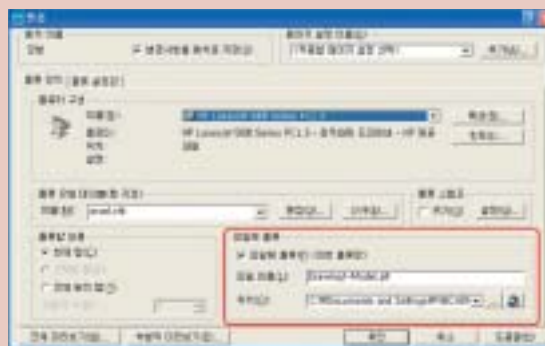


꼭 알아야 할 PLOT 관련 VBA 명령

지난 호에서는 캐드의 윈도우창 제어하기에 대해 알아보았다. 이번 호에서는 꼭 알아두어야 할 플롯 관련 VBA 명령에 대해 자세히 소개한다.



■ 연재순서 ■

제1회 캐드의 윈도우창 제어하기

제2회 꼭 알아야 할 PLOT 관련 VBA 명령

제3회 TEXT 편집 기법(1)

제4회 TEXT 편집 기법(2)

제5회 객체 (Line, Circle 등)의 조작

제6회 레이어와 객체선택 필터링 기법

캐드의 자동화 관련 프로그램이 많지는 않다. Autodesk에서 제공하는 몇 가지 유틸리티와 Express Tools 등들은 사용자의 기대에 많이 떨어지는 수준이다. 특히나 반복적이고 단순한 도면의 출력은 출력도면 매수가 다량이면, 1회성으로 끝나지 않고 필요시마다 지속적으로 이루어지며, 그 형태가 매우 다양하다. 이런 도면의 출력은 캐드 유저들에게 시간적 손실뿐만 아니라, 업무능률 및 사기저하의 효과까지 있어서 자동화가 꼭 이루어져야 한다. AutoCAD의 출력은 버전이 올라가면서 출력방법과 기능이 복잡 다양해지고 있어, 다른 분야 또는 업체의 유저가 작성한 도면을 출력한다는 것은 여간 어려운 일이 아니다. 현재 출력프로그램은 몇 가지의 상업적, 비상업적 프로그램이 있으나, 디자인과 편리성에서만 조금 다를 뿐 기능면에서는 비슷하다고 판단된다. 이런 프로그램들도 각 사용자들이 요구에 정확히 맞아 떨어지지 않아 아쉬움이 많이 있다. 이에 사용자들이 '필요에 따라 간단한 기능의 출력 프로그램을 제작해서 사용하는 것은 어떨까?'하고 생각이 들어 이 글을 써본다.

파일관리 명령

여러 도면을 출력할 때는 파일들을 열고, 닫고, 저장 등 필요한 명령들을 알고 있어야 멀리 출력이 가능해진다. 출력명령들을 알아보기 전에 파일관리 명령들에 대해서 먼저 짚어보고 가보도록 하자.

다음 소스들은 AutoCAD 2002 이상이면 모두 실행되는 코드들이다.

도면 열기(Open)

도면열기에 필요한 매서드(Method)는

Application.Documents.Open [, ReadOnly] (파일이름)

이다. 열기전용으로 도면을 열 때는 ReadOnly 옵션을 덧붙이면 된다. 파일이름은 꼭 따옴표로 둘러 쌓여야 한다. 폴더 및 파일에 띄어쓰기가 포함되어 있어서 따옴표로 둘러 쌓이지 않으면 오류가 발생한다.



이 혁 진

한국철도기술공사 구조설계부 과장으로 근무하고 있으며, 신구대학 외래강사로도 활동하고 있다. 저서로는 <AutoCAD Office>가 있다.
E-Mail | master@vbcad.co.kr
홈페이지 | http://www.vbcad.co.kr

Sub 도면열기예제()

```
Application.Documents.Open ("C:\test.dwg")
```

```
End Sub
```

도면 저장(Save)

도면저장에 필요한 매써드는 아래와 같이 2가지 형태를 취할 수 있다.

