

ActiveX SyGImap control specifications

Is a viewer for fixed and mobile objects over user maps. Properties of this objects (aspect, status, type and position) are defined and actualized from one or more data providers by interchanging messages using different communication protocols. The fixed objects can be also user placed onto the map like a landmark.

SyGImap encapsulates all object data management and representation as well as map managing, displaying and calibrating tasks.

Moreover, it provides support for peer to peer user communication with the data providers and with another SyGImap instances.

Everytime SyGImap is launched (session) it records all incoming messages and data so you can review then at any time even while new data is being procesed.

SyGImap is an ActiveX component that can be used from a programming language in order to develop applications like:

- Fleet management with inter-communication ability (chat) between diffrent units and control points.
- Vehicle navigation.
- Marine, railway and air navigation.
- GPS tracking applications.
- Emergency locator systems.
- Vehicle location and control services (AVL).
- Personal navigation or touring devices.
- Sporting and recreation personal devices.
- Mobil and events Control Panels.
- Part of the Control Point in driving simulators (vehicles, trains, trams...).
- E-learning (sensitive and information maps).
- Displaying and managing of images that requires calibration and/or positioning of mobil and fixed elements.

SyGImap uses UDP and TCP/IP protocols so it can be run within a web explorer to easily develop intranet/internet applications or web map services. Just include this HTML code label:

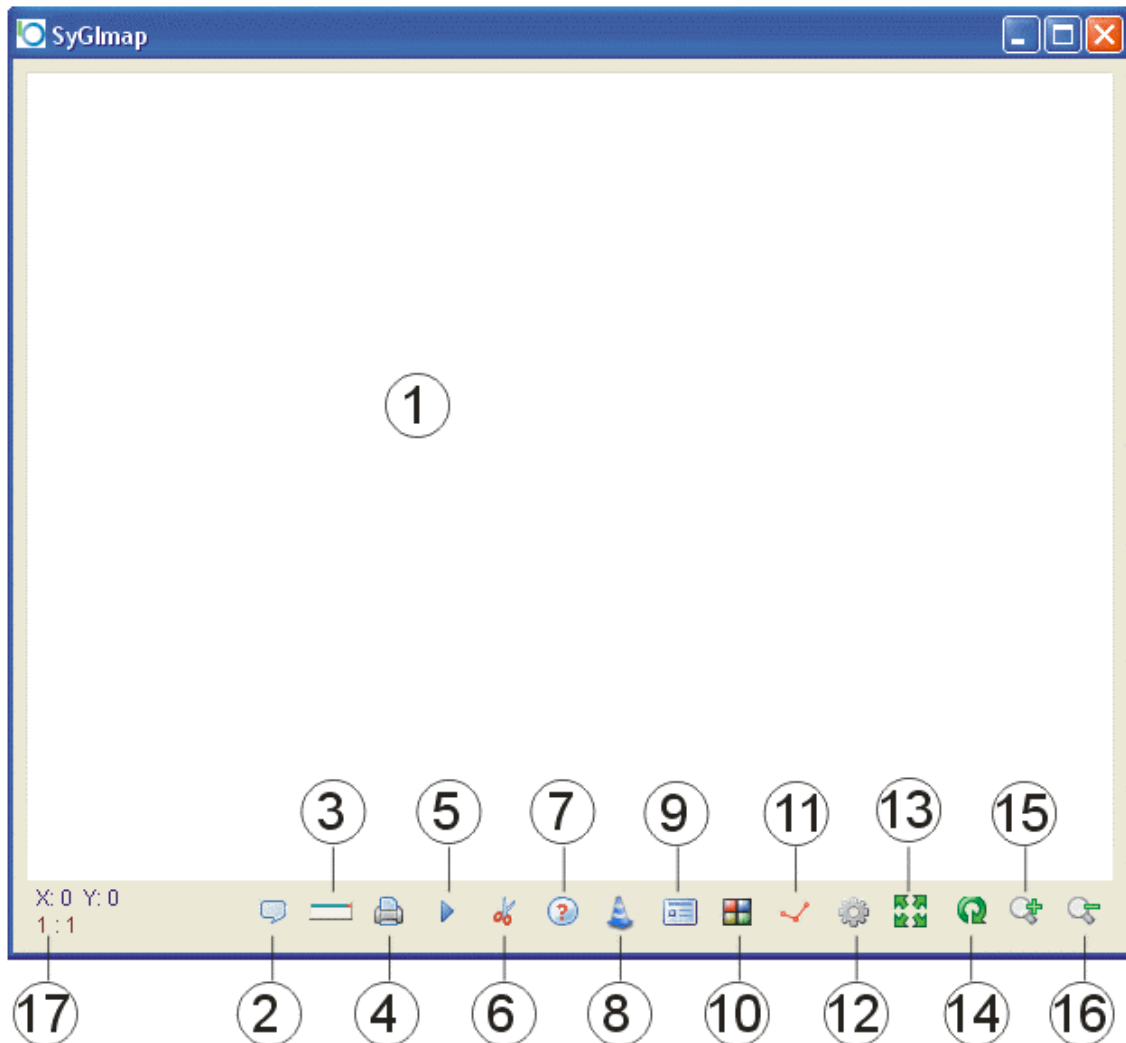
```
<P>  
<OBJECT WIDTH=800 HEIGHT=600 CLASSID="clsid:49A9727E-84CF-463A-9DA2-9092ABE1488C"></OBJECT>  
</P>
```

Install SyGImap ActiveX control executing: **regsvr32 activeSyGImap.ocx**.

Uninstall SyGImap ActiveX control executing: **regsvr32 /u activeSyGImap.ocx**.

You can use (doble clic) the **.bat** files downloaded with the component.

SyGImap does not need a directory structure on disk to work. When it is executed from an application or from a web explorer SyGImap creates a local directory named: **SyGImap**. In this folder can be found internal use files and it is the appropriate place to put your map files and the image files used to personalize certain objects (see later the configuration tool).



1. Objects and maps visualization area.

View center function: do clic with the left button of the mouse on this area to move the window center to the data point. If the map is a raster image the resulting view is also scaled to 1:1 (real size of the image).

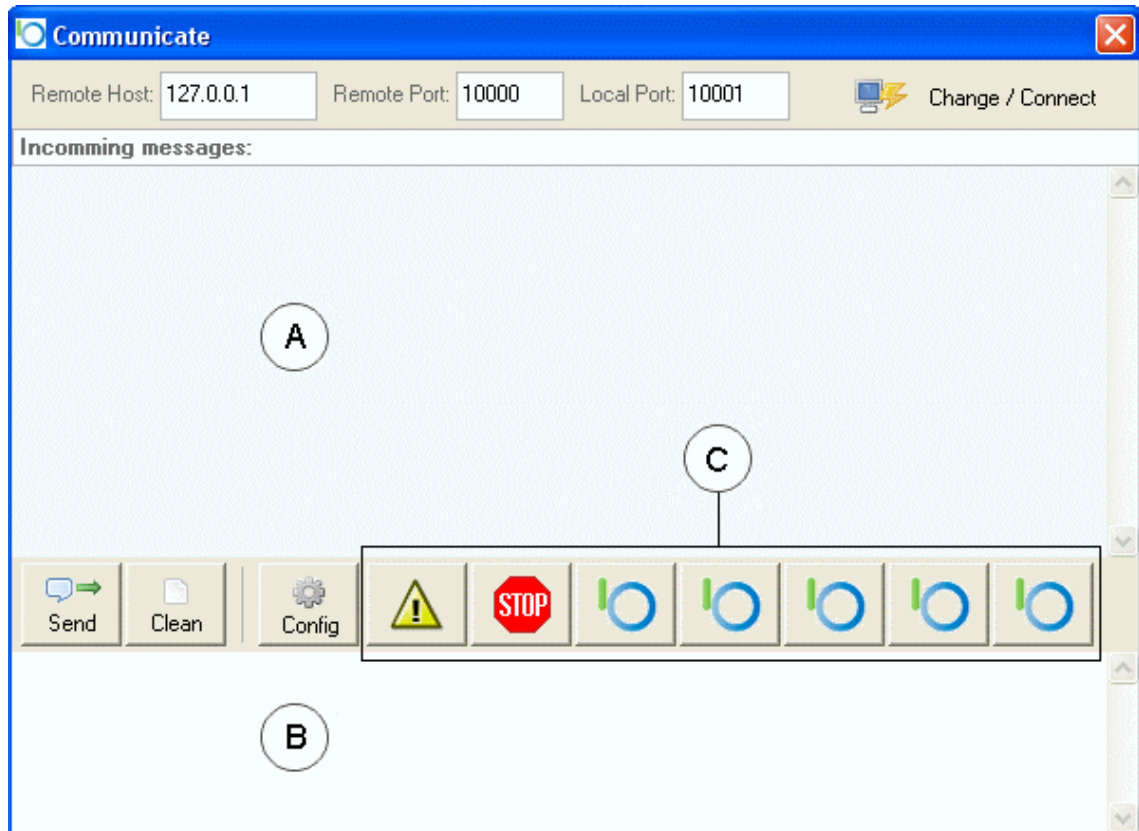
Zoom window: do clic with the left button on this area and drag to obtain the desired zoom window.

