 IntraMap/Objects

### *PROGRAMMERS GUIDE*

IntraMap/Objects3.0

Programmers GUIDE For C#

## (주)한국공간정보통신

#### 2011. 11.

 IntraMap/Objects ***PROGRAMMERS GUIDE***

목 차

1. [INTRAMAP/OBJECTS 소개 4](#_TOC_250013)
2. [데이터 베이스 6](#_TOC_250012)
3. [맵 데이터 셋 18](#_TOC_250011)
4. [GMAPVIEW의 구조 25](#_TOC_250010)
5. [화면 상의 레이어별 색상관리 43](#_TOC_250009)
6. [화면 제어 77](#_TOC_250008)
7. [GEOMETRY 공간 질의 및 SQL을 이용한 데이터 검색 84](#_TOC_250007)
8. [GEOMETRY 간 공간연산 100](#_TOC_250006)
9. [공간 데이터의 편집 129](#_TOC_250005)
10. [RASTER 이미지 편집 142](#_TOC_250004)
11. [SHAPE 파일 가져오기/내보내기 150](#_TOC_250003)
12. [CAD 파일 가져오기/내보내기 159](#_TOC_250002)
13. [화면 출력 167](#_TOC_250001)

[14. 색인 175](#_TOC_250000)

 IntraMap/Objects ***PROGRAMMERS GUIDE*** IntraMap/Objects 3.0 소프트웨어는 ㈜ 한국공간정보통신의 자체기술로 만들어져 그 저작권 및 지적소유권을 ㈜ 한국공간정보통신이 보유하고 있습니다. 본 서의 관련 내용 뿐 아니라 소프트웨어의 구현 방법에 대한 내용 등에 있어 ㈜ 한국공 간정보통신의 허락없이 판매의 목적으로 복제되거나 인용될 수 없습니다.

기술 지원

㈜ 한국공간정보통신 부설 한국공간정보연구소

주 소 : 서울시 강남구 대치2동 943-10 J&A 빌딩 8층 전 화 : (02)2138-1031

팩 스 : (02)2138-1035

E-Mail : [webmaster@ksic.net](mailto:webmaster@ksic.net)

Web Site

인터넷 홈페이지 : [http://www.ksic.net](http://www.ksic.net/) 고객지원 홈페이지 : [http://helpdesk.ksic.net](http://helpdesk.ksic.net/)

 IntraMap/Objects ***PROGRAMMERS GUIDE***

# IntraMap/Objects 소개

I n t r a M a p / Ob j e c t s 는 자사 포맷 인 X Ma p 및 A c c e s s , S p a t i a l S e r v e r (IntraMap Server)와 연동하여 GIS 데이터 관리기능을 지원하는 ActiveX Component 기반의 GIS 개발자 라이브러리이다. 주목 할 점은 개발자 Library라 는 점이다. 일반 사용자가 대상이 아닌 GIS관련 서비스를 하기 위한 개발자를 위한 Library이다. 개발자는 해당 Library를 이용하여 손쉽고 빠르게 GIS 시스템 개발이 가능한 특징을 가지고 있다. 그러므로 제품이 EXE가 아닌 DLL형태로 제 공 된다.

## 개발 환경

IntraMap/Objects는 ActiveX로 구성되어 VC++/VB/Delphi/C++Builder/ VC.net 등 COM모델이 구동될 수 있는 환경에서 구현될 수 있다.

## 특징

1. 다양한 GIS 데이터 포맷을 지원

IntraMap/Objects는 기존 IntraMap/Viewer의 모든 공간데이터 데이터베이스 와 연동한 GIS데이터를 관리 및 SHP/DWG,DXF/XMap등의 파일포맷을 지원하며 항공 사진, 위성 영상 및 각종 이미지 파일을 지원한다.

* XMap \* AutoCAD DWG, DXF(R12~R14) \* ESRI Shape(.shp)
* GeoTiff/Tiff, JPG, GIF, BMP
* IntraMap Server \* ACCESS



 IntraMap/Objects ***PROGRAMMERS GUIDE***

1. GIS 기능

지도 데이터 제어의 기본적인 기능이 모두 포함되어 있으며, 레이어 스타일을 제어 하기 위한 폼도 지원된다. 개발자가 구현하기 까다로운 공간 검색기능인 Clipping, Merge, Union, Intersect, Buffering 등을 함수 형태로 제공 되어 개발자 는 필요한 기능의 함수를 사용함으로써 필요한 기능을 손쉽게 구현 할 수 있다. 또한 공간데이터 베이스 연산인 Select, Update, Insert, Delete기능을 함수 형태 로 제공 하며 모든 데이터 인터페이스를 데이터베이스 형태로 제공하기 때문에 보다 직관적인 인터페이스로 개발자가 쉽게 어플리케이션을 개발 할 수 있다.

## 개발 지원

본사에서 개발자 교육을 통한 개발 지원 및 온라인 도움말 구축 등 다양한 개발 지원정책을 통해 개발자가 개발하기에 더욱 쉬운 환경을 지원한다.

## 향후 개발 방향

앞으로 IntraMap/Objects는 IntraMap/Server와 연계된 시스템을 구축함으로서 보다 강력한 필터링 및 공간 분석 기능을 제공할 예정이며, Mobile과 Unix계열 등에 이르기까지 사용 가능한 제품군으로 발전시켜 나갈 계획이다.



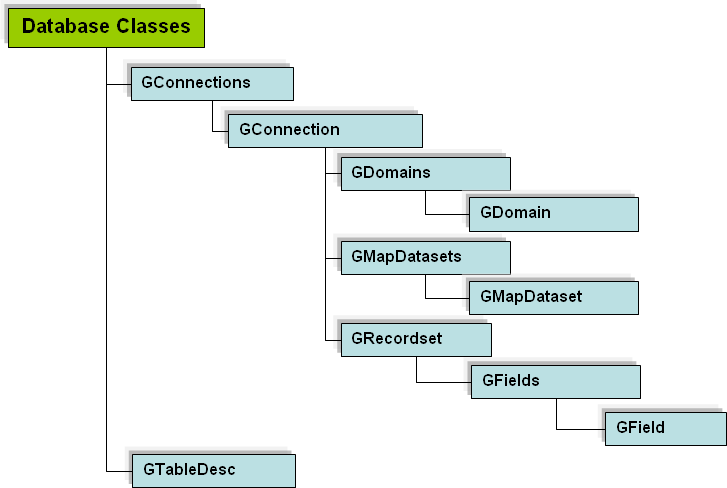
 IntraMap/Objects ***PROGRAMMERS GUIDE***

# 데이터 베이스

1장에서의 개발환경 등록 및 셋팅을 정확하게 수행하였다면 C#에서 IntraMap/Objects 3.0을 활용하여 응용 프로그램을 작성할 준비가 완료된 것이 다. 이제는 실제로 공간 데이터가 저장되어 있는 데이터베이스에 연결하여 응용 프로그램을 작성하는 방법과 기능별 구현 예제들을 다루도록 하겠다.

## 새로운 데이터베이스 연결

공간 데이터 파일 또는 공간 데이터베이스로부터 데이터를 다루기 위해서는 우선적으로 이들에 대한 연결 과정이 필요하다. IntraMap/Objects 3.0에서 이러 한 연결과 관리를 위하여 GConnections Object를 제공한다.





 IntraMap/Objects ***PROGRAMMERS GUIDE*** GConnections는 여러 개의 데이터베이스 연결을 가질 수 있으며 각 데이터베 이스의 생성 등 모든 데이터베이스의 연결(GConnection)들을 관리할 수 있는 프라퍼티와 메소드들을 제공한다.

DB 1

GConnection 1

DB n

GConnection n

DB 2

GConnection 2

**GConnections**



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* GConnections

Property

|  |  |
| --- | --- |
| Name | Description |
| Count | 연결된 데이터베이스 개수를 반환합니다. |

Method

|  |  |
| --- | --- |
| Name | Description |
| Add | 새로운 데이터베이스 연결을 추가합니다. |
| Insert | 인덱스를 주어, 해당위치에 데이터베이스 연결을 삽입  합니다. |
| Item | 인덱스를 주어 해당하는 위치의 데이터베이스 연결정  보를 반환합니다. |
| NewConnection | 새로운 데이터베이스 연결을 생성합니다. |
| Remove | 인덱스를 주어 해당 위치의 데이터베이스 연결을 삭  제합니다. |
| RemoveAll | 모든 데이터베이스 연결정보를 삭제합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* GConnection

Property

|  |  |
| --- | --- |
| Name | Description |
| Database | 데이터베이스명을 설정하거나 반환합니다. |
| ID | 데이터베이스 접속을 위한 ID를 반환합니다. |
| IsConnected | 데이터베이스 접속여부를 반환합니다. |
| Password | 데이터베이스 접속을 위한 암호를 설정하거나 반환합  니다. |
| Server | 데이터베이스 서버를 설정하거나 반환합니다. |
| User | 사용자명을 설정하거나 반환합니다. |

Method

|  |  |
| --- | --- |
| Name | Description |
| AddMapDataset | 새로운 맵데이터셋을 mdb 파일에 추가합니다. |
| BeginTrans | 데이터베이스의 트랜잭션 작업을 위한 초기화 작업을  진행합니다. |
| Commit | 데이터베이스 커밋 작업을 수행합니다. |
| Connect | 데이터베이스에 접속합니다. |
| DeleteMapDataset | mdb 파일에 존재하는 맵데이터셋을 삭제합니다 |
| Disconnect | 데이터베이스 접속을 해제 합니다. |
| Execute | SQL 질의문을 수행합니다. |
| GetMapDomains | 연결된 데이터베이스에 존재하는 도메인 정보를 반환합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

|  |  |
| --- | --- |
| GetMapDatasets | 연결된 데이터베이스에 존재하는 맵데이터셋 정보를  반환합니다. |
| OpenRecordset | SLQ 질의를 통해 검색된 레코드 셋을 반환합니다. |
| Rollback | 데이터베이스 롤백 작업을 수행합니다. |
| UpdateConnection | 데이터베이스와의 연결 정보를 업데이트합니다 |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

데이터베이스 연결

아래 예제는 Microsoft ACCESS의 파일 DB에 GConnections Object를 이용하 여 연결하는 예제로 필요한 테이블이 존재하는 MDB 파일이 존재하여야 한다. GConnections의 변수명은 GConns으로 선언하였으며, 전체 프로젝트 내에서 사 용되도록 전역 변수로 선언하였다.

public static GConnections GConns = new GConnections(); [예제 2-1]

Using IMapObject3Lib;

namespace DBConnection

{

Public partial class Form1 : Form

{

// 전역 변수 선언

// 프로젝트의 데이터베이스 연결들을 관리

public static GConnections GConns = new GConnections(); public Form1()

{

InitializeComponent();

}

// Form\_Load 부분 설정

private void Form1\_Load(object sender, EventArgs e)

{

 IntraMap/Objects ***PROGRAMMERS GUIDE***

// MDB와의 연결 정보를 설정하여 GConnections에 추가할 변수

GConnection GConn;

// 새로운 데이터베이스(MDB) 연결 생성

GConn = GConns.NewConnection("MSACCESS");

// 연결 권한 설정 GConn.User = ""; GConn.Password = "";

// 데이터베이스 파일 경로

string sPath = Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles); GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb";

// 데이터베이스 연결 시도

GConn.Connect();

// GConnectioins에 연결 추가

GConns.Add(GConn);

위 코드를 실행하면 GConnections가 GConnection을 하나 생성시키고 새로 생성된 GConnection이 db.mdb의 데이터베이스 연결 정보를 갖게 된다. 예제에서는 MDB 파일에 연결하기 위하여 GConnections의 NewConnection와 GConnection의 Database 메소드는 매개변수로 “MSACCESS”와 경로를 설정하였 지만, 네트워크상의 IntraMap/Server에 접근하기 위해서는 아래와 같이 설정해 주어야 한다. 또한, User와 Password로 접근 권한을 부여할 수도 있다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***

GConn = GConns.NewConnection("IMAPSERVER");

// 연결 권한 설정 GConn.User = ""; GConn.Password = "";

// 서버의 위치

GConn.Server = "서버의 IP 주소 : 포트 번호";

서버로 연결될경우 GConnections에서 IntraMapSevert용 Connection을 생성한 다

연결될 User와 Password를 입력하고 Severt에 IP주소 포트번호를 입력한다.



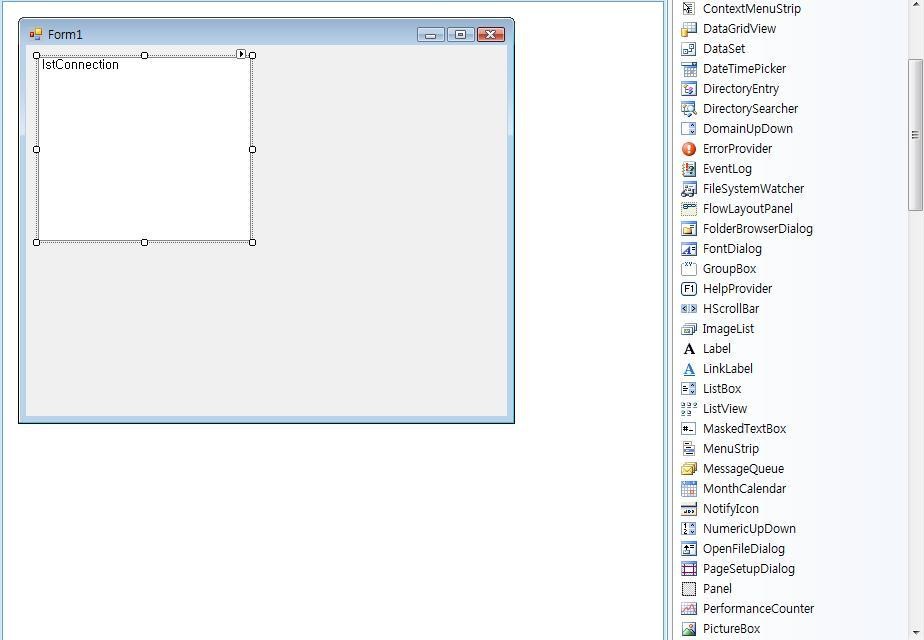
 IntraMap/Objects ***PROGRAMMERS GUIDE***

## 데이터베이스 연결 관리

GConnections가 GConnectoin을 여러 개 생성하게 되면 GConnections는 각 각의 연결에 대하여 식별할 수 있는 정보(ID)를 가지게 되며, 다양한 프라퍼티와 메소드들로 이들에 대한 관리과 활용을 용이하게 하고 있다.

GConnection 정보 얻어오기

다음의 예제는 GConnections가 생성하여 가지고 있는 GConnection의 식별 (ID) 정보를 가져와 ListBox에 목록의 형태로 구현하는 간단한 예제이다. 사용자 들이 혼동하지 않도록 앞의 예제 코드와 함께 실었으며, 추가된 코드에 관한 설 명은 볼드체로 표시하였다.

다음예제를 실행하기 전 Visual Studio에서 ListBox컨트롤을 추가시켜야 한다. ListBox컨트롤의 변수명으로 lstConnection을 설정해준다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***

[예제 2-2]

using IMapObjects3Lib; namespace 2\_MapDataset

{

public partial class Form1 : Form

{

public static GConnectionsClass GConns = new GConnectionsClass();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

GConnection GConn = GConns.NewConnection("MSACCESS"); GMapDatasets mdatasets;

GMapDataset mDataset;

string sPath = Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles);

GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb"; GConn.User = "";

GConn.Password = "";



 IntraMap/Objects ***PROGRAMMERS GUIDE***

GConn.Connect();

GConns.Add(GConn);

mdatasets = GConns.Item(0).GetMapDatasets(); lstMapDataset.Items.Clear();

for (int i = 0; i < mdatasets.Count; i++)

{ lstMapDataset.Items.Add(i + " " + mdatasets.GetMapDataset(i).Name.ToString());

}

}

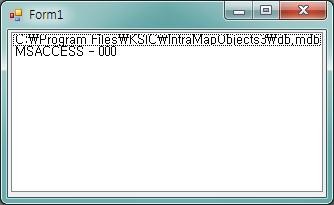
}

}

위 코드를 수행하면 GConnections에 생성된 GConnection의 ID, 데이터베이스 경로가 리스트 박스에 삽입된다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***



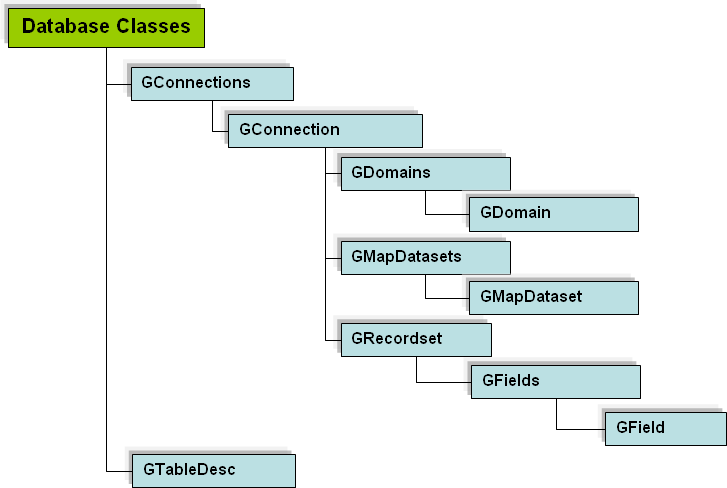


 IntraMap/Objects ***PROGRAMMERS GUIDE***

# 맵 데이터 셋

GMapDatasets는 GConnections으로 연결된 데이터베이스 내의 공간 데이터테 이블들에 대한 정보를 가지고 있는 오브젝트이며, 하나의 공간 데이터 테이블에 대한 정보를 가지고 있는 GMapDataset들의 집합이다.

GMapDataset는 데이터베이스에 저장되어 있는 각각의 MapDataset의 정보에 접근하기위한 ActiveX 개체이다. 그리고 MapDataset란 공통된 타입 및 속성을 가지는 데이터들의 집합을 가르킨다.



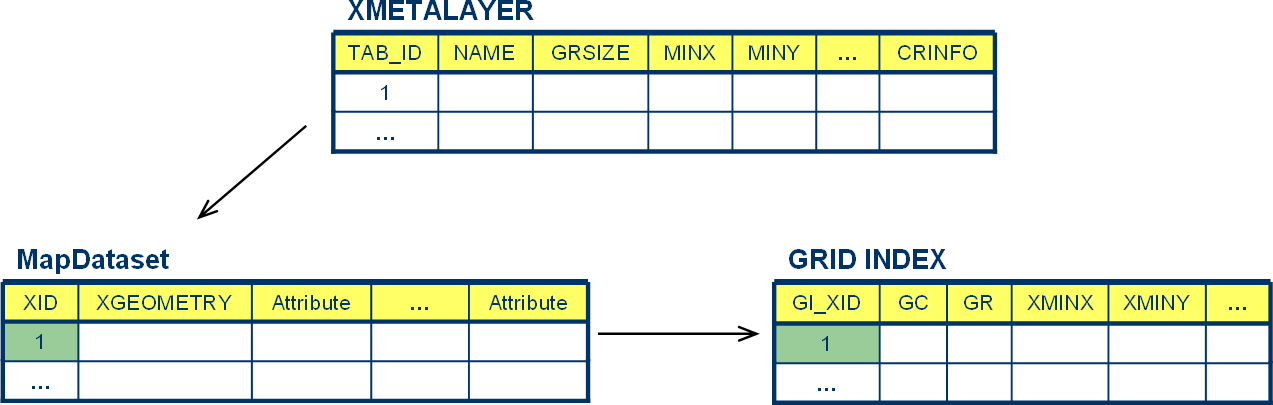


 IntraMap/Objects ***PROGRAMMERS GUIDE***

MapDataset의 물리적 저장구조를 살펴보면, GMapDataset은 크게 XMETALAYER, XDOMAIN, Grid Index, MapDataset 테이블로 이루어진다.

XMETALAYER 테이블은 데이터베이스에 존재하는 MapDataset들의 메타 정보 (인덱스 아이디, 지도의 이름, 경계, 축척 등)를 저장한다. Grid Index 테이블은 MapDataset의 그리드 인덱스 정보 저장한다. 그리드 인덱스 정보란 지도 영역 을 일정 크기의 격자로 나누어 지도 상 개체들의 위치를 격자의 행, 열을 통해 저장 및 검색하는 것을 뜻하며, 이를 바탕으로 빠른 지도표출이 가능하다. XDOMAIN 테이블은 MapDataset이 생성될 시에 정의된 지도 공간좌표 상의 범 위 정보를 담고있다. 도메인 영역은 그리드 인덱스 정보의 저장이 이루어지는 경계이므로 도메인 영역 외부에 위치하는 데이터 추가 시에 화면에 표시되지 않는다. MapDataset 테이블은 공간 객체들의 인덱스 아이디, 속성 정보, 지오메 트리 정보 등을 저장한다.

다음의 그림은 GMapDataset 구성 테이블 간의 관계를 보여준다.





 IntraMap/Objects ***PROGRAMMERS GUIDE*** IntraMap/Objects 3.0에서 제공하는 GMapDatasets와 GMaptDataset의 속성과 메소드를 정리해보면 다음의 표와 같다.

* GMapDatasets

Property

|  |  |
| --- | --- |
| Name | Description |
| Count | 전체 맵데이터셋의 개수를 반환합니다. |

Method

|  |  |
| --- | --- |
| Name | Description |
| GetMapDataset | 인덱스를 주어 해당위치의 맵데이터셋을 반환합니다. |

* GMapDataset

Property

|  |  |
| --- | --- |
| Name | Description |
| CreateDate | 맵데이터셋의 날짜를 생성합니다. |
| desc | 맵데이터셋의 설명을 반환합니다. |
| DomainID | 맵데이터셋의 도메인 아이디를 반환합니다. |
| Extent | 맵데이터셋의 전체영역을 반환합니다. |
| GridSize | 맵데이터셋의 그리드 크기를 반환합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

|  |  |
| --- | --- |
| ID | 맵데이터셋의 ID를 반환합니다. |
| Name | 맵데이터셋의 이름을 반환합니다. |
| Scale | 맵데이터셋의 축척정보를 설정 및 반환합니다 |
| Type | 맵데이터셋의 타입을 반환합니다. |
| XOffset | 맵데이터셋의 시작위치의 X좌표를 반환합니다. |
| YOffset | 맵데이터셋의 시작위치의 Y좌표를 반환합니다. |

Method

|  |  |
| --- | --- |
| Name | Description |
| CreateLayer | 맵데이터셋의 레이어를 반환합니다. |
| GetRecords | 현재 맵데이터셋으로터 쿼리문을 통해 해당하는 레코  를 검색하여 반환합니다. |
| Update | 맵데이터셋의 정보를 업데이트합니다. |



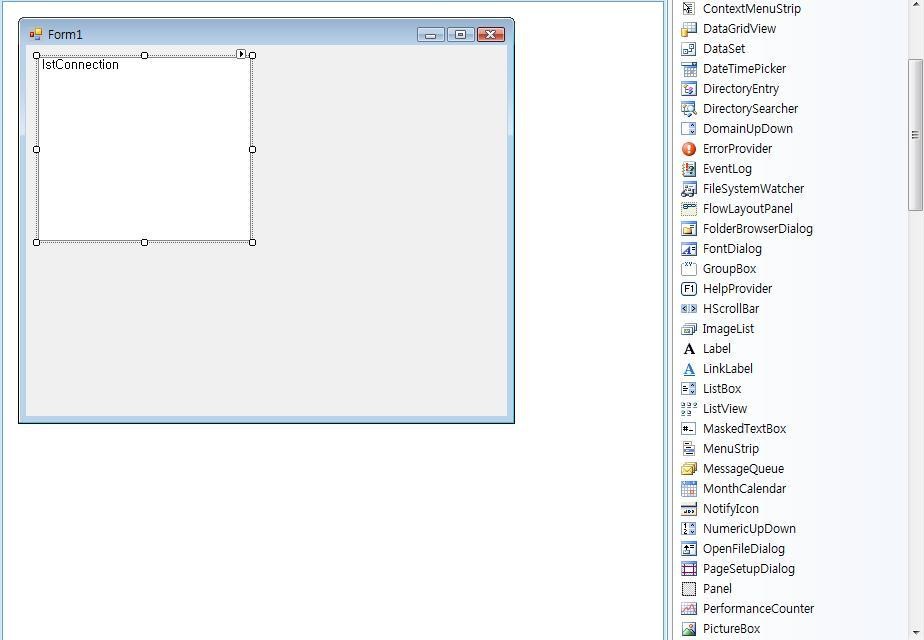
using IMapObjects3Lib; namespace 2\_MapDataset

{

 IntraMap/Objects ***PROGRAMMERS GUIDE***

맵 데이터 셋 정보 얻어오기

다음 예제는 GMapDatasets의 정보를 GConnection에서 얻어오고, GMapDatasets를 통하여 GMapDataset 정보를 얻어오는 기능을 구현한 예제이 다. 본 예제에서 GMapDatasets과 GMapDataset의 변수명은 각각 mDatasets와 mDataset으로 설정하였다.

다음예제를 실행하기 전 Visual Studio에서 ListBox컨트롤을 추가시켜야 한다. ListBox컨트롤의 변수명으로 lstMapDataset 을 설정해준다.

[예제 3-1]



 IntraMap/Objects ***PROGRAMMERS GUIDE***

public partial class Form1 : Form

{

public static GConnectionsClass GConns = new GConnectionsClass();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

GConnection GConn = GConns.NewConnection("MSACCESS"); GMapDatasets mdatasets;

GMapDataset mDataset;

string sPath = Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles);

GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb"; GConn.User = "";

GConn.Password = ""; GConn.Connect();

GConns.Add(GConn);

mdatasets = GConns.Item(0).GetMapDatasets();



 IntraMap/Objects ***PROGRAMMERS GUIDE***

lstMapDataset.Items.Clear();

for (int i = 0; i < mdatasets.Count; i++)

{

lstMapDataset.Items.Add(i + " " + mdatasets.GetMapDataset(i).Name.ToString());

}

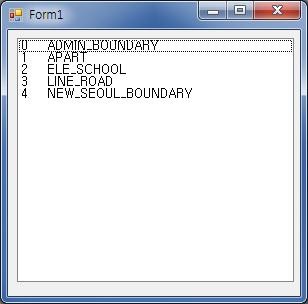
}

}

}

위 예제에서는 GConns.Item(0)으로 GConnections의 첫번째 GConnection 정

보를 가지고 왔지만, 여러 개의 GConnection이 연결된 경우, 사용자는 index를 설정하여 임의의 GConnection 정보를 가져올 수 있다. 다음 그림은 위의 예제 를 수행하여 lstMapdataset 리스트 박스에 추가된 GMapDataset의 목록들을 나 타낸다.

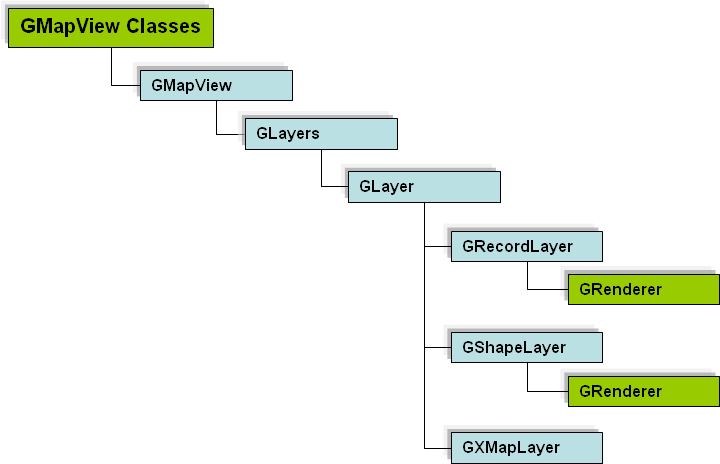




 IntraMap/Objects ***PROGRAMMERS GUIDE***

# GMapView의 구조

GMapView는 공간 데이터베이스의 지도 데이터를 그래픽 화면으로 보여 주고 화면 제어 및 공간 질의를 수행할 수 있는 클래스로서 다양한 속성, 메소드, 이 벤트를 제공한다.