① Thisdrawing.Save

② Thisdrawing.SaveAs 파일이름, 저장파일형태

저장파일형태(AutoCAD 2000~2003 버전, R15일 경우)

VBA 입력값	저장 형태	정수값	2진값
acR12_DXF	AutoCAD 릴리즈12/LT2 DXF (*.dxf)	1	0001
acR13_DWG	AutoCAD 릴리즈13/LT95 DWG (*.dwg)	4	0100
acR13_DXF	AutoCAD 릴리즈13/LT95 DXF (*.dxf) 5	0101	
acR14_DWG	AutoCAD 릴리즈14/LT97 DWG (*.dwg)	8	1000
acR14_DXF	AutoCAD 릴리즈14/LT97 DXF (*.dxf)	9	1001
acR15_DWG	AutoCAD 2000 DWG (*.dwg)	12	1100
acR15_DXF	AutoCAD 2000 DXF (*.dxf)	13	1101
acR15_Template	AutoCAD 2000 도면 템플릿 파일 (*.dwt)	14	1110
acNative	최신 도면 릴리즈의 값 (사용하는 버전)	Var.	Var.

저장파일 형태 (AutoCAD 2004~2006 버전, R16일 경우)

VBA 입력값	저장 형태	정수값	2진값
ac2000_dwg	AutoCAD 2000 DWG (*.dwg)	12	0000 1100
ac2000_dxf	AutoCAD 2000 DXF (*.dxf)	13	0000 1101
ac2000_Template	AutoCAD 2000 도면 템플릿 파일 (*.dwt)	14	0000 1110
ac2004_dwg	AutoCAD 2004 DWG (*.dwg)	24	0001 1000
ac2004_dxf	AutoCAD 2004 DXF (*.dxf)	25	0001 1001
ac2004_Template	AutoCAD 2004 도면 템플릿 파일 (*.dwt)	26	0001 1010
acNative	최신 도면 릴리즈의 값 (사용하는 버전)	Var.	Var.

도면을 저장할 때 Save 매써드를 사용할 경우에는 현재의 저장버전에 맞추어 저장할 수 있으며, SaveAs 매써드를 사용할 경우에는 도면 포맷도 지정할 수 있는 장점이 있다.

위의 저장 파일 형태에서 acNative의 정수(Integer) 값은 현재 캐드버전이 R14일 때 자동으로 8로 지정되며 2000버전(R15) 이상 일 때 자동으로 12, 2004(R16) 이후 버전에는 자동으로 24로 지정된다.

정수값과 2진 값은 VB 프로그래머 또는 내부구조를 이해하기 위한 사용자가 보기 싶도록 참조 값을 적어 놓은 것이므로 VBA 사용자는 참고만 하면 된다. 위의 표에서 2진 값의 형태를 분석해 보면 도면 릴리즈가 올라가면서 비트 레벨(자릿수)이 한 단계씩 증가하였고, DXF 일 경우는 마지막 자리(20)의 비트 값이 1로 세팅되었다. 캐드 도면형식이 R16 이후 전체적인 모습을 바꾸었듯이 저장파일 형태도 4비트 형식에서 8비트로 바뀌었다. Template는 두 번째 자리(21 위치)가 1로 세팅된 것을 확인할 수 있다.

Sub 도면저장예제1()

```
Thisdrawing.Save
```

```
End Sub
```

Sub 도면저장예제2()

```
Thisdrawing.SaveAs "c:\test", acR14_dwg
```

```
End Sub
```

도면 닫기(Close)

도면 닫기에 필요한 매써드는 다음과 같다.

```
Application.Documents.Close ([SaveChanges], FileName)
```

■ SaveChange : True (저장 후 닫기)

False (저장 없이 닫기)

변경된 부분이 있는 도면을 닫을 때 저장(Save) 여부를 설정할 수 있으며, 다른 파일이름으로도 저장이 가능하다. 변경된 부분이 없으면 SaveChange와 FileName은 무시된다.

