 아무 분리자 없이 곧바로 연결시킨다.

3.2 batch file 구조

Dstart

```
RcCnt = {
  ("RC-N+ *"," * [ohm]" )
  ("RC-P+ *"," * [ohm]" )
}
NVth = {
  ("NTR *"," *Vth*" )
}
```

group data 찾기 정보란

```
searching-name = {
  (name, item)
}
```

Dend

Pstart

```
open   o.name=demo o.workdir=D o.wf=dbha*
group  g.name=RcCnt g.group=RcCnt g.split=n1 +
       g.note="N+II 9.0KeV" g.wf=dbha-*[1-9] +
       g.split=n2  g.note="N+II 9.0KeV"      +
       g.wf=dbha-*[1-9]
print  p.group=RcCnt p.outfile=demo p.mode=lot|rpt +
       p.ffmt=ps
```

ettool program list란

```
open   o.name  ...
group  g.name  ...
splist s.group ...
print  p.group ...
```

Pend

* 예약어: Dstart,Dend, Pstart, Pend
* macro : define, include, ...

3.3 command 기술방법 (batch file내에서)

batch file내에서 ettool 명령어는 반드시 줄단위로 기술해야 된다. 즉, 하나의 명령어는 반드시 한 줄 내에서만 유효한 것을 말한다. 여기서 명령어란 주 명령어를 기리킨다. open,group,split,print,name={ - }, 등이다. 각 명령어에는 실행에 필요한 정보를 얻기 위한 종속적인 명령어를 갖고 있다. 이 종속 명령어는 하나의 명령어 줄 내에서만 유효하다. 명령어의 길이는 최대 1024문자까지 지원한다. 사용자가 batch file내에서 하나의 명령어를 여러줄로 나눠서 작성하고자 한다면, 특수문자중 줄 연결문자(+,\)을 이용하면 된다. 또한, 필요시 macro 언어를 사용하여 batch file를 더욱 쉽게 작성할 수 도 있다.

- ettool command lists

main command	sub command lists %syntax: *(command=data)*
open	o.name, o.indir, o.outdir, o.workdir, o.wf (5)
group	g.name, g.fix, g.data, g.split, g.note, g.wf (6)
print	p.group, p.outfile, p.ffmt, p.format, p.opt.mode.[x,y,ffmt,format] (6)
name= { - }	group를 정의하기 위한 group내 item lists 찾기 정보를 기술하는 명령어.

3.4 Macro 언어 사용방법.

macro언어를 batch file에서 사용하면 사용자는 각종 편의 명령어를 이용할 수 있다. 치환명령어(define), 다른 파일을 batch file내로 불러들려오는(inclue) 기능등을 사용하면 batch file를 쉽게 작성할 수 있다. 여기서는 m4 macro 언의의 대표적인 기능인 define,include등의 명령어만을 설명할 것이다. 만일 다른 유용한 명령어를 사용코자 한다면, unix상에서 'man m4'를 이용해 자세한 설명을 얻을 것.

▷ Defining Macros

- syntax: define(name,stuff)

이후 문자열들중 name이란 문자열이 있으면 자동적으로 stuff 내용으로 대체된다. 이때, stuff에 left quote,right quote(`, ‘)를 각각 양쪽에 붙힐수 있다. quote는 stuff 앞에 stuff가 define으로 정의된 경우 stuff을 치환시키지 않고, quote문자만 제거시킨다. 그 다음, 다시 define문에 의해 stuff을 치환시킨다. 아래의 예로 설명하면, 먼저 M은 N으로 치환되고, 다시 N는 100으로 치환시키게 된다.

- example: define(N, 100), define(M,`N')

▷Arguments & User-defined macros

- syntax: define(users-macro,stuff)

사용자 정의 macros는 define에 여러개의 argument을 이용하면 구현할 수 있다. define문의 argument는 입력순에 따라 $1,..,$9의 이름 같는다. 최대 argument의 개수는 9개로 제한되어 있다.

- example: define(bump, $1=($1+$2*$3)) :: bump(x,y,z) ----> x=(x+y*z)

▷ File Manipulation

- syntax: include(filename)

현재 입력화일에 filename의 내용을 복사해 놓는다. C언어에서의 include문과 같은 기능을 한다.

