

MM5와 IDL

IDL에서 MM5를 직접 지원하는 루틴은 제공되지 않지만, IDL 사용자가 개발하여 배포되는 루틴은 존재합니다. MM5IDL이라는 라이브러리인데요, 현재는 다음 링크에서 다운로드 받을 수 있습니다.

<http://www.riv.is/wp-content/uploads/mm5idl>

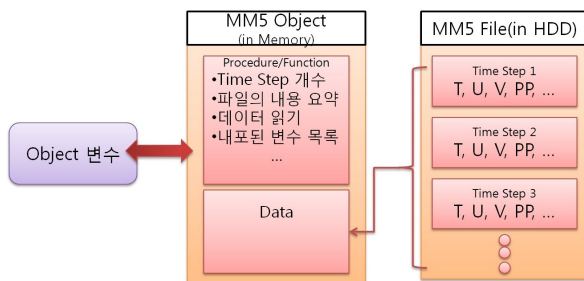
위 페이지에서 bundle 이라고 된 링크를 다운로드 받아 IDL의 검색 경로에 포함시키면 설정이 완료됩니다. 사용 설명서는 위 페이지에서 "read this" 라는 링크를 통해 PDF 문서로 다운로드 받을 수 있습니다. 이 문서에서는 한 장 안에, 좀 더 간단한 내용만 소개하고자 합니다.

Object의 생성

다음과 같이 MM5 파일의 오브젝트를 생성합니다.

IDL> oMM5=obj_new('mm5file', 'MMOUT01_1')

모든 MM5 파일 다루기의 시작은 위 한 줄로 시작합니다. oMM5가 오브젝트 변수입니다. 'mm5file'은 MM5 포맷을 다루는 클래스의 이름이므로 모든 MM5 읽기에서 공통으로 쓰입니다, MMOUT01_1은 파일 이름입니다.



MM5 오브젝트 개요

- MM5 파일과 매칭이 되는 IDL의 Object가 메모리에 생성되고 이를 가리키는 변수(예문에서는 oMM5)를 통해 교신을 합니다. 오브젝트는 사용자가 메모리에서 삭제하거나 IDL을 종료하지 않은 메모리에서 계속 남게 됩니다.
- (개념적으로,) 이 오브젝트 안에 사용자가 하고자 하는 프로시저/함수는 물론 데이터까지 다 들어 있다고 보아도 좋습니다. 이 안에서 필요한 것을 꺼내 쓰면 됩니다.
- 오브젝트는 하나만 만들 수 있는 것이 아닙니다. 메모리만 허용한다면 동시에 매우 많은 여러개를 만들 수도 있으며, 오브젝트형 배열도 있습니다. 그러므로 동시에 여러 개의 MM5 파일을 다루는 것도 가능합니다.

왜 이렇게 시간이 오래 걸릴까요?

파일의 크기가 작다면 모르겠지만, 파일에 포함된 내용이 많은 경우 크기는 수십 GB도 될 수 있습니다. 이런 경우라면 Object를 생성하는 과정(obj_new)이 수분 정도 소요될 수도 있습니다. 이 과정에서 파일을 처음부터 끝까지 한번 스캔하여, 파일의 중요한 지점들을 MM5 오브젝트의 데이터 부분에 연결해 두기 때문입니다(실제 HDD로부터 데이터를 읽는 것은 아닙니다. 사전 조사만 해 두는 것입니다). 그러므로 이 과정이 한번만 실행되고 나면 이후에 해당 파일에 대해 자유롭게 앞/뒤를 오가며 데이터 접근이 가능해집니다. 이후 작업이 더 빠르게 진행되기 때문에 손해 보는 시간은 아니라고 생각하셔도 좋습니다.

간단한 조사

파일의 개요 보기

IDL> oMM5 -> print

-- mm5file information -----

MM5 file: D:\MMOUT_DOMAIN2_0707

Logical unit number = -1

Filetype is: Output, same number of variables per timestep

Number of big-headers = 1

Number of timesteps = 757

Number of variables in each timestep = 52

oMM5는 앞서 생성한 오브젝트 변수 이름입니다. -> 는 오브젝트에 내포된 프로시저나 함수를 호출하기 위한 연산자로 빼기와 부등호는 꼭 붙여 써야 합니다(화살표처럼 보이는 한 덩어리 연산자입니다).

파일 이름 확인

IDL> print, oMM5 -> file()

파일 타입 확인

IDL> print, oMM5 -> filetype()

Time Step 확인

몇 개의 Time Step이 저장되어 있는지 확인

IDL> print, oMM5 -> nt()

253

Time Step의 날짜와 시간

IDL> print, oMM5 -> varDates(0, 0)

2007-05-10_00:00:00.0000

varDates(it, iv)에서 it는 time step을, iv는 변수 번호를 의미합니다.

