GISIZE

User's Guide & Training Manual

GISIZE GEOSPATIAL 2019
FOR WONDERWARE

MAY 11, 2019

LOGITEK
real time solutions

Logitek, S.A.

Ctra de Sant Cugat 63, Esc. B, Planta 1ª

08191 - Rubí - Barcelona - Spain

http://www.logitek.es - http://www.wonderware.es - http://www.GISIZE GEOSpatial.com

Table of Contents

1. Introduction

The objectives of this module are:

- Describe GISIZE GEOSpatial Solution.
- Describe the Training material.
- Introduce the GISIZE Toolbox.
- Explain the system requirements for GISIZE GEOSpatial and Licensing.
- Install the product.
- Describe the architecture.
- Describe the suggested project development workflow.
- Create a startup project as an introduction to the solution.

1.1. Overview

This section introduces the solution, provides the concepts and terminology associated with GISIZE GEOSpatial, and explains how to configure the components to create flexible architectures that meet the different needs of users and projects.

This section also introduces the hands-on labs available in this guide to provide and reinforce the necessary knowledge to install and maintain the product, to create and configure map providers, navigation models, layers, markers, shapes, containers and languages, security settings and how to use GISIZE GEOSpatial with InTouch for System Platform, InTouch OMI and InTouch standalone.

1.1.1. Introduction

GISIZE GEOSpatial for Wonderware provides a software architecture designed to enhance the visualization and navigation of spatial or geographical data in a Geographic Information System (GIS).

GISIZE GEOSpatial provides an interactive add-on for Wonderware System Platform and InTouch Standalone with a set of wizards and visual controls to use within System Platform and InTouch that allows representing all the elements in a huge installation as spatial or geographical data in a Geographic Information System (GIS). It's possible to use the standard map providers (OpenStreetMap, Yahoo, Bing, ArcGIS, Google, etc.) or create your own map or image provider based on vector files.

The key features are:

- Faster project development (better RIO).
- More efficient engineering (greater functionality).
- Simplified and Automated common tasks (fewer problems).
- Extended Wonderware System Platform capabilities (better TCO).

The key points are:

- Increased productivity reduced engineering and deployment time.
- Strong value proposition for Integrators using Wonderware in Infrastructure projects and Geographically Distributed SCADA solutions.

The main features are:

- Interactive navigation with GIS features (zoom, layers, etc).
- Based on standard map/GIS providers or creating a custom map provider based on images or vector files.
- Using typical map providers (OpenStreetMap, Yahoo, Bing, ArcGIS, Google, etc).
- Using OGC Standard (WMS Services) Map Providers, with extended and configurable support to custom URL and Custom Process.
- Using Digital Layout or Vector/Cad files as Map Provider.
- Capable of rendering graphic elements (markers, polygons, polylines and tooltips) with animations.
- Capable of rendering native ArchestrA Graphics with Read/Write capabilities (only in InTouch OMI).
- GUI ready to use (layers selector, map selector, search engine, navigation links, etc.)
- Integrated security (systems and roles to enable or disable the visibility of layers and objects based on the logged user).
- Multilanguage (create your own languages for all the texts created in the system and change it in Runtime).
- Flexible Architecture.
    - Runtime independent from development database.
    - Repository as database or file.
    - Online / Offline mode.
- Aligned with Wonderware strategy.
    - Development through IDE.
    - Using Wonderware Toolkits (OMI SDK for ArchestrA App)
    - Using Wonderware Logger.
    - API oriented (Toolkit).
    - Productivity Tools (export, import, automatic object creation, …).
    - Development and Runtime tools.
    - Real time data integration.
        - ArchestrA Runtime Data (InTouch OMI).
        - MXAccess (InTouch for System Platform).
        - OPC + OIGateway (InTouch Standalone).
    - Templates & Instances.
    - Demos.
        - .NET Controls & ArchestrA Graphics & Demos.
        - ArchestrA App & Wrapper & Demos.

1.1.2. Core Components

GISIZE GEOSpatial provides a software architecture with the following core components:

- GISIZE GEOSpatial Repository:

  The GISIZE GEOSpatial Repository is the complete GISIZE GEOSpatial system consisting of a single logical name (defined by the GISIZE GEOSpatial database) and a collection of map providers, layers, markers, shapes and other components.