- example: include(demo1.tbl)

4. group-data (searching Key for group item-lists)

ettool는 group단위로 작업을 한다. 앞에서 설명했지만, group는 item의 집합이라 했다. 그렇다면, group에 필요한 item의 정보를 필요로 할것이다. group내에는 무수히 많은 item들이 있을 수 있으므로, 이것들을 일 일이 정확히 지정하게 되면 너무나 많은 노력과 힘이 들 것이므로, ettool에서는 group에서 사용할 item에 대한 찾기정보를 주고 이것을 이용해 raw data에서 group내 item를 찾아서 group를 만든다. ettool에서는 이 찾기정보를 group-data로 정의한다. group-data는 batch file내 Dstart,Dend 사이에 ettool에서 사용하고자 하는 수 만큼 작성하게 된다.

group-data내 찾기정보들을 작성할 때, 해당정보에 대한 data spec, 출력시 필요한 각종 문서화 정보도 동시에 지정해 준다.

4.1 KEY 기술 방법

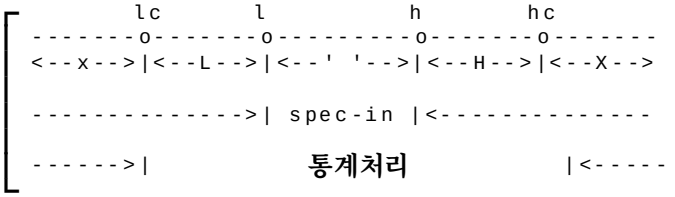
찾기정보에서의 이름,항목란에는 unix에서 사용하는 wild character들을 사용할 수 있다. 다음은 찾기정보 syntax에 대한 내용이다

Syntax) draw(이름, 항목 = lc,l,h,hc, alias) [수식;수식;...]

draw: ettool 출력시 문서화에 정보이다. (2)
- under line: -, . 중 선택하여 입력하면 됨.
- item에 대한 visual 정보: ‘x'이면 item이 출력되지 않는다.

이름,항목: raw data 찾기 정보. 만일 공백를 표함시킬려면 양쪽에 “ 기호를 사용해야 된다.
-“NTR 50/20", "Vth *"

lc,l,h,hc: raw data에 대한 통계처리시 사용 spec data. 출력시 raw data를 spec에 따라 5개의 구간으로 분류하여 표시한다. 분류기호는 ‘x,L, ,H,X' 이다. 'x,X'로 표시된 data는 각종 통계처리시 data로 인정하지 않는다.



x: raw data가 lc spec 이하로 떨어져 있어 data로서의 의미가 없어 버림.
X: raw data가 hc spec 이하로 떨어져 있어 data로서의 의미가 없어 버림.
L: target 값 이하를 표시함.
H: target 값 이상를 표시함.

alias: 출력시 이름,항목을 대체하는 문자열로 사용자가 item명을 변경코자 할 경우 사용. alias문자열 중간에 tab 문자가 있으면 alias는 자동적으로 이것을 기준으로 해 이름, 항목으로 분리되어 사용된다.

수식: raw data에 수식을 대입하여 새로운 data를 얻고자 할 경우 사용. ettool에서는 item 자신의 값은 Z변수로 자동 치환된다. item를 변경코자할 경우는 Z변수로 수식의 결과를 대입하면 된다.

5. ettool programming reference

- [...] : 순서대로 작성해야 된다.
- {...} : option으로 반드시 작성할 필요는 없다.
- <1>: 숫자 data 입력.
- <A>: 문자 data 입력.
- ... : 계속 같은 유형의 data입력.

5.1 open

raw data를 ettool 내로 가져오기 위한 정보를 기술하는 명령어로 반드시 사용해야 되며, 1회만 사용할 수 있는 명령어이다. 본 명령어 실행후 ettool내에는 RDB(raw-data data base) 가 만들어 진다

open 명령어내 사용 가능한 명령어

o.name=<A> o.indir=<A,...> o.outdir=<A> o.workdir=<A> o.wf=<A,...>

name	type	default	description
o.name	string	demo	ettool에서 실행코자하는 프로그램 명, 문서화을 위한 명령어
o.indir	string	"."	raw data가 위치한 directory path
o.outdir	string	"."	ettool 실행후 결과 파일을 저장하기 위한 directory path
o.workdir	string	"."	o.indir o.outdir를 사용하지 않고 raw data,결과 파일의 위치가 같을 경우의 directory path. 만일, o.indir/o.outdir가 이미 입력된 경우에는 무의한 명령어로 처리됨.
o.wf	string	-	raw data을 얻기 위한 찾기정보, 지정된 directory path내 파일중에서 ettool에서 사용할 raw data를 얻기 위한 정보.