2. Communicate.

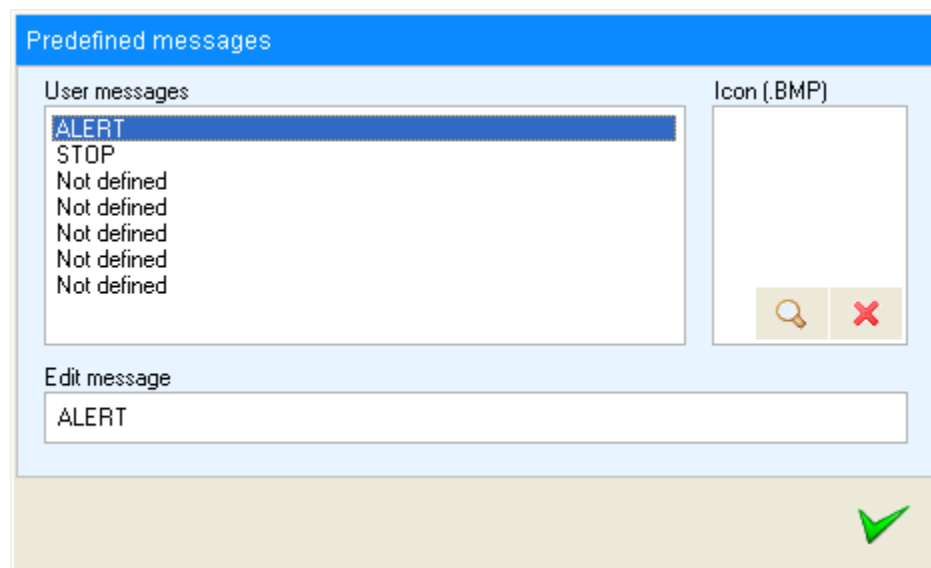
Opens a dialog from where you can stablish a text messages and text predefined based communication (an ALERT and a STOP messages examples are supplied). From here you can interchange messages between the machines using SyGImap and between them and the data providers.

This tool uses the TCP/IP protocol so you must specify: the Remote Host IP to which you desire to stablish the communication, the Local Port and the Remote Port.

Click on 'Change/Connect' button every time you change these parameters.



- A. Show incoming messages. Clic 'Clean' button to clean the content in this area.
- B. Key in this area the text you want to send and then clic 'Send' button or press the <Enter> key.
- C. Predefines messages. Clic a button to send the desired message.
To personalize a message clic the 'Config' button.



A maxim of seven messages are supported.

To create or edit a message: select it from the 'User messages' list, modify the text (Edit message) and add the 'Icon' that must appear on the message button. The image format must be a Windows bitmap (.BMP) and it's size should adjust the target button (maxim of 46x38 pixels).

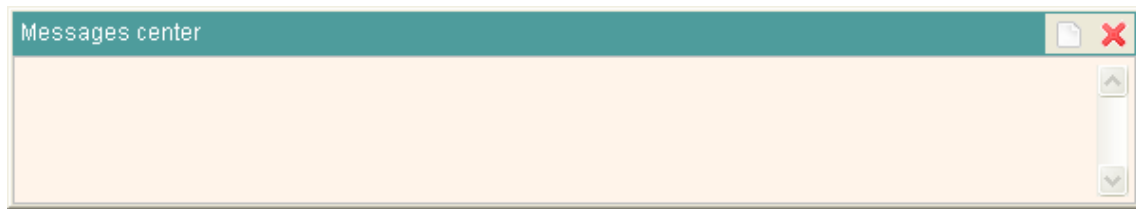
3. Messages Center.

SyGImap uses this dialog to show the standard text messages received from the data provider (see message:

SGI_DATA_MSG) and the notifications to the user (i.e. an overtake occurs between objects).

Use 'Clean content' button to clean the actual text on the window.

Use 'Close' button to hide this dialog.



4. Print view.

Prints the actual view content.

5. Play session.

Everytime SyGImap is launched (session) it records all incoming messages and data so you can review then at any time even while new data is being procesed. The reproduction of a session does not interrupt the reception of incoming data that continues in background.

The session is registered in a file named with the actual date and extension '.REC' on the *SyGImap* folder.

To review a registered session use this tool and select the desired '.REC' file.



Use the time slide bar at the bottom to move throught the session.

To finish a reproduction clic the 'Stop' button placed at the left side of this bar.

When you leave the reproduction SyGImap returns to previous active map.

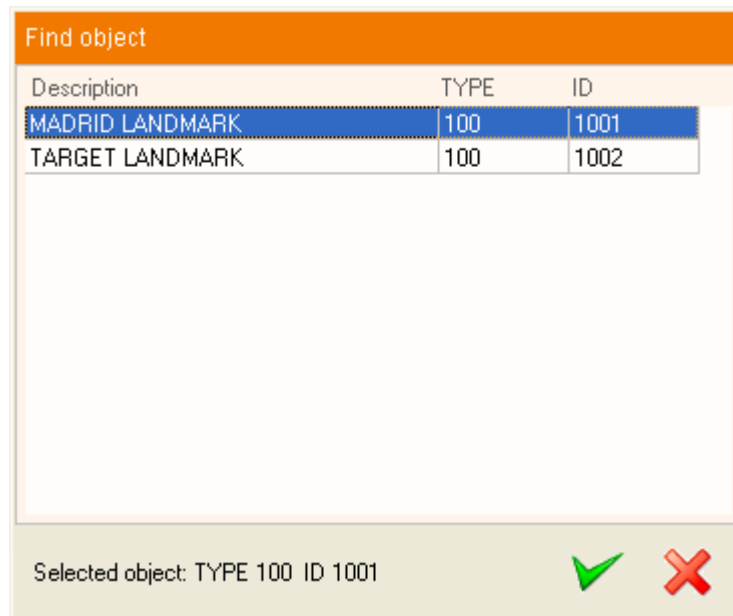
6. Cut active session.

SyGImap advertises you when the session that is being registered now (the active '.REC' file) is selected to review. You must use this tool to divide the session from beginning to this moment.

7. Find object.

Shows a dialog with the list of existing objects on the actual map.

Select the object to find and SyGImap will show it using zoom in function and resalted with a colored circle over it.



8. Place landmark.

With this tool you can place a landmark on the actual map at any time.

Adjust the zoom level you need to place the object and clic the left button of the mouse at the target point.

Set the object's values and change if necessary the image that SyGImap must use to represent it. The image format must be Windows bitmap (.BMP).

Place landmark

Description

Distance

Status

User data 1



User data 2

User data 3

User data 4

User data 5

User data 6

Description: descriptive text for the object.

Distance: integer value, in map units, that SyGImap must use to advertise the user that an overtake has occurred (a mobil object gets a fixed or landmark object) at a distance equal or lower than this value.

User data: You can assign several alphanumeric values to an object. The meaning and understanding of these values is under user criteria.

Status: numeric integer value.

User data 1, 2 y 3: numeric integer values.

User data 4: text value maximum 64 characters.

User data 5: text value maximum 128 characters.

