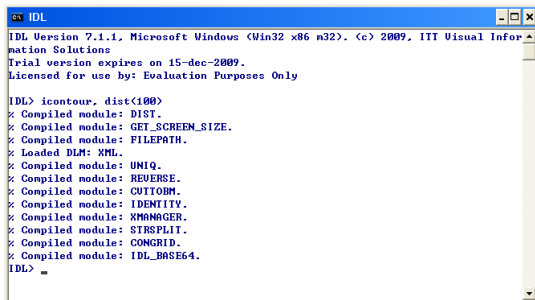


IDL Command Line 모드

애초에 텍스트 기반의 운영체제인 Unix와 Linux는 당연히 Shell 환경에서 IDL이 실행됩니다. 기본(idl로 실행)이 Command Line 모드이고 IDL 워크벤치(idlde로 실행)가 오히려 옵션인 것처럼 느껴질 정도입니다. Mac 역시 내부적으로는 Unix이므로 당연히 Command Line 모드가 지원됩니다. 오직 Windows 만이 Command Line 모드(DOS 창에서 실행)가 지원되지 않았던 것입니다. IDL 7.1이 나오기 전까지 말이지요.

Command Line 모드로 IDL을 실행시키면 컴퓨터에 걸리는 부하가 워크벤치에 비해 현저하게 낮으므로 동시에 여러개의 IDL 세션을 띄워도 문제가 없습니다(한 화면에서 한 사용자가 띄우는 IDL은 무조건 1개의 라이선스로 인식됩니다. 여러개를 띄워도 상관 없습니다).

Multi Process 모드나 이전 명령 매칭(TAB 키)과 같은 기능은 DOS용 IDL에만 구현되어 있는 기능입니다.

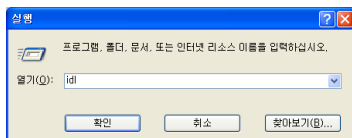


DOS 창에서 실행중인 IDL

Command Line 모드로 실행

다음 중 한 가지 방법으로 시작합니다.

1. 시작 > 모든 프로그램 > IDL 7.1 > IDL Command Line
2. DOS 창에서 idl 또는 idl.exe 입력
3. 시작 > 실행에서 idl 입력



4. C:\Program Files\ITT\IDL71\bin\bin.x86\idl.exe 를 더블클릭

2번과 3번의 방법은 다음 디렉토리가 PATH(경로)로 잡혀 있어야 합니다.

C:\Program Files\ITT\IDL71\bin\bin.x86

CYWIN(Linux 셸과 유사한 Windows용 셸 환경)에서도 잘 작동하므로 xterm과 같이, Linux와 매우 유사한 실행 환경을 구축할 수도 있습니다.

기본적인 시동 옵션

Unix/Linux에서 사용하던 대부분의 시동 옵션을 사용할 수 있다고 생각하면 되겠습니다. -vm 옵션과 -rt 옵션은 idlrt.exe를 통해 따로 구현되어 있습니다.

다음과 같은 옵션이 많이 쓰입니다.

옵션	용도
-32	64비트 머신에서 32비트 모드로 IDL 실행
-args	뒤에 IDL 세션에 넘겨 줄 argument들 나열
-e	뒤에 나오는 IDL 명령문 실행 후 종료

사용예

> idl -32

32비트 모드로 IDL 실행

> idl -args nov 15 2009

이렇게 실행된 IDL 세션에서 COMMAND_LINE_ARGS() 함수를 이용하여 -args 뒤의 내용을 배열로 받을 수 있습니다. 위 예에서는 다음과 같은 결과를 냅니다.

IDL> a=command_line_args(count=ct)

결과는 ct=3 & a=['nov', '15', '2009']

> idl -e "print, 25*10"

250

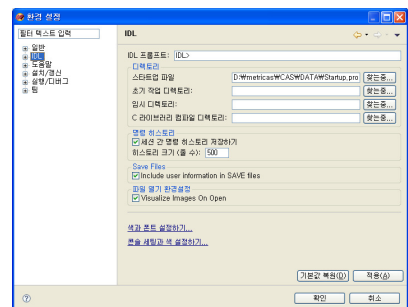
>

-e 옵션 뒤에 나오는 명령을 실행 후 자동으로 IDL을 빠져 나옵니다. 따옴표를 이용하여 띄어쓰기 및 콤마를 사용할 수 있게 되며, &기호를 사용하여 여러개의 명령어를 한번에 넘길 수도 있습니다. 궁극적으로 IDL의 모든 프로그램은 프로시저 이름 한 줄로 시작된다는 점을 생각하면 어떤 것이든 실행시킬 수 있는 것입니다.