 IntraMap/Objects ***PROGRAMMERS GUIDE***

* GMapView

Property

|  |  |
| --- | --- |
| Name | Description |
| AllowLableOverlap | 레이블의 중첩 허용 여부를 설정합니다. |
| AllowZoomEnable | 확대/축소 허용 여부를 설정합니다. |
| BackgroundColor | GMapView의 배경색상을 설정하거나 반환합니다. |
| DirectionSymbolIndex | 방위 심볼의 인덱스 번호를 설정합니다. |
| DirectionSymbolUse | 방위 심볼의 사용여부를 설정합니다. |
| DisplayScale | GMapView의 축척을 설정하거나 반환합니다. |
| DrawStopEnable | 드래그 도중에 다른 이벤트 호출 가능여부를  반환하거나 설정합니다. |
| EditMode | 편집 모드의 타입을 반환하거나 설정합니다. |
| EditSymbol | GMapView 상에서 편집작업을 수행할 때, 편집 대상 객체의 펜, 브러쉬, 아이콘 등의 속성값을  반환하거나 설정합니다. |
| Extent | GMapView의 현재 범위를 반환 또는 설정합니다. |
| HilightEnable | 객체의 하이라이트 가능여부를 설정합니다. |
| HilightSymbol | 객체를 하이라이트하였을 때, 하이라이트된 객 체의 펜, 브러쉬, 아이콘 등의 속성값을 반환하  거나 설정합니다. |
| IconImageList1 | GMapView에서 사용할 첫 번째 아이콘 이미지  리스트를 설정합니다. |
| IconImageList2 | GMapView에서 사용할 두 번째 아이콘 이미지 |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 리스트를 설정합니다. | | | | | | |
| IconImageList3 | GMapView에서 사용할 세 번째 아이콘 이미지  리스트를 설정합니다. | | | | | | |
| Layers | 레이어정보를 설정하거나 반환합니다. | | | | | | |
| MapCursor | GMapview  반환합니다. | 상에서 | 커서의 | 형태를 | | 설정하거나 | |
| MaxDisplayScale | 최대 축척 값을 설정하거나 반환합니다. | | | | | | |
| MinDisplayScale | 최소 축척 값을 설정하거나 반환합니다. | | | | | | |
| MouseMode | MouseMode를 설정하거나 반환 합니다. | | | | | | |
| NoFlashDraw | GMapView 상의 지도를 그려줄 때, 모든 객체  를 동시에 그려줄지 여부를 설정합니다. | | | | | | |
| PaperScale | 화면의 실체 축척을 설정하거나 반환합니다. | | | | | | |
| ScaleBarUse | 스케일바의 사용 여부를 설정하거나 반환합니다. | | | | | | |
| ScrollEnable | GMapView  설정합니다. | 상에서 | 스크롤의 | | 사용가능 | | 여부를 |
| SelectSymbol | GMapview 상에서 사용자가 선택한 객체의 펜, 브러쉬, 아이콘 등의 색상, 유형, 두께 등을 설  정합니다. | | | | | | |
| ShowNextState | ShowNext가 가능한 상태인지를 반환합니다. | | | | | | |
| ShowPreviousState | ShowPrevious가 가능한 상태인지를 반환합니다. | | | | | | |
| Snap | 스내핑 관련 사항을 설정합니다. | | | | | | |
| SnapEnable | 스내핑 가능 여부를 설정합니다. | | | | | | |
| SyncLabelWithLayerOrder | 레이블의 우선순위를 레이어 순서에 따라 조정  할지 여부를 설정합니다. | | | | | | |
| ViewAngle | 지도의 회전각을 설정합니다. | | | | | | |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

Method

|  |  |
| --- | --- |
| Name | Description |
| Cancel | 입력된 점들을 취소합니다. |
| CancelOne | 입력된 마지막 점을 취소합니다. |
| ClearShowStack | 이전보기 스택을 초기화 합니다. |
| Custom\_Refresh | 사용자 모드에서 화면을 Refresh 합니다. |
| DeleteSelectedObjects | 선택 객체를 삭제합니다. |
| DisplayScaleToPaperScale | PaperScale을 DisplayScale로 변환하여 반환합니다. |
| DrawArc | Arc 형태의 도형을 화면에 그립니다. |
| DrawCircle | Circle 형태의 도형을 화면에 그립니다. |
| DrawGeometry | IGeometry 형태에 속하는 도형을 화면에 그립니다. |
| DrawLeftBoxes | 라인의 왼쪽 방향으로 버퍼링하여 박스를 그  립니다. |
| DrawRightBoxes | 라인의 오른쪽 방향으로 버퍼링하여 박스를  그립니다. |
| DrawText | 지정한 위치에 Text를 그립니다. |
| EndEditObject | 객체 편집의 종료를 선언합니다. |
| Fit | 모든 레이어의 전체 영역을 표시합니다. |
| FromMapDistance | 지도상의 거리를 화면상의 거리로 변환합니다. |
| FromMapPoint | 포인트 객체의 지도좌표값을 화면좌표값으로  변환합니다. |
| FromMapXY | X, Y 형태의 지도좌표를 화면좌표로 변환합니다. |
| GetCenter | GMapView 중심의 X, Y 좌표를 반환합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

|  |  |
| --- | --- |
| GetEditingGeometry | 편집되고 있는 Geometry 객체를 반환합니다. |
| GetHilightedObjectInfo | 하이라이트된 객체의 이름과 아이디를 반환합니다. |
| GetHilightedObjectInfoWith  LayerIndex | 하이라이트된 객체의 레이어 인덱스 번호와  아이디를 반환합니다. |
| MouseModeEnd | 마우스 모드의 종료를 선언합니다. |
| Move | GMapView의 크기를 설정하여 이동시킵니다. |
| PaperScaleToDisplayScale | DisplayScale을 PaperScale로 변환하여 반환합니다. |
| PrintMap | 지도의 출력을 설정합니다. |
| PrintMapEx | PrintEnable 속성이 True인 레이어들만을 영역  설정하여 해당 영역을 출력합니다. |
| Redraw | 지도영역 전체를 다시 그립니다. |
| Refresh | 화면을 Refresh 합니다. |
| SetCenter | 중심의 X, Y 좌표를 설정합니다. |
| SetDrawSymbol | GMapview 상에 객체를 그릴 때 객체의 펜, 브  러쉬, 아이콘 등의 속성값을 설정합니다. |
| SetLocation | GMapView에 지도를 그릴 때, 지도의 위치 및  축척을 설정합니다. |
| ShowNext | ShowPrevious 함수에 의해 이동한 위치를 한  번 되돌립니다. |
| ShowPrevious | 이전에 보이던 위치로 이동합니다. |
| StartEditObject | 객체 편집의 시작을 선언합니다. |
| ToMapDistance | 화면상의 거리를 지도상의 거리로 변환합니다. |
| ToMapPoint | 화면상의 X, Y 좌표를 지도상의 좌표로 변환하  여 IGPoint 객체 형태로 반환합니다. |
| ToMapXY | 화면상의 X, Y 좌표를 지도상의 X, Y 형태로 |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

|  |  |
| --- | --- |
|  | 변환하여 반환합니다. |
| ZoomArea | 설정한 범위로 화면을 확대합니다. |
| ZoomIn | 일정배율로 화면을 확대합니다. |
| ZoomOut | 일정배율로 화면을 축소합니다. |

Event

|  |  |
| --- | --- |
| Name | Description |
| AfterDraw | GMapView 상의 지도가 다시 그려질 경우, 지  도가 그려진 후에 발생하는 이벤트입니다. |
| AfterLayerDraw | 하나의 레이어를 그리고 나서 발생하는 이벤트  입니다. |
| BeforeDraw | GMapView에 지도를 그리기 전에 발생하는 이  벤트입니다. |
| BeforeFlush | AfterDraw 발생하기 이전에 발생하는 이벤트  입니다. |
| BeforeLayerDraw | 레이어를 그리기 전에 발생하는 이벤트입니다. |
| BeforeSelectedObjectDraw | 선택된 객체를 그리기 전에 발생하는 이벤트입니다. |
| ConnectionError | 연결 에러가 발생하였을 때 실행되는 이벤트입니다. |
| CustomToolCancel | 사용자 모드에서 Cancel 이벤트가 호출되었을  때 실행되는 이벤트입니다. |
| CustomToolCancelOne | 사용자 모드에서 CancelOne 이벤트가 호출되  었을 때 실행되는 이벤트입니다. |
| CustomToolLButtonDblClk | 사용자 모드에서 LButtonDblClk 이벤트가 호출  되었을 때 실행되는 이벤트입니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

|  |  |
| --- | --- |
| CustomToolLButtonDown | 사용자 모드에서 LButtonDown 이벤트가 호출  되었을 때 실행되는 이벤트입니다. |
| CustomToolLButtonUp | 사용자 모드에서 LButtonUp 이벤트가 호출되  었을 때 실행되는 이벤트입니다. |
| CustomToolMouseMove | 사용자 모드에서 MouseMove 이벤트가 호출  되었을 때 실행되는 이벤트입니다. |
| CustomToolRefresh | 사용자 모드에서 Refresh 이벤트가 호출되었을  때 실행되는 이벤트입니다. |
| DrawFinishedGeometry | 사용자가 GMapView에 Geometry 개체를 그리  고 난 후에 실행되는 이벤트입니다. |
| EditedGeometry | Geometry를 편집할 때 실행되는 이벤트입니다. |
| EmptyOperationClick | 마우스(왼쪽)를 클릭했을 때 실행되는 이벤트  입니다. |
| KeyDown | 키보드의 키가 눌렸을 때 실행되는 이벤트입니다. |
| LButtonDblCkl | 마우스 왼쪽버튼을 두 번 눌렀을 때 발생하는  이벤트입니다. |
| LButtonDown | 마우스 왼쪽버튼을 눌렀을 때 발생하는 이벤트  입니다. |
| LButtonUp | 마우스 왼쪽버튼을 누른 후 띄었을 때 발생하  는 이벤트입니다. |
| MouseMove | 마우스를 움직일 때마다 발생하는 이벤트입니다. |
| RButtonDblCkl | 마우스 오른쪽버튼을 두 번 눌렀을 때 발생하  는 이벤트입니다. |
| RButtonDown | 마우스 오른쪽버튼을 눌렀을 때 발생하는 이벤  트입니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

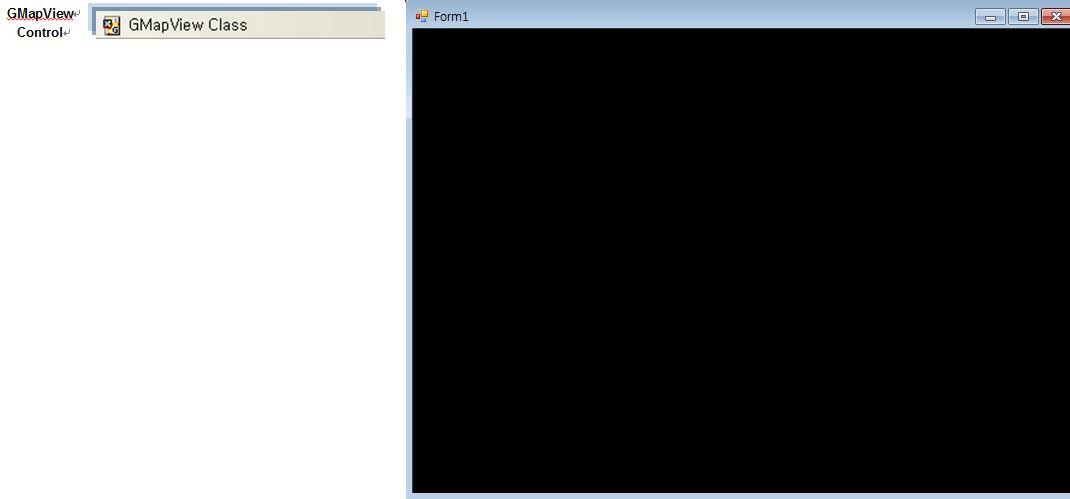
|  |  |
| --- | --- |
| RButtonUp | 마우스 오른쪽버튼을 누른 후 띄었을 때 발생  하는 이벤트입니다. |
| Resize | GMapView의 드로잉 영역의 크기가 변하였을  때 발생하는 이벤트입니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

GMapView에 새로운 레이어 추가

이번 예제는 실제적으로 지도데이터를 디스플레이 하는 예제로서 새로운 레 이어를 GMapView에 추가하는 과정을 보인다. 이에 필요한 레이어 정보는 GMapDataset으로부터 얻을 수 있다. 또한 예제를 수행하기 위해서는 우선적으 로 프로젝트에 GMapView 컨트롤을 추가하여야 한다. GMapView 컨트롤 추가는 앞서 구성요소에 추가한 GMapView 컨트롤을 더블 클릭하거나 또는 드래그하 여 간단히 수행할 수 있다.





 IntraMap/Objects ***PROGRAMMERS GUIDE***

[예제 4-1]

using IMapObjects3Lib;

namespace 1\_GMapview

{

public partial class Form1 : Form

{

public static GConnections GConns = new GConnections(); public static Random m\_objRnd = new Random(100);

public Form1()

{ InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

GConnection GConn = GConns.NewConnection("MSACCESS"); GMapDatasets mDatasets;

GMapDataset mDataset; GRecordLayer RLayer;

string sPath = Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles); GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb";

GConn.User = "";



 IntraMap/Objects ***PROGRAMMERS GUIDE***

GConn.Password = ""; GConn.Connect(); GConns.Add(GConn);

mDatasets = GConns.Item(0).GetMapDatasets(); for (int i = 0; i < mDatasets.Count; i++)

{

mDataset = mDatasets.GetMapDataset(i); RLayer = mDataset.CreateLayer();

GSimpleRenderer oRender = (GSimpleRenderer)RLayer.Renderer; GSymbol oSym = oRender.symbol;

doCreateRndSymbol(oSym, mDataset.Type); axGMapView1.Layers.Add((GLayer)RLayer);

}

axGMapView1.DirectionSymbolUse = true; axGMapView1.DirectionSymbolIndex = 3; axGMapView1.BackgroundColor = Color.Ivory.ToArgb(); axGMapView1.Fit();



 IntraMap/Objects ***PROGRAMMERS GUIDE***

}

private void Form1\_Resize(object sender, EventArgs e)

{

axGMapView1.Fit(); axGMapView1.Refresh();

}

public static void doCreateRndSymbol(GSymbol objSymbol, eGeometryTypes nGeometryType)

{

int nR1 = getRnd(20, 220); int nG1 = getRnd(20, 220); int nB1 = getRnd(20, 220);

if (nR1 == -1 || nG1 == -1 || nB1 == -1)

{

return;

}

objSymbol.Brush.Type = eBrushTypes.ftSolid; objSymbol.Brush.Color = getRGB(255, nR1 + 20, nG1 + 20, nB1 + 20);

objSymbol.Pen.Type = ePenTypes.ltSolid; objSymbol.Pen.Width = 1;

objSymbol.Pen.Color = getRGB(255, nR1 - 20, nG1 - 20, nB1 - 20);

 IntraMap/Objects ***PROGRAMMERS GUIDE***

if (nGeometryType == eGeometryTypes.gtPoint)

{

objSymbol.Icon.Type = eIconTypes.itCircle; objSymbol.Icon.Size = 3;

objSymbol.Icon.Color = getRGB(255, nR1, nG1, nB1);

}

}

private static int getRnd(int min, int max)

{

int nRnd = -1;

nRnd = m\_objRnd.Next(min, max); return nRnd;

}

public static int getRGB(int R, int G, int B)

{ return Color.FromArgb(150, R, G, B).ToArgb();

}

public static int getRGB(int A, int R, int G, int B)

{ return Color.FromArgb(A, R, G, B).ToArgb();

}

}

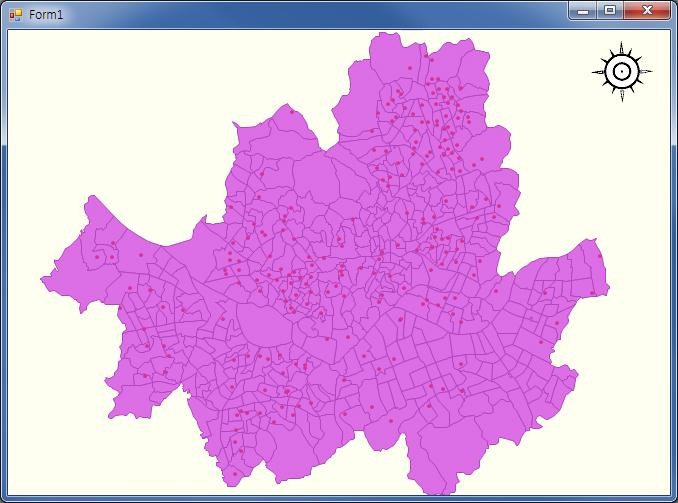
}

GConnection으로부터 가져온 GMapDatasets의 개수만큼 루프를 돌면서



 IntraMap/Objects ***PROGRAMMERS GUIDE*** GRecordLayer 변수인 RLayer에 해당 GMapDataset의 레이어 정보를 가지고 온다. 마지막으로 이 레이어들을 GMapView에 추가 함으로써 디스플레이 할 준 비가 완료된다.

GMapView의 DirectionSymbolUse 속성은 방위 심볼의 표현 여부를 설정하며, DirectionSymbolIndex 속성은 방위 심볼의 종류를 설정한다. 그리고 GMapView 의 BackgroundColor 속성은 GMapView의 배경색을 설정하는 속성값이다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***

좌표 변환

지도 데이터를 디스플레이 하는 화면 좌표와 실세계 객체의 좌표는 서로 다르 기 때문에 이들간의 좌표 변환이 필요하다. 따라서 IntraMap/Objects 3.0은 좌 표 변환을 위하여 다음과 같은 메소드들을 제공하고 있다.

|  |  |
| --- | --- |
| Name | Description |
| FromMapDistance | 지도상의 거리를 화면상의 거리로 변환합니다. |
| FromMapPoint | 포인트 객체의 지도좌표값을 화면좌표값으로 변환  합니다. |
| FromMapXY | X, Y 형태의 지도좌표를 화면좌표로 변환합니다. |
| ToMapDistance | 화면상의 거리를 지도상의 거리로 변환합니다. |
| ToMapPoint | 화면상의 X, Y 좌표를 지도상의 좌표로 변환하여  IGPoint 객체 형태로 반환합니다. |
| ToMapXY | 화면상의 X, Y 좌표를 지도상의 X, Y 형태로 변환하  여 반환합니다. |

다음은 GMapView의 LButtonDown 이벤트를 활용하여 간단하게 사용자가 클 릭하는 지점의 실좌표와 화면좌표를 얻어오는 예제이다.

우선 좌측 도구상자의 TextBox 컨트롤을 추가하고 GMapView를 클릭하여 속 성의 번개모양의 아이콘을 클릭한다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***

[예제 4-2]

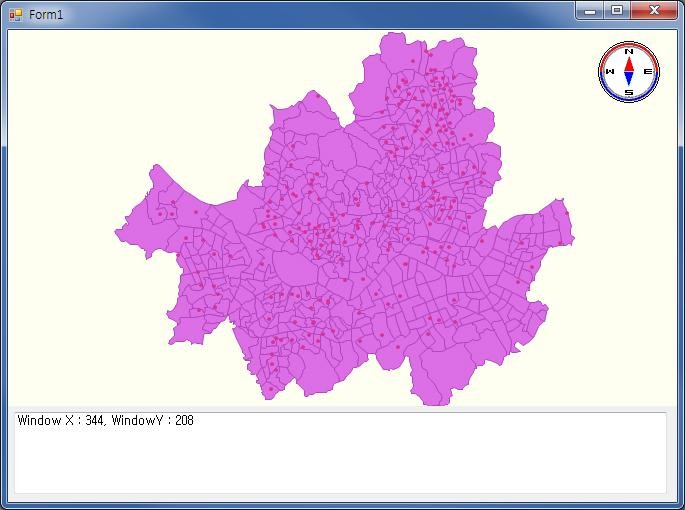
private void axGMapView1\_LButtonDown(object sender, AxIMapObjects3Lib.\_IGMapViewEvents\_LButtonDownEvent e)

{

double WindowX = e.windowX; double WindowY = e.windowY;

textBox1.Text = "Window X : " + WindowX + ", " + "WindowY : " + WindowY;

}





 IntraMap/Objects ***PROGRAMMERS GUIDE*** 다음은 좌표 변환 커맨드 버튼 클릭시에 ToMapXY 메소드를 활용하여 사용자 가 선택한 지점의 화면 좌표를 실좌표로 변환하는 예이다.

[예제 4-3]

private void button1\_Click\_1(object sender, EventArgs e)

{

axGMapView1.ToMapXY(WindowX, WindowY, out MapX, out MapY); textBox4.Text = "Map X : " + MapX;

textBox5.Text = "Map Y : " + MapY;

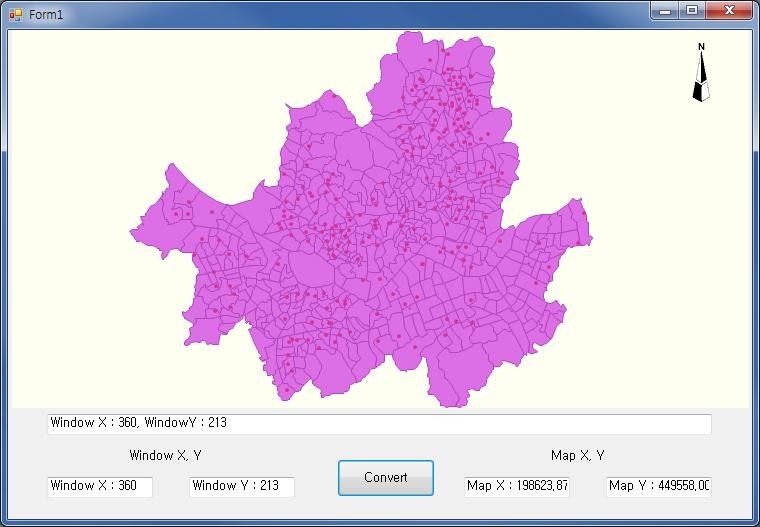
}

또한, 이와 반대로 실좌표를 화면 좌표로 변환하기 위해서는 FromMapXY 메소 드를 활용하면 된다.

GMapView1.FromMapXY(MapX, Map, out winX, out winY);



 IntraMap/Objects ***PROGRAMMERS GUIDE***

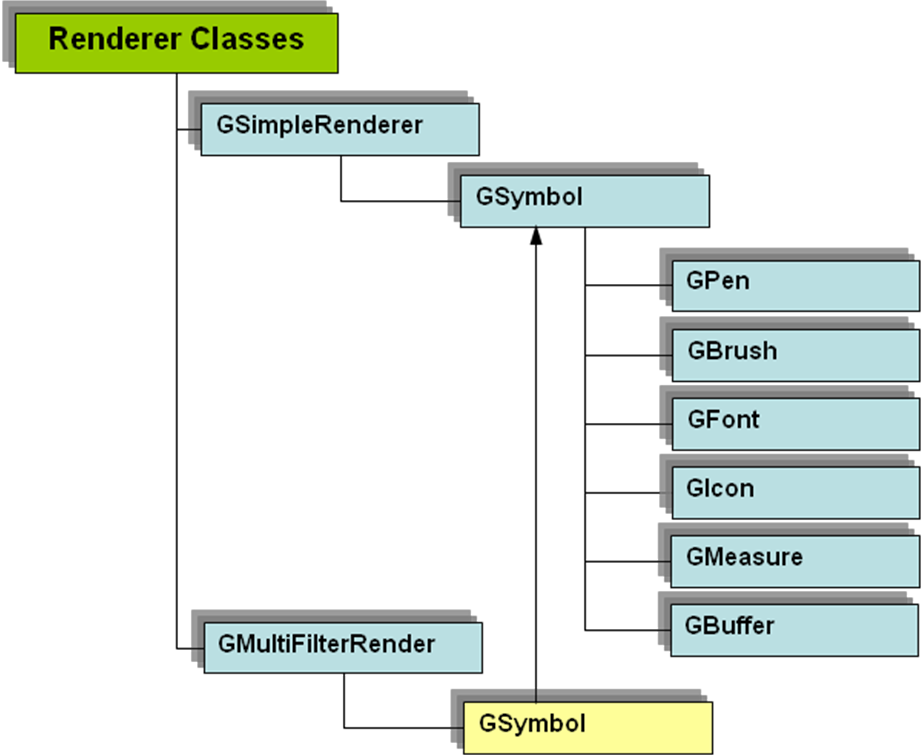




 IntraMap/Objects ***PROGRAMMERS GUIDE***

# 화면 상의 레이어별 색상관리

IntraMap/Objects 3.0은 개발자 및 사용자가 임으로 레이어의 속성을 설정할 수 있는 기능을 제공하고 있다. 다음은 이러한 기능을 가능하게 하는 Graphic Class의 계층 구조를 나타낸다.





 IntraMap/Objects ***PROGRAMMERS GUIDE***

1. 렌더러

렌더러 클래스는 사용자가 임의로 레이어의 속성을 설정할 수 있도록 지원하 며, GMapView 상 레이어의 심볼 속성 변경도 지원한다. 렌더러 클래스를 이용 하여 하나의 레이어에 대한 세부 조건 별 심볼을 지정할 수 있으며, 레이어의 형태에 따라 다양한 형태의 지도 디스플레이가 가능하다.

렌더러 클래스는 GSimpleRenderer와 GMultiFilterRender로 이루어진다.

* GSimpleRenderer

Property

|  |  |
| --- | --- |
| Name | Description |
| Symbol | SimpleRenderer의 심볼을 설정합니다. |
| Type | 렌더러의 타입을 반환합니다. |

* GMultiFilterRender

Property

|  |  |
| --- | --- |
| Name | Description |
| Symbols | MultiFilterRenderer의 심볼을 설정합니다. |
| Filters | Index의 Filter 조건을 설정하거나 반환합니다. |
| Title | Index의 Title을 설정하거나 반환합니다. |
| Type | 렌더러의 타입을 반환합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

Method

|  |  |
| --- | --- |
| Name | Description |
| SetMaxCount | 렌더러 필터의 최대 개수를 설정합니다. |

1. 레이어 심볼 정의

GMapView에 레이어를 그리는 렌더러의 형태는 심볼을 정의함으로써 이루어 진다.

GSymbol Object = GRenderer Object.Symbol;

GSymbol Object는 다양한 프라퍼티와 메소드 그리고 상수를 제공함으로써 사용자가 정의하는 심볼로 레이어를 그릴 수 있는 기능을 제공한다.