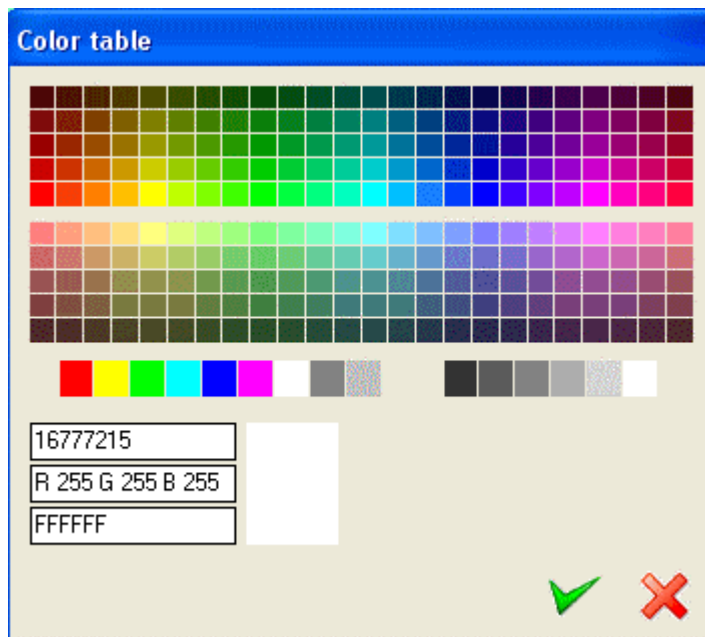
User data 6: text value maximum 255 characters.

9. Show/hide legend.

You can define a image as the map's legend and show or hide it using this tool.
To assign a image as a legend see the 'Configuration' tool.

10. Background color.

Sets the background color of the visualization area.



11. Path.

With this tool you can digitize a path on the map and extract in a text file the coords (X,Y) of it.

To draw the path clic the right button of the mouse over the desired points. While drawing the path you can use the zoom tools to adjust the view as you need.

Clic on this same tool to finish the path and set the file name in the save dialog box that appears. The extension of a path file is: '.PATH'.

12. Configuration.

Shows the configuration dialog box. From this dialog you can personalize the entire SyGImap component properties.

General

Size (pixels) width-height.

Sets the size, in pixels, for the component.

When you use SyGImap in a web browser you must set the size in the HTML code (WIDTH and HEIGHT labels).

Example:

```
<P>
<OBJECT WIDTH=800 HEIGHT=600 CLASSID="clsid:49A9727E-84CF-463A-9DA2-9092ABE1488C"></OBJECT>
</P>
```

Beginning map

Sets the map that must be loaded by default when SyGImap starts. Set this field blank if no map is desired at the begin.

Show this image as legend

Sets the image that must be used as the map's legend. Set this field blank if no legend is desired. See the supported raster graphic formats.

Number of decimals in coords

Set the number of decimals that must be shown when displaying coords.

Fill closed areas with color (only DXF format)

Only with the DXF format: if marked, the closed entity areas will be drawn filled with its border color.

TCP Protocol


If marked, SyGImap will set communication channels with the data providers using the TCP/IP protocol.

Clic the 'Configure...' button to specify the appropriate values.

TCP Configuration

Available communication channels:

Host IP	Remote Port
127.0.0.1	6666



In the 'Host IP' column set the network direction of the data provider. In the 'Remote Port' column set the port number that the data provider uses to communicate.

Each SyGImap instance will receive only those data providers listed in this table.


UDP Protocol

If marked, SyGImap will set communication channels with the data providers using the UDP protocol. Click the 'Configure...' button to specify the appropriate values

UDP Configuration

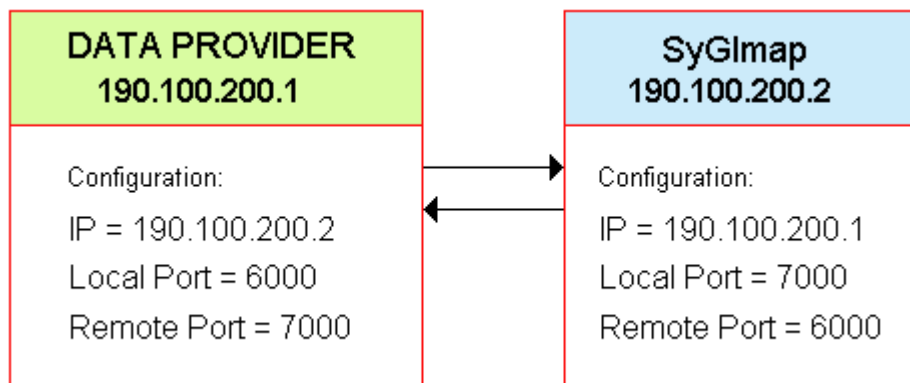
Available communication channels:

IP	Local Port	Remote Port
127.0.0.1	4444	5555



In the 'IP' column set the network direction of the data provider. In the 'Local Port' and 'Remote Port' columns set

the ports to communicate with the data provider. Example:



Each SyGImap instance will receive only those data providers who are listed in this table.

Communications: use silent mode

If marked, SyGImap does not reply to incoming messages from the data provider. See COMMUNICATION MESSAGES later.

Actual map: change with server request

If marked, SyGImap changes its active map when requested from the data provider (See message: SGI_DATA_MAP).

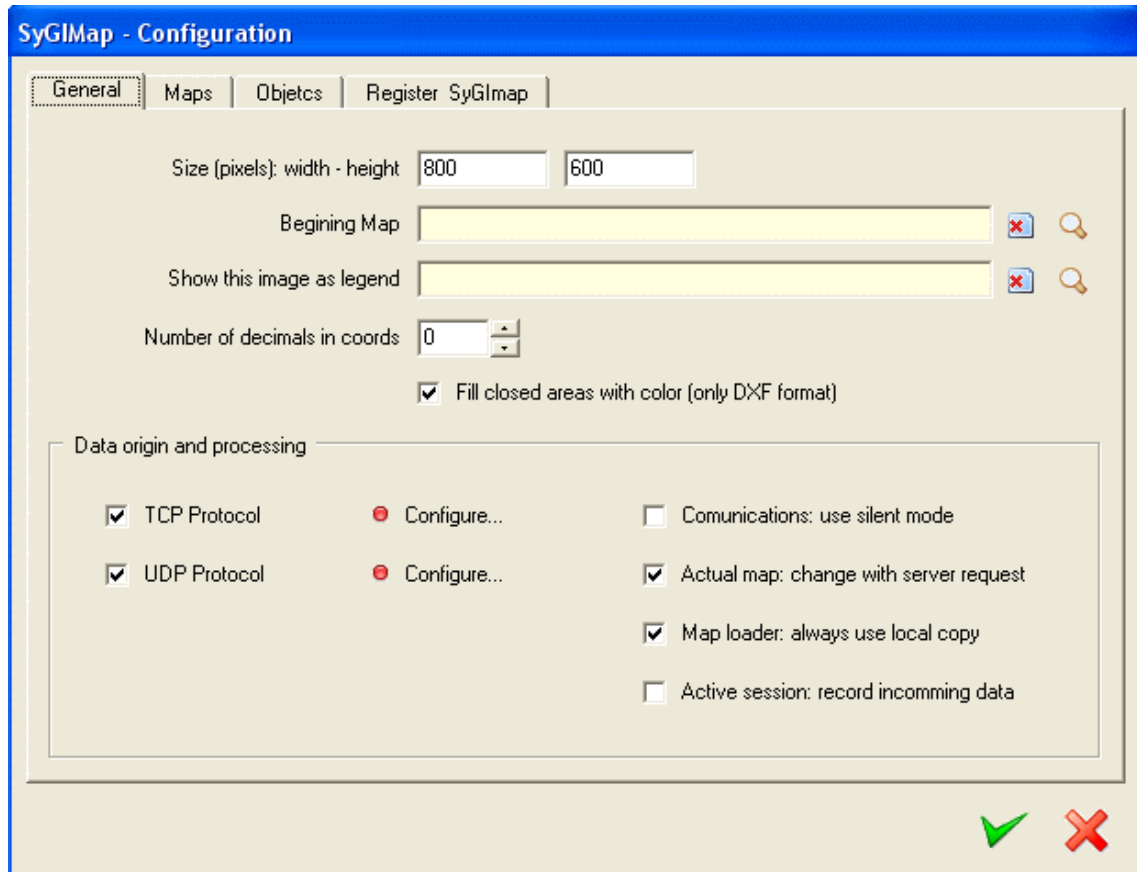
Map loader: always use local copy

SyGImap try to load the maps from the local folder *SyGImap* for a great efficiency in image managing. If the requested map from a data provider is not found in this folder, a copy of the map is made from the specified path (See message: SGI_DATA_MAP).