Sub 도면닫기예제()

```
Thisdrawing.Close (False)
```

```
End Sub
```

Purge 실행

도면의 용량을 줄이는 명령으로 Purge가 있다. 도면을 닫기 전에 자동으로 Purge를 수행하면, 사용자가 일일이 신경 쓰지 않아도 용량 관리를 할 수 있을 것이다. 현재 많은 도면을 자동으로 Purge시켜 주는 프로그램도 공개되어 있다. 아주 간단한 매써드를 사용하면 독자들도 충분히 제작할 수 있는 부분이다. 도면을 Purge하는 매써드는 아래와 같다.

```
Thisdrawing.PurgeAll
```

이 매써드는 명령행(Command Line)에 PURGE를 입력하고 모든 옵션을 선택한 다음 '모든 것을 소거하시겠습니까?' 프롬프트에 예(Yes)를 선택하는 것과 같은 결과를 발생한다. 삭제된 객체는 이 매써드(Method)를 사용하여 소거될 때까지 문서에 남는다.

Sub 용량줄이기 예제()

```
Thisdrawing.PurgeAll
```

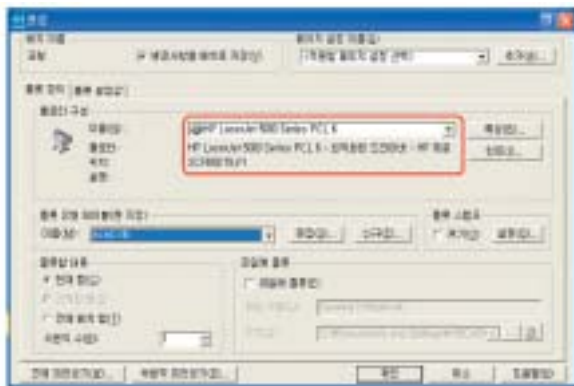
```
End Sub
```

PLOT 관련 명령

VBA 프로그래머들이 어떤 매써드나 프로퍼티(Property)를 어떻게 접목해야 할지 모른다는게 큰 어려움이다. 간단한 명령어를 알고 나면 해결하기 쉬운 문제임에도 불구하고, 그 명령어를 찾기 위해 시간과 노력을 많이 허비하게 마련이다. 특히나 출력에 관한 매써드와 프로퍼티(Property)는 짐작이 되는 단어들이 아니어서 찾기가 더더욱 어렵다. 그래서 출력에 꼭 필요하거나 빈번히 사용되는 매써드와 프로퍼티(Property)를 나열하여 사용자들의 시간을 줄임과 동시에 상당부분 도움이 되도록 기술하여 보았다.

출력장치(Device) 목록 가져오기

출력장치(Device)라 함은 Printer 또는 Plotter를 의미한다. 현재 각자 사용하는 프린터 목록이 다르므로 개인설정과 도면에 맞는 프린터 목록이 필요하다. 이 때 사용하는 매써드로서 Layout 또는 PlotConfiguration에서 받아들여져야 한다.



▲ Plot창에서 출력장치(Device)

```
Set Layout=ThisDrawing.ActiveLayout
출력장치들_목록 = Layout.GetPlotDeviceNames()
```

위의 매써드를 사용하기 전에 layout.RefreshPlotDeviceInfo 명령을 실행하여 Device 정보를 갱신하면 최신의 목록을 취할 수 있다. 아래의 코드를 실행시키면 현재 본인의 출력장치목록을 그림과 같이 직접실행창에서 확인할 수 있다.



▲ 직접 실행창에서 출력장치 목록을 확인

Sub 출력장치목록예제()

```
Dim layout As AcadLayout
Dim 출력장치들_목록 As Variant
Dim i As Integer

Set layout = ThisDrawing.ActiveLayout
출력장치들_목록 = layout.GetPlotDeviceNames()

'목록 갱신
Call layout.RefreshPlotDeviceInfo

'순차적으로 목록 출력
For i = LBound(출력장치들_목록) To UBound(출력장치들_목록)
    Debug.Print 출력장치들_목록(i)
Next

End Sub
```

위의 코드 중에서 Debug.Print 명령은 실행된 결과를 직접 실행창에 나타내도록 하는 코드이다.