* GSymbol

Property

|  |  |
| --- | --- |
| Name | Description |
| Brush | 브러쉬를 설정하거나 반환합니다. |
| Buffer | 버퍼링 관련 사항을 설정합니다. |
| FloorField | 2.5차원으로 표출되는 데이터의 높이 정보를 담고  있는 필드를 반환합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

|  |  |
| --- | --- |
| Font | 폰트를 설정하거나 반환합니다. |
| Icon | 아이콘를 설정하거나 반환합니다. |
| LabelField | 레이블로 표시되는 필드를 설정하거나 반환합니다. |
| LabelMode | 레이블 형태를 설정하거나 반환합니다. |
| LabelMustOverlap | 레이블의 중첩허용 여부를 설정하거나 반환합니다. |
| MaxBuildingShowScale | 빌딩이 표시되는 최대 축척을 설정하거나 반환합니다. |
| MaxLabelShowScale | 레이블이 표시되는 최대 축척을 설정하거나 반환합니다. |
| MaxShowScale | 심볼이 표시되는 최대 축척을 설정하거나 반환합니다. |
| Measure | 각도, 거리 측정값의 표시 형태를 설정하거나 반환합니다. |
| MinBuildingShowScale | 빌딩이 표시되는 최소 축척을 설정하거나 반환합니다. |
| MinLabelShowScale | 레이블이 표시되는 최소 축척을 설정하거나 반환합니다. |
| MinShowScale | 심볼이 표시되는 최소 축척을 설정하거나 반환합니다. |
| Pen | 펜을 설정하거나 반환합니다. |
| Transparent | 심볼의 투명여부를 설정하거나 반환합니다. |

Method

|  |  |
| --- | --- |
| Name | Description |
| CopyValue | Symbol을 복사합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* GPen

Property

|  |  |
| --- | --- |
| Name | Description |
| ByDisplayScale | 선의 두께를 거리단위로 표출할지 설정하거나 반환합니다. |
| CapSize | Cap의 크기를 설정하거나 반환합니다. |
| Color | 선의 색상을 설정하거나 반환합니다. |
| EndCap | Cap의 종료 형태를 설정하거나 반환합니다. |
| StartCap | Cap의 시작 형태를 설정하거나 반환합니다. |
| Type | 선의 형태를 설정하거나 반환합니다. |
| TypeEx | 확장된 선의 형태를 설정하거나 반환합니다. |
| Width | 선의 넓이 값을 설정하거나 반환합니다. |

* GBrush

Property

|  |  |
| --- | --- |
| Name | Description |
| BackColor | 브러쉬의 배경색을 설정하거나 반환합니다. |
| Color | 브러쉬의 색을 설정하거나 반환합니다. |
| Hatch | 해치 형태를 설정하거나 반환합니다. |
| HatchTransparent | 해치의 투명도를 설정하거나 반환합니다. |
| Type | 브러쉬의 채우기 형태를 설정하거나 반환합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* GFont

Property

|  |  |
| --- | --- |
| Name | Description |
| Align | 텍스트의 정렬 여부를 설정하거나 반환합니다. |
| BackColor | 텍스트의 배경색을 설정하거나 반환합니다. |
| Bold | 텍스트의 볼드체 여부를 설정하거나 반환합니다. |
| ByDisplayScale | 텍스트의 크기를 거리단위로 표출할지 설정하거나 반  환합니다. |
| ByGeometryAngle | 텍스트의 각도를 지오메트리 객체에 의해 조정할지  설정하거나 반환합니다. |
| Height | 텍스트의 높이 값을 설정하거나 반환합니다. |
| Hilight | 하이라이트 여부를 설정하거나 반환합니다. |
| Italic | 이탤릭체 여부를 설정하거나 반환합니다. |
| Name | 폰트명을 설정하거나 반환합니다. |
| Strikethrough | 텍스트의 Strikethrough 속성을 설정하거나 반환합니다. |
| TextColor | 텍스트의 색을 설정하거나 반환합니다. |
| Transparent | 텍스트의 투명도를 설정하거나 반환합니다. |
| Underline | 텍스트의 밑줄 사용여부를 설정하거나 반환합니다. |
| Weight | 텍스트의 굵기 값을 설정하거나 반환합니다. |
| Width | 텍스트의 넓이 값을 설정하거나 반환합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* GIcon

Property

|  |  |
| --- | --- |
| Name | Description |
| angle | 아이콘의 각도를 설정하거나 반환합니다. |
| ByDisplayScale | 아이콘의 크기를 거리단위로 표출할지 설정하거나 반  환합니다. |
| Code | 아이콘의 코드를 설정하거나 반환합니다. |
| Color | 아이콘의 색을 설정하거나 반환합니다. |
| Name | 아이콘명을 설정하거나 반환합니다. |
| Size | 아이콘의 크기를 설정하거나 반환합니다. |
| Type | 아이콘의 타입를 설정하거나 반환합니다. |

* GBuffer

Property

|  |  |
| --- | --- |
| Name | Description |
| ByDisplayScale | 아이콘의 크기를 거리단위로 표출할지 설정하거나 반  환합니다. |
| distance | 버퍼링 거리를 설정하거나 반환합니다. |
| FillColor | 버퍼링 영역의 채우기 색을 설정하거나 반환합니다. |
| LineColor | 버퍼링 영역의 경계선 색을 설정하거나 반환합니다. |
| Type | 버퍼링의 타입를 설정하거나 반환합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* GMeasure

Property

|  |  |
| --- | --- |
| Name | Description |
| AngleMeasureStyle | 각도 표현 타입를 설정하거나 반환합니다. |
| DistanceMeasureStyle | 거리 표현 타입를 설정하거나 반환합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

레이어 심볼 적용

다음 예제는 앞에서 사용한 db.mdb에 들어있는 6개의 레이어들 중 ADMIN\_BOUNDARY와 LINE\_ROAD 레이어의 심볼을 설정하고, 그 심볼로 GMapView에 사용자가 원하는 라인 타입과 색상으로 디스플레이 하는 예제이 다.

[예제 5-1]

using IMapObjects3Lib; namespace 1\_SimpleRenderer

{

public partial class Form1 : Form

{

public static GConnections GConns = new GConnections();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

GConnection GConn = GConns.NewConnection("MSACCESS"); GMapDatasets mDatasets;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

GMapDataset mDataset; GRecordLayer RLayer; GSimpleRenderer SRender; GSymbol Sym;

string sPath = Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles);

GConn.Database =

sPath + "\\KSIC\\IntraMapObjects3\\db.mdb";

GConn.Connect(); GConns.Add(GConn);

mDatasets = GConn.GetMapDatasets();

mDataset = mDatasets.GetMapDataset("ADMIN\_BOUNDARY"); RLayer = mDataset.CreateLayer();

RLayer.Visible = true;

SRender = (GSimpleRenderer)RLayer.Renderer; Sym = SRender.symbol;

Sym.Brush.Type = eBrushTypes.ftNull; Sym.Pen.Type = ePenTypes.ltDash; Sym.Pen.Color = getRGB(0, 0, 255);

Sym.Pen.Width = 1; axGMapView1.Layers.Add((GLayer)RLayer);



 IntraMap/Objects ***PROGRAMMERS GUIDE***

mDataset = mDatasets.GetMapDataset("LINE\_ROAD"); RLayer = mDataset.CreateLayer();

RLayer.Visible = true;

SRender = (GSimpleRenderer)RLayer.Renderer; Sym = SRender.symbol;

Sym.Brush.Type = eBrushTypes.ftNull; Sym.Brush.Color = getRGB(255, 255, 0); Sym.Pen.Type = ePenTypes.ltSolid; Sym.Pen.Color = getRGB(255, 0, 0);

Sym.Pen.Width = 2; axGMapView1.Layers.Add((GLayer)RLayer);

axGMapView1.ScaleBarUse = true; axGMapView1.BackgroundColor = Color.Ivory.ToArgb();

axGMapView1.Fit();

}

private void Form1\_Resize(object sender, EventArgs e)

{ axGMapView1.Fit(); axGMapView1.Refresh();

}

public static int getRGB(int R, int G, int B)

{ return Color.FromArgb(150, R, G, B).ToArgb();



 IntraMap/Objects ***PROGRAMMERS GUIDE***

}

public static int getRGB(int A, int R, int G, int B)

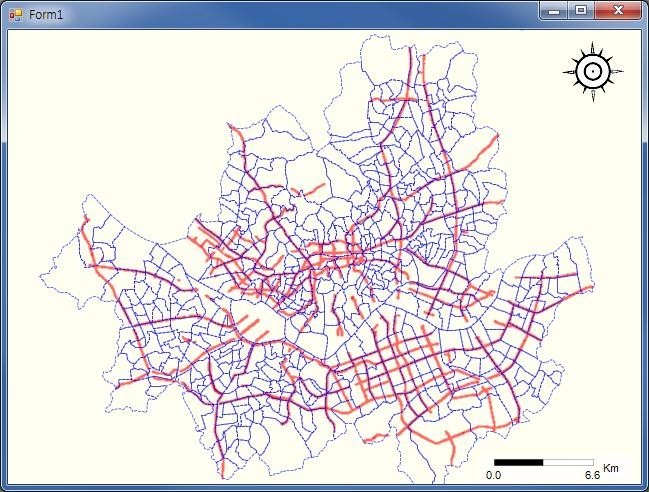
{ return Color.FromArgb(A, R, G, B).ToArgb();

}

}

}

위의 예제에서 LINE\_ROAD 레이어에 해당하는 도로 내부와 외곽선에 대해서 는 빨간색 실선으로 표현하고, ADMIN\_BOUNDARY 구경계 레이어에 대해서는 내부가 비어있는 파란색 점선으로 외곽선을 표현함으로써 레이어의 구분을 더 욱 명확하게 나타내었다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***

멀티 필터 렌더러의 활용

다음 예제는 앞에서 사용한 db.mdb에 들어있는 6개의 레이어들 중 NEW\_SEOUL\_BOUNDARY와 LINE\_ROAD 레이어의 심볼을 특정 필드의 조건에 따라 다르게 표출되도록 설정하고, 설정된 심볼로 GMapView에 사용자가 원하 는 라인 타입과 색상으로 디스플레이 하는 예제이다.

멀티 필터 렌더러는 하나의 렌더러에 여러 개의 필터별로 심볼을 설정할 수 있도록 해주며, MDB 파일이나 IntraMap/Server에 들어있는 GRecordLayer에서만 사용가능하다. 멀터 필터 렌더러의 활용방법은 아래와 같다.

GMultiFilterRenderer MRender = New GmultiFilterRenderer();

RLayer.Renderer = (GRenderer)MRender; MRender.SetMaxCount(Number of filter);

MRender.set\_Filters(0, Condition1); MRender.get\_Symbols(0).Pen.Color = getRGB(100, 0, 0 );

MRender.set\_Filters(1, Condition2); MRender.get\_Symbols(1).Brush.Color = getRGB(90, 0, 0);

[예제 5-2]

using IMapObjects3Lib;

namespace 2\_MultiFilterRenderer



 IntraMap/Objects ***PROGRAMMERS GUIDE***

{

public partial class Form1 : Form

{

public static GConnections GConns = new GConnections();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

GConnection GConn = GConns.NewConnection("MSACCESS"); GMapDatasets mDatasets;

GMapDataset mDataset; GRecordLayer RLayer;

GMultiFilterRenderer MRender1 = new GMultiFilterRenderer(); GMultiFilterRenderer MRender2 = new GMultiFilterRenderer();

GConn.User = ""; GConn.Password = ""; string sPath =

Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles); GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb";

GConn.Connect();



IntraMap/Objects ***PROGRAMMERS GUIDE***

GConns.Add(GConn);

mDatasets = GConns.Item(0).GetMapDatasets();

mDataset = mDatasets.GetMapDataset("NEW\_SEOUL\_BOUNDARY"); RLayer = mDataset.CreateLayer();

RLayer.Renderer = (GRenderer)MRender1; MRender1.SetMaxCount(3);

MRender1.set\_Filters(0, "GU\_NAME = '도봉구'"); MRender1.get\_Symbols(0).Pen.Type = ePenTypes.ltSolid; MRender1.get\_Symbols(0).Pen.Color = getRGB(255, 0, 0);

MRender1.get\_Symbols(0).Pen.Width = 2; MRender1.get\_Symbols(0).Brush.Type = eBrushTypes.ftHatch; MRender1.get\_Symbols(0).Brush.Hatch = eHatchTypes.htBDiagonal; MRender1.get\_Symbols(0).Brush.Color = getRGB(50,0, 255, 0);

// 알파값 변경 가능 [4/8/2011] MRender1.get\_Symbols(0).LabelMode = eLabelModes.lmHas; MRender1.get\_Symbols(0).LabelField = "GU\_NAME";

MRender1.set\_Filters(1, "GU\_NAME = '용산구'"); MRender1.get\_Symbols(1).Pen.Type = ePenTypes.ltSolid; MRender1.get\_Symbols(1).Pen.Color = getRGB(0, 255, 0);

MRender1.get\_Symbols(1).Pen.Width = 2; MRender1.get\_Symbols(1).Brush.Type = eBrushTypes.ftHatch; MRender1.get\_Symbols(1).Brush.Hatch = eHatchTypes.htBMCircle;



IntraMap/Objects ***PROGRAMMERS GUIDE*** MRender1.get\_Symbols(1).Brush.Color = getRGB(50,0, 0, 255); MRender1.get\_Symbols(1).LabelMode = eLabelModes.lmHas; MRender1.get\_Symbols(1).LabelField = "GU\_NAME";

MRender1.set\_Filters(2, "GU\_NAME = '서초구'"); MRender1.get\_Symbols(2).Pen.Type = ePenTypes.ltSolid; MRender1.get\_Symbols(2).Pen.Color = getRGB(0, 0, 255);

MRender1.get\_Symbols(2).Pen.Width = 2; MRender1.get\_Symbols(2).Brush.Type = eBrushTypes.ftHatch; MRender1.get\_Symbols(2).Brush.Hatch = eHatchTypes.htBDiagonal; MRender1.get\_Symbols(2).Brush.Color = getRGB(0, 255, 0); MRender1.get\_Symbols(2).LabelMode = eLabelModes.lmHas; MRender1.get\_Symbols(2).LabelField = "GU\_NAME";

axGMapView1.Layers.Add((GLayer)RLayer);

mDataset = GConns.Item(0).GetMapDatasets().GetMapDataset("LINE\_ROAD"); RLayer = mDataset.CreateLayer();

RLayer.Renderer = (GRenderer)MRender2; MRender2.SetMaxCount(3);

MRender2.set\_Filters(0, "ACRES < 5"); MRender2.get\_Symbols(0).Pen.Type = ePenTypes.ltSolid; MRender2.get\_Symbols(0).Pen.Color = getRGB(255, 0, 0);



IntraMap/Objects ***PROGRAMMERS GUIDE*** MRender2.get\_Symbols(0).Pen.Width = 2; MRender2.get\_Symbols(0).Brush.Type = eBrushTypes.ftHatch; MRender2.get\_Symbols(0).Brush.Color = getRGB(255, 255, 0);

MRender2.set\_Filters(1, "5 <= ACRES and ACRES < 10"); MRender2.get\_Symbols(1).Pen.Type = ePenTypes.ltSolid; MRender2.get\_Symbols(1).Pen.Color = getRGB(0, 255, 0);

MRender2.get\_Symbols(1).Pen.Width = 2; MRender2.get\_Symbols(1).Brush.Type = eBrushTypes.ftHatch; MRender2.get\_Symbols(1).Brush.Color = getRGB(255, 255, 0);

MRender2.set\_Filters(2, "10 <= ACRES"); MRender2.get\_Symbols(2).Pen.Type = ePenTypes.ltSolid; MRender2.get\_Symbols(2).Pen.Color = getRGB(0, 0, 255);

MRender2.get\_Symbols(2).Pen.Width = 2; MRender2.get\_Symbols(2).Brush.Type = eBrushTypes.ftHatch; MRender2.get\_Symbols(2).Brush.Color = getRGB(50,255, 255, 0);

axGMapView1.Layers.Add((GLayer)RLayer);

axGMapView1.DirectionSymbolUse = true; axGMapView1.BackgroundColor = Color.Ivory.ToArgb(); axGMapView1.Fit();

 IntraMap/Objects ***PROGRAMMERS GUIDE***

}

private void Form1\_Resize(object sender, EventArgs e)

{

axGMapView1.Fit(); axGMapView1.Refresh();

}

public static int getRGB(int R, int G, int B)

{

return Color.FromArgb(150, R, G, B).ToArgb();

}

public static int getRGB(int A, int R, int G, int B)

{

return Color.FromArgb(A, R, G, B).ToArgb();

}

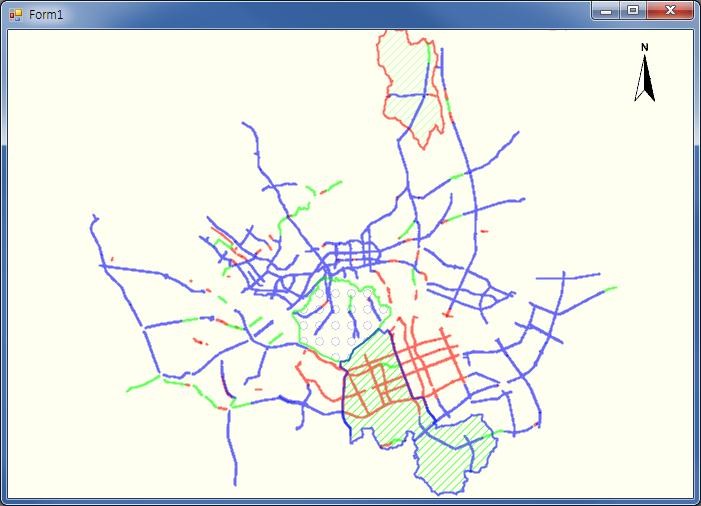
}

}

위의 예제에서 NEW\_SEOUL\_BOUNDARY 구경계 레이어는 구의 이름에 따라 심볼의 색상 및 형태가 변화되어 표현되었으며, LINE\_ROAD 레이어에 대해서는 테이블 구성 필드 중 acres 필드값의 범위에 따라 라인의 색상 및 형태가 변화 되어 표현되었다.



 IntraMap/Objects ***PROGRAMMERS GUIDE*** GMapView의 Fit 메소드는 실제 지도 데이터의 영역과 GMapView가 보이고 있는 영역을 동일하게 함으로써 전체 보기의 기능을 제공한다.





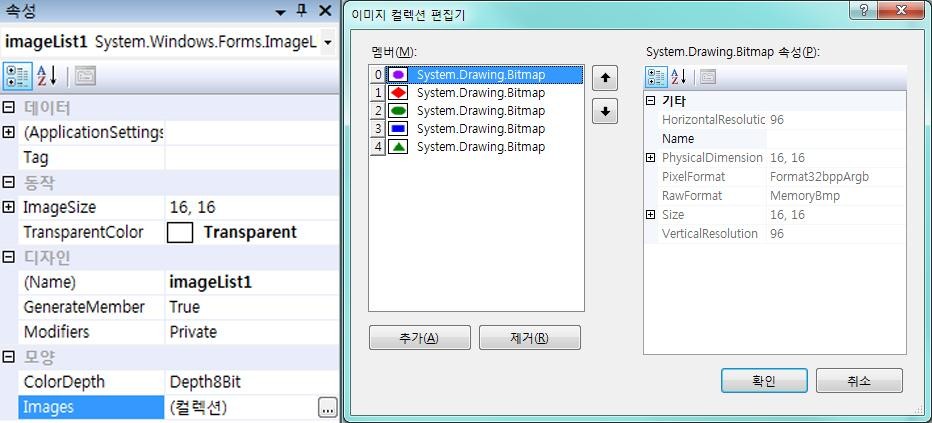
 IntraMap/Objects ***PROGRAMMERS GUIDE***

이미지 아이콘의 활용

다음 예제는 db.mdb에 들어있는 6개의 레이어들 중 APART 레이어의 레코드 중에서 특정 이름을 가지는 레코드들만을 선택하여 레코드의 심볼을 이미지 아 이콘으로 설정하는 예제이다. 레코드의 심볼을 이미지 아이콘으로 설정하기 위 하여 이미지 리스트를 생성하고 이를 GMapView의 이미지 리스트와 연결하는 작업이 필요하다.

GMapView의 이미지 리스트와 이미지 리스트 콘트롤을 연결하는 방법은 아래 와 같으며, 다음 그림은 이미지 리스트에 이미지를 추가한 모습을 보여주는 화 면이다.

axGMapView1.IconImageList1 = (int)imageList1.Handle;





 IntraMap/Objects ***PROGRAMMERS GUIDE***

[예제 5-3]

using IMapObjects3Lib; namespace 3\_ImageList

{

public partial class Form1 : Form

{

public static GConnections GConns = new GConnections();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

axGMapView1.IconImageList1 = (int)imageList1.Handle;

GConnection GConn = GConns.NewConnection("MSACCESS"); GConn.User = "";

GConn.Password = "";

string sPath = Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles);



 IntraMap/Objects ***PROGRAMMERS GUIDE***

GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb";

GConn.Connect(); GConns.Add(GConn);

GRecordLayer RLayer = GConns.Item(0).GetMapDatasets().GetMapDataset("APART").CreateLayer();

GMultiFilterRenderer RENDER = new GMultiFilterRenderer(); RLayer.Renderer = (GRenderer)RENDER; RENDER.SetMaxCount(5);

{

RENDER.set\_Filters(0, "BLDG\_NAME like '한양\*'"); GSymbol SYM = new GSymbol();

SYM = RENDER.get\_Symbols(0); SYM.Icon.Type = eIconTypes.itImage; SYM.Icon.Code = 0;

}

{

RENDER.set\_Filters(1, "BLDG\_NAME like '현대\*'"); GSymbol SYM = new GSymbol();

SYM = RENDER.get\_Symbols(1); SYM.Icon.Type = eIconTypes.itImage;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

SYM.Icon.Code = 1;

}

{

RENDER.set\_Filters(2, "BLDG\_NAME like '한신\*'"); GSymbol SYM = new GSymbol();

SYM = RENDER.get\_Symbols(2); SYM.Icon.Type = eIconTypes.itImage; SYM.Icon.Code = 2;

}

{

RENDER.set\_Filters(3, "BLDG\_NAME like '개포\*'"); GSymbol SYM = new GSymbol();

SYM = RENDER.get\_Symbols(3); SYM.Icon.Type = eIconTypes.itImage; SYM.Icon.Code = 3;

}

{

RENDER.set\_Filters(4, "BLDG\_NAME like '경남\*'"); GSymbol SYM = new GSymbol();

SYM = RENDER.get\_Symbols(4);



 IntraMap/Objects ***PROGRAMMERS GUIDE***

SYM.Icon.Type = eIconTypes.itImage; SYM.Icon.Code = 4;

}

RLayer.Visible = true; axGMapView1.Layers.Add((GLayer)RLayer);

axGMapView1.BackgroundColor = Color.WhiteSmoke.ToArgb(); axGMapView1.Fit();

}

private void Form1\_Resize(object sender, EventArgs e)

{

axGMapView1.Fit(); axGMapView1.Refresh();

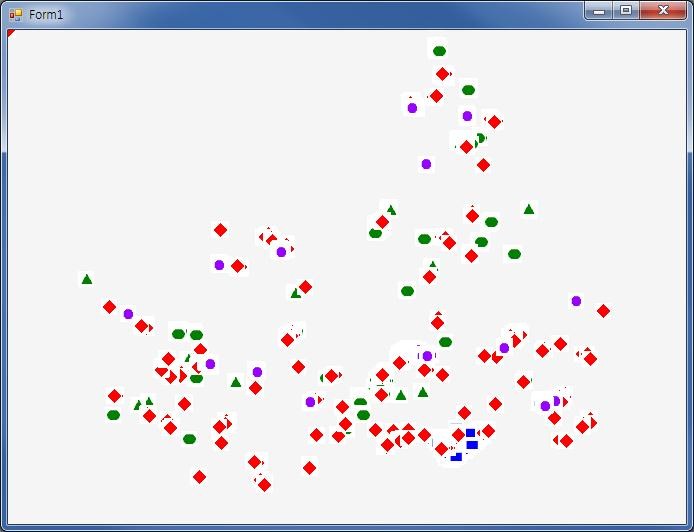
}

}

}



 IntraMap/Objects ***PROGRAMMERS GUIDE*** 위의 예제에서 APART 레이어의 모든 레코드 중 6개의 필터 조건에 해당하 는 레코드들만이 GMapView에 표출되었으며, 각각의 레코드들은 해당 필터의 아이콘 코드에 해당하는 이미지로 표현되었다. 이러한 이미지 아이콘의 사용 은 시각적인 효과를 증가시킬 수 있을뿐 아니라 다른 속성을 가지는 객체간 구분을 용이하게 해준다.





 IntraMap/Objects ***PROGRAMMERS GUIDE***

레이블의 설정 및 디스플레이 범위 설정

다음 예제는 앞에서 사용한 db.mdb에 들어있는 6개의 레이어들 중 LINE 레이 어의 레이블을 원하는 형태로 설정하여 GMapView에 디스플레이 하는 예제이 다. 디스플레이된 LINE 레이어는 사용자가 각각의 버튼을 클릭함에 따라 다른 형태의 레이블로 GMapView 상에 디스플레이 될 것이며, 또한, 사용자가 레이블 의 최대/최소 디스플레이 범위를 설정을 하게되면 이에 따라 GMapView 상에서 LINE 레이어가 디스플레이 유무가 변하는 과정을 다음 예제를 통하여 살펴볼 수 있다.