If marked, this option causes map loading from the local folder if it exists. When this option is not marked the map is always actualized from the data provider.

Active session: record incoming data

If marked, SyGImap registers automatically all incoming data and messages from the data provider making possible their later review.



Maps

A map is any kind of user image, scheme or graphic on which objects are located and refreshed. Note that the use of SyGImap is not only restricted to geographic maps. Due to its calibration property it can be used for any other purpose which requires a calibrated image and/or the ability to locate coordinated objects onto it.

SyGImap is multimap: it works with an unlimited number of maps.

Supported maps: vectorial (DXF format) and raster (see later the supported raster formats).

Coords system: SyGImap supports any type of coords system. Vectorial maps use their own DXF internal coords system. Raster maps must be georeferenced or calibrated: at the moment you add the image to the image library you must give the extreme coords values of the map. By default, the extreme coords are the values in pixels of image's width and height.

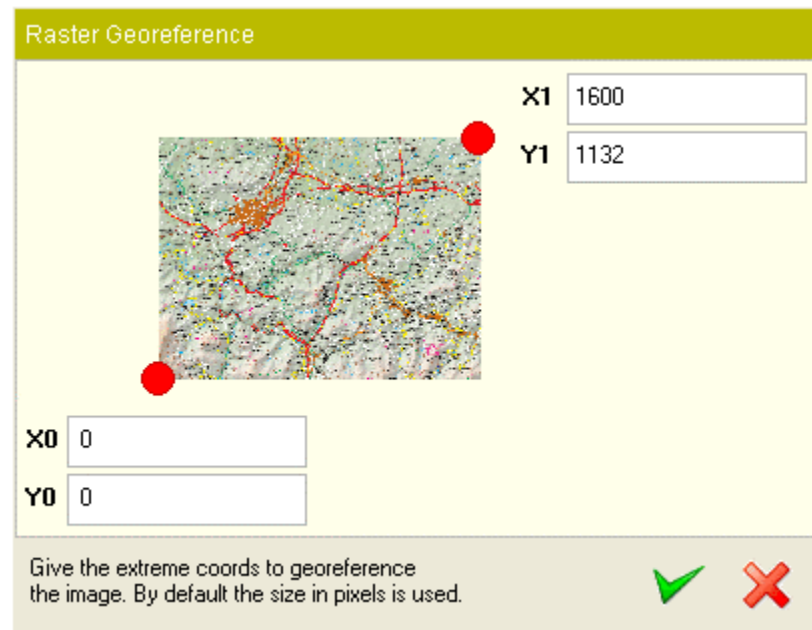
Quality of the map: it depends only on the user image. SyGImap handles data raster with a 'tiled in memory' technics optimized to show large images in an efficient and fast way.

Map library

It contains the list of local available maps in SyGImap. Use the 'Add map' and 'Delete map' buttons to add and delete maps in library.

When adding a raster map SyGImap shows you a calibration dialog to specify the extreme coords of the image. This will permit SyGImap to locate the objets over the map.

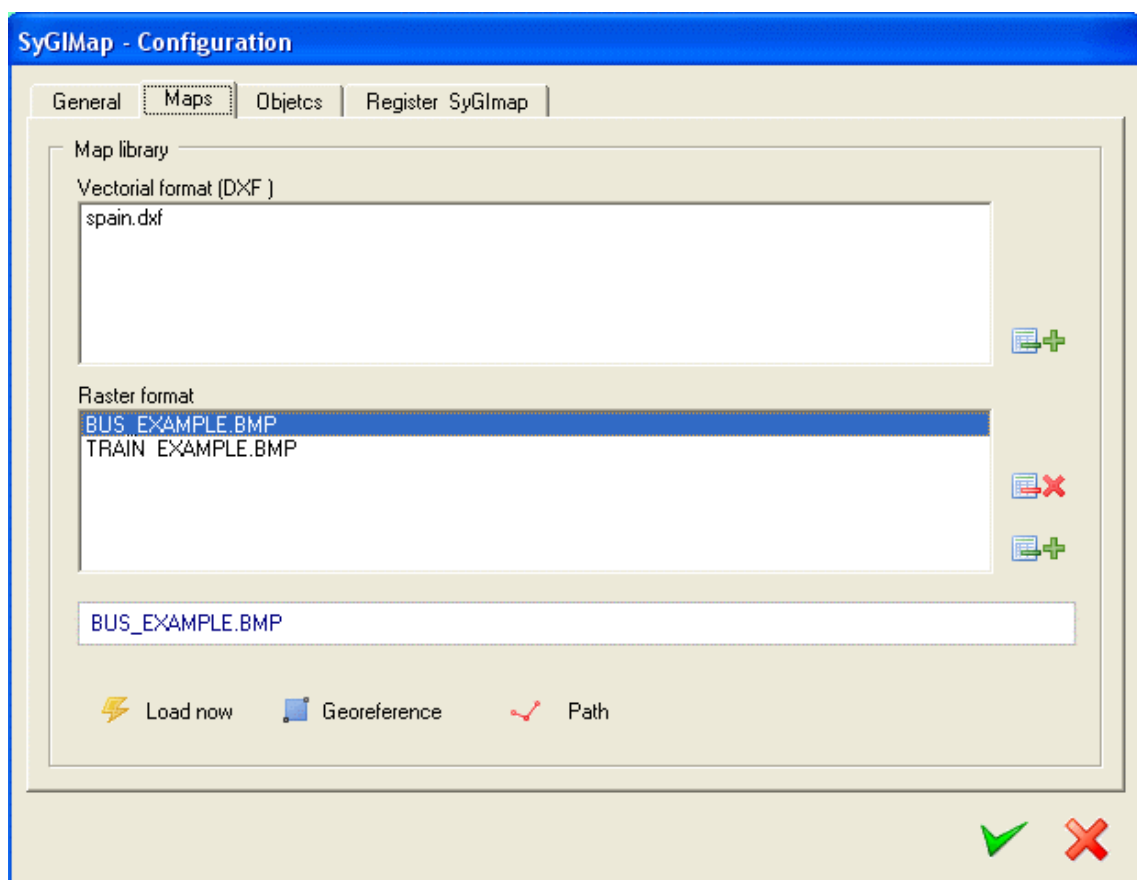
By default, the extreme coords values are the image width an height sizes (in pixels).



Clic 'Load now' button to change the active map with the selected one from the list of available maps.

Clic 'Georeference' button to modify the calibration values of the selected raster map.

Clic the 'Path' button to digitize a path on the selected map in order to obtain a file with its (X,Y) coords.

















Objects

An object element consists on both a image (bitmap) and a group of alphanumeric data and is placed, based on this data, over the user map.

SyGImap objects can be: fixed (landmarks) and mobiles. The difference between them is the way how SyGImap informs the user and the data providers their position and the instant in which a mobil object overtakes a fixed object.

Object types supplied by default:

	Landmark		Walker
type 100		type 107	
	Vehicle		Plane
type 101		type 108	
	Ambulance		Ship
type 102		type 109	
	Bombers vehicle		Train
type 103		type 110	
	Bus		Tram
type 104		type 111	
	Truck		Emergency vehicle
type 105		type 112	
	Van		SyGImap uses this icon to display objects with undefined or not available image file
type 106			

Previous associated bitmaps can be customized and replaced by any other image. Clic the search button to select the image file for the object. The image format must be Windows bitmap (.BMP).

Objects of the same type are discriminated by a numeric value (object's ID). For example, you can send to a map different objects of type TRAIN each of them with its ID number and a set of data (see message: SGI_DATA_OBJECT).