- 예제)

1) o.name=demo1 o.indir=/rdb/csp5hb/dbha-14,/rcp/csp5hb/dbha-12 + o.outdir=./OUT o.wf=dbha*

2) o.name=demo2 o.workdir=/rdb/csp5hb/dbha-14 o.outdir=./OUT o.wf=dbha*

3) o.name=demo3 o.workdir=./Work o.wf=dbha*

5.2 group

group을 만들고, group내 split들을 만들어 내는 명령어.

group 명령어내 사용 가능한 명령어

g.name=<A> g.data=<A,...> g.fix {[g.split=<A> g.note=<A> g.wf=<A,...>],...}

name	type	default	description
g.name	string	-	group의 이름으로 여러개의 group를 사용코자 할 경우 다른 group과 구별하기 위한 이름.
g.data	string	-	group를 생성시키기 위한 group-data를 지정. Dstart,Dend 영역내 정의된 group-data를 사용해야 됨. ***
g.fix	toggle sw		?
g.split	string	-	group내 split을 정의하는 명령어. 여러개의 split를 사용하게 됨에 따른 여타 split과 구별하기 위한 이름. 순서의 주의.
g.note	string	" "	split의 문서화를 위한 명령어. FAB진행시 적용된 split 조건을 기술해 준다. 결과물 상단에 표시가 된다.
g.wf	string	-	ettool내에 raw-data DB에 있는 파일중에서 split에 필요한 wafer 이름을 지정하는 명령어.

- 예제)

1) g.name=RcCntN+ g.data=RcCntNPlus +
   g.split=CntNplus1 g.note="N+II 1.5Kev" g.wf=dbha-140[1-5] +
   g.split=CntNplus2 g.note="N+II 2.5Kev" g.wf=dbha-140[6-9] +
   g.split=CntNplus3 g.note="N+II 3.5Kev" g.wf=dbha-141[1-5] +
   g.split=CntNplus4 g.note="N+II 4.5Kev" g.wf=dbha-141[6-9]

2) g.name=RcCntP+ g.fix g.data=RcCntP+ +
   g.split=CntPplus1 g.note="P+II 1.5Kev" g.wf=dbha-140[1-5] +
   g.split=CntPplus2 g.note="P+II 2.5Kev" g.wf=dbha-140[6-9] +
   g.split=CntPplus3 g.note="P+II 3.5Kev" g.wf=dbha-141[1-5] +
   g.split=CntPplus4 g.note="P+II 4.5Kev" g.wf=dbha-141[6-9]

3) g.name=RcCntP+ g.fix g.data=RcCntP+

5.3 split

group내 split들을 만들어 내는 명령어.

group 명령어내 사용 가능한 명령어

s.group=<A> {[s.split=<A> s.note=<A> s.wf=<A,...>],...}

name	type	default	description
s.group	string	-	group의 이름으로 앞에서 만들어 진 group의 이름.
s.split	string	-	split을 정의하는 명령어. 여러개의 split를 사용하게 됨에 따른 여타 split과 구별하기 위한 이름
s.note	toggle sw		split의 문서화를 위한 명령어. FAB진행시 적용된 split 조건을 기술해 준다. 결과물 상단에 표시가 된다.
s.wf	string	-	ettool내에 raw-data DB에 있는 파일중에서 split에 필요한 wafer 이름을 지정하는 명령어.

- 예제)

1) s.split=CntNplus1 s.note="N+II 1.5Kev" s.wf=dbha-140[1-5] +
   s.split=CntNplus2 s.note="N+II 2.5Kev" s.wf=dbha-140[6-9] +
   s.split=CntNplus3 s.note="N+II 3.5Kev" s.wf=dbha-141[1-5] +
   s.split=CntNplus4 s.note="N+II 4.5Kev" s.wf=dbha-141[6-9]

2) s.split=CntPplus1 s.note="P+II 1.5Kev" s.wf=dbha-140[1-5] +
   s.split=CntPplus2 s.note="P+II 2.5Kev" s.wf=dbha-140[6-9] +
   s.split=CntPplus3 s.note="P+II 3.5Kev" s.wf=dbha-141[1-5] +
   s.split=CntPplus4 s.note="P+II 4.5Kev" s.wf=dbha-141[6-9]

5.4 print

출력을 하기 위한 명령어.

group 명령어내 사용 가능한 명령어

p.group=<A> p.outfile=<A> p.mode=<*type*|...> {p.format=<A>} {p.ffmt=<ps|txt>}
{p.opt.*type*.*command*=<A>}

* type: lot,lot2,dat,dat2,sum,sum1,map,map2,rpt
* command: x,y,ffmt,format,lists

name	type	default	description
p.group	string	-	출력하고자 하는 group의 이름.
p.outfile	string	group 이름	출력하고자 하는 파일이름.
p.mode	toggle sw	-	원하는 출력형식을 ' |'를 사용하여 여러개 입력.
p.format	string	%1.2e	출력시 data의 값 표현 양식을 입력.
p.ffmt	string	txt	출력시 출력화일의 형식을 입력하는 것으로 ps,txt가 있음.
p.opt.*xx.xx*	string		출력시 출력형식별로 세밀한 option를 지정코자 할 경우 사용.