출력장치(Device) 새로 갱신(Refresh)

위에서 설명한 것처럼 출력장치목록을 갱신하는데 사용된다.

```
Set Layout=ThisDrawing.ActiveLayout
Layout.RefreshPlotDeviceInfo
```

출력장치(Device) 설정

출력장치 목록 중에서 출력하고자 하는 출력장치를 설정하는 속성(Property) 값은 아래와 같다. 사용 출력장치 이름은 따옴표(“)안에 포함시켜 기재하여야 한다.

```
Set Layout=ThisDrawing.ActiveLayout
Layout.ConfigName = "출력장치이름"
```

아래의 코드는 "HP LaserJet 5000 Series PCL 6" 제품으로 설정되는 예제이다.

Sub 출력장치설정예제()

```
Dim layout As AcadLayout

Set layout = ThisDrawing.ActiveLayout
layout.ConfigName = "HP LaserJet 5000 Series PCL 6"

End Sub
```

본인이 사용하지 않는 장치를 선택하였을 경우에는 아래와 같은 에러 메시지가 발생하므로 사용시에 주의하기 바라며, ComboBox 등을

이용하여 사용자가 지정된 목록 외에는 선택할 수 없도록 조치를 해 놓으면 좋을 듯 싶다.



▲ 목록에 없는 장치를 선택하였을 경우 에러

스타일 테이블 목록 가져오기(Plot Style Table)

캐드의 출력유형을 정하는 스타일 테이블(STB 또는 CTB)을 가져 오는 기능이다. 현재 열려 있는 도면의 Layout에서 사용 가능한 테이블의 목록을 불러온다.

```
Set layout = ThisDrawing.ActiveLayout
테이블_목록 = layout.GetPlotStyleTableNames()
```



▲ 목록에 없는 장치를 선택하였을 경우 에러

Sub 스타일테이블목록예제()

```
Dim layout As AcadLayout
Dim 테이블_목록 As Variant
Dim i As Integer

Set layout = ThisDrawing.ActiveLayout
테이블_목록 = layout.GetPlotStyleTableNames()

'목록을 직접실행창에 보여줌
For i = LBound(테이블_목록) To UBound(테이블_목록)
    Debug.Print 테이블_목록(i)
Next i

End Sub
```

스타일 테이블 설정(Plot Style Table)

출력하려는 스타일 테이블 이름을 지정한다. 간혹 몇몇의 도면은 스타일 테이블 특성을 받아들이지 못해서 에러(Error)를 발생하기도 하는데, 이는 도면자체에 문제가 있는 것으로 판단된다. 완벽한 STB 또는 CTB를 작성하지 못한 경우에도 에러(Error)가 발생하므로 새로

운 테이블 생성시 유의하기 바란다.

```
Set Layout=ThisDrawing.ActiveLayout
Layout.StyleSheet = "테이블 이름"
```

아래는 AutoCAD를 배포할 때 기본적으로 설치되어 있는 'acad.ctb'로 테이블을 설정하는 예제이다.

Sub 스타일테이블설정예제()

```
Dim layout As AcadLayout

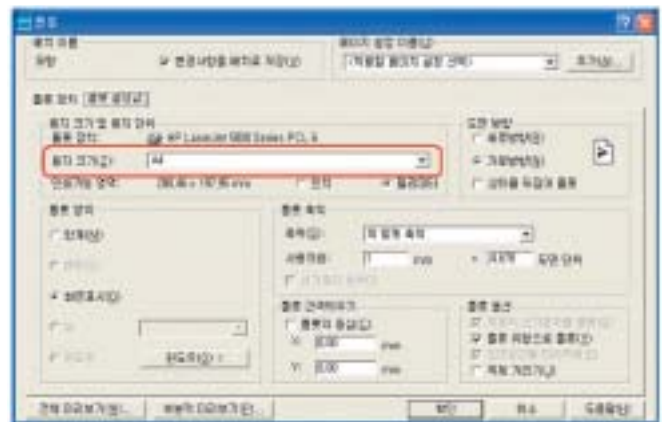
Set layout = ThisDrawing.ActiveLayout
layout.StyleSheet = "acad.ctb"

End Sub
```