[예제 5-4]

using IMapObjects3Lib; namespace 4\_Label\_Scale

{

public partial class Form1 : Form

{

public static GConnections GConns = new GConnections(); public GSymbol Sym = new GSymbol();

public Form1()

{

InitializeComponent();



 IntraMap/Objects ***PROGRAMMERS GUIDE***

}

private void Form1\_Load(object sender, EventArgs e)

{

GConnection GConn = GConns.NewConnection("MSACCESS"); GMapDatasets mDatasets;

GMapDataset mDataset; GRecordLayer RLayer; GSimpleRenderer SRender;

string sPath = Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles); GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb";

GConn.Connect(); GConns.Add(GConn);

mDatasets = GConn.GetMapDatasets(); mDataset = mDatasets.GetMapDataset("LINE "); RLayer = mDataset.CreateLayer();

RLayer.Visible = true;

SRender = (GSimpleRenderer)RLayer.Renderer; Sym = SRender.symbol;

Sym.Pen.Type = ePenTypes.ltDash; Sym.Pen.Color = getRGB(0, 255, 0);



 IntraMap/Objects ***PROGRAMMERS GUIDE***

Sym.Pen.Width = 1;

Sym.Font.Name = "Arial"; Sym.Font.Height = 11.0; Sym.Font.TextColor = Color.Black.ToArgb();

axGMapView1.Layers.Add((GLayer)RLayer);

axGMapView1.DirectionSymbolUse = true; axGMapView1.DirectionSymbolIndex = 4; axGMapView1.ScaleBarUse = true; axGMapView1.BackgroundColor = Color.Ivory.ToArgb(); axGMapView1.Fit();

txtScale.Text = axGMapView1.PaperScale.ToString();

}

private void axGMapView1\_AfterDraw(object sender, AxIMapObjects3Lib.\_IGMapViewEvents\_AfterDrawEvent e)

{ txtScale.Text = axGMapView1.PaperScale.ToString();

}

private void Form1\_Resize(object sender, EventArgs e)

{

axGMapView1.Fit(); axGMapView1.Refresh();



 IntraMap/Objects ***PROGRAMMERS GUIDE***

}

private void cmdScale\_Click(object sender, EventArgs e)

{

try

{

Sym.MaxShowScale = Convert.ToDouble(txtMax.Text); Sym.MinShowScale = Convert.ToDouble(txtMin.Text); axGMapView1.Redraw();

}

catch(Exception err)

{ MessageBox.Show(err.Message);

}

}

private void cmdHas\_Click(object sender, EventArgs e)

{

Sym.LabelMode = eLabelModes.lmHas; Sym.LabelField = "LINK\_ID";

axGMapView1.Redraw();

}

private void cmdOnly\_Click(object sender, EventArgs e)

{



 IntraMap/Objects ***PROGRAMMERS GUIDE***

Sym.LabelMode = eLabelModes.lmOnly; Sym.LabelField = "LINK\_ID";

axGMapView1.Redraw();

}

private void cmdMirror\_Click(object sender, EventArgs e)

{

Sym.LabelMode = eLabelModes.lmMirror; Sym.LabelField = "LINK\_ID";

axGMapView1.Redraw();

}

private void cmd2Side\_Click(object sender, EventArgs e)

{

Sym.LabelMode = eLabelModes.lm2Side; Sym.LabelField = "([LINK\_ID],[LINK\_ID])";

axGMapView1.Redraw();

}

private void cmdBridge\_Click(object sender, EventArgs e)

{

Sym.LabelMode = eLabelModes.lmBridge;

Sym.LabelField = "([LINK\_ID]-[LINK\_ID])([LINK\_ID]-[LINK\_ID])";



 IntraMap/Objects ***PROGRAMMERS GUIDE***

axGMapView1.Redraw();

}

public static int getRGB(int R, int G, int B)

{

return Color.FromArgb(150, R, G, B).ToArgb();

}

public static int getRGB(int A, int R, int G, int B)

{

return Color.FromArgb(A, R, G, B).ToArgb();

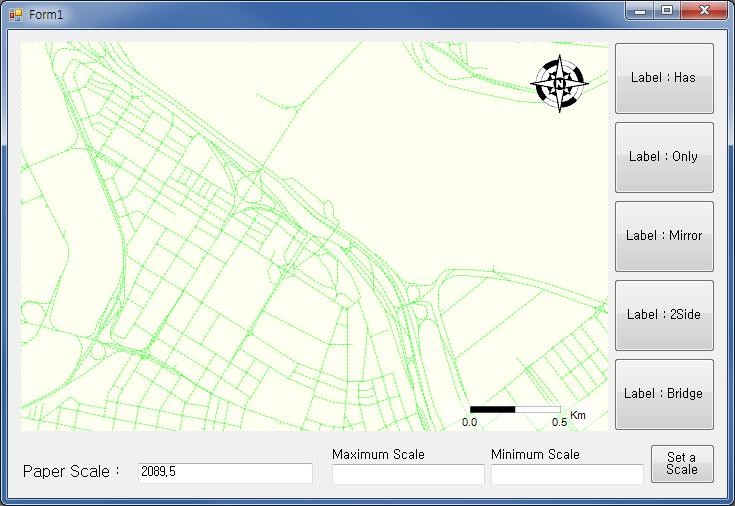
}

}

}

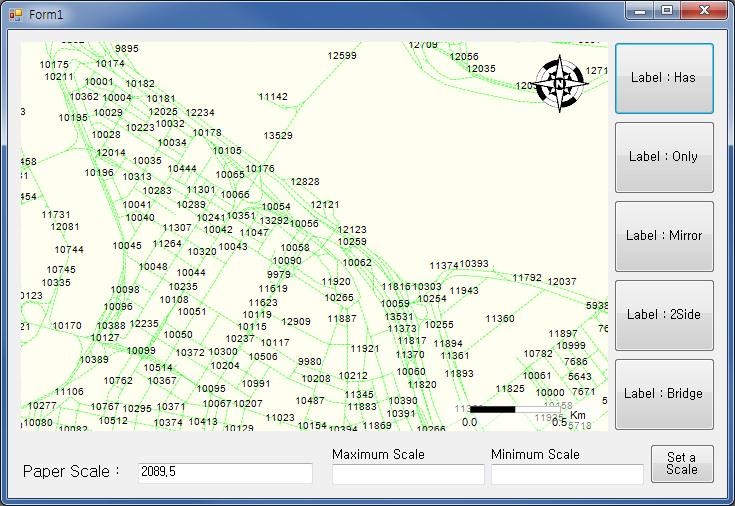
위의 예제와 같이 LINE 레이어의 레이블을 원하는 형태로 설정하여 아래 그림과 같이 GMapView에 디스플레이 할 수 있으며, LINE 레이어 각각의 버 튼을 클릭하여 다른형태의 레이블로 디스플레이되는 것을 확인할 수 있다.





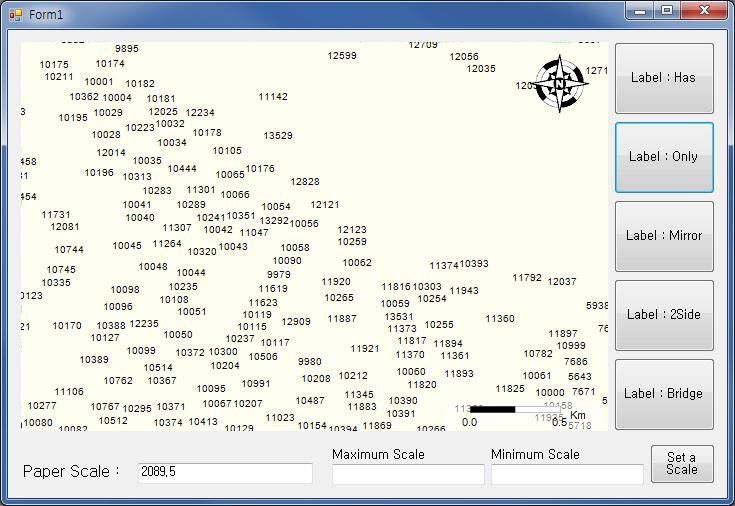
IntraMap/Objects

***PROGRAMMERS GUIDE***

기본 화면

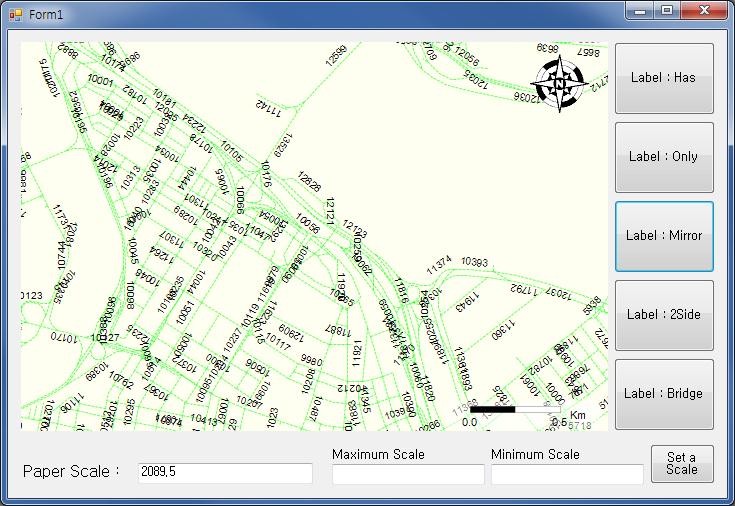
Label Mode : Has





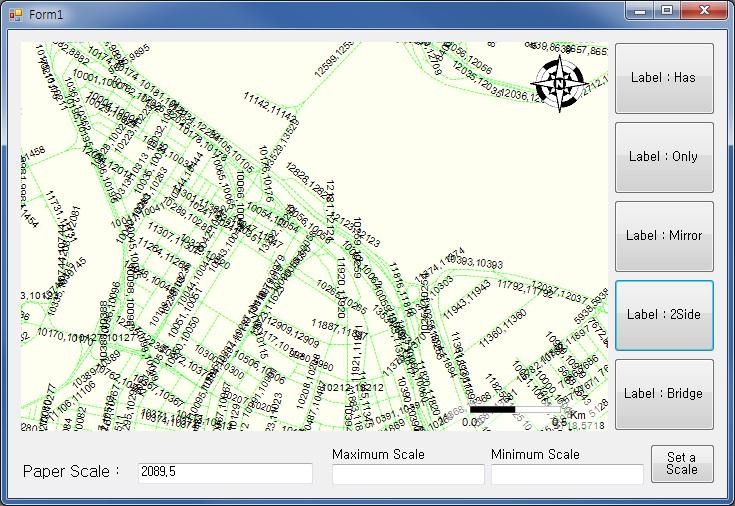
IntraMap/Objects

***PROGRAMMERS GUIDE***

Label Mode : Only

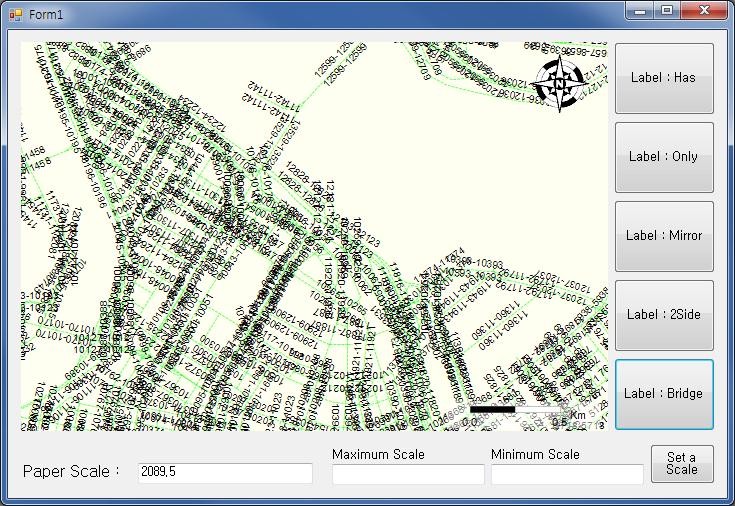
Label Mode : Mirror





IntraMap/Objects

***PROGRAMMERS GUIDE***

Label Mode : 2Side

Label Mode : Bridge



 IntraMap/Objects ***PROGRAMMERS GUIDE***

# 화면 제어

GMapView 상에서 디스플레이된 지도데이터를 조작하기 위해서 IntraMap/Objects 3.0은 다양한 화면제어 기능과 마우스 모드를 제공한다. 이번 장에서는 가장 기본적인 화면 제어 기능인 확대, 축소, 전체보기 등과 마우스 모드에 대하여 알아본다.

1. 기본 기능

화면에 지도데이터가 디스플레이되면 사용자는 다양한 공간 분석의 목적으로 지도데이터를 조작하게 된다. 이때 가장 기본적인 기능이 지도데이터의 확대, 축소, 이동, 전체보기 등이며, 이러한 기능은 GMapView가 제공하는 Method를 통하여 간단히 수행될 수 있다.

// 지도데이터 확대

axGMapView.ZoomIn();

//지도데이터 축소

axGMapView.ZoonOut();

//지도데이터 전체 보기

axGMapView.Fit();



 IntraMap/Objects ***PROGRAMMERS GUIDE***

1. 마우스 모드

프로그램 실행시 기본적으로 윈도우가 제공하는 마우스 모드외에 IntraMap/Objects 3.0은 GMapView 상에서 나타낼 수 있는 다양한 마우스 모드 를 제공한다. 다음은 IntraMap/Objects 3.0 Contants 중 eMouseMode의 기능을 나타낸다.

* eMouseMode Constants

|  |  |
| --- | --- |
| Type | Description |
| mmNormal | 기본 마우스 모드로 설정합니다. |
| mmCalcArea | 마우스 모드를 ‘영역재기’ 모드 형태로 설정합니다. |
| mmCalcAreaEx | 마우스 모드를 ‘영역재기’ 모드 형태로 설정하 며, 마우스 모드 종료 후에 DrawFinishedGeom  etry 이벤트가 호출됩니다. |
| mmCalcDistance | 마우스 모드를 ‘거리재기’ 모드 형태로 설정합니다. |
| mmCalcDistanceEx | 마우스 모드를 ‘거리재기’ 모드 형태로 설정하 며, 마우스 모드 종료 후에 DrawFinishedGeo  metry 이벤트가 호출됩니다. |
| mmCustom | 마우스 모드를 사용자 모드로 설정합니다. |
| mmDrawArc | 마우스 모드를 원호그리기 모드로 설정합니다. |
| mmDrawArcLine | 마우스 모드를 선 및 원호그리기 모드로 설정  합니다. (Ctrl 키 누를 경우 : 원호 그리기) |
| mmDrawCircle | 마우스 모드를 원그리기 모드로 설정합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

|  |  |
| --- | --- |
| mmDrawLine | 마우스 모드를 선그리기 모드로 설정합니다. |
| mmDrawMultiMove | 마우스 모드를 객체이동 모드로 설정합니다. (버텍스 선택 : 버텍스이동 모드,  객체 선택 : 객체이동 모드) |
| mmDrawPoint | 마우스 모드를 점그리기 모드로 설정합니다. |
| mmDrawPolygon | 마우스 모드를 면그리기 모드로 설정합니다. |
| mmDrawPolygonSplit | 마우스 모드를 폴리곤 분할 그리기 모드로 설  정합니다. |
| mmDrawRectangle | 마우스 모드를 직사각형그리기 모드로 설정합니다. |
| mmEditMove | 마우스 모드를 객체이동 모드로 설정합니다. |
| mmEditMoveResize | 마우스 모드를 객체이동 및 크기조작 모드로  설정합니다. |
| mmEditSelect | 마우스 모드를 객체선택 모드로 설정합니다. |
| mmEditVertex | 마우스 모드를 버텍스 편집 모드로 설정합니다. |
| mmEditVertexAdd | 마우스 모드를 버텍스 추가 모드로 설정합니다. |
| mmEditVertexMove | 마우스 모드를 버텍스 이동 모드로 설정합니다. |
| mmEditVertexRemove | 마우스 모드를 버텍스 삭제 모드로 설정합니다. |
| mmMapRotate | 마우스 모드를 지도회전 모드로 설정합니다. |
| mmMouseModeOff | 마우스 모드를 중지 모드로 설정합니다. |
| mmPanning | 마우스 모드를 이동 모드로 설정합니다. |
| mmScroll | 마우스 모드를 스크롤 모드로 설정합니다. |
| mmZoomIn | 마우스 모드를 화면확대 모드로 설정합니다. |
| mmZoomOut | 마우스 모드를 화면축소 모드로 설정합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

마우스 모드에 따른 기능

다음 예제는 IntraMap/Objects 3.0이 제공하는 기본적인 화면 제어 기능과 마 우스 모드 Constants에 따른 마우스 포인터의 타입과 각 기능들을 구현한 예제 이다.

[예제 6-1]

private void cmd\_ZoomIn\_Click()

{

// 화면 확대

axGMapView1.ZoomIn();

}

private void cmd\_ZoomOut\_Click()

{

// 화면 축소

axGMapView1.ZoomOut();

}

private void cmd\_FullExtent\_Click()

{

// 전체 보기

axGMapView1.Fit();

}

private void cmd\_mmNormal\_Click()

{

// 기본 모드



 IntraMap/Objects ***PROGRAMMERS GUIDE***

axGMapView1.MouseMode = eMouseMode.mmNormal;

}

private void cmd\_Panning\_Click()

{

// 이동 모드

axGMapView1.MouseMode = eMouseMode.mmPanning;

}

private void cmd\_mmZoomIn\_Click()

{

// 확대 모드

axGMapView1.MouseMode = eMouseMode.mmZoomIn;

}

private void cmd\_mmZoomOut\_Click()

{

// 축소 모드

axGMapView1.MouseMode = eMouseMode.mmZoomOut;

}

private void cmd\_mmDrawPoint\_Click()

{

// Point 그리기 모드

axGMapView1.MouseMode = eMouseMode.mmDrawPoint;

}

private void cmd\_mmDrawLine\_Click()

{

// Line 그리기 모드



 IntraMap/Objects ***PROGRAMMERS GUIDE***

axGMapView1.MouseMode = eMouseMode.mmDrawLine;

}

Private void cmd\_DrawPolygon\_Click()

{

// Polygon 그리기 모드

axGMapView1.MouseMode = eMouseMode.mmDrawPolygon;

}

private void cmd\_mmDrawRectangle\_Click()

{

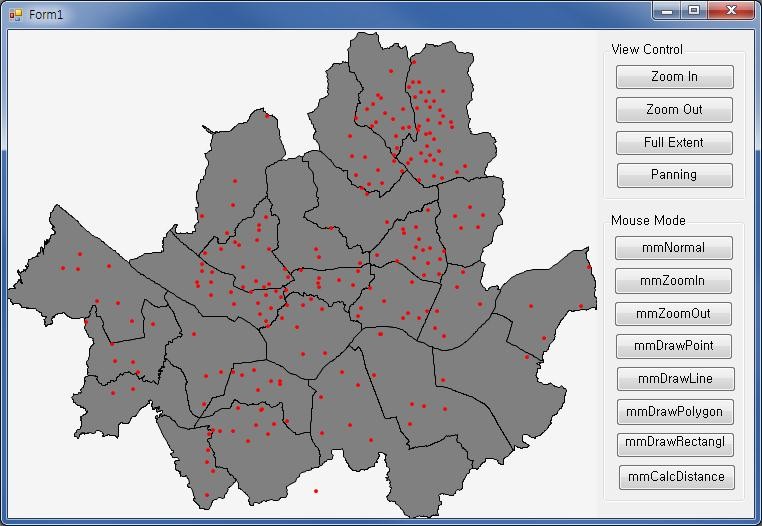
// Rectangle 그리기 모드

axGMapView1.MouseMode = eMouseMode.mmDrawRectangle;

}



 IntraMap/Objects ***PROGRAMMERS GUIDE***

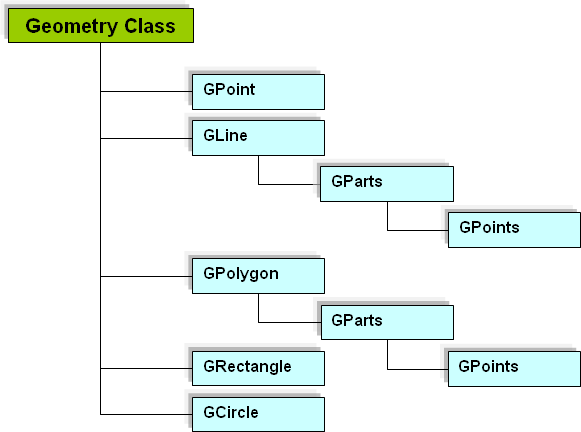




 IntraMap/Objects ***PROGRAMMERS GUIDE***

# Geometry 공간 질의 및 SQL을 이용한 데이터 검색

IntraMap/Objects 3.0에서는 다음의 다이어그램과 같이 다양한 Geometry 타 입들을 제공하며, 각 Geometry에 관한 프라퍼티와 메소드를 지원한다. 또한, Geometry 간의 다양한 공간 연산을 수행하는 메소드를 지원한다. IntraMap/Objects 3.0에서 데이터베이스에 저장하기 위한 기본 개체로는 GPoint, GLine, GPolygon이 있으며, 이를 보조하기 위하여 확장된 개체로는 GRectangle, GCircle 등이 있다.





 IntraMap/Objects ***PROGRAMMERS GUIDE***

* Geometry

Property

|  |  |
| --- | --- |
| Name | Description |
| Type | 지오메트리 타입을 반환합니다. |
| Extent | 지오메트리 객체의 영역을 반환합니다. |

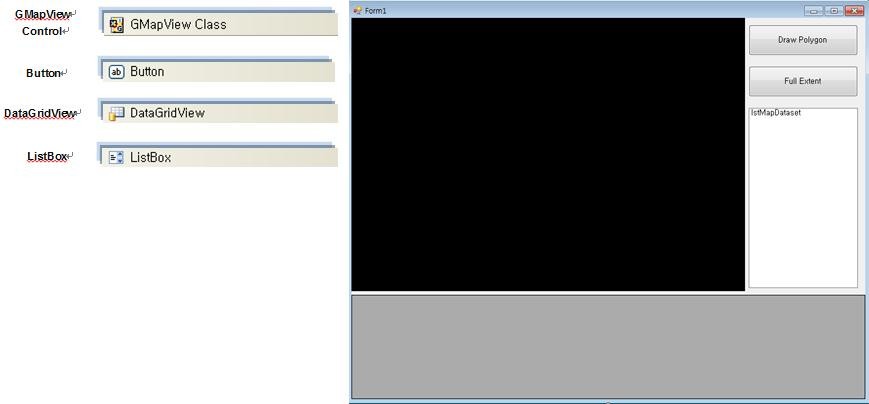
Method

|  |  |
| --- | --- |
| Name | Description |
| Buffer | 일정거리만큼 버퍼링한 영역을 생성하여 반환합니다. |
| CalculateBound | 객체가 분포하는 영역좌표를 계산하여 반환합니다. |
| Center | 중심을 계산하여 반환합니다. |
| Contain | 선택 객체가 입력 객체를 포함하는 지 여부를 반환합니다. |
| DistanceTo | 주어진 점으로부터 객체까지 가장 가까운 거리를 반  환합니다. |
| GetCrossPoints | 객체간의 교차되는 점을 반환합니다. |
| Intersect | 선택 객체가 입력 객체와 교차하는 지를 반환합니다. |
| Move | 설정한 거리만큼 객체를 이동시킵니다. |
| Within | 객체가 입력 객체에 포함되는 지를 반환합니다. |

 IntraMap/Objects ***PROGRAMMERS GUIDE***

1. GPolygon을 이용한 공간 질의

다음 예제는 db.mdb의 ELE\_SCHOOL 레이어와 사용자가 설정한 GPolygon 개 체간의 within 연산을 수행하는 예제이다. 본 예제에서는 GPolygon 개체에 포함 된 ELE\_SCHOOL 포인트 개체들의 속성 리스트를 DataGridView 콘트롤과 연결 하여 보여 줄 것이다.



db.mdb 파일에 연결하면 6개의 레이어 정보를 가져올 수 있다. 본 예제에서는 이 레이어들 중에 ELE\_SCHOOL의 Point 객체 레이어 정보에 대하여 공간 질의 를 수행하는 예제로 ELE\_SCHOOL 레이어만을 GMapView에 다음과 같은 심볼을 설정하여 디스플레이한다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***

[예제 7-1-1]

public static GConnections GConns = new GConnections(); Geometry geo;

private void Form1\_Load(object sender, EventArgs e)

{

GMapDatasets mdatasets;

GConnection GConn = GConns.NewConnection("MSACCESS"); GConn.User = "";

GConn.Password = "";

string sPath = Environment.GetFolderPath(

Environment.SpecialFolder.ProgramFiles);

GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb"; GConn.Connect();

GConns.Add(GConn);

mdatasets = GConns.Item(0).GetMapDatasets(); lstMapDataset.Items.Clear();

for (int i = 0; i < mdatasets.Count; i++)

{

lstMapDataset.Items.Add(i + " " + mdatasets.GetMapDataset(i)

.Name.ToString());

 IntraMap/Objects ***PROGRAMMERS GUIDE***

}

GRecordLayer RLayer = GConns.Item(0).GetMapDatasets()

.GetMapDataset("ELE\_SCHOOL").CreateLayer();

GSimpleRenderer Render = (GSimpleRenderer)RLayer.Renderer; GSymbol Sym = Render.symbol;

Sym.Icon.Type = eIconTypes.itCircle;

Sym.Icon.Color = getRGB(100,0, 255, 0);

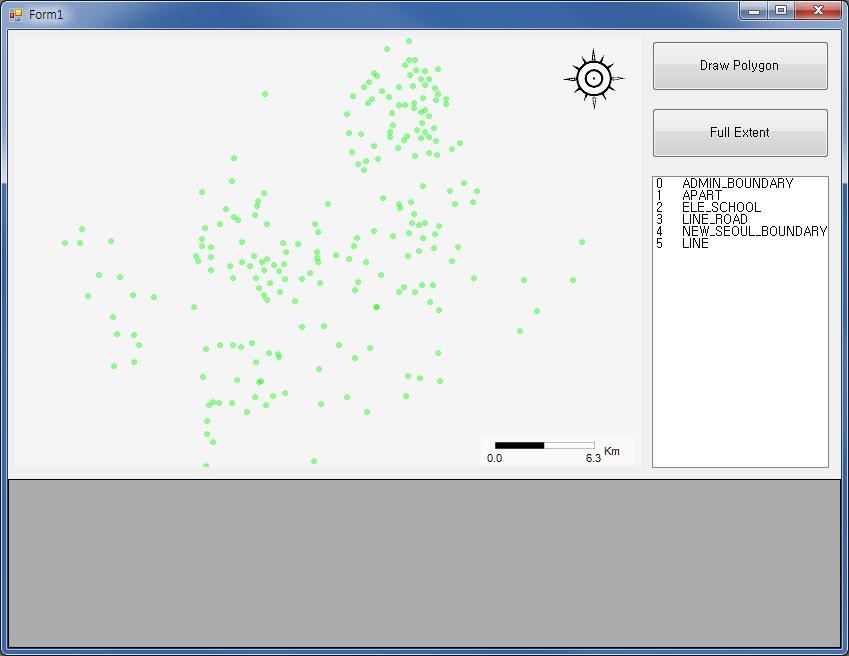
Sym.Icon.Size = 5; axGMapView1.Layers.Add((GLayer)RLayer);

axGMapView1.DirectionSymbolUse = true; axGMapView1.DirectionSymbolIndex = 3; axGMapView1.ScaleBarUse = true; axGMapView1.BackgroundColor = Color.WhiteSmoke.ToArgb(); axGMapView1.Fit();

}

위 코드를 수행하면 ELE\_SCHOOL의 포인트 레이어만 설정한 심볼대로 디스플 레이 된다.





IntraMap/Objects

***PROGRAMMERS GUIDE***

이제 디스플레이된 ELE\_SCHOOL 레이어 위에 공간 질의를 수행할 Polygon 개 체를 사용자가 그릴 수 있도록 하여야 한다. 사용자가 그린 Polygon 개체의 Geometry 정보는 GMapView의 DrawFinishedGeometry 이벤트를 통하여 얻을 수 있으며, 화면 갱신은 GMapView의 AfterDraw 이벤트를 통하여 수행하면 된 다.

 IntraMap/Objects ***PROGRAMMERS GUIDE*** 우선 사용자가 GMapView에 개체를 그리기 위해서는 마우스 모드를 변경하여 야 한다. 본 예제에서는 Polygon 객체로 공간 질의를 수행함으로 Draw Polygon 커맨드 버튼에 다음과 같이 마우스 모드를 설정한다.

axGMapView1.MouseMode = eMouseMode.mmDrawPolygon;

다음은 GMapView 상에서 사용자가 Polygon 개체를 그리고 난 후에 사용자가 그린 객체의 Geometry 정보를 얻어오고, 이를 바탕으로 공간검색을 수행하기 위한 DrawFinishedGeometry 이벤트이다.