- 예 제)

1) p.group=RcCntPplus p.outfile=Rc_CntPP p.mode=dat|lot|rpt +
p.format=%.2U p.ffmt=ps

2) p.group=RcCntPplus p.outfile=Rc_CntPP p.mode=dat|lot|rpt +
p.format=%.2U p.opt.rpt.ffmt=ps p.opt.rpt.lists=mi,me,ma

6.0 group-data

group내 item를 선언하기 위한 명령어.(batch file내 Dstart,Dend내에서 사용)

```
group-data 명령어
 name = {
     draw( cmt1,  cmt2  = lc, l, h, hc, alias) [수식;...]
       ㉠     ㉡      ㉢            ㉣      ㉤      ㉥
     ...
 }
```

name	type	default	description
name	string	-	group-data의 이름을 기술하는 부분
㉠ draw	char	'_'	item의 출력시 문서화 기능제공, 2가지 상태 전환 가능 - underline(-,.): item출력후 다음 줄에 line를 긋는다. - visual s/w (x): 출력시 해당 item을 제외시킨다.
㉡ cmt1	string		raw data의 " 이름 " 부분의 data를 찾기 위한 찾기 정보. (wild character 사용 가능: *,?)
㉢ cmt2	toggle sw	-	raw data의 " 항목 " 부분의 data를 찾기 위한 찾기 정보. (wild character 사용 가능: *,?)
㉣ spec	float		data의 통계처리를 위한 spec data 입력 부분.
㉤ alias	string		출력화일에서 raw data내 이름 대신 다른 이름으로 사용코자할 경우 사용.
㉥ 수식	string		data를 변경코자할 경우 사용.

6.1 사용방법

ettool는 자체내 계산기 프로그램을 내장하고 있다. 계산능력뿐만 아니라 a-z변수까지 변수를 허용하며, 변수내 값는 ettool 종료시 까지 유효하다. 사용자가 이 계산기 기능을 사용하면 raw data을 변경시킬 수 있을 뿐만 아니라, raw data에 존재하지 않는 새로운 data를 만들어 낼 수도 있다.

계산기내 변수중 z변수는 사용자 변수가 이니다. z변수는 ettool에서 사용하는 것으로 data 자체가 z변수가 된다. ettool 실행과정을 통해 수식의 처리방법을 설명해 보면 다음과 같다.

6.2 적용과정

▷ group를 만든다.

group생성시 명령어에서 지정한 group-data를 이용해서 만든다. group-data내 찾기정보를 이용 해 group에서 사용하는 item를 DB에서 검색할 때 item이 존재할 경우와 존재하지 않을 경우를 있을 것이다. ettool는 기본적으로 item이 존재하지 않으면 group내에 item를 추가할 수 없다. 그러나 group명령어에서 g.fix를 사용했다면 DB에 해당 item이 없어도 찾기정보를 item으로 가정해 group내 item으로

추가한다.

▷ group내 split를 만든다.

raw data DB내 해당 wafer lists를 찾아냈다.

- raw data DB에서 data를 찾아서 split table에 복사한다.
- group-data에서 수식이 입력됐다면, split table내 해당 data를 수식에 대입하여 처리한다.
- split table의 data를 spec data를 이용해 통계자료를 얻는다.

▷ group를 출력한다.

6.3 적용시 주의사항

- 변수중 Z는 ETTOOL 자신이 사용하고 있으므로 절대로 사용자 변수로 사용되면 안된다.
- 변수는 ETTOOL내 모든 GROUP에서 계속 사용 가능하나, ERROR가 발생할 수 있는 소지가 있으므로 한 GROUP에서 사용한 변수는 다른 GROUP에서 ACCESS할 수 없다고 가정하자.
- ETTOOL는 ITEM에서 수식을 적용하기에 앞서 자신의 값을 먼저 Z변수로 치환시킨다. 또한, 수식의 결과를 ITEM의 값으로 반영코자 한다면 Z변수로 치환시킨다.
- 여러개의 수식을 기술할 때는 수식간의 분리를 위해 수식 마지막에 ‘;’기호를 넣는다.
- 수식에서 0으로 어떻값을 나누게 되면 ERROR가 발생된다.
- 수식 사용 예

수식 기술 예	설 명	변수값 변동
("Ntr 50/5 ","Vth") [a=z]	변수A에 Vth값을 치환시키다. Vth=0.5로 가정	A=0.5, Z=0.5
("Ntr 50/10","Vth") [b=z]	변수B에 Vth값을 치환시키다. Vth=1.5로 가정	B=1.5, Z=1.5
("Ntr Delta","Diff-Vth") [z=a-b] .	Z변수에 A-B를 계산하여 치환시키다. Z변수값을 ITEM내로 치환시킨다.	Z=-1.0, ITEM내로 치환 .
("Ntr 50/50","Vth") [c=z;z=z*10] .	C변수에 Vth값을 치환시킨다. Z변수는 자신의 값에 10을 곱해서 치환. Z변수값을 ITEM내로 치환시킨다.(VTH:0.3)	C=0.3, Z=3.0 Z변수 ITEM내로 치환