- Map Provider

  A GEOSpatial database with Geo-registered map images

- Markers

  A pointer to a specific coordinate on a map that is represented as an image

- Shapes

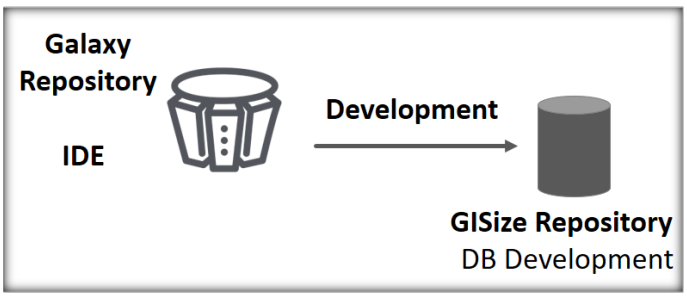
  A collection of specific coordinates on a map that are represented as a polyline if they are open or a polygon if they are closed

- Layer

  Can be conceptually interpreted as a transparent level where makers and shapes are attached. All marker and shape instances belong to a layer.

1.1.3. Architecture

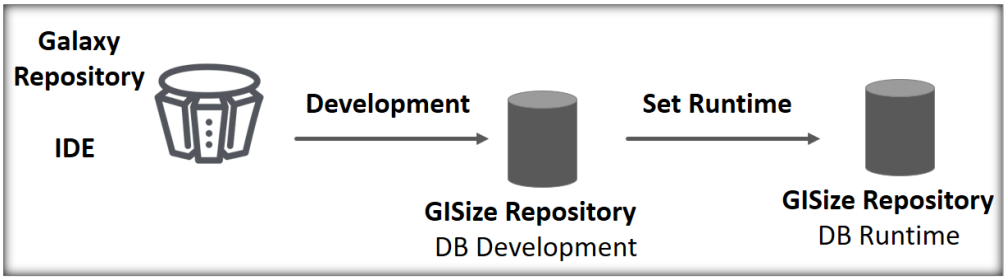
The GISIZE GEOSpatial development tools are embedded in the ArchestrA IDE within the GISIZE Toolbox. This tools are configured to use the local machine as repository server, though it generally uses the Galaxy Repository (GR) platform, but it's also possible to have the repository in an external SQL Server.



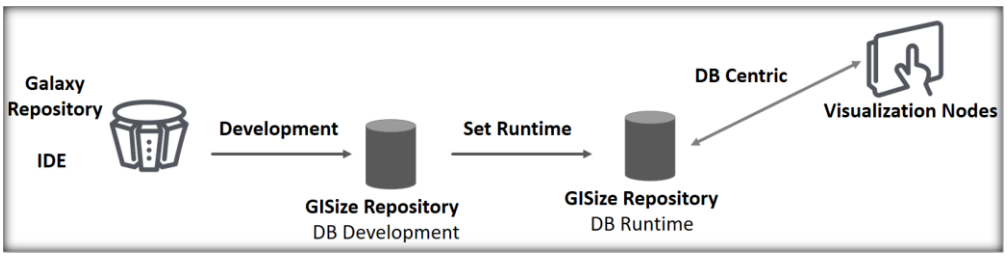
The GISIZE GEOSpatial components are configurable, allowing the creation of flexible architectures that meet the different needs of users and projects.

Runtime independent from development database

The GISIZE GEOSpatial Tools works always with a repository considered as a development environment, but there is the possibility to create a duplicated repository that can be used only at runtime.

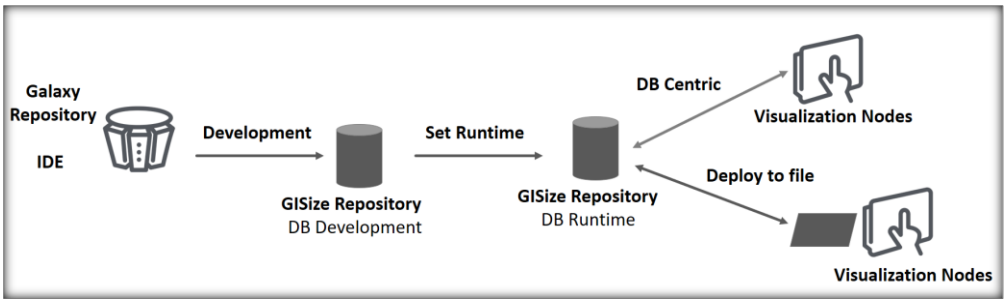


This feature allows the developer to have 2 environments running in parallel, one for development or testing purposes without affecting the second environment in production.