[예제 7-1-2]



// 질의 수행 후 레코드의 존재 여부 확인

// 사용자가 입력한 지오메트리 정보를 이용하여 공간 질의 수행

GRecordset gRs = mdataset.GetRecords("", eSpatialQueryTypes.sqtWithin, geo);

gRs.MoveFirst();

AxIMapObjects2Lib.\_IGMapViewEvents\_DrawFinishedGeometryEvent e)

{

// 사용자가 그린 지오메트리 정보 저장

geo = e.geometry;

// 공간질의를 수행할 MapDataset 설정

GMapDataset mdataset =

GConns.Item(0).GetMapDatasets().GetMapDataset("ELE\_SCHOOL");

sender,

axGMapView1\_DrawFinishedGeometry(object

void

private



 IntraMap/Objects ***PROGRAMMERS GUIDE***

if (gRs.RecordCount > 0)

{

dataGridView1.Rows.Clear(); dataGridView1.Columns.Clear();

// DataGridView 크기 설정

for (int k = 0; k < gRs.Fields.Count; k++)

{

dataGridView1.Columns.Add(

k.ToString(), gRs.Fields.Item(k).Name.ToString());

}

for (int i = 0; i < gRs.RecordCount; i++)

{

dataGridView1.Rows.Add();

for (int j = 0; j < gRs.Fields.Count; j++)

{

// 필드명이 “XGEOMETRY” 인 경우에만 “XGEOM”으로 변경

if (gRs.Fields.Item( j).Name.ToString() == "XGEOMETRY")

{

dataGridView1.Rows[i].Cells[ j].Value = "XGEOM";

}

else

dataGridView1.Rows[i].Cells[ j].Value = gRs.Fields.Item( j).Value;

}

gRs.MoveNext();



 IntraMap/Objects ***PROGRAMMERS GUIDE***

}

}

// 결과 레코드셋을 반환하고 초기화

gRs = null;

}

위 코드를 수행한 후에는 GMapView의 AfterDraw 이벤트를 이용하여 사용자 가 그린 지오메트리 개체를 디스플레이하여야 한다.

[예제 7-1-3]

//DrawPolyGon버튼 클릭 이벤트 설정

Private void cmdDrawPolygon\_Click (object sender, EventArgs e)

{ axGMapView1.MouseMode = eMouseMode.mmDrawPolygon;

}

private void

axGMapView1\_AfterDraw (object sender, AxIMapObjects3Lib.\_IGMapViewEvents\_AfterDrawEvent e)

{

// 사용자가 그린 지오메트리 객체 존재 여부 확인

if (geo != null)

{

// 지오메트리 객체 표출을 위한 심볼의 속성값 설정

GSymbol Sym = new GSymbol(); Sym.Pen.Color = getRGB(255, 0, 0);

Sym.Brush.Color = getRGB(0, 0, 255); Sym.Brush.Type = eBrushTypes.ftHatch;



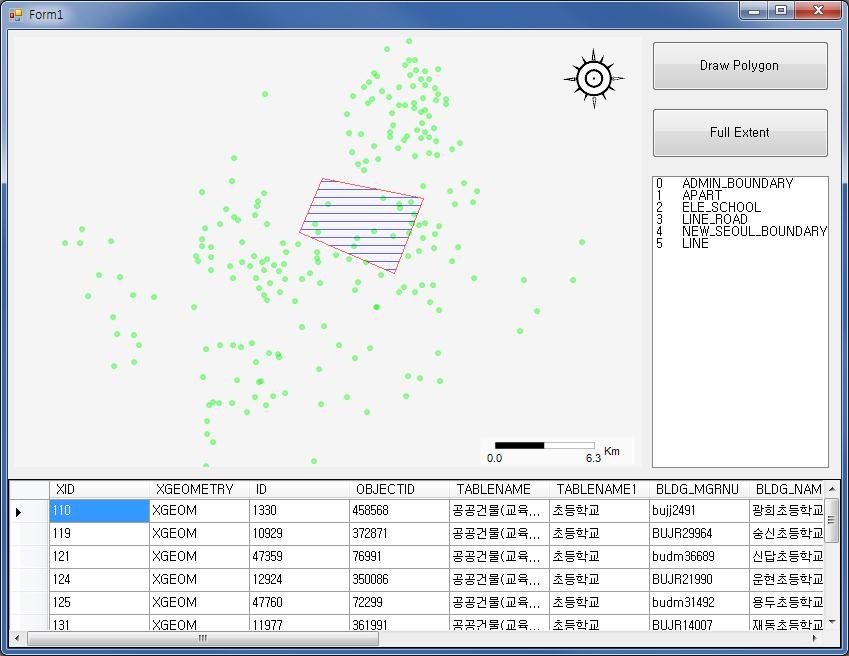
 IntraMap/Objects ***PROGRAMMERS GUIDE***

Sym.Brush.Hatch = eHatchTypes.htHorizontal;

// 설정한 심볼로 지오메트리 객체 표출 axGMapView1.SetDrawSymbol(Sym); axGMapView1.DrawGeometry(geo);

}

}





 IntraMap/Objects ***PROGRAMMERS GUIDE***

1. SQL 쿼리문을 이용한 데이터 검색

IntraMap Object 3.0은 Geometry간의 공간 질의뿐만 아니라 접속된 데이터베 이스에 SQL문과 부가적인 여러 명령어들을 활용하여 대화형으로 처리할 수 있 다. 이번 장에서는 SQL 쿼리를 통하여 해당하는 레코드를 검색할 수 있는 OpenRecordset과 GetRecords에 대해서 알아본다.

SQL문을 이용한 데이터 검색

이번 예제에서는 GetRecords과 OpenRecordset을 활용하여 사용자 입력하는 SQL 쿼리문에 의한 데이터 검색을 다룬다. 다음 코드는 Execute 커맨드 버튼을 클릭 시에 APART 레이어에 대하여 쿼리가 수행되는 코드이다.

[예제 7-2-1]

private void cmdExecute\_Click (object sender, EventArgs e)

{

// “APART” 레이어의 MapDataset 정보를 반환 GMapDataset mdataset = GConns.Item(0).GetMapDatasets().GetMapDataset("APART");

// MapDateset에서 “BLDG\_NAME = ‘벽산아파트’” 만족하는 레코드 반환

GRecordset gRs = mdataset.GetRecords("BLDG\_NAME='벽산아파트'", eSpatialQueryTypes.sqtNone, "");

// 이하는 결과 레코드셋을 DataGridView에 넣어주는 코드

gRs.MoveFirst();



 IntraMap/Objects ***PROGRAMMERS GUIDE***

if (gRs.RecordCount > 0)

{

dataGridView1.Rows.Clear(); dataGridView1.Columns.Clear();

for (int k = 0; k < gRs.Fields.Count; k++)

{

dataGridView1.Columns.Add(k.ToString(), gRs.Fields.Item(k).Name.ToString());

}

for (int i = 0; i < gRs.RecordCount; i++)

{

dataGridView1.Rows.Add();

for (int j = 0; j < gRs.Fields.Count; j++)

{

if (gRs.Fields.Item( j).Name.ToString() == "XGEOMETRY")

{

dataGridView1.Rows[i].Cells[ j].Value = "XGEOM";

}

else

dataGridView1.Rows[i].Cells[ j].Value = gRs.Fields.Item( j).Value;

}

gRs.MoveNext();

}

}



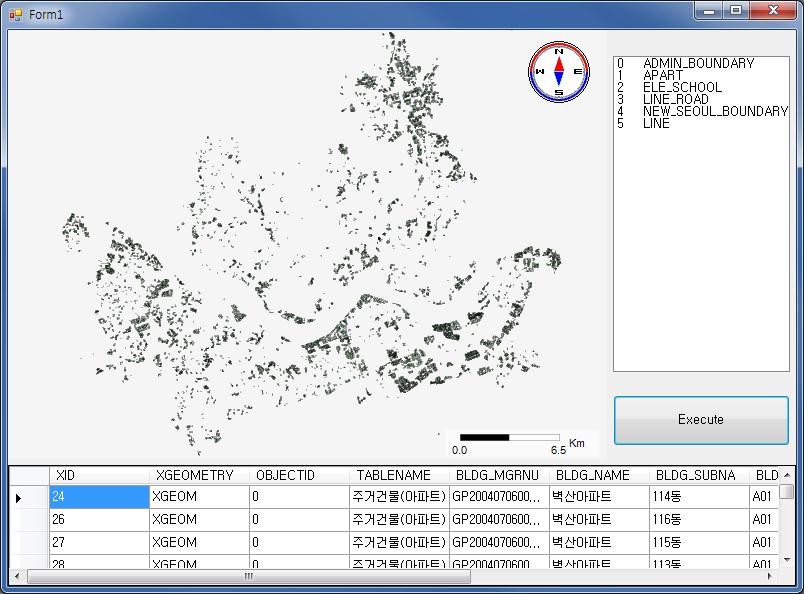
 IntraMap/Objects ***PROGRAMMERS GUIDE***

// 검색된 레코드를 하이라이트하기 위해 화면을 다시 그려줌

axGMapView1.Refresh();

}

위 코드에서 사용된 GetRecords는 대상 레이어의 객체에 대하여 검색 조건에 해당하는 레코드 정보를 반환하는 역할을 수행한다. 여기에서는 filter 조건으로 BLDG\_NAME이 “벽산아파트”인 객체들만을 검색할 수 있도록 설정하였다.

다음 그림은 BLDG\_NAME이 “벽산아파트”인 개체들이 검색된 그림이다.



 IntraMap/Objects ***PROGRAMMERS GUIDE*** 여기에 검색된 레코드들에 대하여 사용자들이 위치를 쉽게 인식할 수 있도 록 해당 객체들을 하이라이트 시켜주는 것은 GIS 응용 프로그램에서 필수적인 기능이다. 앞의 코드에서 axGMapView1.Refresh() 가 이를 수행할 수 있도록 GMapView의 AfterDraw 이벤트에 다음과 같은 코드를 추가한다.

[예제 7-2-2]

private void axGMapView1\_AfterDraw

(object sender, AxIMapObjects3Lib.\_IGMapViewEvents\_AfterDrawEvent e)

{

// Geometry 정보를 저장할 변수 선언

Geometry geo;

// 검색된 객체를 그리기 위한 심볼의 속성값 설정

GSymbol Sym = new GSymbol(); Sym.Pen.Color = getRGB(100,255, 0, 0);

Sym.Pen.Width = 5;

Sym.Brush.Color = getRGB(100,255, 0, 0); Sym.Brush.Type = eBrushTypes.ftSolid; axGMapView1.SetDrawSymbol(Sym);

// 레코드의 존재 여부 판단

if (gRs.RecordCount > 0)

{

// 첫 번재 레코드로 이동

gRs.MoveFirst();

for (int i = 0; i < gRs.RecordCount; i++)

{ // 검색된 레코드의 지오메트리 정보 반환

geo = (Geometry)gRs.Fields.Item("XGEOMETRY").Value;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

// 선택된 객체를 설정한 심볼로 그림

axGMapView1.DrawGeometry(geo);

// 순서대로 모든 레코드 객체를 그리기 위하여 이동

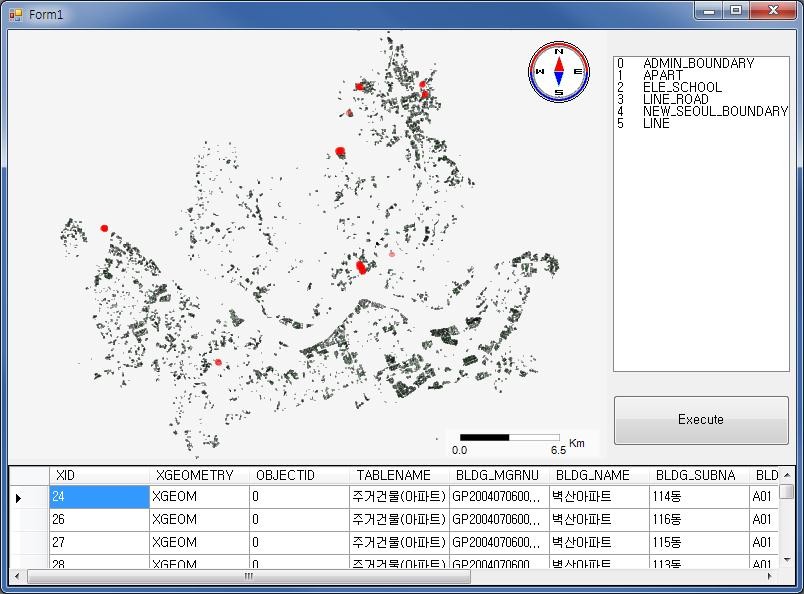
gRs.MoveNext();

}

}

}

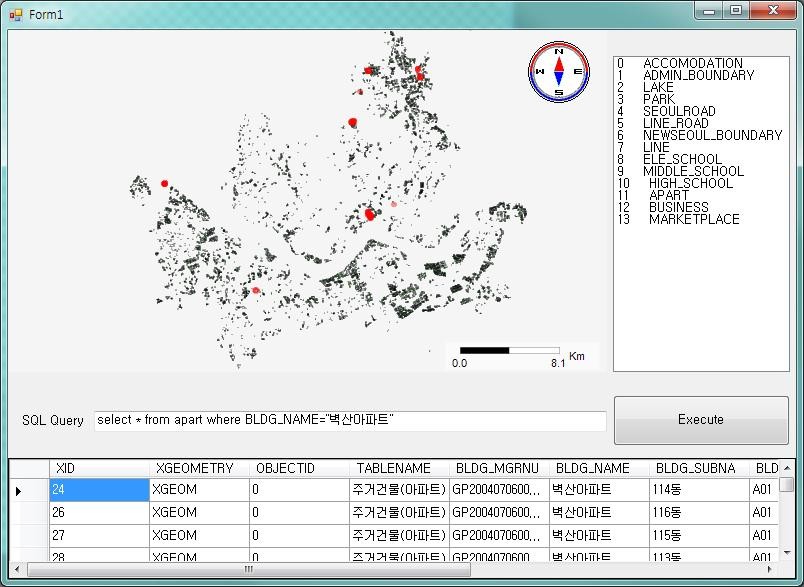
위 코드는 질의 검색 후, GMapView1.Refresh에 의하여 수행된다. 다음은 선택 된 레코드들을 하이라이트 시킨 그림이다.





 IntraMap/Objects ***PROGRAMMERS GUIDE*** GetRecords는 해당 레이어의 맵데이터셋으로부터 필터 조건을 설정하여 공간 데이터를 검색하는 메소드였다. 이와 유사한 메소드로 OpenRecords는 SQL 질 의를 통해 검색된 레코드 셋을 반환하는 기능을 제공한다. 다음은 텍스트 박스 (QueryText)를 활용하여 사용자의 SQL문을 입력 받아 레코드셋 정보를 가져오 는 예이다.

GRecordset gRs = GConns.Item(0).OpenRecordset(QueryText.Text)





 IntraMap/Objects ***PROGRAMMERS GUIDE***

# Geometry 간 공간연산

IntraMap Object 3.0에서는 다양한 공간연산기능을 제공하고 있다. 공간연산기 능은 공간 상에 분포한 자료들의 다양한 공간관계를 분석할 수 있도록 해주며, 데이터들의 유형에 따라 각각의 유형에 적합한 공간연산기능을 제공해준다.

IntraMap Object 3.0에서의 대표적인 공간연산기능을 살펴보면 Geometry 간 공통적인 공간연산함수로 DistanceTo, Buffer, GetCrossPoints 등이 있으며, GPolygon 간의 공간연산함수로 Union, SplitByPoints, Difference, Intersection, ExclusiveOr 등이 있다. 아래의 그림은 IntraMap Object 3.0에서의 다양한 공간 연산기능을 보여준다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***

< Geometry 간 공통 연산함수>



Result

DistanceTo

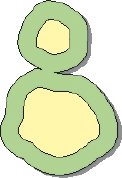
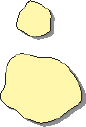
Original

A

A

Distance

**DistanceTo**



Result

Buffering

Base

**Buffer**

Result

GetCrossPoints

Original

=

A

B

+

A

Cross Points

**Result = A.GetCrossPoints(B)**



 IntraMap/Objects ***PROGRAMMERS GUIDE***

< GPolygon 공간연산함수>

Result

Original

A

=

B

+

**A.Union B**

A

Result

Original

C

B

A

A

Points

**A.SplitbyPoints Points, B, C**



Result

Difference

Original

A

=

A

+

A

B

**A.Difference B**



 IntraMap/Objects ***PROGRAMMERS GUIDE***



Result

Intersection

Original

=

A

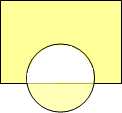
A

+

A

B

**A.Intersection B**



Result

ExclusiveOr

Original

A

=

A

+

A

B

**A.ExclusiveOr B**



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* 1. GPolygon 공간연산기능

이번 예제에서는 GPolygon의 공간연산 메소드와 DrawFinishedGeometry 이 벤트를 활용하여, Split by Line, Split by Polygon, Union 등의 공간연산기능을 구 현해보는 예제이다. 본 예제에서는 NEW\_SEOUL\_BOUNDARY 폴리곤 개체를 바 탕으로 공간연산 대상 개체를 선택한 후에 대상 개체에 대하여 원하는 공간 연 산기능을 수행하는 과정을 보여줄 것이다.

다음 코드는 폼을 생성한 후에 연결된 데이터베이스에서 NEW\_SEOUL\_BOUNDARY 레이어를 GMapView상으로 불러오고 원하는 심볼로 표출하는 내용을 보여준다.

[예제 8-1-1]

// 전역 변수 선언

public static GConnections GConns = new GConnections();

// RGB색상 설정을 위한 getRGB 함수 선언

public static int getRGB(int R, int G, int B)

{

return Color.FromArgb(150, R, G, B).ToArgb();

}

public static int getRGB(int A, int R, int G, int B)

{

return Color.FromArgb(A, R, G, B).ToArgb();

}



 IntraMap/Objects ***PROGRAMMERS GUIDE***

public void Form1\_Load(object sender, EventArgs e)

{

// Form\_Load 부분 설정

GConnection GConn = GConns.NewConnection("MSACCESS"); GMapDatasets mdatasets;

GConn.User = ""; GConn.Password = ""; string sPath =

Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles); GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb"; GConn.Connect();

GConns.Add(GConn);

mdatasets = GConns.Item(0).GetMapDatasets();

// NEW\_SEOUL\_BOUNDARY의 GLayer 정보 반환

GRecordLayer RLayer = GConns.Item(0).GetMapDatasets()

.GetMapDataset("NEW\_SEOUL\_BOUNDARY").CreateLayer();

GSimpleRenderer Render = (GSimpleRenderer)RLayer.Renderer; GSymbol Sym = Render.symbol;

Sym.Pen.Color = getRGB(100,0, 0, 255);

Sym.Pen.Width = 1;

Sym.Pen.Type = ePenTypes.ltDash;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

Sym.Brush.Color = getRGB(100,255, 255, 0); Sym.Brush.Type = eBrushTypes.ftSolid;

// GMapView에 NEW\_SEOUL\_BOUNDARY 레이어 추가 axGMapView1.Layers.Add((GLayer)RLayer); axGMapView1.DirectionSymbolUse = true; axGMapView1.DirectionSymbolIndex = 1; axGMapView1.ScaleBarUse = true;

axGMapView1.BackgroundColor = Color.WhiteSmoke.ToArgb(); axGMapView1.Fit();

}

위 코드를 수행하면 GMapView 상에 NEW\_SEOUL\_BOUNDARY 레이어의 폴리 곤 데이터들은 노란색으로 채워지며 파란 점선의 경계선을 가지게 된다. 데이터 간의 공간연산을 위해서는 공간연산 대상 데이터를 선택한 후에 선택 데이터와 의 연산을 위한 공간객체를 그리는 작업이 필요하다. 다음의 코드는 데이터의 선택을 위한 마우스 모드의 설정과 Split by Line, Split by Polygon, Union 등의 공간연산 수행에 필요한 준비를 위한 버튼을 설정하는 부분이다.

[예제 8-1-2]

// Button별 구분을 위한 전역 변수 설정

Boolean CheckOn1, CheckOn2, CheckOn3;

// 객체 선택을 위한 하이라이트 버튼

private void cmdSelect\_Click(object sender, EventArgs e)



 IntraMap/Objects ***PROGRAMMERS GUIDE***

{

// 하이라이트 가능여부 설정

axGMapView1.HilightEnable = true;

// 사용자 모드로 마우스 모드설정

axGMapView1.MouseMode = eMouseMode.mmCustom;

}

// Union 버튼

private void cmdUnion\_Click(object sender, EventArgs e)

{

// 버튼 별 구분을 위한 변수 설정

CheckOn1 = true; CheckOn2 = false; CheckOn3 = false;

// 편집모드 확인

if (axGMapView1.EditMode.Equals(eEditModes.emNone) == true)

{ return;

}

// 마우스 모드의 설정

axGMapView1.MouseMode = eMouseMode.mmDrawPolygon;

}

// Split By Polygon 버튼

private void cmdSplitByPoly\_Click(object sender, EventArgs e)



 IntraMap/Objects ***PROGRAMMERS GUIDE***

{

// 버튼 별 구분을 위한 변수 설정

CheckOn1 = false; CheckOn2 = true; CheckOn3 = false;

if (axGMapView1.EditMode.Equals(eEditModes.emNone) == true)

{ return;

}

axGMapView1.MouseMode = eMouseMode.mmDrawPolygon;

}

// Split By Line 버튼

private void cmdSplitByLine\_Click(object sender, EventArgs e)

{

// 버튼 별 구분을 위한 변수 설정

CheckOn1 = false; CheckOn2 = false; CheckOn3 = true;

if (axGMapView1.EditMode.Equals(eEditModes.emNone) == true)

{ return;

}

axGMapView1.MouseMode = eMouseMode.mmDrawLine;

}



 IntraMap/Objects ***PROGRAMMERS GUIDE***

위의 코드 중 개체선택 버튼을 선택한 후에 GMapView 상에서 공간연산을 수행할 개체를 클릭하게 되면, 사용자 모드 (Custom Mode)의 LButtonDown 이 벤트가 호출된다. LButtonDown 이벤트에서는 공간연산의 결과가 반영될 수 있 도록 편집모드를 설정하고 내용과 공간연산의 대상이 되는 선택개체의 정보를 가져오는 내용이 들어간다. 다음의 코드는 사용자 모드에서 마우스 클릭에 의해 호출되는 LButtonDown 이벤트 부분이다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***

[예제 8-1-3]

// 객체 선택 및 편집 모드 설정을 위한 사용자 모드의 설정

private void axGMapView1\_CustomToolLButtonUp(object sender,

AxIMapObjects3Lib.\_IGMapViewEvents\_CustomToolLButtonUpEvent e)

{

GMapDataset mdataset; String strLayerName; int nXID;

// 선택 개체의 정보 획득

axGMapView1.GetHilightedObjectInfo(out strLayerName, out nXID);

mdataset = GConns.Item(0).GetMapDatasets().GetMapDataset(strLayerName);

selectedRS =

mdataset.GetRecords("XID = " + nXID.ToString(), eSpatialQueryTypes.sqtNone, "");

selectedRS.MoveFirst();

// 편집 모드 및 편집 대상 필드의 설정 axGMapView1.StartEditObject ((Geometry)selectedRS.Fields.Item("XGEOMETRY").Value);

}



 IntraMap/Objects ***PROGRAMMERS GUIDE***

객체가 선택된 후, Split by Line, Split by Polygon, Union 중 하나의 연산이 수 행될 것인데 수행될 공간연산의 구분은 CheckOn1, CheckOn2, CheckOn3 변수 의 True/False 여부로 이뤄진다. 각각의 공간연산은 개체가 그려진 후에 실행되 어야하므로 DrawFinishedGeometry 이벤트를 통하여 수행된다. 공간연산을 수행 하기 위해서는 LButtonDown 이벤트에 의해 선택된 레코드셋의 정보와 사용자 에 의해 그려진 개체의 정보를 저장시켜놓은 후 이를 바탕으로 연산이 수행되 어야 한다. 다음의 코드는 각각의 공간연산이 수행되는 과정을 보여주고 있다.

[예제 8-1-4]

// LButtonDown 이벤트에 의해 선택된 레코드셋을 위한 전역 변수 설정

GRecordset selectedRS;

// DrawFinishedGeometry 이벤트를 통한 공간연산 수행

private void axGMapView1\_DrawFinishedGeometry(object sender, AxIMapObjects3Lib.\_IGMapViewEvents\_DrawFinishedGeometryEvent e)

// 선택 객체를 위한 변수 선언

GPolygon selectedPolygon;

// 선택 객체를 위한 변수 설정

selectedPolygon = (GPolygon)selectedRS.Fields.Item("XGEOMETRY").Value;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

// 공간 연산 - Union

if (CheckOn1 == true)

{

// 사용자가 그린 객체를 위한 변수 선언

GPolygon tmpPolygon1;

// 사용자가 그린 객체를 위한 변수 설정

tmpPolygon1 = (GPolygon)e.geometry;

// Union 공간연산 수행

selectedPolygon.Union(tmpPolygon1);

selectedRS.Edit();

// 연산 수행 결과 반환

selectedRS.Fields.Item("XGEOMETRY").Value = selectedPolygon;

// 레코드셋 업데이트

selectedRS.Update();

// 공간연산 결과 반영을 위해 지도 다시 그림

axGMapView1.Redraw();

}

// 공간 연산 – Erase by Polygon

if (CheckOn2 == true)

{



 IntraMap/Objects ***PROGRAMMERS GUIDE***

// 사용자가 그린 객체를 위한 변수 선언

GPolygon tmpPolygon2;

// 공간 연산 결과 저장을 위한 변수 선언 (비교차 영역)

GPolygon differPolygon;

// 사용자가 그린 객체를 위한 변수 설정

tmpPolygon2 = (GPolygon)e.geometry;

// 공간 연산 결과 저장을 위한 변수 설정 (교차 영역)

differPolygon = (GPolygon)selectedRS.Fields.Item("XGEOMETRY").Value;

// 예외 처리 : 교차 영역 없을 경우

if (selectedPolygon.Intersect(e.geometry) == false)

{

MessageBox.Show("Error!"); return;

}

// 교차 영역을 위한 공간 연산 수행

selectedPolygon.Intersection(tmpPolygon2);

// 비교차 영역을 위한 공간 연산 수행

differPolygon.Difference(tmpPolygon2);

selectedRS.Edit();

// 교차 영역 연산 수행 결과 반환

selectedRS.Fields.Item("XGEOMETRY").Value = differPolygon;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

// 레코드셋 업데이트

selectedRS.Update();

// 레코드셋 업데이트

selectedRS.Update();

// 공간연산 결과 반영을 위해 지도 다시 그림

axGMapView1.Redraw();

}

// 공간 연산 – Spilt by Line

if (CheckOn3 == true)

{

// 공간 연산 결과 저장을 위한 변수 선언 (왼쪽 영역)

GPolygon leftPolygon;

// 공간 연산 결과 저장을 위한 변수 선언 (오른쪽 영역)

GPolygon rightPolygon;

// 사용자가 그린 객체를 위한 변수 선언

GLine DrawnLine;

// 공간 연산 수행을 위한 GPoints 변수 선언

GPoints points;

// 사용자가 그린 객체를 위한 변수 설정

DrawnLine = (GLine)e.geometry;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

// 예외 처리 : 교차 영역 없을 경우

if (selectedPolygon.Intersect(e.geometry) == false)

{ MessageBox.Show("Error!"); return;

}

// 공간 연산 수행을 위한 변수 설정

points = DrawnLine.Parts.Item(0);

// 공간 연산 수행

selectedPolygon.SplitByPoints(points, out leftPolygon, out rightPolygon);

selectedRS.Edit();

// 공간 연산 수행 결과 (왼쪽 영역) 반환

selectedRS.Fields.Item("XGEOMETRY").Value = leftPolygon;

// 레코드셋 업데이트

selectedRS.Update();

selectedRS.AddNew();

// 공간 연산 수행 결과 (오른쪽 영역) 반환

selectedRS.Fields.Item("XGEOMETRY").Value = rightPolygon;

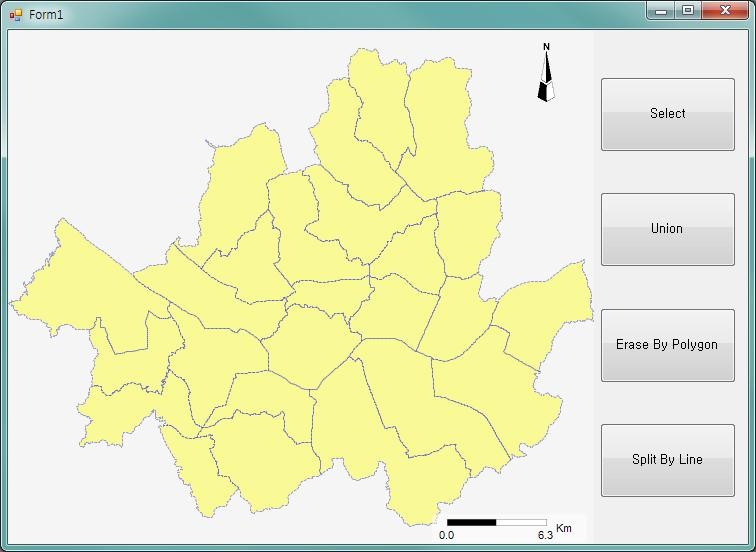
// 레코드셋 업데이트 selectedRS.Update(); axGMapView1.Redraw();

}



 IntraMap/Objects ***PROGRAMMERS GUIDE***

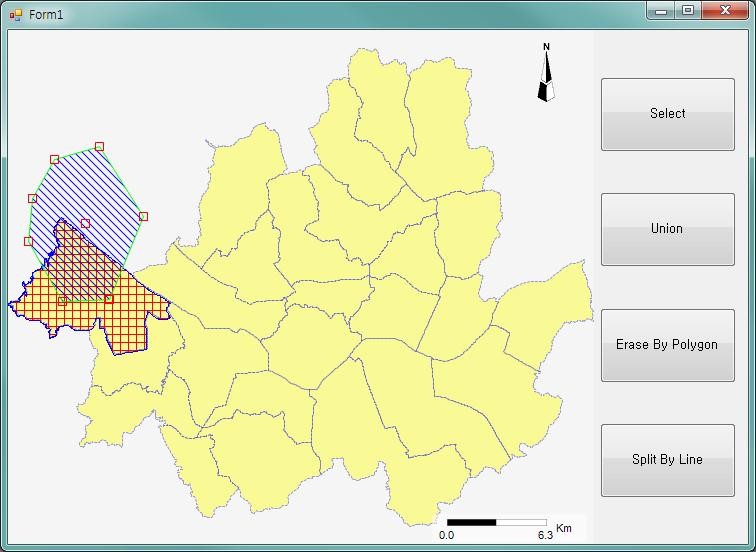
위 코드는 공간연산 실행 후, axGMapView1.Redraw() 에 의하여 수행된다. 다 음은 공간 객체를 선택한 후에 각각의 공간연산을 실행하는 과정을 보여주는 그림이다.