7. ERROR MESSAGE

- 아래의 오류멧세지중 ‘o' 기호로 시작하는 멧세지는 ettool s/w 실행 중단의 의미
- 아래의 오류멧세지중 ‘-' 기호로 시작하는 멧세지는 ettool s/w 경고의 의미

7.1 batch file error

● 시스템내 특정화일의 삭제가 않됩니다. 멧세지의 파일을 수작업으로 삭제한 후 재 실행.

o ERROR: file remove error '/tmp/ettool.pgm'
o ERROR: file remove error '/tmp/ettool.dat'
o ERROR: file remove error '/tmp/ettool.io'
o system() error... : system-command

● MACRO실해시 장애가 발생하였습니다. 해당 FILE이 존재하지 않읍니다.

o ERROR: file open error '/tmp/ettool.dat'
o ERROR: file open error '/tmp/ettool.pgm'

● BATCH FILE내 Dstart,Dend,Pstart,Pend 예약어의 사용에 문제가 있읍니다.

o IO-FILE Format error ~Checking IO-FILE....
o ERROR: IO-FILE Format error...

● string는 ETTOOL의 주명령어가 아닙니다. Batch File를 확인하세요

- ERROR: Undefined command 'string' ignored.

● command는 ETTOOL의 명령어가 아닙니다. Batch File를 확인하세요

- ERROR: OPEN - Illegal sub-command 'command'
- ERROR: GROUP - Illegal sub-command 'command'
- ERROR: SPLIT - Illegal sub-command 'command'
- ERROR: PRINT - Illegal sub-command. 'command'

● SPLIT 명령어 사용에 문제가 있습니다. s.group 명령어가 없거나 중복 사용.

o Duplicate 's.group'
o No declaration 's.group'

● PRINT 명령어 사용에 문제가 있습니다. p.gruop,p.mode 명령어가 없읍니다.

o ERROR: PRINT - No declaration GROUP-NAME
o ERROR: PRINT - No declaration PMODE

● PRINT 명령어 사용에 문제가 있습니다. 출력형식 이름이 틀렸습니다.

- ERROR: Illegal option in PRINT 'command'
- ERROR: Undefined print mode 'p.mode,subcmd'

● GROUP-DATA내 dataline에 문제가 있습니다.

- Warning: Illegal data for 'dataline'
- ERROR: paragraph missmatch'(',')' in 'dataline'

● ETTOOL 실행을 위한 COMPUTER MEMORY가 부족합니다. 다른 PROCESS을 종료한 다음 실행해 보시요.

o Memory allocation failure at Malloc_command_node
o Memory allocation failure at Strings_copy
o Error: Memory allocation failure... Add_sdbl_list
o Error: Memory allocation failure... Make_spec_dat

7.2 raw data loading error

● RAW DATA FILE이 존재하지 않거나 열수 없는 상태입니다. 파일 상태를 파악한 후 재 실행 할 것.

o ERROR: File open fail "rawdata-filename"
o ERROR: Data file open fail "rawdata-filename"
o ERROR: Data file table is empty... Path: dirs
● RAW DATA FILE에 문제가 있습니다. RAW DATA의 Header 부분을 확인한 후 개선이 인된면, 개발자에게 문의 할 것.
o ERROR: Undefined location data
- WARING: Undefined item data - 'name item'
- WARING: Data reading error - rawdata-name
o ERROR: Data file header error : filename
● ETTOOL 실행을 위한 COMPUTER MEMORY가 부족합니다. 다른 PROCESS을 종료한 다음 실행해 보시요.
o memory allocation failure.... ~ progrma quit : save_item_data
o memory allocation failure.... ~ progrma quit : save_wafer_data
- ERROR: Data file table is full "group-name"