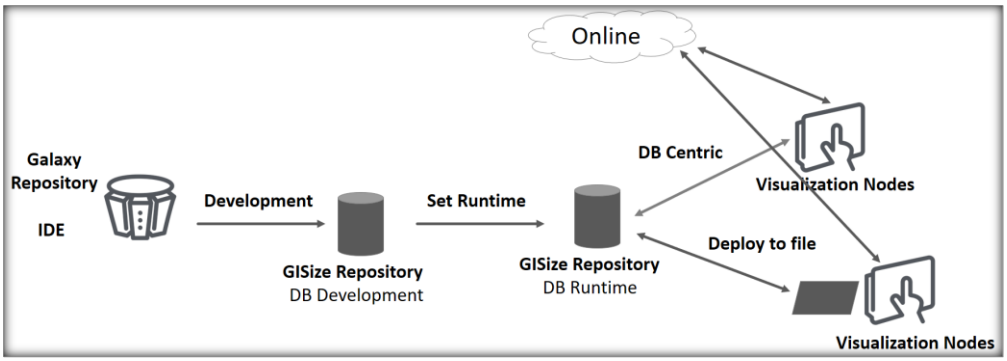
Repository as database or file

In environments without access to an SQL server, there is also the possibility to create a standalone repository file independent from any repository server, which also can be used as runtime repository for a GISIZE GEOSpatial control.

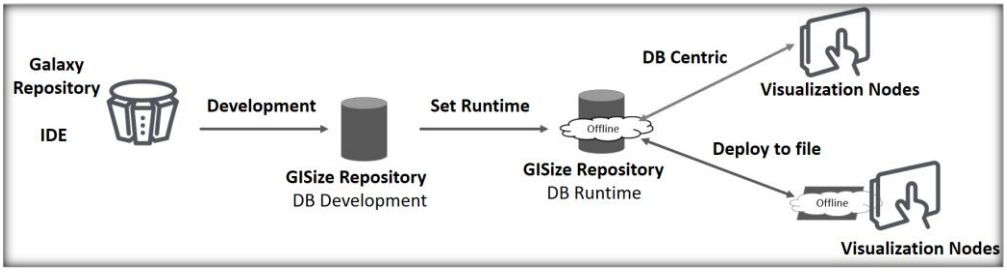


Online / Offline mode

The visualization nodes can be connected to the Map Server if there is connectivity.



There is also the possibility to download specific areas inside the GISIZE GEOSpatial repository to work offline, without connectivity to the Map Server.



In this scenario, an isolated visualization node can have the full GISIZE GEOSpatial repository and the maps locally without connectivity to SQL Server o Map Server.

1.2. Course introduction

The GISIZE GEOSpatial 2019 for Wonderware course is a 1-day, instructor-led class designed to provide a fundamental understanding of how GISIZE GEOSpatial enhance the visualization and navigation of spatial or geographical data in a Geographic Information System (GIS).

1.2.1. Hands-on labs

This User's Guide offers hands-on labs to provide and reinforce the necessary knowledge to install and maintain the product, to create and configure map providers, navigation models, layers, markers, shapes, containers and languages, security settings and to use GISIZE GEOSpatial with InTouch for System Platform, InTouch OMI and InTouch standalone.

1.2.2. Objectives

Upon completion of this labs, you will be able to:

- Create and maintain repositories.
- Create and configure map providers using standard map providers or WMS servers or create your own map provider using a vector file.
- Create and configure layers and navigation models.
- Model the repository using markers and shapes.
- Integrate GISIZE GEOSpatial controls in InTouch for System Platform, InTouch OMI and InTouch standalone.
- Configure multi-language and security settings.