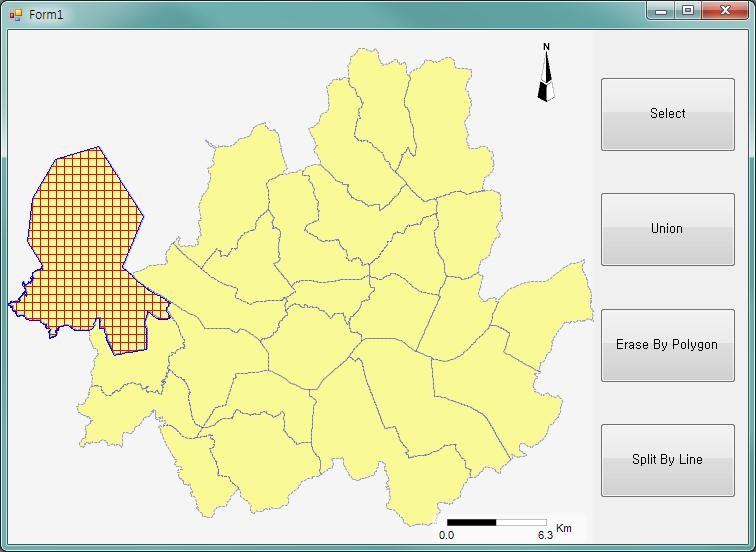
< NEW\_SEOUL\_BOUNDARY >



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* Union 연산

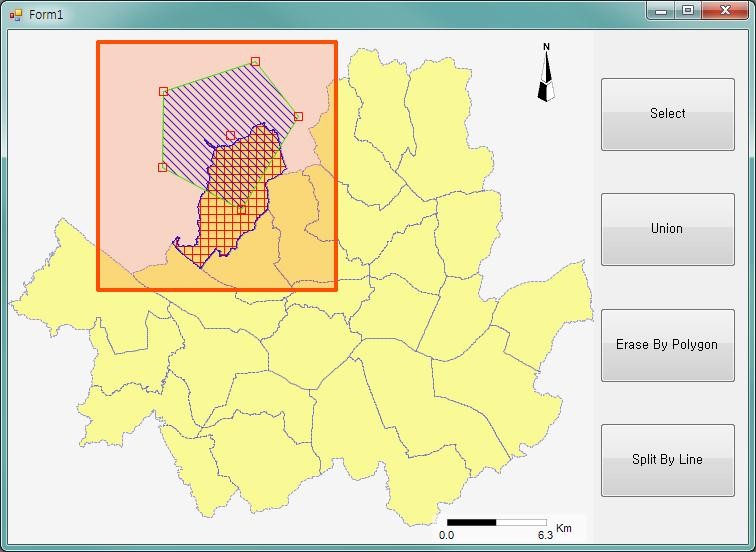
데이터 입력



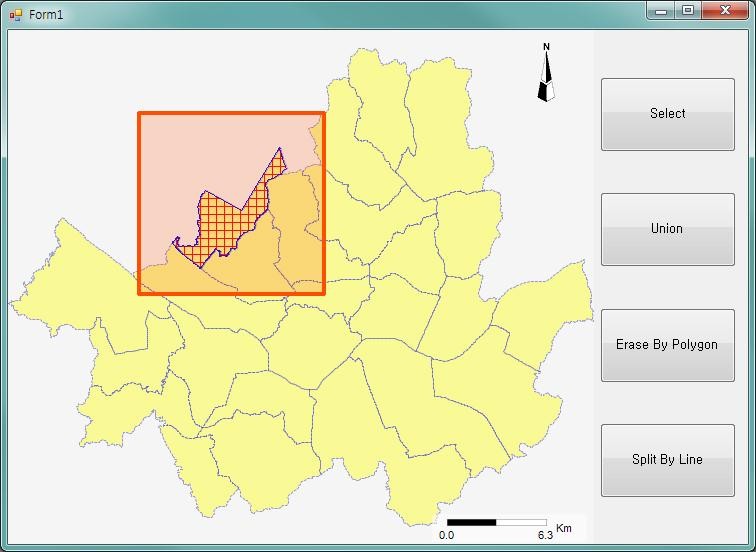
Union 수행 결과



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* EraseByPolygon 연산

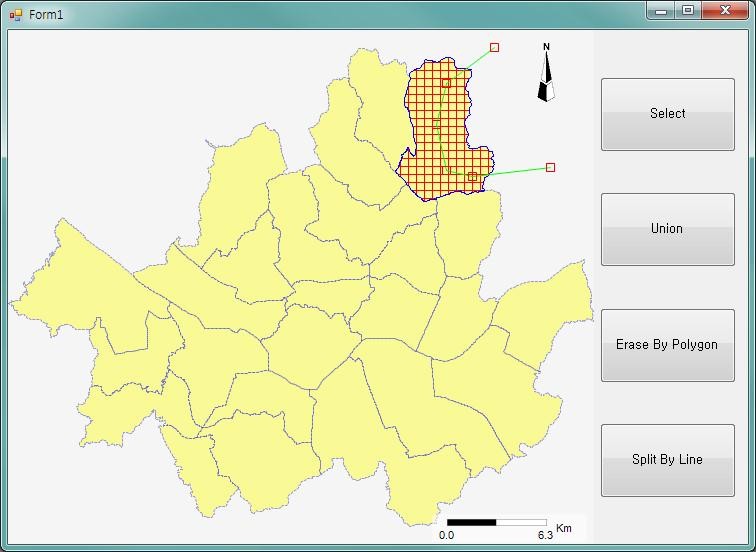
데이터 입력



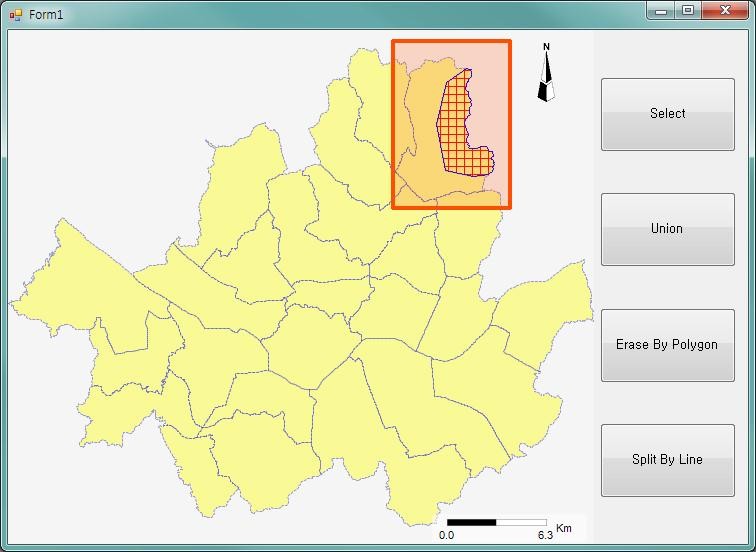
EraseByPolygon 수행 결과



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* SplitByLine 연산

데이터 입력



SplitByLine 수행 결과



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* 1. Geometry 공통 공간연산 - Buffering Geometry 공통 공간연산기능

이번 예제에서는 Geometry 공통 공간연산 메소드인 Buffer와 AfterDraw, CustomToolLButtonUp 이벤트를 활용하여 객체를 선택하고 선택객체에서 일정 범위를 보여주는 Buffer 공간연산기능을 구현해보는 예제이다. 본 예제에서는 ELE\_SCHOOL 포인트 개체를 GMapView 상에 원하는 심볼로 나타낸 후에 사용 자가 개체를 선택하고 대상 개체에 대하여 Buffer 공간 연산기능을 수행하는 과 정을 보여줄 것이다.

다음 코드는 폼을 생성한 후에 연결된 데이터베이스에서 ELE\_SCHOOL 레이어 를 GMapView상으로 불러오고 원하는 심볼로 표출하는 내용을 보여준다.

[예제 8-2-1]

public static GConnections GConns = new GConnections();

// 선택 객체의 레코드 정보 저장을 위한 전역 변수 선언 GRecordset selectedRS;

// 버퍼링 영역 저장을 위한 전역 변수 선언 GPolygon bufPoly;

// getRGB 함수

int getRGB(int R, int G, int B)

{

return Color.FromArgb(255, R, G, B).ToArgb();

}



 IntraMap/Objects ***PROGRAMMERS GUIDE***

public void Form1\_Load(object sender, EventArgs e)

{

// Form\_Load 부분 설정

GConnection GConn = GConns.NewConnection("MSACCESS"); GMapDatasets mDatasets;

GMapDataset mDataset; GRecordLayer RLayer; GSimpleRenderer SRender; GSymbol Sym = new GSymbol();

string sPath = Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles); GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb";

GConn.Connect(); GConns.Add(GConn);

mDatasets = GConn.GetMapDatasets();

// ELE\_SCHOOL의 MapDataset 정보 반환

mDataset = mDatasets.GetMapDataset("ELE\_SCHOOL");

// 레이어 정보 반환

eGeometryTypes a = mDataset.Type; RLayer.Visible = true;

SRender = (GSimpleRenderer)RLayer.Renderer; Sym = SRender.symbol;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

// 아이콘 타입 설정

Sym.Icon.Type = eIconTypes.itRectanlge; Sym.Icon.Size = 10;

Sym.Icon.Color = Color.Blue.ToArgb();

// GMapView에 ELE\_SCHOOL 레이어 추가

axGMapView1.Layers.Add((GLayer)RLayer);

axGMapView1.BackgroundColor = Color.Ivory.ToArgb(); axGMapView1.Fit();

}

위 코드를 수행하면 GMapView 상에 ELE\_SCHOOL 레이어의 포인트 데이터들 은 녹색 직사각형으로 표현된다. 공간연산 수행을 위해서는 공간연산 대상 데이 터를 선택한 후에 공간연산 결과 객체를 그리는 작업이 필요하다. 다음의 코드 는 데이터의 선택을 위한 마우스 모드의 설정과 선택 객체의 정보를 얻어오는 부분이다.

[예제 8-2-2]

// 객체 선택을 위한 하이라이트 버튼

private void cmdSelect\_Click(object sender, EventArgs e)

{

// 하이라이트 가능 여부 설정

axGMapView1.HilightEnable = true;

// 사용자 모드로 마우스 모드 전환



 IntraMap/Objects ***PROGRAMMERS GUIDE***

axGMapView1.MouseMode = eMouseMode.mmCustom;

}

// 선택 객체의 정보 획득

private void axGMapView1\_CustomToolLButtonUp(object sender, AxIMapObjects3Lib.\_IGMapViewEvents\_CustomToolLButtonUpEvent e)

{

GMapDataset mdataset1;

// 선택 객체의 레이어 이름을 위한 변수

String strLayerName;

// 선택 객체의 아이디 저장을 위한 변수

int nXID;

axGMapView1.GetHilightedObjectInfo(out strLayerName, out nXID);

mdataset1 = GConns.Item(0).GetMapDatasets().GetMapDataset(strLayerName);

// 선택 객체 정보를 레코드셋 변수에 저장

selectedRS = mdataset1.GetRecords("XID = " + nXID.ToString(), eSpatialQueryTypes.sqtNone, "");

selectedRS.MoveFirst();

axGMapView1.StartEditObject((Geometry)selectedRS.Fields.Item ("XGEOMETRY").Value);



 IntraMap/Objects ***PROGRAMMERS GUIDE***

// 일반 모드로 마우스 모드 전환

axGMapView1.MouseMode = eMouseMode.mmNormal;

}

위의 코드를 통하여 공간연산을 수행할 객체를 선택할 수 있게되며, 선택된 객체의 정보는 GetHilightedObjectInfo 이벤트를 통하여 획득된다. 선택된 객체 의 레코드 정보는 GRecordset 변수에 저장되며, 이 정보 중 일부는 10-3의 예 제를 통하여 GPoint 변수에 저장이 될 것이다.

다음 예제는 선택된 객체의 레코드 정보를 GPoint 변수에 저장하는 부분과 이를 바탕으로 Buffering을 수행하는 내용을 보여준다. 또한, Buffering 결과로 생성되는 GPolygon 객체의 심볼을 설정하는 내용도 살펴볼 수 있다.

[예제 8-2-3]

// 선택 객체를 바탕으로 Buffering 연산 수행

private void cmdBuffer\_Click(object sender, EventArgs e)

{

try

{

// 선택 객체 저장을 위해 GPoint 변수 선언

GPoint selectedPoint;

// GRecordset 정보를 GPoint 변수에 저장 selectedPoint = (GPoint)selectedRS.Fields.Item("XGEOMETRY").Value;

// Buffering 연산 수행



 IntraMap/Objects ***PROGRAMMERS GUIDE***

bufPoly = selectedPoint.Buffer(1000);

// Buffering 결과 반영을 위해 지도 다시 그림

axGMapView1.Redraw();

}

Catch( Exception err )

{ MessageBox.Show(err,Message);

}

}

// Buffering 결과 GPolygon의 심볼 설정

private void axGMapView1\_AfterDraw(object sender, AxIMapObjects3Lib.\_IGMapViewEvents\_AfterDrawEvent e)

{

// Buffering 결과 표출을 위한 심볼의 속성 설정

GSymbol Sym = new GSymbol(); Sym.Pen.Color = getRGB(255, 0, 255);

Sym.Pen.Width = 1;

Sym.Brush.Color = getRGB(255, 0, 255); Sym.Brush.Type = eBrushTypes.ftHatch; Sym.Brush.Hatch = eHatchTypes.htCross;

// DrawSymbol과 심볼 속성 설정값과의 연결

axGMapView1.SetDrawSymbol(Sym);



 IntraMap/Objects ***PROGRAMMERS GUIDE***

if (bufPoly ==null)

{

}

else

{

// Buffering 연산 결과 그리기

axGMapView1.DrawGeometry((Geometry)bufPoly);

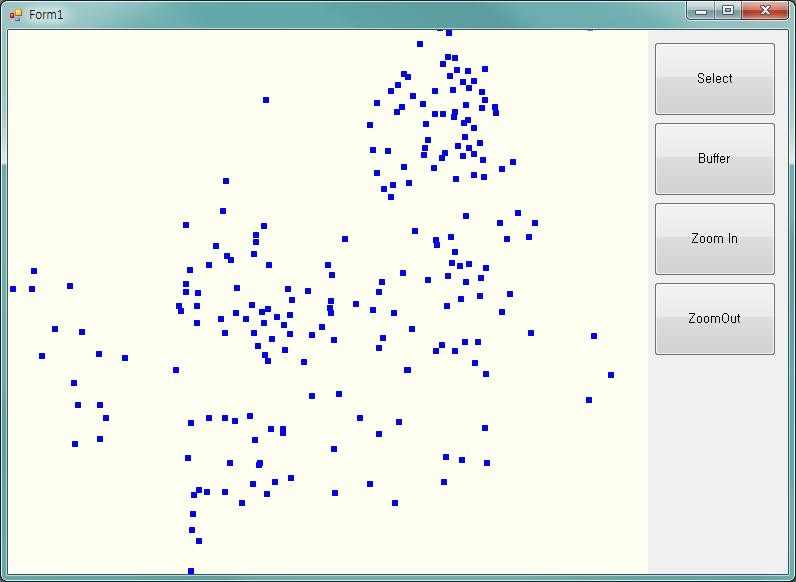
}

}

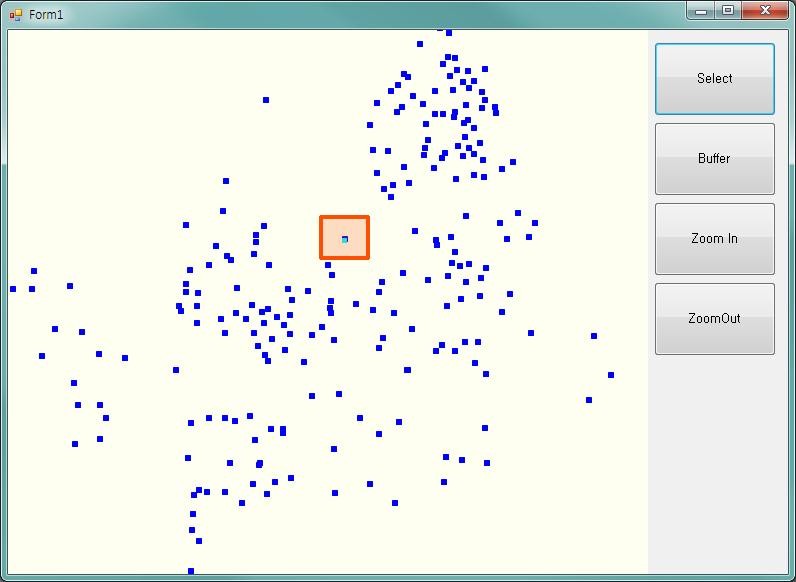
위 코드는 공간연산 실행 후, axGMapView1.Redraw() 에 의하여 수행된다. 다 음은 공간 객체를 선택한 후에 Buffering 공간연산이 실행되는 과정을 보여주는 그림이다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***



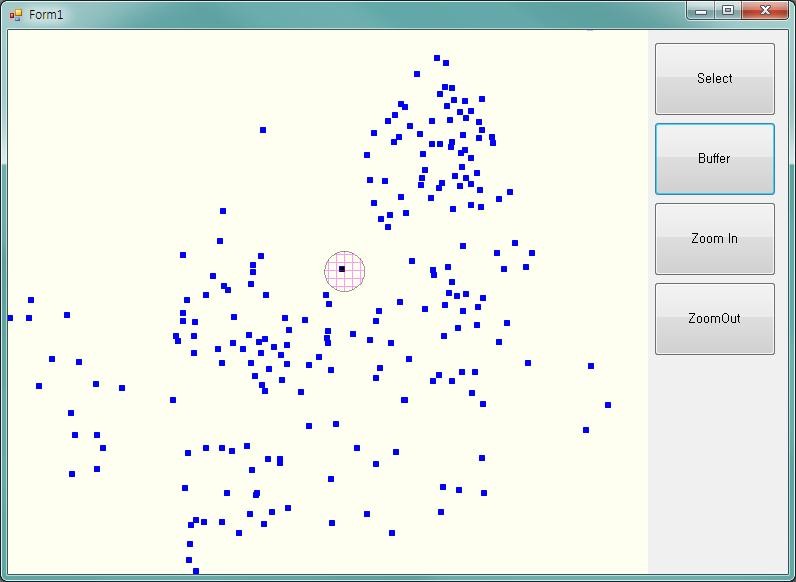
ELE\_SCHOOL 레이어

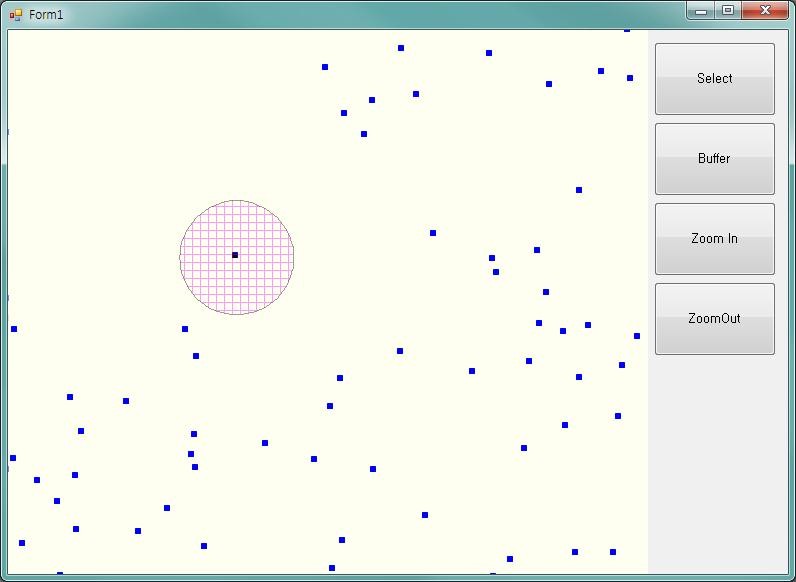


객체 선택



 IntraMap/Objects ***PROGRAMMERS GUIDE***





Buffering 연산 수행 결과



 IntraMap/Objects ***PROGRAMMERS GUIDE***

# 공간 데이터의 편집

실세계의 도로, 건물 등은 시간에 흐름에 따라 건설 또는 확장으로 인하여 많 은 변화가 일어난다. 실세계를 모델링한 GIS는 이러한 변화에 따라 삽입, 삭제, 수정등을 통하여 최신의 데이터를 유지할 수 있어야 한다.

IntraMap Object 3.0은 이러한 공간 데이터를 개발자 및 사용자가 쉽게 편집할 수 있도록 다양한 기능들을 제공한다.

1. 새로운 공간 객체의 추가

다음 예제는 db.mdb의 APART 레이어에 대하여 새로운 폴리곤 객체를 추가하 는 예제이다. 새로운 객체를 GMapView상에 그리고 Insert를 수행하면 테이블에 Geometry 정보가 추가된다.

다음은 공간 객체 편집을 위해 db.mdb 파일내의 APART 폴리곤 레이어를 디 스플레이하는 코드이다.

[예제 9-1]

// 전역 변수 선언

public static GConnections GConns = new GConnections(); GRecordset selectedRS;

// RGB색상 설정을 위한 getRGB 함수 선언 public static int getRGB(int R, int G, int B)

{ return Color.FromArgb(150, R, G, B).ToArgb();

}



 IntraMap/Objects ***PROGRAMMERS GUIDE***

private void Form1\_Load(object sender, EventArgs e)

{

// Form\_ Load 부분 설정

GConnection GConn = GConns.NewConnection("MSACCESS"); GMapDatasets mdatasets;

GConn.User = ""; GConn.Password = "";

string sPath = Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles);

GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb"; GConn.Connect();

GConns.Add(GConn);

mdatasets = GConns.Item(0).GetMapDatasets(); GRecordLayer RLayer =

GConns.Item(0).GetMapDatasets().GetMapDataset("APART").CreateLayer();

GSimpleRenderer Render = (GSimpleRenderer)RLayer.Renderer; GSymbol Sym = Render.symbol;

Sym.Pen.Color = getRGB(0, 255, 0);

Sym.Pen.Width = 1;

Sym.Pen.Type = ePenTypes.ltSolid; Sym.Brush.Color = getRGB(0, 255, 0);



 IntraMap/Objects ***PROGRAMMERS GUIDE***

Sym.Brush.Type = eBrushTypes.ftSolid; axGMapView1.Layers.Add((GLayer)RLayer);

axGMapView1.DirectionSymbolUse = true; axGMapView1.DirectionSymbolIndex = 2; axGMapView1.ScaleBarUse = true; axGMapView1.BackgroundColor = getRGB(225,255,255); axGMapView1.Fit();

}

 IntraMap/Objects ***PROGRAMMERS GUIDE***



위 그림은 설정한 심볼로 디스플레이된 APART 레이어이다. 다음은 이 레이어 에 새로운 아파트 공간 개체를 추가하는 과정으로 GMapView에 폴리곤 개체를 추가하기 위하여 마우스 모드를 mmDrawPolygon로 변경한다.

[예제 9-2]

private void cmdDrawPolygon\_Click(object sender, EventArgs e)

{

// 마우스 모드를 폴리곤 그리기로 설정

axGMapView1.MouseMode = eMouseMode.mmDrawPolygon;

}

편집 모드의 상수는 다음과 같다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***

* eEditModes

|  |  |
| --- | --- |
| Type | Description |
| emNone | 편집 모드가 아닙니다. |
| emObject | 개체단위 편집모드 입니다. |

다음은 사용자가 그린 폴리곤을 편집하기 위한 버텍스 관련 코드이다.

[예제 9-3]

private void cmdMoveVertex\_Click(object sender, EventArgs e)

{

// 편집 모드 판별

if (axGMapView1.EditMode.Equals(eEditModes.emNone) == true)

{

return;

}

// 마우스 모드를 버텍스 이동으로 설정

axGMapView1.MouseMode = eMouseMode.mmEditVertexMove;

}



 IntraMap/Objects ***PROGRAMMERS GUIDE***

private void cmdRemoveVertex\_Click(object sender, EventArgs e)

{

// 편집 모드 판별

if (axGMapView1.EditMode.Equals(eEditModes.emNone) == true)

{

return;

}

// 마우스 모드를 버텍스 삭제로 설정

axGMapView1.MouseMode = eMouseMode.mmEditVertexRemove;

}

다음은 기존의 공간 객체 편집을 저장하기나 사용자가 그린 공간 객체를 추가 하는 코드이다.