Objects of the same type are displayed with the same image. To assign an image to a specific ID inside the same type of object you must copy the image file to the local directory *SyGImap* and name this file as: 'TYPE_N1_eN2.BMP' where:

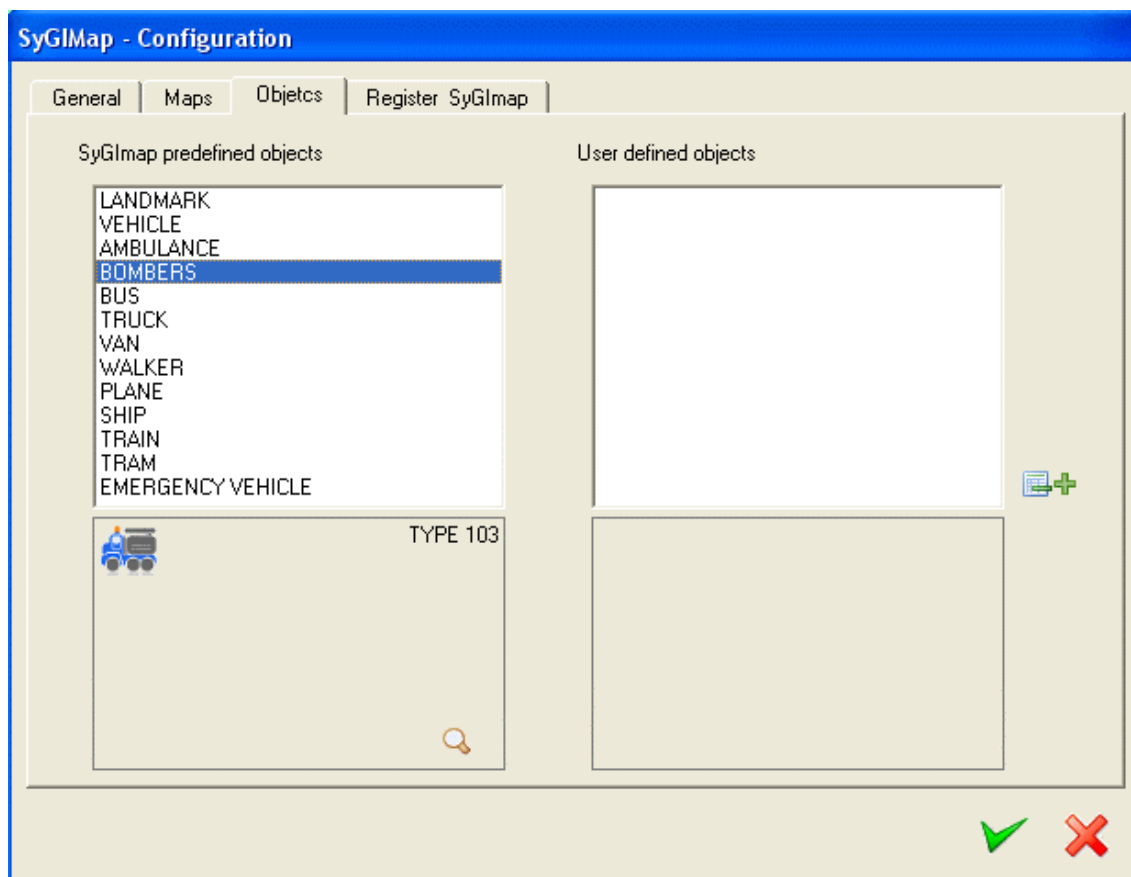
N1 = integer number equal to the object's type
N2 = integer number equal to the object's ID

Example:

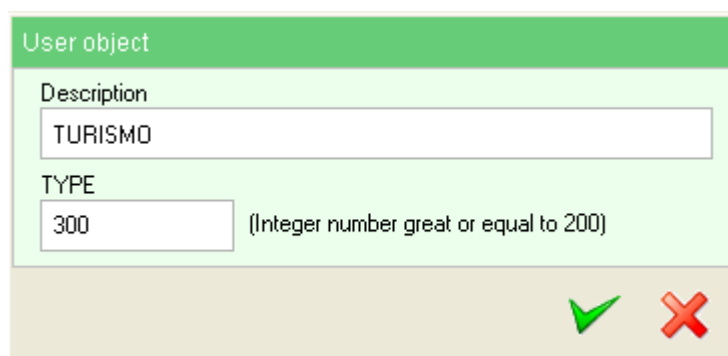
'TYPE_104_e1.BMP' is the image that SyGImap must assign to the object of type 104 (bus) and ID 1.

When a data provider sends an object message to SyGImap the object is created if it not exists, incoming data are assigned and then positioned. If the object just exists the data and position are actualized.

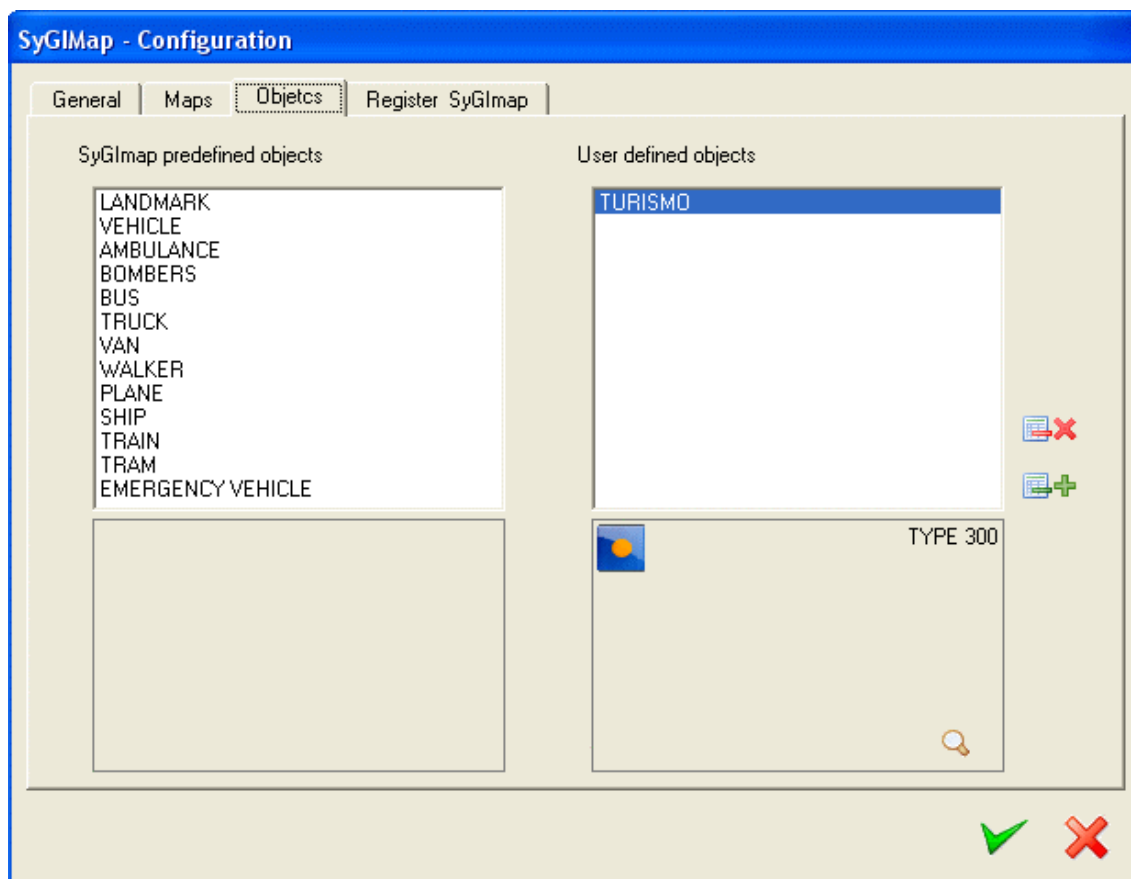
SyGImap supports an unlimited number of object types and unlimited number of type IDs.



User can add new object types and SyGImap automatically incorporate them to its object's system. To add a new object clic the 'Add object' button. In the next dialog box set the 'Description' or name for the object and the numeric value for the TYPE (greater or equal to 200). Clic the accept button to finish.



Then, customize the image for this type of objects using the search button. The image format must be Windows bitmap (.BMP).