7.3 core part error

● ETTOOL 실행을 위한 COMPUTER MEMORY가 부족합니다. 다른 PROCESS을 종료한 다음 실행해 보시요.
o ERROR:Can't allocation make_group_node
o ERROR:Can't memory reallocation 'make_group_node'
o ERROR:Can't memory allocation 'make_split_node'
o ERROR:Can't memory reallocation 'make_split_node'
o Error:Can't allocation cp_raw_edb_split
● 해당 SPLIT의 RAW-DATA가 하나도 없습니다. 무시하고 실행시킵니다(SPLIT를 만들지 않읍니다).
- WARRING: Not searched wafer-lists (Group:%s, Split:%s)
● GROUP내의 ITEM에서 RAW-DATA상의 DATA가 아닌 가상의 DATA의 수가 초과했습니다.(최대100 허용)
o ERROR: memory boundary excess '%s'
● GROUP내의 ITEM수가 하나도 없거나 GROUP-DATA에 정의되어 않음. BATCH FILE를 확인해 볼 것.
- Warring :: searched item is 0
o Error:Not found group : group_name
● 알 수 없는 오류입니다. 개발자를 호출할 것.
o Error, ... "mesage" in function()

7.4 output part error

● PRINT 명령어 오류입니다. 입력된 OUTPUT DIRECTORY가 존재하지 않습니다. BATCH FILE,DIRECTORY를 확인한 후 재 실행 하시오.
o PRINT ERROR.... out_dir
- Warring :: unknown print mode.. %d
- Warring :: K1LINE-FORMAT, Not Support ....
● PRINT 명령어 오류입니다. GRUOP이 없거나 출력될 파일이 열리지 않읍니다.
- Error:Can't found group : group_name
- Cannot file open : out_file

7.5 utility part error

● '수식' 오류입니다.
- Numerical Recipes run-time error... - error_text
- ...now exiting to system...
- Error, ... "message" in func_name()
● BATCH FILE내에서 사용하신 명령어줄의 입력형태에 오류가 있읍니다.

o Input stram error at BSParse() !
o Input stram error at toknize() !

8. DEMO

▷ DEMO1.I file

```
! DEMO1.I
! KIM,JongTae  97/05/26
!
define(N_SPLIT,``
  g.split=n1 g.wf=*0[234] +



  g.split=n2 g.wf=*0[56] +
  g.split=n5 g.wf=*1[789],*20 +
  g.split=n6 g.wf=*2[1-5]
  ''
)
define(groupPrint,
  group g.name=$1 g.fix g.data=$1 + $2
  print p.group=$1 p.mode=$3 p.format=%1.1U
)
include(DEMO1.TBL)
Pstart
  open o.indir=/rdb/XX,/rdb/csp5hb/dbha-14 o.outdir=. o.wf=dbha-????
  groupPrint(RcCnt,N_SPLIT,dat|lot|rpt)
  groupPrint(RcCntN+,N_SPLIT,dat|lot|plt)
Pend
```

▷ DEMO1.TBL

```
!97/05/26,KIM,JongTae
!DEMO1.TABLE(TBL)
Dstart
  ALL = {
      ("*","*")
  }
  RcCnt = {
     x("RC-N+ 0.30X0.30  ","R*"          = 1,2,3,1000)
     x("RC-N+ 0.35X0.35  ","R*"          = 1,2,3,1000)
     x("RC-N+ 0.40X0.40  ","R*"          = 1,2,3,1000)
     x("RC-N+ 0.45X0.45  ","R*"          = 1,2,3,1000)
     x("RC-N+ 0.50X0.50  ","R*"          = 1,2,3,1000)
     x("RC-N+ 0.55X0.55  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.30X0.30  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.35X0.35  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.40X0.40  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.45X0.45  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.50X0.50  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.55X0.55  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.30X0.30  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.35X0.35  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.40X0.40  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.45X0.45  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.50X0.50  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.55X0.55  ","R*"          = 1,2,3,1000)
  }
  RcCntN+ = {
      ("RC-N+ 0.30X0.30  ","R*"          = 1,2,3,1000)
      ("RC-N+ 0.35X0.35  ","R*"          = 1,2,3,1000)
      ("RC-N+ 0.40X0.40  ","R*"          = 1,2,3,1000)
      ("RC-N+ 0.45X0.45  ","R*"          = 1,2,3,1000)
      ("RC-N+ 0.50X0.50  ","R*"          = 1,2,3,1000)
      ("RC-N+ 0.55X0.55  ","R*"          = 1,2,3,1000)
  }
  RcCntP+ = {
      ("RC-P+ 0.30X0.30  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.35X0.35  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.40X0.40  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.45X0.45  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.50X0.50  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.55X0.55  ","R*"          = 1,2,3,1000)
  }
Dend
```