용지 이름(Paper Size) 가져오기

용지이름은 출력장치마다 다르게 지정되어 있고, 그 크기도 다양하다. 각 출력장치에서 사용할 수 있는 용지 이름을 가지고 와서 적용해야 에러(Error) 없이 정확한 결과값을 얻을 수 있다. 용지이름을 가져 오기에 앞서 늘 출력장치(Device)를 설정하는 것을 잊지 말도록 하자.

```
Set Layout=ThisDrawing.ActiveLayout
용지이름들=Layout.GetCanonicalMediaNames()
```



▲ Plot창에서 용지크기(Paper Size)

용지이름은 프로그램내에서 인식하는 이름과 사용자에게 보여지는 이름이 서로 다르다. GetCanonicalMediaNames()은 프로그램 내에서 인식하는 이름이고, 이 이름을 가지고 GetLocaleMediaName()이란 Property를 다시 한 번 적용시켜 중복 사용하면 사용자에게 보여지는 이름들이 나타난다. 아래의 예제에 두 가지 이름들을 동시에 나타내어 이를 비교할 수 있게 만들었으며, 출력장치의 종류에 따라 GetCanonicalMediaNames()에서 얻은 이름과 GetLocaleMediaName()에서 얻은 이름이 서로 같을 수도 있다.

Sub 용지이름목록예제()

```
Dim layout As AcadLayout
Dim 용지_목록 As Variant
Dim i As Integer

Set layout = ThisDrawing.ActiveLayout

'출력장치 우선 지정
layout.ConfigName = "HP LaserJet 5000 Series PCL 6"

'용지 목록
용지_목록 = layout.GetCanonicalMediaNames()

For i = LBound(용지_목록) To UBound(용지_목록)
    Debug.Print 용지_목록(i)
    Debug.Print layout.GetLocaleMediaName(용지_목록(i))
Next i

End Sub
```

용지 설정(Paper Size)

용지이름을 지정할 때 사용하는 속성(Property)이다. 용지이름 설정 전에 출력장치를 한 번 더 지정하여 줌으로서 사용자 부주의로 인한 에러(Error)를 줄일 수 있다.

```
Set Layout=Thisdrawing.ActiveLayout
Layout.CanonicalMediaName = "용지이름"
```

Sub 용지크기설정예제()

```
Dim layout As AcadLayout

Set layout = ThisDrawing.ActiveLayout

'출력장치 우선 지정
layout.ConfigName = "HP LaserJet 5000 Series PCL 6"

'용지 지정
layout.CanonicalMediaName = "A3"

End Sub
```

용지 방향(Paper Orientation)

용지의 방향을 설정할 때 사용한다. 용지의 방향이라기 보다는 용지의 각도가 더 이해하기 쉬우리라 생각된다. 용지의 각도는 0도, 90도, 180도, 270도의 4가지만 설정이 가능하다.

```
Set Layout=Thisdrawing.ActiveLayout
Layout.PlotRotation = <설정값>
```

설정값

VBA 입력값	설명	정수값
ac0degrees	용지회전 0도	0
ac90degrees	용지회전 90도	1
ac180degrees	용지회전 180도	2
ac270degrees	용지회전 270도	3



▲ Plot창에서 용지방향 (Paper Rotate)

용지 단위 설정 (Paper Units)

단위를 설정할 때 사용하는 속성(Property)이다. 우리나라의 사용자는 대부분 밀리미터(Millimeter)를 기본값으로 사용하면 되겠다.

```
Set Layout=Thisdrawing.ActiveLayout
Layout.PaperUnits = <설정값>
```

다음의 예제는 용지방향 및 단위를 변경하는 예제이다. 아래를 실행시키면 용지방향을 90도 회전하며, 단위는 Inch를 사용하게 된다.