User objects of the same type are displayed with the same image. To assign an image to a specific ID inside the same type of user object you must copy the image file to the local directory *SyGImap* and name this file as: 'TYPE_N1_eN2.BMP' where:

- N1 = integer number equal to the object's type
- N2 = integer number equal to the object's ID

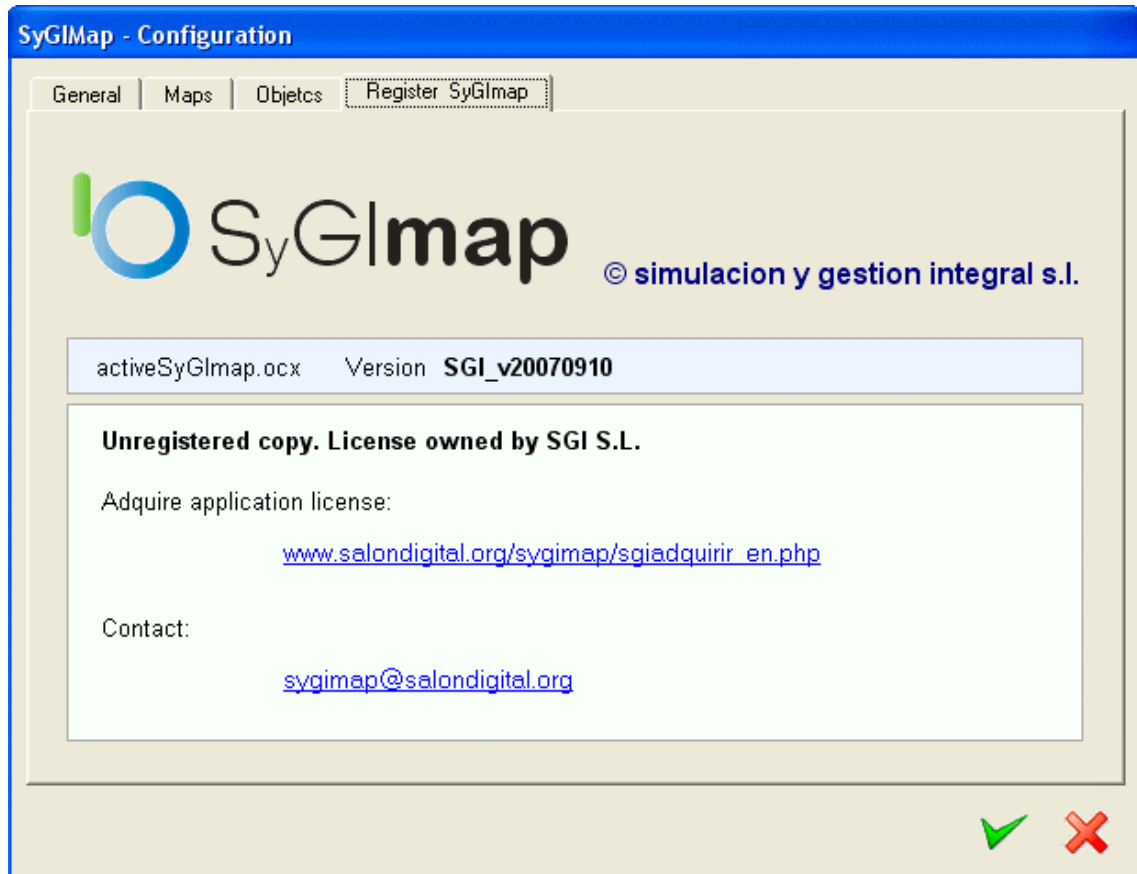
Example:

'TYPE_300_e56.BMP' is the image that SyGImap must assign to the object of type 300 (TURISMO) and ID 56.

Clic the 'Delete object' button to delete the selected object from the list.

Objects can be linked to a particular map or attached to all the maps (see message: SGI_DATA_OBJECT).

Register SyGImap



SyGImap ActiveX control is distributed under customized license: your data will appear in this dialog as **the owner of the license** and the texts displaying our contact data and the mark over the map will be deleted.

To do this you may send us:

- Your name or company name
- Your document number or the identity number of the company
- An email address to contact you

Please, note that:

- If you use the component to develop your own application, you must purchase a number of individual licenses equal to the instances of SyGImap you plane to install.
- If you use the component to develop applications for third parties, you must acquire a number of individual licenses equal to the instances of SyGImap you plane to install to your client to personalize them with his name.

For more than one license we have special offers and best prices.

To obtain a registered copy contact with **simulación y gestión integral s.l.** at the web site:

www.salondigital.org/sygimap/sgiadquirir_en.php

or use this email:

13. Fit all.

Displays the map fitted in the display area.

14. Last view (only DXF format).

Shows the previous view of the map.

15. Zoom in.

Increase the zoom level of the map (decrease the scale factor).

16. Zoom out.

Decrease the zoom level of the map (increase the scale factor).

17. Coords and active scale.

Shows the cursor position in map coords and the active viewing scale of the map.

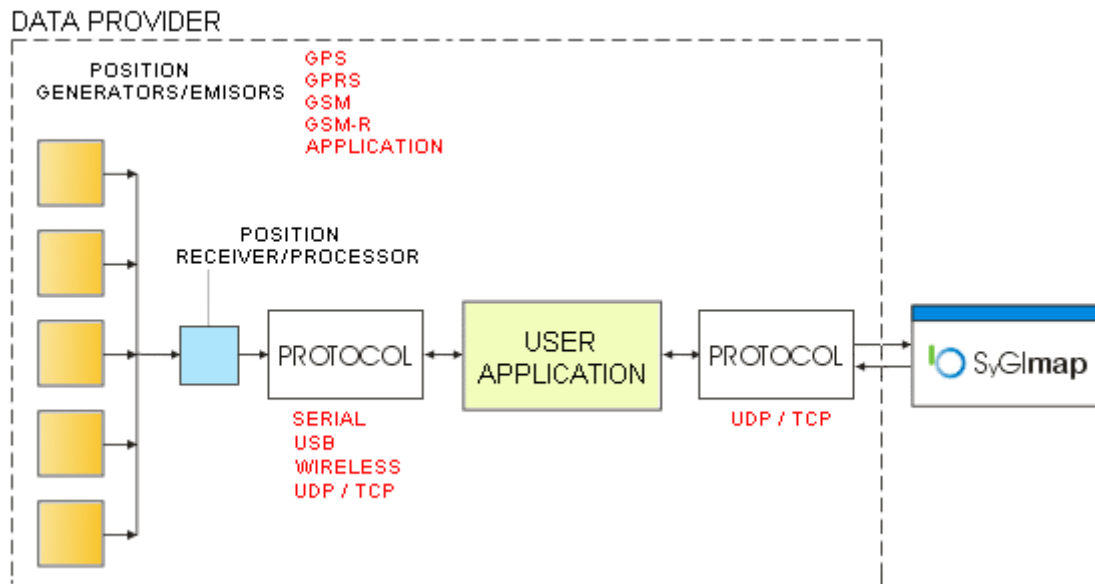
Communications and structure of the messages

SyGImap interacts with the data providers using UDP and TCP/IP protocols (also Serial RS-232c protocol under demand). All of them can be employed simultaneously at the same SyGImap instance. In this case, take care about port number conflicts.

The number of available active conexions depends on operative system requeriments and the user application needs.

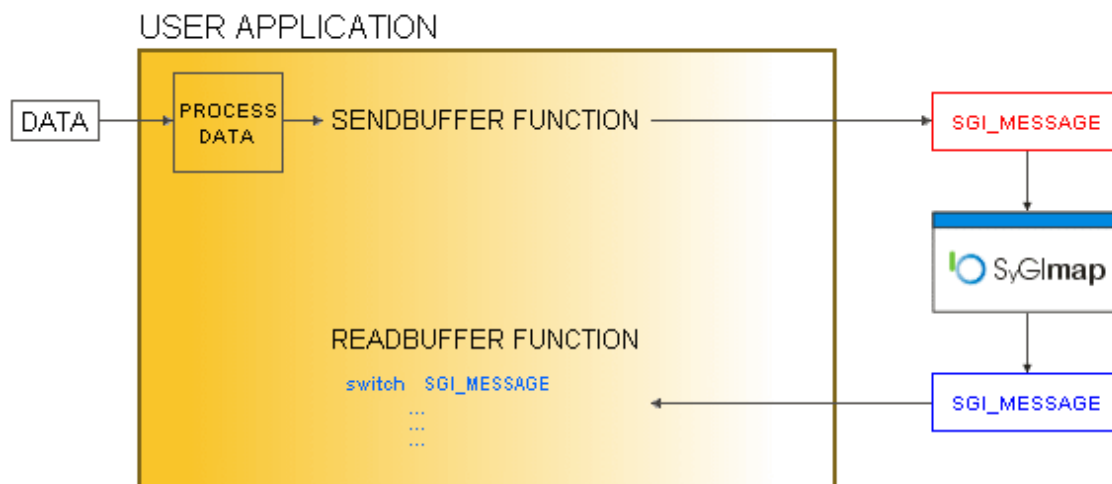
A data provider is a set of a data generation and/or position capture system and a program or application that process those data and interchanges messages with SyGImap.