▷ macro 진행후의 batch file.

```
! DEMO1.I
! KIM,JongTae  97/05/26
!

!97/05/26,KIM,JongTae
!DEMO1.TABLE(TBL)
Dstart
  ALL = {
      ("*","*")
  }
  RcCnt = {
     x("RC-N+ 0.30X0.30  ","R*"          = 1,2,3,1000)
     x("RC-N+ 0.35X0.35  ","R*"          = 1,2,3,1000)
     x("RC-N+ 0.40X0.40  ","R*"          = 1,2,3,1000)
     x("RC-N+ 0.45X0.45  ","R*"          = 1,2,3,1000)
     x("RC-N+ 0.50X0.50  ","R*"          = 1,2,3,1000)
     x("RC-N+ 0.55X0.55  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.30X0.30  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.35X0.35  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.40X0.40  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.45X0.45  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.50X0.50  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.55X0.55  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.30X0.30  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.35X0.35  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.40X0.40  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.45X0.45  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.50X0.50  ","R*"          = 1,2,3,1000)
      ("RC-GP 0.55X0.55  ","R*"          = 1,2,3,1000)
  }
  RcCntN+ = {
      ("RC-N+ 0.30X0.30  ","R*"          = 1,2,3,1000)
      ("RC-N+ 0.35X0.35  ","R*"          = 1,2,3,1000)
      ("RC-N+ 0.40X0.40  ","R*"          = 1,2,3,1000)
      ("RC-N+ 0.45X0.45  ","R*"          = 1,2,3,1000)
      ("RC-N+ 0.50X0.50  ","R*"          = 1,2,3,1000)
      ("RC-N+ 0.55X0.55  ","R*"          = 1,2,3,1000)
  }
  RcCntP+ = {
      ("RC-P+ 0.30X0.30  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.35X0.35  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.40X0.40  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.45X0.45  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.50X0.50  ","R*"          = 1,2,3,1000)
      ("RC-P+ 0.55X0.55  ","R*"          = 1,2,3,1000)
  }
Dend

Pstart
  open o.indir=/rdb/XX,/rdb/csp5hb/dbha-14 o.outdir=. o.wf=dbha-????
  group g.name=RcCnt g.fix g.data=RcCnt +
  g.split=n1 g.wf=*0[234] +
  g.split=n2 g.wf=*0[56] +
  g.split=n5 g.wf=*1[789],*20 +
  g.split=n6 g.wf=*2[1-5]


  print p.group=RcCnt p.mode=dat|lot|rpt p.format=%1.1U

  group g.name=RcCntN+ g.fix g.data=RcCntN+ +
  g.split=n1 g.wf=*0[234] +
  g.split=n2 g.wf=*0[56] +
  g.split=n5 g.wf=*1[789],*20 +
  g.split=n6 g.wf=*2[1-5]


  print p.group=RcCntN+ p.mode=dat|lot|plt p.format=%1.1U

Pend
```

▷ 실행결과.

==> RcCnt.dat <==

```
> GROUP:RcCnt  SPLIT:n1  NOTE:""
>   OWF:dbha-1403,dbha-1404 (2)
```

1. Item-Name	:	RC-P+ 0.30X0.30	RC-P+ 0.40X0.40	RC-P+ 0.50X0.50	RC-GP 0.30X0.30	RC-GP 0.40X0.40
1. Item-Size	:	R [ohm/cn]	R [ohm/cn]	R [ohm/cn]	R [ohm/cn]	R [ohm/cn]
2. Item-Name	:	RC-P+ 0.35X0.35	RC-P+ 0.45X0.45	RC-P+ 0.55X0.55	RC-GP 0.35X0.35	
2. Item-Size	:	R [ohm/cn]	R [ohm/cn]	R [ohm/cn]	R [ohm/cn]	
dbha-1403 06_03	:	437.3MX	307.8MX	361.3MX	7.1KX	3.2KX	1.5KX	482.4MX	405.1MX	33.2 H
dbha-1403 06_06	:	415.9MX	342.8MX	12.5KX	4.0KX	1.4KX	818.9 H	505.9MX	553.7MX	9.5 H
dbha-1404 06_03	:	371.4MX	349.2MX	88.9KX	7.0KX	2.9KX	1.2KX	550.5MX	531.7MX	12.2 H
dbha-1404 06_06	:	403.4MX	346.6MX	11.6KX	3.5KX	1.4KX	821.7 H	607.7MX	89.0MX	7.4 H
<RESULT> MEAN		0.0	0.0	0.0	0.0	0.0	820.3	0.0	0.0	15.6
MEDIAN		0.0	0.0	0.0	0.0	0.0	820.3	0.0	0.0	10.8
SIGMA		0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	11.9
MIN		0.0	0.0	0.0	0.0	0.0	818.9	0.0	0.0	7.4
MAX		0.0	0.0	0.0	0.0	0.0	821.7	0.0	0.0	33.2
TOTAL		4	4	4	4	4	4	4	4	4

==> RcCnt.lot <==

```
> GROUP:RcCnt  SPLIT:n1  NOTE:""
>   OWF:dbha-1403,dbha-1404 (2)
```