[예제 9-4]

private void cmdEditFeature\_Click(object sender, EventArgs e)

{

// 편집 객체의 Recordset 정보 저장을 위한 변수 선언

GRecordset editRS;

// 편집 모드 판별



 IntraMap/Objects ***PROGRAMMERS GUIDE***

if (axGMapView1.EditMode.Equals(eEditModes.emNone) == true)

{

return;

}

// 편집 대상 레이어의 레코드셋 정보 반환

editRS = GConns.Item(0).GetMapDatasets().GetMapDataset("APART").GetRecords( "", eSpatialQueryTypes.sqtNone, "");

// Recordset 편집 선언

editRS.Edit();

// 편집 대상 객체의 geometry 정보 추가 editRS.Fields.Item("XGEOMETRY").Value = axGMapView1.EndEditObject();

// 레코드셋 업데이트

editRS.Update();

// 편집 결과 반영을 위해 지도 다시 그림

axGMapView1.Redraw();

}

private void cmdAddFeature\_Click(object sender, EventArgs e)

{

GRecordset editRS;

 IntraMap/Objects ***PROGRAMMERS GUIDE***

if (axGMapView1.EditMode.Equals(eEditModes.emNone) == true)

{

return;

}

editRS = GConns.Item(0).GetMapDatasets().GetMapDataset("APART").GetRecords( "", eSpatialQueryTypes.sqtNone, "");

// Recordset 추가 선언

editRS.AddNew();

// 추가된 객체의 geometry 정보 추가 editRS.Fields.Item("XGEOMETRY").Value = axGMapView1.EndEditObject();

// 레코드셋 업데이트

editRS.Update();

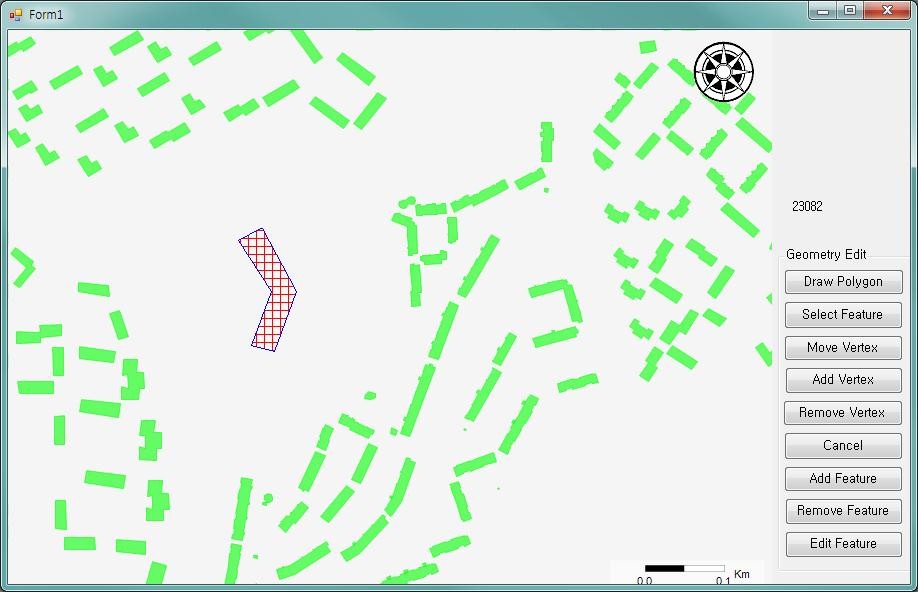
// 편집 결과 반영을 위해 지도 다시 그림

axGMapView1.Redraw();

}

위 예제를 수행하면 아래의 그림과 같이 기존의 공간 객체를 편집하거나, 새 로운 공간 객체를 추가할 수 있다.





IntraMap/Objects

***PROGRAMMERS GUIDE***





 IntraMap/Objects ***PROGRAMMERS GUIDE***

1. 공간 개체의 삭제

다음은 사용자가 선택한 객체를 삭제하는 예제이다. 여기에서는 사용자가 마우 스 이용하여 삭제하고자 하는 객체를 식별하기 위하여 GRecordLayer의 HilightEnable 프라퍼티와 GMapView의 GetHilightedObjectInfo 메소드를 활용한 다. 그리고 마우스 모드를 사용자 모드로 설정하고, 이에 대한 이벤트를 설정하 여 하이라이트된 객체를 삭제하면 된다.

[예제 9-5]

private void cmdSelect\_Click (object sender, EventArgs e)

{

// 하이라이트 가능 여부 설정

axGMapView1.HilightEnable = true;

// 마우스 모드를 사용자 모드로 설정

axGMapView1.MouseMode = eMouseMode.mmCustom;

}



 IntraMap/Objects ***PROGRAMMERS GUIDE***

다음은 하이라이트된 객체를 선택시 발생하는 이벤트이다.

[예제 9-6]

private void axGMapView1\_CustomToolLButtonUp(object sender, AxIMapObjects3Lib.\_IGMapViewEvents\_CustomToolLButtonUpEvent e)

{

GMapDataset mdataset;

// 레이어 이름 저장을 위한 변수 선언

String strLayerName;

// 선택 객체의 아이디 저장을 위한 변수 선언

int nXID;

// 하이라이트된 객체의 데이터셋명과 ID를 반환

axGMapView1.GetHilightedObjectInfo(out strLayerName, out nXID);

mdataset = GConns.Item(0).GetMapDatasets().GetMapDataset(strLayerName);

// 선택 객체의 레코드 정보를 반환

selectedRS = mdataset.GetRecords("XID = " + nXID.ToString(), eSpatialQueryTypes.sqtNone, "");

selectedRS.MoveFirst();



 IntraMap/Objects ***PROGRAMMERS GUIDE***

// 선택 객체의 편집 시작 선언 axGMapView1.StartEditObject((Geometry)selectedRS.Fields.Item("X GEOMETRY").Value);

}

private void cmdRemoveFeature\_Click(object sender, EventArgs e)

{

// 편집모드 판별

if (axGMapView1.EditMode.Equals(eEditModes.emNone) == true)

{

return;

}

// 선택 객체 삭제

selectedRS.Delete();

// 객체 편집 종료 선언

axGMapView1.EndEditObject();

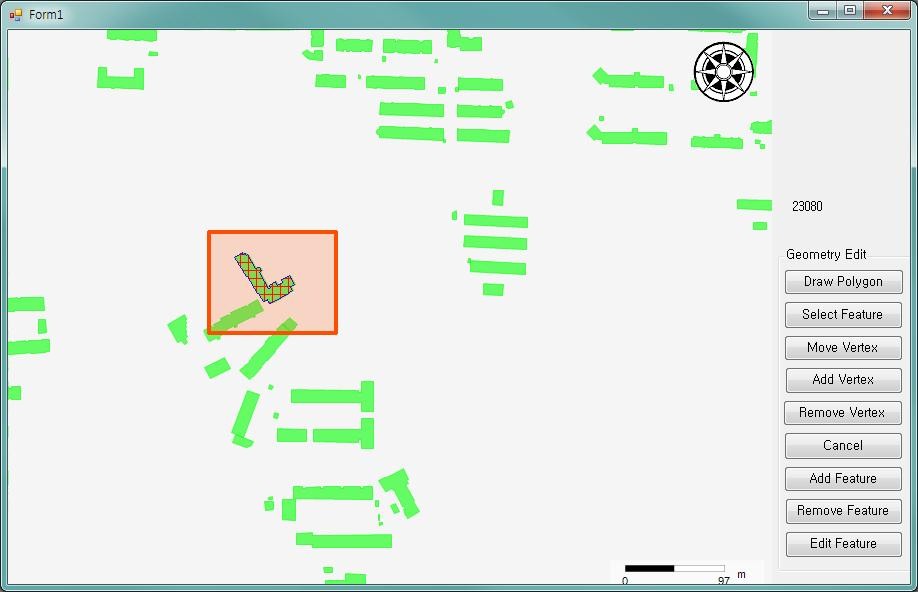
// 편집 결과를 반영하기 위해 지도 다시 그림

axGMapView1.Redraw();

selectedRS = null;

}





IntraMap/Objects

***PROGRAMMERS GUIDE***





 IntraMap/Objects ***PROGRAMMERS GUIDE***

# RASTER 이미지 편집

IntraMap/Objects 3.0은 Raster 이미지 데이터 포맷 jpeg, bmp, png 파일을 open하여 이동, 회전, 스케일 변환 기능을 제공한다.

Raster 이미지 열기 및 편집

다음 예제는 db.mdb의 ADMIN\_BOUNDARY 레이어에 새로운 Raster 이미지 파일을 추가하여 편집하는 예제이다. 본 예제에서는 사용자가 임의로 Raster 이 미지 파일을 OpenFileDialog를 이용하여 GMapView 상에 불러온 후 마우스 모 드를 통하여 이동/회전/스케일 변환 기능을 수행할 수 있다.

[예제 10]

public partial class Form1 : Form

{

//이미지 편집 모드 enum eTMmode

{

eNone, eMove, eRotate, eScale,

}

public static GConnections GConns = new GConnections(); public static GImagineLayer oImage = new GImagineLayer();

Double mMMMapX1; Double mMMMapY1;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

Double dCenterX; Double dCenterY; eTMmode mmode;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

GConnection GConn = GConns.NewConnection("MSACCESS"); GMapDatasets mDatasets;

GMapDataset mDataset; GRecordLayer RLayer; GSimpleRenderer SRender; GSymbol Sym;

string sPath = Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles); GConn.Database = sPath + "\\KSIC\\IntraMapObjects3\\db.mdb";

GConn.Connect(); GConns.Add(GConn);

mDatasets = GConn.GetMapDatasets();

mDataset = mDatasets.GetMapDataset("ADMIN\_BOUNDARY"); RLayer = mDataset.CreateLayer();



 IntraMap/Objects ***PROGRAMMERS GUIDE***

RLayer.Visible = true;

SRender = (GSimpleRenderer)RLayer.Renderer; Sym = SRender.symbol;

Sym.Brush.Type = eBrushTypes.ftNull; Sym.Pen.Type = ePenTypes.ltDash; Sym.Pen.Color = getRGB(0, 0, 255);

Sym.Pen.Width = 1;

axGMapView1.Layers.Add((GLayer)RLayer); axGMapView1.ScaleBarUse = true; axGMapView1.BackgroundColor = Color.Ivory.ToArgb();

mmode = eTMmode.eNone; axGMapView1.Fit();

}

public static int getRGB(int R, int G, int B)

{

return Color.FromArgb(150, R, G, B).ToArgb();

}

public static int getRGB(int A, int R, int G, int B)

{

return Color.FromArgb(A, R, G, B).ToArgb();

}



 IntraMap/Objects ***PROGRAMMERS GUIDE***

//파일 오픈

private void button\_RasterOpen\_Click(object sender, EventArgs e)

{

OpenFileDialog oDlg = new OpenFileDialog();

oDlg.Filter = "Raster Files(.bmp)|\*.bmp|Raster Files(.jpg)|\*.jpg|Raster Files(.png)|\*.png"; oDlg.FilterIndex = 1;

if (oDlg.ShowDialog() == DialogResult.OK)

{

axGMapView1.Layers.Remove(1); oImage.Open(oDlg.FileName); GRectangle rt = new GRectangle();

rt.Left = 189119.0;

rt.Right = 206171.0;

rt.Top = 456574.0;

rt.Bottom = 446241.0;

oImage.SetArea(rt.Left, rt.Bottom, rt.Right, rt.Top); axGMapView1.Layers.Add((GLayer)oImage); axGMapView1.Redraw();

}

}

 IntraMap/Objects ***PROGRAMMERS GUIDE***

//이미지 이동

private void button\_Move\_Click(object sender, EventArgs e)

{

axGMapView1.MouseMode = eMouseMode.mmCustom; axGMapView1.HilightEnable = false;

mmode = eTMmode.eMove;

}

//이미지 회전

private void button\_Rotate\_Click(object sender, EventArgs e)

{

axGMapView1.MouseMode = eMouseMode.mmCustom; axGMapView1.HilightEnable = false;

mmode = eTMmode.eRotate;

}

//이미지 스케일

private void button\_Scale\_Click(object sender, EventArgs e)

{

axGMapView1.MouseMode = eMouseMode.mmCustom; axGMapView1.HilightEnable = false;

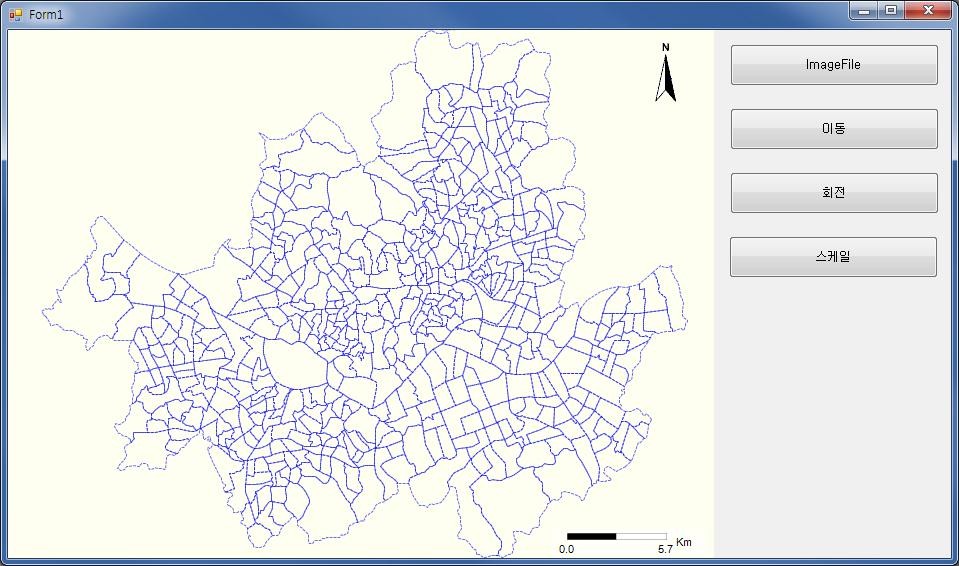
mmode = eTMmode.eScale;

}

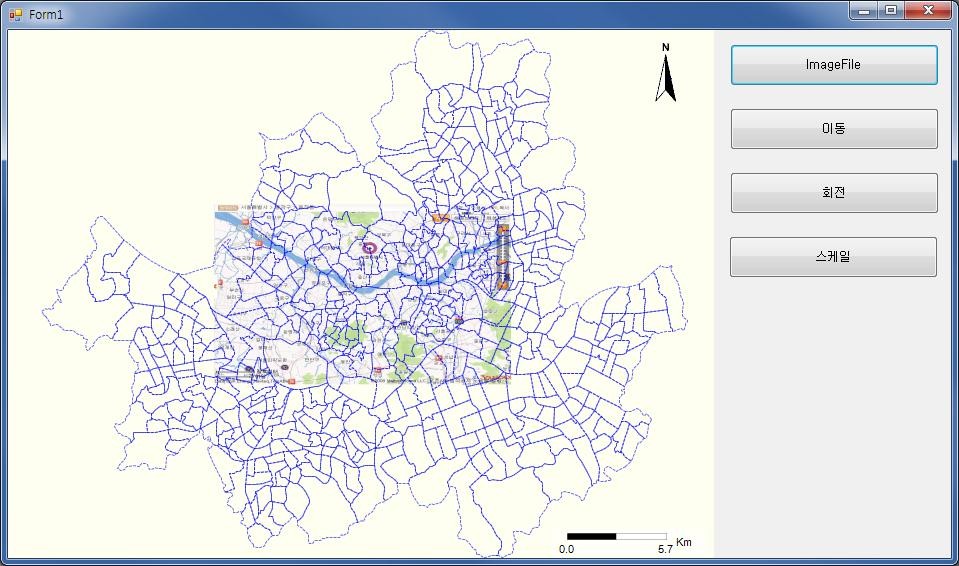
위의 예제에서 PrintMap에 대한 결과 그림은 다음과 같다.



 IntraMap/Objects ***PROGRAMMERS GUIDE***



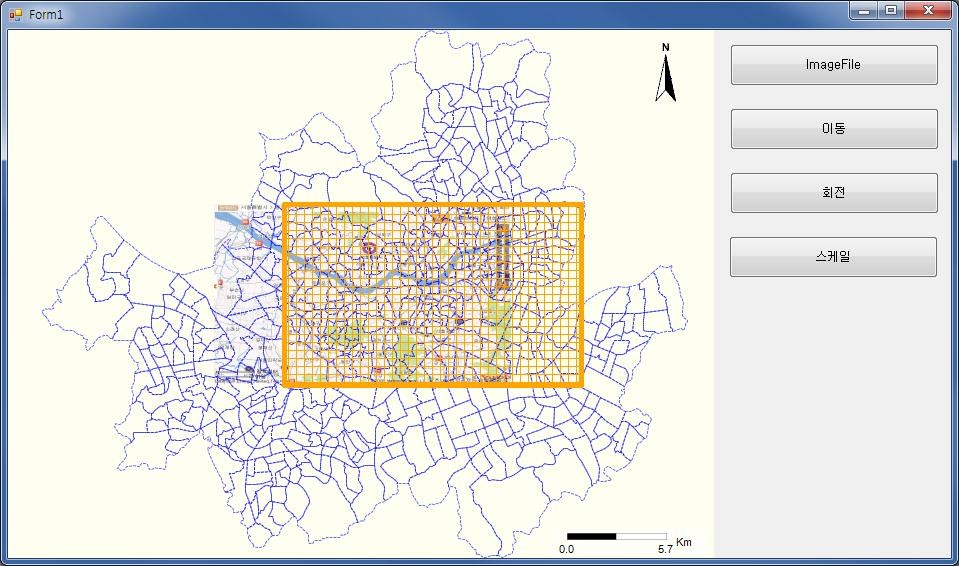
기본 화면



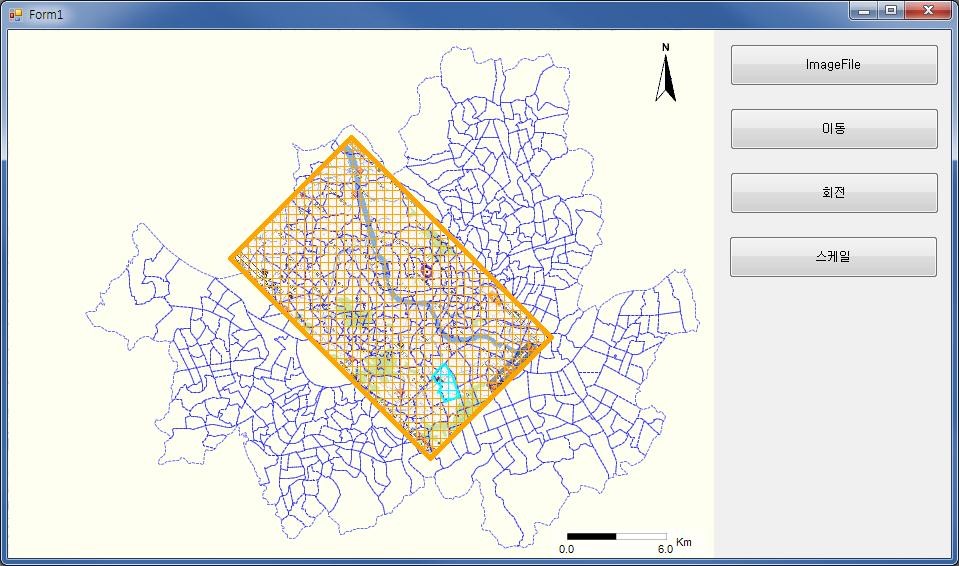
Raster 이미지 불러오기



 IntraMap/Objects ***PROGRAMMERS GUIDE***



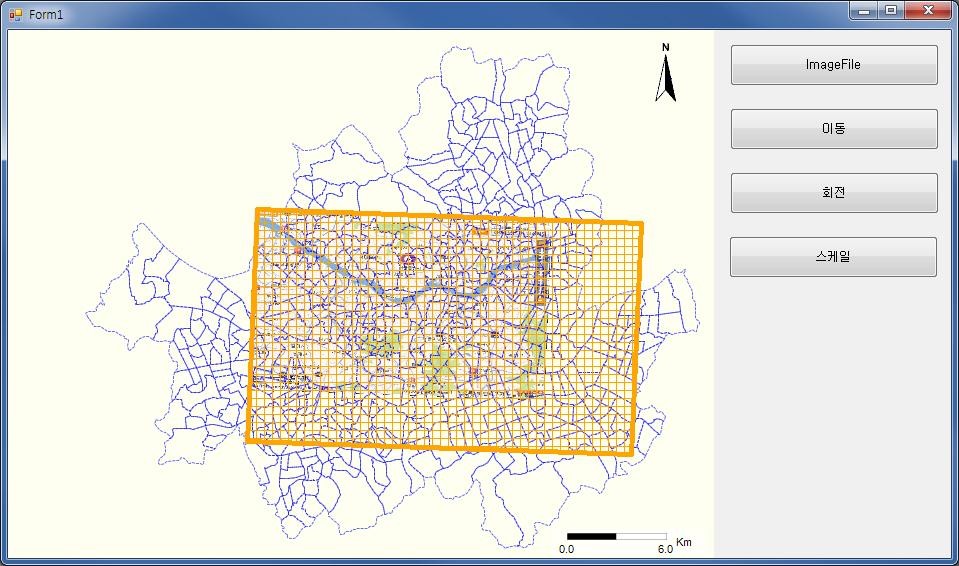
Raster 이미지 이동



Raster 이미지 회전



 IntraMap/Objects ***PROGRAMMERS GUIDE***



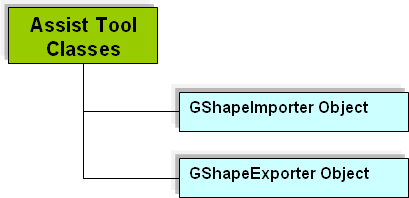
Raster 크기 변환



 IntraMap/Objects ***PROGRAMMERS GUIDE***

# Shape 파일 가져오기/내보내기

IntraMap/Objects 3.0은 ESRI사의 공간 데이터 포맷인 Shape 파일을 GShapeImporter와 GShapeExporter Object를 통하여 Import, Export 할 수 있는 기능을 제공한다.





 IntraMap/Objects ***PROGRAMMERS GUIDE*** 다음은 GShapeImporter와 GShapeExporter가 제공하는 프라퍼티와 메소드를 나타낸다.

* GShapeImporter

Property

|  |  |
| --- | --- |
| Name | Description |
| Connection | Shape 파일 가져오기를 위한 데이터베이스 연결을 설  정하거나 반환합니다. |
| CurrentPosition | 가져오기 상태를 알려주는 값을 설정하거나 반환합니다. |
| desc | XMETALAYER에 들어갈 내용을 설정하거나 반환합니다. |
| DomainExtent | 가져오기를 위한 Shape 파일의 도메인 영역을 설정하  거나 반환합니다. |
| DomainID | 가져오기를 위한 Shape 파일의 도메인 아이디를 반환합니다. |
| DomainName | 가져오기를 위한 Shape 파일의 도메인 이름을 설정하  거나 반환합니다. |
| FileName | 가져오기를 수행할 Shape 파일의 이름을 설정하거나  반환합니다. |
| geometryType | 지오메트리 타입을 반환합니다. |
| GridSize | 그리드인덱스의 그리드 크기를 설정하거나 반환합니다. |
| hMessageBox | 메시지박스 컨트롤의 핸들값을 설정합니다. |
| hProgressBar | 진행상태바 컨트롤의 핸들값을 설정합니다. |
| InitialGridSize | Shape 파일의 계산된 그리드 사이즈를 반환합니다. |
| NeedDomainInfo | 도메인 정보의 필요여부를 반환합니다. |
| ShapeExtent | 가져오기를 수행할 Shape 파일의 영역을 반환합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

|  |  |
| --- | --- |
| ShapeRecordCount | 가져오기를 수행할 Shape 파일의 레코드 개수을 반환  합니다. |
| TableName | 가져오기를 통해 생성될 테이블명을 설정하거나 반환  합니다. |
| UseExistMapDataset | 기존 맵데이터셋의 사용여부를 설정하거나 반환합니다. |

Method

|  |  |
| --- | --- |
| Name | Description |
| Execute | Shape Import 작업을 수행합니다. |

* GShapeExporter

Property

|  |  |
| --- | --- |
| Name | Description |
| Connection | Shape 내보내기를 위한 데이터베이스 연결을 설정하  거나 반환합니다. |
| FileName | 내보내기 작업을 통해 생성할 Shape 파일의 이름을  설정하거나 반환합니다. |
| Filter | 내보내기를 수행할 데이터셋의 필터조건을 설정하거  나 반환합니다. |
| TableName | Shape 파일로 내보내기 작업을 수행할 테이블명을 설  정하거나 반환합니다. |



 IntraMap/Objects ***PROGRAMMERS GUIDE***

Method

|  |  |
| --- | --- |
| Name | Description |
| Execute | Shape Export 작업을 수행합니다. |

Shape 파일 가져오기 (Import)

다음 예제는 연결된 데이터베이스로 GShapeImporter를 통하여 Shape 파일을 Import하는 예제이다. 본 예제에서는 사용자가 임의로 Shape 파일을 선택하여 ACCESS 내로 가져오거나, ACCESS 내의 Shape 파일을 특정 디렉토리로 내보내 는 기능을 구현하였다.

예제에서 가져오기/내보내기의 실행은 Import 버튼, Export 버튼이 클릭되었을 때 수행되도록 하였다.

[예제 11-1]

private void cmdImport\_Click(object sender, EventArgs e)

{

try

{

GConnection GConn; GMapDatasets mDatasets; GMapDataset mDataset; GRecordLayer RLayer;

GConn = (GConnectionClass)GConns.NewConnection("MSACCESS");



 IntraMap/Objects ***PROGRAMMERS GUIDE***

GConn.User = ""; GConn.Password = "";

string sPath = Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles);

GConn.Database = Application.StartupPath + "\\..\\..\\Shape Files\\qdb.mdb";

if (System.IO.File.Exists(GConn.Database) == true)

{ System.IO.File.Delete(GConn.Database);

}

CreateDatabase(GConn.Database);

GConn.Connect(); GConns.Add(GConn);

string[] stringArray = new string[] { "APART" }; for (int i = 0; i < stringArray.Length; ++i)

{

GShapeImporter sImporter = new GShapeImporter();

sImporter.Connection = GConns.Item(0); sImporter.TableName = stringArray[i];

sImporter.filename = Application.StartupPath + "\\..\\..\\Shape



 IntraMap/Objects ***PROGRAMMERS GUIDE***

Files\\" + stringArray[i] + ".shp"; sImporter.Connection.UpdateConnection(); sImporter.Execute();

sImporter = null;

}

for (int i = 0; i < GConn.GetMapDatasets().Count ; ++i)

{

mDatasets = GConn.GetMapDatasets(); mDataset = mDatasets.GetMapDataset(i); RLayer = mDataset.CreateLayer();

GSimpleRenderer oRender = (GSimpleRenderer)RLayer.Renderer; GSymbol oSym = oRender.symbol;

doCreateRndSymbol(oSym, mDataset.Type);

axGMapView1.Layers.Add((GLayer)RLayer);

}

axGMapView1.Fit();

}

catch(Exception err)

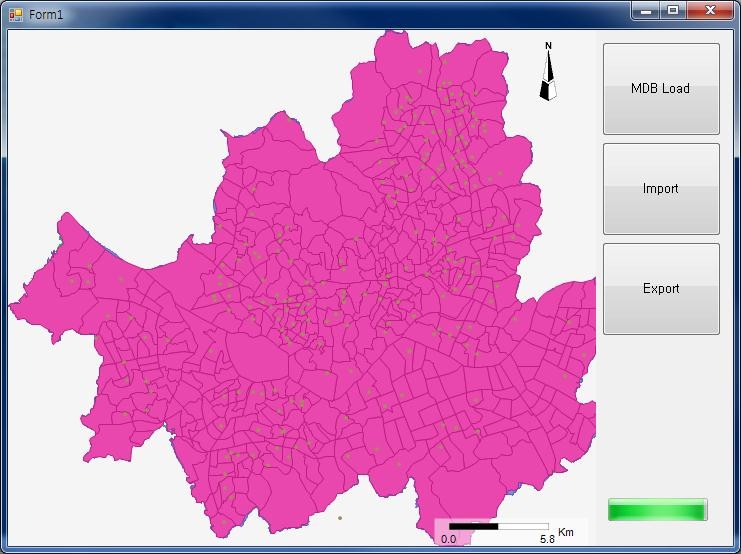
{ MessageBox.Show(err.Message);

}

}



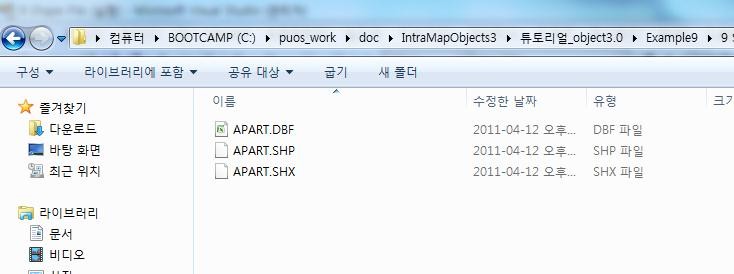




IntraMap/Objects

***PROGRAMMERS GUIDE***

데이터베이스 연결, 테이블 이름 등을 이용하여 가져오기를 수행할 Shape 파 일, 테이블명, 데이터베이스를 설정하고, Execute 메소드를 실행하면 가져오기가 수행된다. 다음은 예제에서 사용한 db.mdb에 APART.shp 파일의 가져오기가 수 행된 그림이다.