Schema of a generic data provider:



SyGImap can selectively display incoming objects data from a data provider or from several data providers simultaneously.

The user application sends and receives data to/from SyGImap instances through UDP and/or TCP protocols.



Structure of messages

Every message is formed by two parts:

1. A message identifier: is an integer number identifying the message whose data is nested.
2. Message body: is a data structure with the content of the message.

For example, the change active map message must be sent as follows:

```
const
  MAX_SIZE = 4096;
  SGI_DATA_MAP = 705;
  MAP_TIPO_RAS = 2;

type
  TCadena = array [1..MAX_SIZE] of char;

  struct_DATA_MAP = record
    ip: string[16];
    tipo: integer;
    map: string[255];
  end;

var
  id, msgSize: integer;
  buf: TCadena;
  ms_data_map: struct_DATA_MAP;

begin
  //clear buffer
  fillchar(buf,MAX_SIZE,#0);

  //set the ID of this message
  id := SGI_DATA_MAP;
  move(buf,id,sizeof(id));

  //fill the message data
  ms_data_map.ip := '192.100.200.1'; //my data server application ip address
  ms_data_map.map := '\\192.100.200.1\my maps\amap.tif'; //location of the map on the data server
  ms_data_map.tipo := MAP_TIPO_RAS; //map type is raster
  move(buf[5],ms_data_map,sizeof(ms_data_map));

  //now send the buffer using the socket component or function
  msgSize := sizeof(id)+sizeof(ms_data_map);
  Socket.SendBuf(buf, msgSize);
end;
```

The next example shows a function to process incoming messages (from SyGImap instances) in the data provider:

```
function Socket_procesa_msg(buf: TCadena): integer;
var
  id: integer;
  ms_data_connect: struct_DATA_CONNECT;
  ms_replay_object: struct_DATA_REPLY_OBJECT;
  ms_data_overtake: struct_DATA_OVERTAKE;
begin
  //buf is the variable that contains the incoming message

  //extract message ID
  move(buf,id,sizeof(id));

  case id of
    SGI_CONNECT_OK:
      begin
        //extract SyGImap response to SGI_CONNECT message:
        move(buf[5],ms_data_connect,sizeof(ms_data_connect));

      end;
    SGI_DATA_OK:
      begin
        //SyGImap has just received message SGI_DATA_MSG sent from this server

      end;
    SGI_DATA_OBJECT_OK:
      begin
        //extract SyGImap response to SGI_DATA_OBJECT and SGI_DATA_POSITION messages:
        move(buf[5],ms_replay_object,sizeof(ms_replay_object));

      end;
    SGI_DATA_OVERTAKE:
      begin
        //SyGImap sent a landmark overtake message
        move(buf[5],ms_data_overtake,sizeof(ms_data_overtake));

      end;
    SGI_DISCONNECT:
      begin
        //SyGImap instance has disconnected

      end;
  end;
end;
```

Predefined consts and data structs used by SyGImap are shown before for Object Pascal (Delphi) and C languages.

Initialization messages

Connection message (SGI_CONNECT)

Is sent by the data provider to a SyGImap instance to verify if it is connected.

CONTENIDO DEL MENSAJE			
DATA	TYPE	SIZE (bytes)	DESCRIPTION
700	Integer	4	Message identifier

Replay message to a connection message (SGI_CONNECT_OK)

Is sent by the SyGImap instance as response to a connection message received from a data provider.

CONTENIDO DEL MENSAJE			
DATA	TYPE	SIZE (bytes)	DESCRIPTION
800	Integer	4	Message identifier
ip	Char array	16	Network ip of the SyGImap instance
map	Char array	255	Active map name in the SyGImap instance

Finalization messages

End of connection message (SGI_DISCONNECT)

Is sent by the SyGImap instance to its data providers when finishing execution.

CONTENIDO DEL MENSAJE			
DATA	TYPE	SIZE (bytes)	DESCRIPTION
701	Integer	4	Message identifier

Messages sent by the user application to SyGImap

Full object data message (SGI_DATA_OBJECT)

The data provider sends this message to the desired SyGImap instances with the full set of object data. At the SyGImap side the object is created if it not exists, incoming data are assigned and then positioned. If the object just exists the data and position are actualized.

CONTENIDO DEL MENSAJE			
DATA	TYPE	SIZE (bytes)	DESCRIPTION
702	Integer	4	Message identifier
typeObject	Integer	4	Type of object
idObject	Integer	4	ID number of object
status	Integer	4	Object status
isLandMark	Integer	1	TRUE=is a landmark FALSE=is not a landmark
distance	Integer	4	If it is a landmark, distance to notify the overtake by a mobile object
description	Char array	64	Name or description of the object
map	Char array	128	Name of the map the object is linked to. Set blank if the object is not linked to a particular map.
X	Char array	16	Coord x value. Number format: decimal separator = '.' (point) and without thousand separator.
Y	Char array	16	Coord y value. Number format: decimal separator = '.' (point) and without thousand separator.
Z	Char array	16	Coord z value. Number format: decimal separator = '.' (point) and without thousand separator.
velocity	Char array	16	Object's velocity value. Number format: decimal separator = '.' (point) and without thousand separator.
User1	Integer	4	Additional user data
User2	Integer	4	Additional user data
User3	Integer	4	Additional user data
User4	Char array	64	Additional user data
User5	Char array	128	Additional user data
User6	Char array	255	Additional user data

Objetc data position message (SGI_DATA_POSITION)

The data provider sends this message to the desired SyGImap instances with the object data position and status. If the object exists SyGImap sets the new position and satus values.

CONTENIDO DEL MENSAJE			
DATA	TYPE	SIZE (bytes)	DESCRIPTION
704	Integer	4	Message identifier
typeObject	Integer	4	Type of object
idObject	Integer	4	ID number of object
status	Integer	4	Object status
X	Char array	16	Coord x value. Number format: decimal separator = '.' (point) and without thousand separator.
Y	Char array	16	Coord y value. Number format: decimal separator = '.' (point) and without thousand separator.
Z	Char array	16	Coord z value. Number format: decimal separator = '.' (point) and without thousand separator.
velocity	Char array	16	Object's velocity value. Number format: decimal separator = '.' (point) and without thousand separator.

Change active map message (SGI_DATA_MAP)

The data provider sends this message to the desired SyGImap instances to display the specified map on them.

CONTENIDO DEL MENSAJE			
DATA	TYPE	SIZE (bytes)	DESCRIPTION
705	Integer	4	Message identifier
Ip	Char array	16	Data provider ip address
Tipo	Integer	4	1=vectorial, 2=raster
Map	Char array	255	Name of the map that the SyGImap instance must to display. If the <i>map</i> parameter contains the full path of the map file, SyGImap will make a copy of it in its local directory and will use it in next calls. If the <i>map</i> parameter does not contain the full path of the map file, SyGImap will try to load it from its local directory.

Text message (SGI_DATA_MSG)

The data provider uses this message to send notifications to the user. The SyGImap instance will show them to the user in a dialog (*Messages Center*).

CONTENIDO DEL MENSAJE			
DATA	TYPE	SIZE (bytes)	DESCRIPTION
703	Integer	4	Message identifier
Ip	Char array	16	Data provider ip address
msg	Char array	255	Text with the notification that SyGImap must show to the user in the SyGImap instance

Messages sent by SyGImap to the user application

Reply message to full object data and object position messages (SGI_DATA_OBJECT_OK)

The SyGImap instance sends this message to the data provider when receives the full object data message (SGI_DATA_OBJECT) and the object position message (SGI_DATA_POSITION).

If SyGImap is configured to work in *silent mode* this message is not sent to the data provider (see the configuration tool).

CONTENIDO DEL MENSAJE			
DATA	TYPE	SIZE (bytes)	DESCRIPTION
802	Integer	4	Message identifier
typeObject	Integer	4	Type of incoming object
IdObject	Integer	4	ID number of incoming object
Status	Integer	4	Incoming object status

Replay message to a text message (SGI_DATA_OK)

The SyGImap instance sends this message to the data provider when receives a text or user notification message (SGI_DATA_MSG).

If SyGImap is configured to work in *silent mode* this message is not sent to the data provider (see the configuration tool).

CONTENIDO DEL MENSAJE			
DATA	TYPE	SIZE (bytes)	DESCRIPTION
801	Integer	4	Message identifier

Overtake message (SGI_DATA_OVERTAKE)

The SyGImap instance sends this message every time an object of type mobile overtakes the position of an object

of type fixed (landmark). This occurs when the distance between both objects is equal or less than the value of the parameter **distance** in the landmark object.