설정값

VBA 입력값	설명	정수값
acInches	inch로 설정	0
acMillimeters	mm로 설정	1
acPixels	Pixel로 설정	2



▲ Plot창에서 용지단위(Paper Units)

Sub 용지방향_단위설정예제()

```
Dim layout As AcadLayout
layout.PlotRotation = ac90degrees
layout.PaperUnits = acInches

End Sub
```

출력영역(Plot Area)

출력하고자 하는 영역을 지정할 때 사용하는 속성(Property)으로,

특히 Window로 출력하려면 좌표값을 요구한다. Windows로 출력하는 경우는 다음 항목에서 다시 언급토록 하겠다.

```
Set Layout=ThisDrawing.ActiveLayout
Layout.PlotType = <설정값>
```

설정값

VBA 입력값	설명	정수값
acDisplay	Display영역으로 설정	0
acExtents	Extents영역으로 설정	1
acLimits	Limits영역으로 설정	2
acView	View영역으로 설정	3
acWindow	Window영역으로 설정	4
acLayout	Layout영역으로 설정	5



▲ Plot창에서 출력영역 (Plot Area)

윈도우 영역으로 출력

윈도우영역으로 출력을 하려면 AutoCAD에 출력영역을 좌표로 알려주어야 한다. 출력영역을 설정하는 PlotType을 acWindow로 설정하고 2차원 좌표값을 제공하는 방법을 예제를 통해 알아보자.

```
Layout.SetWindowToPlot Point1, Point2
```

Sub 윈도우영역출력예제()

```
Dim layout As AcadLayout
Dim Point1(0 To 1) As Double
Dim Point2(0 To 1) As Double
Set layout = ThisDrawing.ActiveLayout
```

'출력좌표값 설정(2차원 값)

```
Point1(0) = 0
Point1(1) = 0
Point2(0) = 841
Point2(1) = 594
```

```
layout.SetWindowToPlot Point1, Point2
```

'윈도우로 출력영역을 정함

```
layout.PlotType = acWindow
```

```
End Sub
```

축척설정(Scale)

축척은 '표준축척'과 '사용자축척' 두 가지를 설정할 수 있다. 출력 자동화에서는 축척 설정이 잘 못 설정된 경우, 일부 또는 백지로 출력될 수 있으므로 주의해서 작성해야 한다. 표준축척은 1:1, 1:10, 1:100 등 보편적인 축척값을 가지고 있으나, 프로그래머는 특히 '꼭 맞게 축척(ScaleToFit)'의 값이 acScaleToFit(정수값=0)임을 알아두어야 한다. 예제는 제일 뒤쪽에 소개되는 Sample Code를 참조하기 바란다.

● 꼭 맞게 축척(ScaleToFit)

```
Set Layout=ThisDrawing.ActiveLayout
Layout.StandardScale=acScaleToFit
```

● 사용자 축척 설정

```
Set Layout=ThisDrawing.ActiveLayout
Layout.SetCustomScale, <축척1>, <축척2>
```

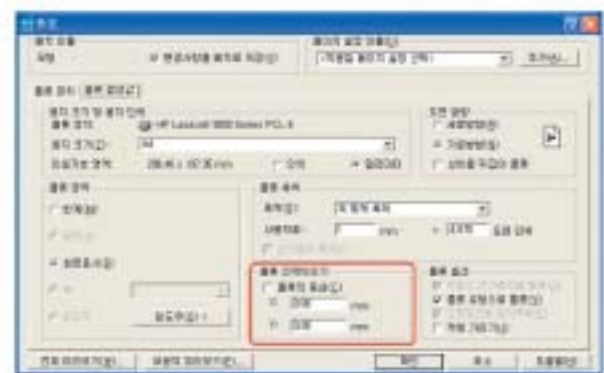


▲ Plot창에서 축척(PLOT Scale)

출력간격(Origin) 조정

출력 시 종이의 특정위치로 출력내용을 이동시키고자 할 때 사용하는 기능으로 X좌표와 Y좌표값을 주어 그 값만큼 위치를 이동할 수 있다. 좌표값은 2차원 좌표계를 사용하여야 한다.

```
Set Layout=ThisDrawing.ActiveLayout
Layout.PlotOrigin = <좌표>
```