 IntraMap/Objects ***PROGRAMMERS GUIDE***

Shape 파일 내보내기 (Export)

다음 예제는 연결된 데이터베이스로 GShapeExporter를 통하여 Shape 파일을 내보내는 예제이다. 본 예제에서는 사용자가 설정한 데이터베이스 연결의 모든 Shape 파일을 내보낼 수 있도록 하였다.

[예제 11-2]

private void cmdExport\_Click(object sender, EventArgs e)

{

try

{

GShapeExporter sExporter = new GShapeExporter(); sExporter.Connection = GConns.Item(0); sExporter.TableName =

GConns.Item(0).GetMapDatasets().GetMapDataset("APART").Name; sExporter.filename = Application.StartupPath + "\\..\\..\\Shape Files\\"

+ GConns.Item(0).GetMapDatasets().GetMapDataset("APART").Name + ".shp";

sExporter.Execute();

}

catch(Exception err)

{ MessageBox.Show(err.Message);

}

MessageBox.Show("Export Completed!", Application.ProductName,

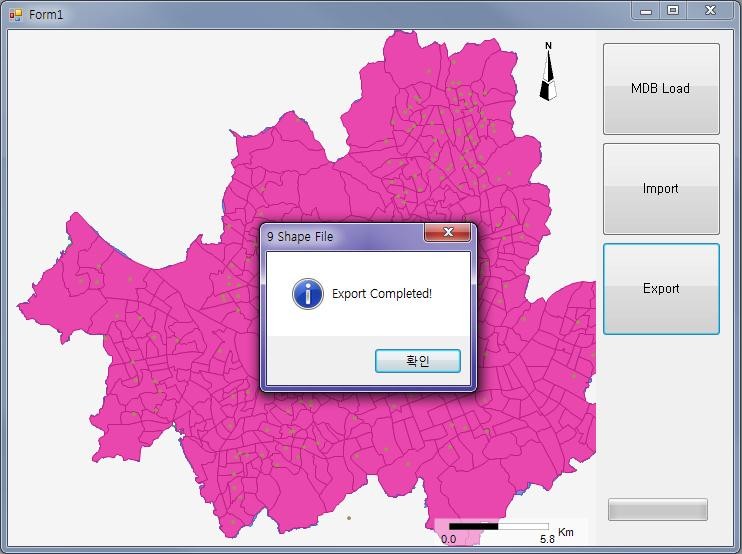


 IntraMap/Objects ***PROGRAMMERS GUIDE***

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

GShapeExporter의 파일 이름, 테이블 이름, 데이터베이스 연결 등의 설정을 통 하여 생성될 Shape 파일명과 내보내기가 수행될 테이블명 및 데이터베이스 연 결을 설정하고, Execute 메소드를 실행하면 내보내기가 수행된다.





 IntraMap/Objects ***PROGRAMMERS GUIDE***

# CAD 파일 가져오기/내보내기

IntraMap Object 3.0은 AutoCAD 社 의 데이터 포맷인 dwg(dxf) 파일을 GCadImporter와 GCadExporter Object를 통해서 Import, Export 할 수 있는 기 능을 제공한다.

CAD 파일 가져오기(Import)

다음 예제는 연결된 데이터베이스로 GCadImporter를 통하여 CAD 파일을 Import하는 예제이다. 본 예제에서는 사용자가 임의로 CAD 파일을 선택하여 ACCESS 내로 가져오는 기능을 구현하였다.

[예제 12-1]

namespace \_12\_1.DwgImport

{

public partial class Form1 : Form

{

public static GConnections gConns = new GConnections(); public static Random m\_objRnd = new Random(100); public static string RESOURCE\_PATH =

System.IO.Path.GetDirectoryName(Application.ExecutablePath) + "\\..\\..\\..\\..\\ResourceFile\\";

public string m\_strDwgTableName;

public Form1()

{



 IntraMap/Objects ***PROGRAMMERS GUIDE***

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

GConnection gConn = gConns.NewConnection("MSACCESS"); GMapDatasets gDataSets;

gConn.User = ""; gConn.Password = "";

gConn.Database = RESOURCE\_PATH + "db.mdb"; gConn.Connect();

gConns.Add(gConn);

gDataSets = gConns.Item(0).GetMapDatasets(); ConnectionDBToList(); axGMapView1.DirectionSymbolUse = true; axGMapView1.DirectionSymbolIndex = 3; axGMapView1.BackgroundColor = Color.Ivory.ToArgb(); axGMapView1.Fit();

}

//연결된 Connection에서 Table리스트를 뽑아낸다. public void ConnectionDBToList()

{

int iTableCount = gConns.Item(0).GetMapDatasets().Count; listBox\_LayerTable.Items.Clear();



 IntraMap/Objects ***PROGRAMMERS GUIDE***

for (int i = 0; i < iTableCount; i++)

{

GMapDataset gDTSet = gConns.Item(0).GetMapDatasets().GetMapDataset(i); listBox\_LayerTable.Items.Add(gDTSet.Name);

}

}

public static void doCreateRndSymbol(GSymbol objSymbol, eGeometryTypes nGeometryType)

{

int nR1 = getRnd(20, 220); int nG1 = getRnd(20, 220); int nB1 = getRnd(20, 220);

if (nR1 == -1 || nG1 == -1 || nB1 == -1)

{

return;

}

objSymbol.Brush.Type = eBrushTypes.ftSolid; objSymbol.Brush.Color = getRGB(255, nR1 + 20, nG1 + 20, nB1 + 20); objSymbol.Pen.Type = ePenTypes.ltSolid;

objSymbol.Pen.Width = 1;

objSymbol.Pen.Color = getRGB(255, nR1 - 20, nG1 - 20, nB1 - 20); if (nGeometryType == eGeometryTypes.gtPoint)

{

objSymbol.Icon.Type = eIconTypes.itCircle;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

objSymbol.Icon.Size = 3;

objSymbol.Icon.Color = getRGB(255, nR1, nG1, nB1);

}

}

private void button\_OpenDwg\_Click(object sender, EventArgs e)

{

OpenFileDialog cDialog = new OpenFileDialog(); cDialog.Filter = "Dwg Files(\*.dwg, \*.dxf) | \*dwg; \*.dxf"; Uri cDwgPath = new Uri(RESOURCE\_PATH); cDialog.InitialDirectory = cDwgPath.LocalPath;

if(cDialog.ShowDialog() == DialogResult.OK)

{

textBox\_DwgName.Text = cDialog.FileName;

}

}

private void button1\_Click(object sender, EventArgs e)

{

int iSelectList = listBox\_LayerTable.SelectedIndex;

if (iSelectList == -1)

{ return;

}

else



 IntraMap/Objects ***PROGRAMMERS GUIDE***

{

String strTableName = (string)listBox\_LayerTable.SelectedItem; GMapDataset gDTset = gConns.Item(0).GetMapDatasets().GetMapDataset(strTableName); GRecordLayer gLayer = gDTset.CreateLayer();

gLayer.Visible = true;

GSimpleRenderer oRenderer = (GSimpleRenderer)gLayer.Renderer; GSymbol oSym = oRenderer.symbol; doCreateRndSymbol(oSym, gDTset.Type); axGMapView1.Layers.Add((GLayer)gLayer); axGMapView1.Fit();

axGMapView1.Redraw();

}

}

//DwgImport

private void button\_DwgImport\_Click(object sender, EventArgs e)

{

String strDwgFileName = textBox\_DwgName.Text; GCADImporter cCadImporter = new GCADImporter(); cCadImporter.Connection = gConns.Item(0); cCadImporter.filename = strDwgFileName;

m\_strDwgTableName = System.IO.Path.GetFileNameWithoutExtension(strDwgFileName); int iDwgLayerCount = cCadImporter.LayerCount;

for( int i = 0; i < iDwgLayerCount; i++)

{



 IntraMap/Objects ***PROGRAMMERS GUIDE***

string strDwgObjectName = cCadImporter.GetLayerName(i); cCadImporter.Execute(i, m\_strDwgTableName); m\_strDwgTableName = cCadImporter.TableName;

}

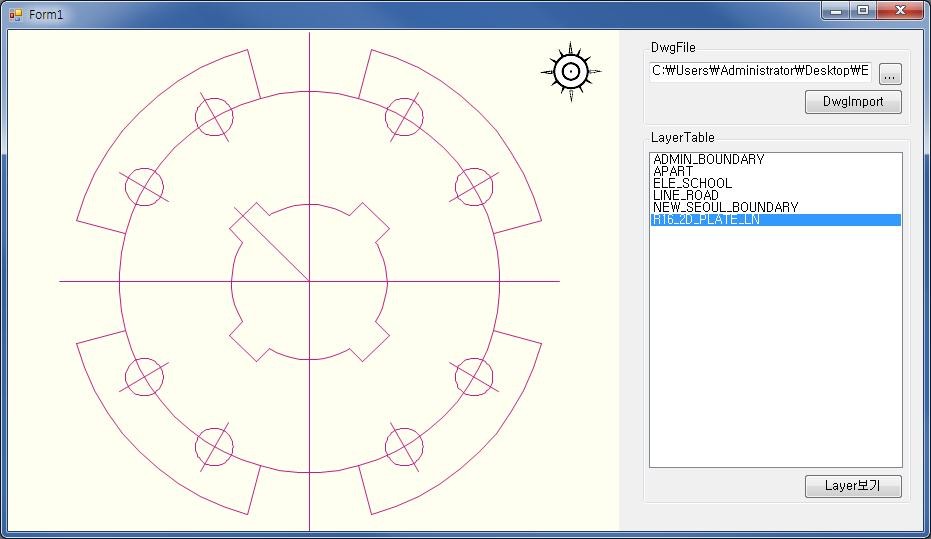
//Import한 테이블 보기 ConnectionDBToList();

}

}

}

위의 예제에서 샘플 예제에 포함된 r16\_2d\_plate.dwg 파일을 Import한 결과 그림은 다음과 같다.





 IntraMap/Objects ***PROGRAMMERS GUIDE***

CAD 파일 내보내기(Export)

다음 예제는 연결된 데이터베이스로 GCadExporter를 통하여 CAD 파일을 내 보내는는 예제이다. 본 예제에서는 CAD 파일을 선택하여 사용자가 설정한 데이 터베이스로 내보내는 기능을 구현하였다.

[예제 12-2]

//DwgExport

private void button\_DwgExport\_Click(object sender, EventArgs e)

{

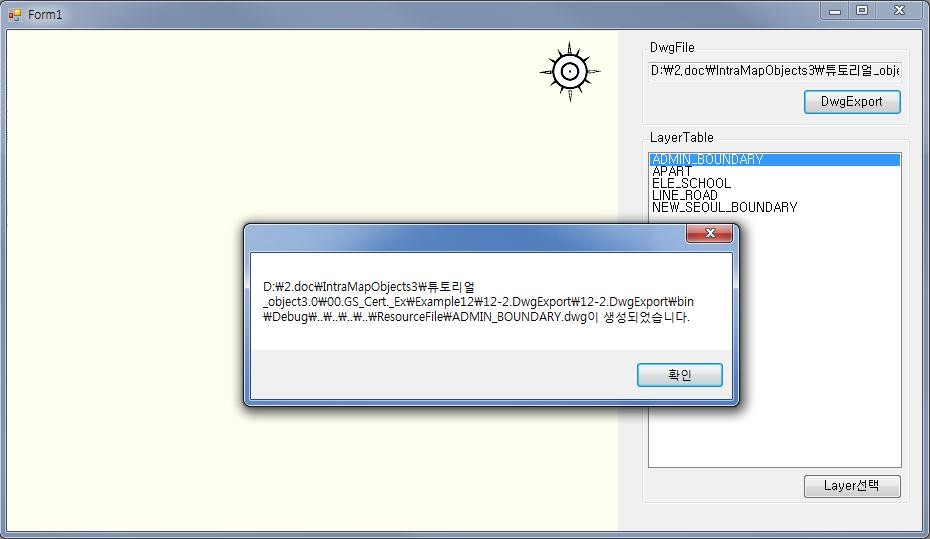
GCADExporter cadExporter = new GCADExporter(); cadExporter.filename = textBox\_DwgName.Text + ".dwg"; cadExporter.Connection = gConns.Item(0); cadExporter.TableName = (String)listBox\_LayerTable.SelectedItem; cadExporter.Execute();

string strDwgFileName = cadExporter.filename; MessageBox.Show(strDwgFileName + "이 생성되었습니다.");

}



 IntraMap/Objects ***PROGRAMMERS GUIDE*** 위의 예제에서 샘플 예제에 포함된 r16\_2d\_plate.dwg 파일을 Import한 결과 그림은 다음과 같다.





 IntraMap/Objects ***PROGRAMMERS GUIDE***

# 화면 출력

IntraMap Object 3.0은 GMapView 상에서 현재 디스플레이 된 지도 데이터의 영역을 선택하여 일반적인 화면 출력 및 A0 용지 사이즈 출력을 위한 대용량 이미지 파일로 출력하는 기능을 제공한다.

일반 화면 출력

이번 예제에서는 디스플레이 된 지도데이터를 그림파일로 출력하는 방법에 대하여 다룬다. 화면 출력 기능은 디스플레이된 지도의 영역을 선택하여 bmp 파일 형식으로 출력한다.

다음 코드는 폼을 생성한 후에 연결된 데이터베이스에서 ADMIN\_BOUNDARY 레이어를 GMapView 상으로 불러온 후, 출력할 영역을 설정하여 사용자가 출력 버튼을 클릭하면 설정된 디렉토리로 그림파일이 출력되는 것을 확인할 수 있다.

[예제 13-1]

//지정한 영역 선택 확인

private void axGMapView1\_AfterDraw(object sender, AxIMapObjects3Lib.\_IGMapViewEvents\_AfterDrawEvent e)

{

if(m\_LayerExtant != null)

{

GSymbol gSym = new GSymbol();

gSym.Brush.Type = eBrushTypes.ftHatch;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

gSym.Brush.Hatch = eHatchTypes.htCross; gSym.Brush.Color = Color.Yellow.ToArgb();

axGMapView1.SetDrawSymbol(gSym); axGMapView1.DrawGeometry((Geometry)m\_LayerExtant);

}

textBox\_DisplayScale.Text = (axGMapView1.DisplayScale \* 0.0001).ToString();

}

//일반 영역 출력

private void button\_Print\_Click(object sender, EventArgs e)

{

if (m\_LayerExtant == null)

{

MessageBox.Show("영역을 먼저 선택 해주세요!"); return;

}

GRectangle gRect = new GRectangle(); GPoints gPts = new GPoints();

Geometry gGeo = (Geometry)m\_LayerExtant;

gRect.Bottom = gGeo.Extent.Bottom; gRect.Left = gGeo.Extent.Left;



 IntraMap/Objects ***PROGRAMMERS GUIDE***

gRect.Right = gGeo.Extent.Right; gRect.Top = gGeo.Extent.Top;

GPoint gPt1 = new GPoint(); gPt1.x = gRect.Left;

gPt1.y = gRect.Top;

GPoint gPt2 = new GPoint(); gPt2.x = gRect.Right;

gPt2.y = gRect.Bottom;

double dPrintScale = Convert.ToDouble(textBox\_PrintScale.Text); String strBmpName = String.Format("{0}\\Test{1}\_{2}\_{3}.bmp",

RESOURCE\_PATH, textBox\_PrintScale.Text, textBox\_Width.Text,

textBox\_Height.Text);

int dWidth = Convert.ToInt32(textBox\_Width.Text); int dHeight = Convert.ToInt32(textBox\_Height.Text);

ScreenCapture oCapture = new ScreenCapture();

bool bCheck = oCapture.GMapViewRectToFile(axGMapView1, strBmpName, ImageFormat.Bmp, gPt1, gPt2, dWidth, dHeight, dPrintScale \* 10000);

MessageBox.Show("이미지가" + RESOURCE\_PATH + " 폴더에 저장되었습니다.");

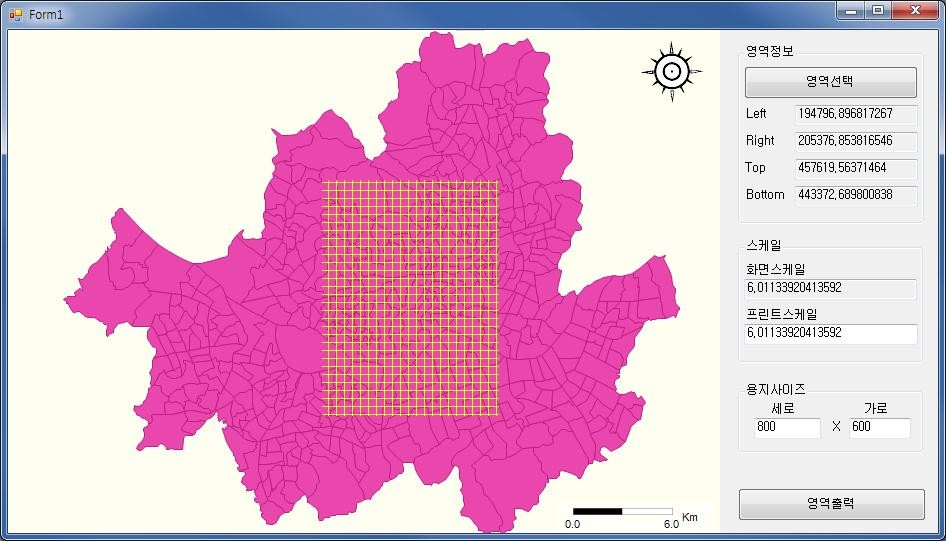


 IntraMap/Objects ***PROGRAMMERS GUIDE***

//선택영역 해제 axGMapView1.EndDrawGeometry(); m\_LayerExtant = null; axGMapView1.Redraw();

}

위의 예제에 대한 결과 그림은 다음과 같다.



일반 화면 출력



 IntraMap/Objects ***PROGRAMMERS GUIDE***

대용량 화면 출력

다음 코드는 ADMIN\_BOUNDARY 레이어 화면을 대용량(A0) 사이즈로 출력할 수 있는 예제이다.

[예제 13-2]

//지정한 영역 선택 확인

private void axGMapView1\_AfterDraw(object sender, AxIMapObjects3Lib.\_IGMapViewEvents\_AfterDrawEvent e)

{

if(m\_LayerExtant != null)

{

GSymbol gSym = new GSymbol(); gSym.Brush.Type = eBrushTypes.ftHatch; gSym.Brush.Hatch = eHatchTypes.htCross; gSym.Brush.Color = Color.Yellow.ToArgb(); axGMapView1.SetDrawSymbol(gSym);

axGMapView1.DrawGeometry((Geometry)m\_LayerExtant);

}

textBox\_DisplayScale.Text = (axGMapView1.DisplayScale \* 0.0001).ToString();

}

//대용량 영역 출력

private void button\_Print\_Click(object sender, EventArgs e)

{

if (m\_LayerExtant == null)

{

MessageBox.Show("영역을 먼저 선택해주세요");



 IntraMap/Objects ***PROGRAMMERS GUIDE***

return;

}

GRectangle gRect = new GRectangle(); GPoints gPts = new GPoints();

Geometry gGeo = (Geometry)m\_LayerExtant;

gRect.Bottom = gGeo.Extent.Bottom; gRect.Left = gGeo.Extent.Left; gRect.Right = gGeo.Extent.Right; gRect.Top = gGeo.Extent.Top;

GPoint gPt1 = new GPoint(); gPt1.x = gRect.Left;

gPt1.y = gRect.Top;

GPoint gPt2 = new GPoint(); gPt2.x = gRect.Right;

gPt2.y = gRect.Bottom;

ScreenCapture oCapture = new ScreenCapture(); oCapture.SetProgressBar(ref progressBar1);

double dPrintScale = Convert.ToDouble(textBox\_PrintScale.Text);

String strBmpName = String.Format("{0}\\Test{1}\_{2}\_{3}.bmp", RESOURCE\_PATH, textBox\_PrintScale.Text,



 IntraMap/Objects ***PROGRAMMERS GUIDE***

textBox\_Width.Text, textBox\_Height.Text); int dWidth = Convert.ToInt32(textBox\_Width.Text);

int dHeight = Convert.ToInt32(textBox\_Height.Text);

bool bCheck = oCapture.GMapViewRectToFile2(axGMapView1, strBmpName, ImageFormat.Bmp, gPt1, gPt2, dWidth, dHeight, dPrintScale \* 10000);

MessageBox.Show("이미지가" + RESOURCE\_PATH + " 폴더에 저장되었습니다.");

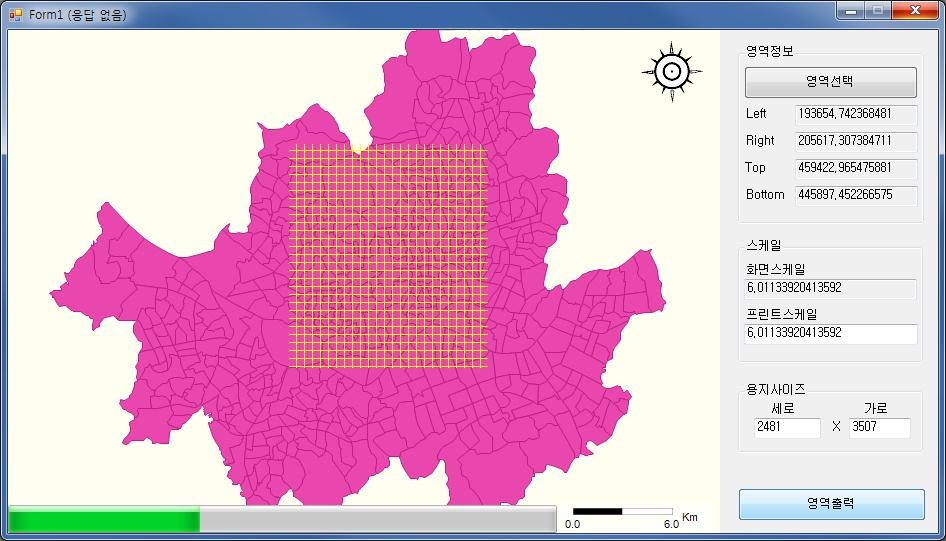
//선택영역 해제 axGMapView1.EndDrawGeometry(); m\_LayerExtant = null; axGMapView1.Redraw();

}



 IntraMap/Objects ***PROGRAMMERS GUIDE***

위의 예제에 대한 결과 그림은 다음과 같다.



대용량 화면 출력



 IntraMap/Objects ***PROGRAMMERS GUIDE***

# 색인

Ａ

AfterDraw 이벤트............................................................................................................................................ 92, 97

Ｂ

Buffer 메소드 120

Ｄ

DrawFinishedGeometry 이벤트..............................................................................................................90, 111

Ｅ

eEditModes 133

eMouseMode Constants 78

Ｆ

FromMapXY 41



IntraMap/Objects

***PROGRAMMERS GUIDE***

Ｇ

GBrush 47

GBuffer 49

GConnection 9

GConnection 정보 얻어오기 14

GConnections 8

Geometry 85

Geometry 간 공간연산 100

Geometry 공통 연산함수 101

Geometry에 의한 공간 질의 84

GetHilightedObjectInfo 메소드 138

GFont 48

GIcon 49

GMapDataset 20

GMapDataset 구성 테이블 간 관계 19

GMapDatasets 20

GMapView 26

GMapView의 구조 25

GMeasure 50

GMultiFilterRender 44

GPen 47

GPolygon 공간연산 104

GPolygon 공간연산함수 102

GPolygon 이용 공간 질의 86

 IntraMap/Objects ***PROGRAMMERS GUIDE***

Grid Index 테이블 19

GShapeExporter 152

GShapeImporter 151

GSimpleRenderer 44

GSymbol 45

Ｈ

HilightEnable 프라퍼티 138

Ｉ

IntraMap/Objects 소개 4

IntraMap/Server 연결 12

Ｌ

LButtonDown 이벤트 ..................................................................................................................................39, 109

Ｍ

MapDataset 물리적 저장구조 19

MapDataset 테이블 19

MDB 파일 연결 11

|  |  |
| --- | --- |
| IntraMap | /Objects ***PROGRAMMERS GUIDE*** |
|  | |
| Ｏ | |

OpenRecords 메소드 99

Ｓ

Shape 파일.................................................................................................................................................... 142, 150

Shape 파일 가져오기............................................................................................................................... 142, 153

Shape 파일 가져오기/내보내기 150

Shape 파일 내보내기 157

SQL 쿼리문 이용 데이터 검색 94

Ｔ

ToMapXY 41

Ｘ

XDOMAIN 테이블 19

XMETALAYER 테이블 19

|  |  |
| --- | --- |
| IntraMap | /Objects ***PROGRAMMERS GUIDE*** |
|  | |
| ㄱ | |

개발 지원 5

개발 환경 4

공간 개체 삭제 138

공간 객체 추가 129

공간 데이터 편집 129

ㄷ

데이터베이스 6

데이터베이스 연결 11

디스플레이 범위 설정 68

ㄹ

레이블 설정 68

레이어 심볼 적용 51

레이어 심볼 정의 45

레이어 추가 33

렌더러 44

|  |  |
| --- | --- |
| IntraMap | /Objects ***PROGRAMMERS GUIDE*** |
|  | |
| ㅁ | |

마우스 모드 78

마우스 모드에 따른 기능 80

맵 데이터 셋 18

맵 데이터 셋 정보 얻어오기 22

멀티 필터 렌더러의 활용 55

ㅅ

실세계 좌표 39

ㅇ

이미지 리스트 콘트롤 연결 62

이미지 아이콘의 활용 62

ㅈ

좌표 변환 39

지도데이터 조작 77





IntraMap/Objects

***PROGRAMMERS GUIDE***

ㅎ

화면 상 레이어별 색상관리 43

화면 제어 77

화면 좌표 39