CONTENIDO DEL MENSAJE			
DATA	TYPE	SIZE (bytes)	DESCRIPTION
803	Integer	4	Message identifier
typeTarget	Integer	4	Type of the landmark object overtaken by the mobile object.
idTarget	Integer	4	ID number of the landmark object overtaken by the mobile object.
typeObject	Integer	4	Type of the object that overtakes the landmark.
idObject	Integer	4	ID number of the object that overtakes the landmark.
X	Char array	16	Coord x value of the overtake point. Number format: decimal separator = '.' (point) and without thousand separator.
Y	Char array	16	Coord y value of the overtake point. Number format: decimal separator = '.' (point) and without thousand separator.
Z	Char array	16	Coord z value of the overtake point. Number format: decimal separator = '.' (point) and without thousand separator.
velocity	Char array	16	Mobile object velocity value at the overtake point. Number format: decimal separator = '.' (point) and without thousand separator.

Pdefined consts and data structs

Object Pascal (Delphi)

Archive: sygimap_messages.pas

```
unit sygimap_messages;

interface

uses Windows, Classes, SysUtils;

const

//define IDs and consts

//map types

MAP_TIPO_NONE = 0;
MAP_TIPO_DXF = 1; //vectorial
MAP_TIPO_RAS = 2; //raster

//message IDs

SGI_CONNECT      = 700;
SGI_DISCONNECT   = 701;
SGI_DATA_OBJECT  = 702;
SGI_DATA_MSG     = 703;
SGI_DATA_POSITION = 704;
SGI_DATA_MAP     = 705;

SGI_CONNECT_OK   = 800;
SGI_DATA_OK      = 801;
SGI_DATA_OBJECT_OK = 802;
SGI_DATA_OVERTAKE = 803;

//predefined object types:

//landmark
TYPE_LANDMARK    = 100;

//basic objects
```

```
TYPE_VEHICLE      = 101;
TYPE_AMBULANCE    = 102;
TYPE_BOMBER       = 103;
TYPE_BUS          = 104;
TYPE_TRUCK        = 105;
TYPE_VAN          = 106;
TYPE_WALKER       = 107;
TYPE_PLANE        = 108;
TYPE_SHIP         = 109;
TYPE_TRAIN        = 110;
TYPE_TRAM         = 111;
TYPE_EMERGENCY    = 112;
```

```
//user object types:
//ID > 200
```

```
//...
```

```
//...
```

```
//define message structs
```

```
type
```

```
    struct_DATA_OBJECT = record
```

```
        idObjetc: integer;
        typeObjetc: integer;
        status: integer;
```

```
        isLandMark: boolean;
        distance: integer;
```

```
        description: string[64];
```

```
        map: string[128];
```

```
        x: string[16];
        y: string[16];
        z: string[16];
        velocity: string[16];
```

```
        user1: integer;
        user2: integer;
        user3: integer;
```

```
        user4: string[64];
        user5: string[128];
        user6: string[255];
```

```
    end;
```

```
    struct_DATA_POSITION = record
```

```
        idObjetc: integer;
        typeObjetc: integer;
        status: integer;
```

```
        x: string[16];
        y: string[16];
        z: string[16];
        velocity: string[16];
    end;
```

```
    struct_DATA_REPLY_OBJECT = record
```

```
        idObjetc: integer;
        typeObjetc: integer;
```

```
        status: integer;
    end;

    struct_DATA_OVERTAKE = record

        idTarget: integer;
        typeTarget: integer;

        idObjetc: integer;
        typeObjetc: integer;

        x: string[16];
        y: string[16];
        z: string[16];
        velocity: string[16];
    end;

    struct_DATA_MSG = record

        ip: string[16];
        msg: string[255];
    end;

    struct_DATA_MAP = record

        ip: string[16];
        tipo: integer;
        map: string[255];
    end;

    struct_DATA_CONNECT = record

        ip: string[16];
        map: string[255];
    end;

    struct_DATA_REPLY_HIT = record

        idHit: integer;

        idObjetc: integer;
        typeObjetc: integer;
        status: integer;

        distance: string[16];
    end;

implementation

end.
```

C, C++

Archive: sygimap_messages.h

```
#ifndef SYGIMAP_MESSAGES_H
#define SYGIMAP_MESSAGES_H

//define IDs and consts

//map types

#define MAP_TIPO_NONE 0
#define MAP_TIPO_DXF 1 //vectorial
#define MAP_TIPO_RAS 2 //raster

//message IDs
```

```
#define SGI_CONNECT          700
#define SGI_DISCONNECT      701
#define SGI_DATA_OBJECT     702
#define SGI_DATA_MSG        703
#define SGI_DATA_POSITION   704
#define SGI_DATA_MAP        705

#define SGI_CONNECT_OK      800
#define SGI_DATA_OK        801
#define SGI_DATA_OBJECT_OK  802
#define SGI_DATA_OVERTAKE   803
```

```
//predefined object types:
```

```
//landmark
```

```
#define TYPE_LANDMARK      100
```

```
//basic objects
```

```
#define TYPE_VEHICLE      101
#define TYPE_AMBULANCE    102
#define TYPE_BOMBER       103
#define TYPE_BUS          104
#define TYPE_TRUCK        105
#define TYPE_VAN          106
#define TYPE_WALKER       107
#define TYPE_PLANE        108
#define TYPE_SHIP         109
#define TYPE_TRAIN        110
#define TYPE_TRAM         111
#define TYPE_EMERGENCY    112
```

```
//user object types:
```

```
//ID > 200
```

```
//...
```

```
//...
```

```
//define message structs
```

```
struct struct1{
    int idObjetc;
    int typeObjetc;
    int status;

    bool isLandMark;
    int distance;

    char description[64];

    char map[128];

    char x[16];
    char y[16];
    char z[16];
    char velocity[16];

    int user1;
    int user2;
    int user3;

    char user4[64];
    char user5[128];
    char user6[255];
}

typedef struct1 struct_DATA_OBJECT;
```



```
struct struct2{
    int idObjetc;
    int typeObjetc;
    int status;

    char x[16];
    char y[16];
    char z[16];
    char velocity[16];
}

typedef struct2 struct_DATA_POSITION;

struct struct3{
    int idObjetc;
    int typeObjetc;
    int status;
}

typedef struct3 struct_DATA_REPLY_OBJECT;

struct struct4{
    int idTarget;
    int typeTarget;

    int idObjetc;
    int typeObjetc;

    char x[16];
    char y[16];
    char z[16];
    char velocity[16];
}

typedef struct4 struct_DATA_OVERTAKE;

struct struct5{
    char ip[16];
    char msg[255];
}

typedef struct5 struct_DATA_MSG;

struct struct6{
    char ip[16];
    int tipo;
    char map[255];
}

typedef struct6 struct_DATA_MAP;

struct struct7{
    char ip[16];
    char map[255];
}

typedef struct7 struct_DATA_CONNECT;

struct struct8{
    int idHit;

    int idObjetc;
```

```
int typeObjetc;  
int status;  
  
char distance[16];  
}  
  
typedef struct8 struct_DATA_REPLY_HIT;  
  
#endif
```

Supported graphic formats

Vectorial format:

- **Autodesk DXF (*.dxf)**, R11/R12/LT2 versions

Raster format:

- **TIFF images (*.tif; *.tiff)**
- **GFI fax images (*.fax)**
- **SGI images (*.bw, *.rgb, *.rgba, *.sgi)**
- **Autodesk images files (*.cel; *.pic)**
- **Truevision images (*.tga; *.vst; *.icb; *.vda; *.win)**
- **ZSoft Paintbrush images (*.pcx, *.pcc; *.scr)**
- **Kodak Photo-CD images (*.pcd)**
- **Portable pixel/gray map images (*.ppm, *.pgm, *.pbm)**
- **Dr. Halo images (*.cut, *.pal)**
- **CompuServe images (*.gif)**
- **SGI Alias/Wavefront images (*.rla, *.rpf)**
- **Standard Windows bitmap images (*.bmp, *.rle, *.dib)**
- **Photoshop images (*.psd, *.pdd)**
- **Paintshop Pro images (*.psp)**

- **Portable network graphic images (*.png)**