1.2.3. Audience

Application developers, engineers, system integrators, consultants, and other individuals who need to use GISIZE GEOSpatial in infrastructure projects.

1.2.4. Prerequisites

- Knowledge of using the ArchestrA IDE.
- Experience using and creating ArchestrA Symbols.
- Basic knowledge of using and creating scripts.
- Familiarity with the Deployment Model and the Plant model.
- Knowledge of InTouch OMI.

1.2.5. Training files

The Training files used in the labs can be found in the 'Program Files/GISize/Training' path of the computer after install the product.

The Wonderware packages are exported from System Platform 2017 Update 3, so you need at least this version to use these packages.

1.2.6. Course Outline

1 - Introduction

1.1 - Overview

This section introduces the solution, provides the concepts and terminology associated with GISIZE GEOSpatial, and explains how to configure the components to create flexible architectures that meet the different needs of users and projects.

1.2 - Course Introduction

This section describes the GISIZE GEOSpatial for Wonderware 2019 course, its objectives, intended audience prerequisites, and the course agenda.

1.3 - System requirements and Licensing

This section describes the Wonderware products compatible with GISIZE GEOSpatial, the system requirements necessary to use the product, the licensing model and the additional requirements to use GISIZE GEOSpatial.

1.4 - Installation

This section describes the process to install and import the objects needed to use it.

1.5 - Lab 01 - Installing the product

1.6 – The GISIZE Toolbox

This section provides an introduction to the GISIZE Toolbox and its options.

1.7 - Administration Panel

This section explains the operations that affect the whole repository.

1.8 - Lab 02 - Getting Started

2 - Map Providers

2.1 - Overview

This section explains the different types of maps and how to configure it.

2.2 - Online Map Providers

This section explains how to connect to a WMS Server that uses the OGC Standards.

2.3 -Lab 03 - Creating an online map provider

2.4 -Lab 04 - Creating an online map provider with MapBox

2. 5 - Vectorial Map Providers

This section explains how to create a custom Map Provider based on a SVG file.

2.6 - Lab 05 - Creating a vectorial map provider

1.3. System requirements and licensing

This section describes the Wonderware products compatible with GISIZE GEOSpatial, the system requirements necessary to use the product, the licensing model and the additional requirements to use GISIZE GEOSpatial.

1.3.1. Compatibility

GISIZE GEOSpatial 2019 is compatible with:

- Wonderware System Platform 2012 R2 and later versions
- InTouch OMI 2017 Update 2 and later versions
- Wonderware InTouch 2014 R2 and later versions

GISIZE GEOSpatial 2019 incorporates the .NET Framework 4. For this reason, this release is only compatible with:

- .NET Framework 4.0 and later versions.
- Application Server 2012 R2 and later versions.

1.3.2. Operating Systems

GISIZE GEOSpatial 2019 for Wonderware Application Server is supported on all operating systems that are supported by Application Server 2012 R2 and later versions.

1.3.3. Hardware Requirements

To run GISIZE GEOSpatial 2019 for Wonderware Application Server, we recommend as a minimum the following hardware configuration:

- Computer with 2 gigahertz (GHz) or higher dual core processor (32-bit and 64-bit) clock speed.
- 4 gigabyte (GB) of RAM or higher (3 GB minimum supported; may limit performance and some features)
- 10 gigabytes (GB) of available hard disk space
- XGA (1024 x 768) or higher-resolution video adapter and monitor
- Keyboard and mouse or compatible pointing device.

1.3.4. Licensing

Specific license is required to use GISIZE GEOSpatial 2019 for Wonderware Application Server at run time, although is it not required for configuration or at developing time. A valid license must be applied with the 'License Utility' located at GISIZE GEOSpatial folder at Windows menu.