NAME	SIZE [UNIT]	MEAN	MEDIAN	SIGMA	MIN	MAX	TOT	MIC	LSL	USL	MXC	SPEC-DATA			
RC-P+ 0.30X0.30	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	4	.	.	.	4	1	2	3	1K
RC-P+ 0.35X0.35	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	4	.	.	.	4	1	2	3	1K
RC-P+ 0.40X0.40	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	4	.	.	.	4	1	2	3	1K
RC-P+ 0.45X0.45	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	4	.	.	.	4	1	2	3	1K
RC-P+ 0.50X0.50	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	4	.	.	.	4	1	2	3	1K
RC-P+ 0.55X0.55	R [ohm/cn]	820.3	820.3	2.0	818.9	821.7	4	.	.	2	2	1	2	3	1K
RC-GP 0.30X0.30	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	4	.	.	.	4	1	2	3	1K
RC-GP 0.35X0.35	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	4	.	.	.	4	1	2	3	1K
RC-GP 0.40X0.40	R [ohm/cn]	15.6	10.8	11.9	7.4	33.2	4	.	.	4	.	1	2	3	1K
RC-GP 0.45X0.45	R [ohm/cn]	6.1	6.2	264.6m	5.8	6.4	4	.	.	4	.	1	2	3	1K
RC-GP 0.50X0.50	R [ohm/cn]	5.4	5.4	121.8m	5.3	5.6	4	.	.	4	.	1	2	3	1K
RC-GP 0.55X0.55	R [ohm/cn]	5.1	5.0	73.4m	5.0	5.2	4	.	.	4	.	1	2	3	1K

```
> GROUP:RcCnt  SPLIT:n5  NOTE:""
>   OWF:dbha-1420 (1)
```

NAME	SIZE [UNIT]	MEAN	MEDIAN	SIGMA	MIN	MAX	TOT	MIC	LSL	USL	MXC	SPEC-DATA			
RC-P+ 0.30X0.30	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	2	.	.	.	2	1	2	3	1K
RC-P+ 0.35X0.35	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	2	.	.	.	2	1	2	3	1K
RC-P+ 0.40X0.40	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	2	.	.	.	2	1	2	3	1K
RC-P+ 0.45X0.45	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	2	.	.	.	2	1	2	3	1K
RC-P+ 0.50X0.50	R [ohm/cn]	690.5	690.5	36.1	665.0	716.1	2	.	.	2	.	1	2	3	1K
RC-P+ 0.55X0.55	R [ohm/cn]	538.7	538.7	130.0	446.7	630.6	2	.	.	2	.	1	2	3	1K
RC-GP 0.30X0.30	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	2	.	.	.	2	1	2	3	1K
RC-GP 0.35X0.35	R [ohm/cn]	0.0	0.0	0.0	0.0	0.0	2	.	.	1	1	1	2	3	1K
RC-GP 0.40X0.40	R [ohm/cn]	7.0	7.0	279.3m	6.8	7.2	2	.	.	2	.	1	2	3	1K
RC-GP 0.45X0.45	R [ohm/cn]	5.9	5.9	160.5m	5.8	6.0	2	.	.	2	.	1	2	3	1K
RC-GP 0.50X0.50	R [ohm/cn]	5.4	5.4	125.2m	5.3	5.5	2	.	.	2	.	1	2	3	1K
RC-GP 0.55X0.55	R [ohm/cn]	5.1	5.1	100.4m	5.0	5.2	2	.	.	2	.	1	2	3	1K

==> RcCnt.rpt <==

```
> GROUP:RcCnt  SPLIT:n1  NOTE:""
>   OWF:dbha-1403,dbha-1404 (2)
```

ITEM	MEDIAN	AVG	SDEV
RC-P+ 0.30X0.30 R [ohm/cn]	0.0	0.0	0.0
RC-P+ 0.35X0.35 R [ohm/cn]	0.0	0.0	0.0
RC-P+ 0.40X0.40 R [ohm/cn]	0.0	0.0	0.0
RC-P+ 0.45X0.45 R [ohm/cn]	0.0	0.0	0.0
RC-P+ 0.50X0.50 R [ohm/cn]	0.0	0.0	0.0
RC-P+ 0.55X0.55 R [ohm/cn]	820.3	820.3	2.0
RC-GP 0.30X0.30 R [ohm/cn]	0.0	0.0	0.0
RC-GP 0.35X0.35 R [ohm/cn]	0.0	0.0	0.0
RC-GP 0.40X0.40 R [ohm/cn]	10.8	15.6	11.9
RC-GP 0.45X0.45 R [ohm/cn]	6.2	6.1	264.6m
RC-GP 0.50X0.50 R [ohm/cn]	5.4	5.4	121.8m
RC-GP 0.55X0.55 R [ohm/cn]	5.0	5.1	73.4m