▲ Plot창에서 출력간격(PLOT Origin)

Sub 출력간격예제()

```
Dim layout As AcadLayout
Dim 좌표(0 To 1) As Double

좌표(0) = 5 : 좌표(1) = -1

Set layout = ThisDrawing.ActiveLayout
layout.PlotOrigin = 좌표

End Sub
```

참고

AutoCAD 2005 이하 버전에서는 X좌표와 Y좌표가 바뀌어져 있다. 2005버전부터는 버그가 수정되어 정상적으로 작동된다.

중앙에 출력(Center the Plot)

종이의 중앙에 출력내용을 위치시킬 때 사용하는 기능이다. 이 기능을 사용하면 자동으로 플롯간격픽우기(Plot Origin)의 X, Y값을 자동으로 계산하여 적용시킨다. 예제는 제일 뒤쪽에 소개되는 Sample Code를 참조하기 바란다.

```
Set Layout=Thisdrawing.ActiveLayout
Layout.CenterPlot = <설정값>
```

- 설정값 : True (중앙에 출력)
- False (출력간격(Origin)에 의거 출력)



▲ Plot창에서 중앙에 출력(Plot Center)

출력매수

도면을 여러장 출력할 때 사용하는 기능이다. 명령창에서 수동으로 출력할 때 Plot 창에는 없는 기능으로 프로그래밍에서만 가능하다.

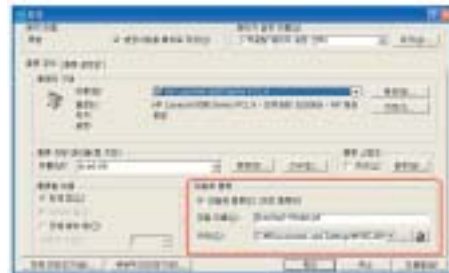
```
ThisDrawing.Plot.NumberOfCopies = <출력매수>
```

파일로 출력(Plot to File)

출력장치로 출력하는 것이 아니라, PLT 파일 형식으로 저장매체에 저장해 두었다가 필요할 때마다 PLT파일을 출력장치에 보내어 출력 결과를 얻어내는 방법으로, 용량이 큰 파일 또는 스타일테이블(Style Table) 설정이 복잡한 파일 등을 미리 PLT 파일로 출력해 놓으면 용

이하게 재출력할 수 있을 뿐만 아니라, PLT 파일의 용량이 적어 출력 장치에 빠르게 전달되어 작업시간을 많이 줄일 수 있다.

```
ThisDrawing.Plot.PlotToFile(<파일이름>)
```



▲ Plot 창에서 파일에 출력(PlotToFile)

출력장치로 출력

직접 지정된 출력장치에 출력할 때 사용된다. 출력자동 프로그램의 제일 뒤에 쓰여져야 한다.

```
ThisDrawing.Plot.PlotToDevice
```

VBA로 작성한 Plot Sample Code

Sub 출력예제()

```
Application.Documents.Open ("C:\test.dwg")

Dim Layout As AcadLayout
Set Layout = ThisDrawing.ActiveLayout

'출력장치 설정
Layout.ConfigName = "HP LaserJet 5000 Series PCL 6"

'스타일테이블 설정
Layout.StyleSheet = "acad.ctb"

'용지설정
Layout.CanonicalMediaName = "A3"

'출력영역 설정
Layout.PlotType = acExtents

'도면축척을 1:10으로 설정
Layout.SetCustomScale , 1 , 10

'중앙에 출력 옵션 설정
Layout.CenterPlot = True

'출력매수를 2매로 설정
ThisDrawing.Plot.NumberOfCopies = 2

'설정한 출력장치로 출력을 시작함
ThisDrawing.Plot.PlotToDevice

'도면의 용량 줄임
ThisDrawing.PurgeAll

'도면을 저장하고 끝냄
ThisDrawing.Close (True)

End Sub
```