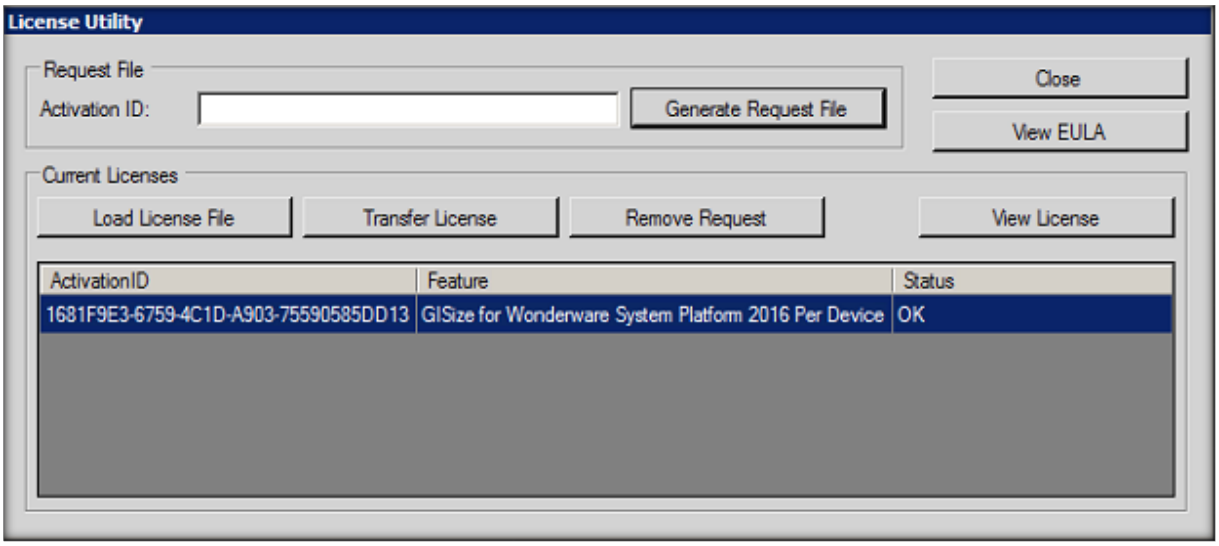
A GISIZE GEOSpatial control running without a license in the machine will show different watermark texts and messages indicating that the product is not licensed.

1.3.4.1. License Utility

To use the GISIZE GEOSpatial control at runtime it is necessary to have a license. This chapter will explain how to request and apply a license.

The first step is open the License Utility. This can be done by clicking the License button at the GISIZE toolbox or by the Windows' start menu navigating to 'All Programs/GISIZE/License Utility'.

Once opened, the License Utility will look like this:



The bottom table will show the current requests/licenses and their status. By selecting a license and clicking the button license will open a popup window with the details of the selected license.

To obtain a valid license, a request file must be generated with a valid Activation ID. To do that, type the Activation ID in the text field and click Generated Request File button. After a few seconds a request file (GRE) will be generated at the desired location.

Once obtained a request file (GRE), this file must be send to the vendor who will generate and send back a valid license file (GLI).

Once obtained a license file (GLI), the license must be loaded in corresponding computer. To do that, click the button Load License File and indicate the license file. The license will be applied successfully as far as the previous request is found and no hardware modification is made.

An active license can be transferred to another computer. To do that previously it must be removed from the current computer. Click in the 'Transfer License' button to remove the license and generate a transfer license request file (GTE). This file must be send to the vendor who will free the license, allowing to use it in another request petition.

A request petition can be removed clicking the Remove Request button.

1.3.4.2. GISIZE GEOSpatial for Wonderware System Platform

GISIZE GEOSpatial for Wonderware System Platform license is based on the I/O count of the System Platform Galaxy and the total number of System Platform Visualization Clients.

A GISIZE GEOSpatial for Wonderware System Platform license is limited by:

- (i) the size of the Application Name Space, which in turn is limited by the I/O Count, or the Site Count, as defined in the Wonderware License File, Wonderware License Certificate or Wonderware Order Form, and
- (ii) the total number of System Platform Visualization Clients, as the sum of the InTouch for System Platform licenses and the InTouch for System Platform RDS licenses, as defined in the Wonderware License File, Wonderware License Certificate or Wonderware Order Form.

Additional requirements:

- GISIZE GEOSpatial for Wonderware System Platform includes application development tools that are used to develop applications for deployment of the Wonderware System Platform Software. GISIZE GEOSpatial for Wonderware development tools works in combination with a Wonderware Development Studio or Advanced Development Studio license. The development tools may only be used on the same device as the Wonderware Development Studio license.
- GISIZE GEOSpatial for Wonderware System Platform runtime tools works in combination with a Wonderware Supervisory Client license. The runtime tools may only be used on the same device or server as the Wonderware Supervisory Client license.
- GISIZE GEOSpatial for Wonderware System Platform supports several MS SQL Server versions and editions (see ReadMe file) - Customers need to purchase separately or own Microsoft SQL Server and CAL licenses.