TAB 키를 이용한 명령어 매칭

명령문의 일부를 입력하고 TAB 키를 치면, 이전에 실행했던 명령문 중에 같은 패턴으로 시작하는 행을 자동으로 입력하여 줍니다. TAB 키를 한번 더 누르면 그 이전에 같은 패턴으로 입력한 행을 보여 줍니다. 이런 식으로 IDL이 기억하는 한 계속 거슬러 올라갈 수 있습니다.

IDL 워크벤치의 환경설정 > IDL의 히스토리 크기(줄 수) 설정에 따라 기억하는 과거 입력 내용



환경설정>IDL>히스토리 크기

의 분량이 다를 수 있습니다. 디폴트 설정은 최근 500 행입니다.

히스토리 기능 사용

커서 키 위쪽 방향을 이용하여 이전에 입력한 명령행을 거슬러 올라갈 수 있습니다. :h(콜론h)를 입력하여 히스토리 기억 설정만큼의 최근 입력 내용을 조회할 수 있습니다.

```
IDL> :h
```

도움말 시동

워크벤치와 마찬가지로 물음표를 입력하여 도움말을 시동하고 원하는 도움말을 조회할 수 있습니다.

```
IDL> ?
```

```
IDL> ? plot
```

다중 프로세스 모드 (MP 모드)

콜론 (:) 으로 시작하는 프로세스 번호를 이용하여 창 하나에서 다수의 IDL 프로세스를 사용할 수 있습니다. 이 기능은 UNIX 계열에는 없습니다. IDL DOS 환경의 보너스 기능입니다. UNIX 계열에서는 터미널을 여러개 열어 쓰는 방법밖에는 없습니다.

```
IDL> :
```

```
0
```

콜론만 입력하면 현재 창에서 작동하고 있는 IDL의 프로세스 번호를 나열합니다. 처음에는 0번 프로세스만 보일 것입니다.

```
IDL> :1 idl
```

```
1:IDL>
```

:n idl을 실행하면 n번 프로세스의 IDL이 실행되고 프롬프트가 n번 프로세스를 알아볼 수 있도록 바뀝니다.

```
1:IDL> :
```

```
0 1
```

0번과 1번 프로세스를 운용 중입니다. 현재는 1번 프로세스에서 작업을 하고 있지만 프로세스 간 이동은 자유롭습니다.

```
1:IDL> :0
```

```
IDL> :1
```

```
1:IDL>
```

각각의 프로세스는 exit로 빠져나올 수 있습니다.

```
1:IDL> exit
```

```
IDL session 1 is about to die.
```

```
IDL>
```

프로세스 번호는 0번(기본 프로세스)을 제외하고는 자유롭게 지정할 수 있습니다. 실제 운영체제가 부여하는 프로세스 번호와는 무관합니다.

```
1:IDL> :3456 idl
```

```
3456:IDL> :
```

```
0 1 3456
```

콜론을 두 개 연속으로 붙여 쓰면 현재 창에서 운용 중인 프로세스의 상태를 볼 수 있습니다.

```
3456:IDL> ::
```

```
0 32-bit idle
```

```
1 32-bit idle
```

```
3456 32-bit idle
```

다중 프로세스의 활용

다중 프로세스를 시동할 때에도 -e, -32, -args 등의 옵션을 사용할 수 있습니다.

```
3456:IDL> :22 idl -args 2009 11 30
```

```
22:IDL> print, command_line_args()
```

```
2009 11 30
```

프로세스에 직접 명령 내리기

```
22:IDL> :3456 for l=0L, 100000L do a=randomu(-1, 10000)
```

프로세스가 제법 오래 실행될 것입니다. 실행 중에 프로세스 상태를 확인해 볼 수 있습니다.

```
22:IDL> ::
```

```
0 32-bit idle
```

```
1 32-bit idle
```

```
3456 32-bit busy: for l=0L, 1000000 ....
```

```
22 32-bit idle
```

물론 3456 프로세스가 진행되는 동안 다른 프로세스에서 IDL 작업을 계속할 수 있습니다.

실행 중인 프로세스 멈출(Ctrl+C)

3456 프로세스가 의미 없이 너무 오래 걸린다면 Ctrl+C로 멈출 수도 있습니다. 다만 현재 로그인되어 있는 프로세스가 3456번이 아니라면 해당 프로세스로 들어가야 합니다.

```
22:IDL> :3456
```

```
3456:IDL> print, 'Hello' ;현재 3456은 for... 하느라 바쁩니다.
```

```
IDL session 3456 is busy.
```

```
3456:IDL> ;이 상태에서 Ctrl+C를 눌러 프로세스를 멈출 수 있습니다.
```

```
% Interrupted at: $MAIN$
```