Time Step의 경과 시간(분 단위)

IDL> print, oMM5 -> varTimes(4, 0)

다음과 같이 MM5 파일에 포함된 모든 예보 시각을 출력할 수 있습니다.

```
IDL> nts=oMM5->nt( )
IDL> for l=0, nts-1 do print, oMM5->varDates(l, 0)
2007-05-10_00:00:00.000
2007-05-10_01:00:00.000
2007-05-10_02:00:00.000
...
```

변수 확인

파일에 포함된 전체 변수들 확인

```
IDL> print, oMM5 -> varNames(0)
U V T Q CLW RNW ICE SNOW RAD TEND W PP
PSTARCGRS ... SOIL T 6 T2 Q2 U10 V10 ALBD
SLMO SFEM SFZ0 THERIN SFHC SCFX SIGMAH
```

0은 Time Step의 번호를 의미합니다. 이를 표시하지 않으면 모든 Time Step의 변수들을 출력하는데요, 대부분의 MM5 파일은 모든 Time Step이 같은 변수를 가지므로 varNames(0) 과 같이 사용하는 것이 일반적입니다. 0번 Time Step이 없는 경우는 없지요.

변수의 개수

```
IDL> print, oMM5 -> nv( )
```

파일에 포함된 일부 변수들 확인

varNames(it, iv)의 형태로 사용하며 iv는 스칼라 또는 벡터를 쓸 수 있습니다.

```
IDL> print, oMM5 -> varNames(0, 0)
U
IDL> print, oMM5 -> varNames(0, [0, 1, 2])
U V T
```

MM5의 데이터 읽기

netCDF나 HDF와 마찬가지로, MM5 역시 파일에 대한 조사만 끝나면 실제 데이터를 읽는 것은 극히 간단합니다. data() 함수를 이용합니다.

n번째 time step의 모든 변수를 구조체로 읽기

```
IDL> all=oMM5->data(8) ;8번째 Time Step
IDL> help, all, /STRUCT
** Structure <30ceab8>, 52 tags, length=45207684, data
length=45207684, refs=1:
U          FLOAT      Array[178, 148, 33]
V          FLOAT      Array[178, 148, 33]
T          FLOAT      Array[178, 148, 33]
Q          FLOAT      Array[178, 148, 33]
RAD_TEND   FLOAT      Array[178, 148, 33]
...
```

참고) MM5 변수 이름에서 공백으로 된 부분은 IDL 구조체 필드 이름에서 자동으로 '_'로 변환됩니다(RAD_TEND를 보세요). 구조체 필드 이름에 공백을 쓸 수 없기 때문입니다.

n번째 time step의 지정 변수들을 구조체로 읽기

```
IDL> vars=oMM5->data(6, ['U', 'V', 'T']) ;n=6일 때
IDL> help, vars, /struct
** Structure <30d7fe0>, 3 tags, length=10432224, data
length=10432224, refs=1:
U          FLOAT      Array[178, 148, 33]
V          FLOAT      Array[178, 148, 33]
T          FLOAT      Array[178, 148, 33]
```

n번째 time step의 단일 변수를 배열로 읽기

```
IDL> t=oMM5->data(3, 'T', /ARRAY)
IDL> help, t
T          FLOAT      = Array[178, 148, 33]
```

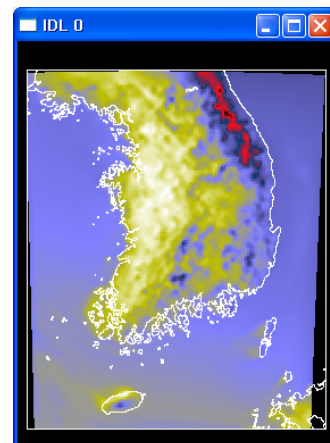
지정 변수들을 오브젝트로 할당

일반적으로 쓰이지는 않지만, 각 변수에 대해 가장 자세한 정보를 추출할 수 있습니다. vObjArr 키워드를 이용하여 각 변수를 오브젝트로 다시 받아 냅니다.

```
IDL> vars=oMM5->data(2, ['U', 'V'], vObjArr=oV)
IDL> help, oV
OV          OBJREF     = Array[2]
```

예제

7번째 Time Step의 변수를 읽어 기온을 디스플레이



```
file='D:\MMOUT_091108'
oMM5=obj_new('mm5file', file)
all=oMM5->data(7)
;imagemap 은 Liam Gumley의 공개 루틴입니다.
imagemap, all.[*,*,32], all./atitcrs, all./longicrs
map_continents, /HIRES, /COAST
;oMM5 메모리에서 제거.
obj_destroy, oMM5
```