When a MS SQL Server license delivered with other Wonderware products is used for the GISIZE GEOSpatial for Wonderware database as well, customers have the following purchase options to secure proper Microsoft CAL licensing:

- WW Core w MS Core licenses from WW CALs sheet
- WW CAL w MS CAL from WW CALs sheet
- SQL Core from Microsoft
- SQL CAL from Microsoft

1.3.4.3. GISIZE GEOSpatial for Wonderware InTouch

GISIZE GEOSpatial for Wonderware InTouch license is based on the tag count of the InTouch license.

A GISIZE GEOSpatial for Wonderware InTouch license is limited by the tag count of the InTouch application defined in the Wonderware License File, Wonderware License Certificate or Wonderware Order Form.

For additional clients with GISIZE GEOSpatial for Wonderware InTouch licenses for the same InTouch project:

- The price for the second and third units will be 50% of the price of the original license.
- The price from the fourth unit on will be 25% of the price of the original license.

Additional requirements:

- GISIZE GEOSpatial for Wonderware InTouch includes application development tools that are used to develop applications for the Wonderware InTouch Software. GISIZE GEOSpatial for Wonderware development tools works in combination with a Wonderware Development Studio license. The development tools may only be used on the same device as the Wonderware Development Studio license.
- GISIZE GEOSpatial for Wonderware InTouch runtime tools works in combination with a Wonderware InTouch license. The runtime tools may only be used on the same device or server as the Wonderware InTouch license.
- GISIZE GEOSpatial for Wonderware InTouch supports several MS SQL Server versions and editions (see ReadMe file) - Customers need to purchase separately or own Microsoft SQL Server and CAL licenses.

When a MS SQL Server license delivered with other Wonderware products is used for the GISIZE GEOSpatial for Wonderware database as well, customers have the following purchase options to secure proper Microsoft CAL licensing:

- WW Core w MS Core licenses from WW CALs sheet
- WW CAL w MS CAL from WW CALs sheet
- SQL Core from Microsoft
- SQL CAL from Microsoft

1.4. Installation

This section describes the process to install the product and import the objects in the Galaxy.

1.4.1. Installation process

The application is supplied as an installer that can be executed from any folder with a user with administrator rights on the machine

During the installation, the EULA must be read and accepted. Also it is mandatory to have the ArchestrA IDE closed, otherwise installation will halt and ask to close it.

Please note that if you are migrating from a previous version to 2019, the names of Wonderware objects for InTouch OMI have changed.

To minimize the changes to be made in the galaxy, it is recommended to follow the below procedure:

- *Take a backup of the Galaxy*
- *Export the objects:* **GISizeWPF**, **GISizeWrapper**, **GISizeWPF_Beta**, **GISizeWPFWrapper_Beta**, **GISize_WPFSettings**, **GISizeManager**
- Rename the current objects:
  - **GISizeWPF** to **GISizeWPF_Legacy**
  - **GISizeWrapper** to **GISizeWrapper_Legacy**
  - **GISizeWPF_Beta** to **GISIZE_GEOSpatialApp**
  - **GISizeWPFWrapper_Beta** to **GISIZE_GEOSpatialWrapper**
  - **GISizeWPFSettings** to **GISIZE_GEOSpatialSettings**
  - **GISizeManager** to **GISIZE_GEOSpatialManager**
- Delete the renamed objects **GISizeWPF_Legacy**, **GISizeWrapper_Legacy**, **GISIZE_GEOSpatialApp**, **GISIZE_GEOSpatialWrapper**, **GISIZE_GEOSpatialSettings**
- Import the new package available in the installation folder

Afterwards, you must choose the type of installation, between 'Development' or 'Runtime' mode.

- The development mode will install the full product including the integration with IDE.
- The runtime mode will install only the necessary tools for licensing.

If the 'Development' mode is selected, then you can configure the default GIS Repository Server connection in the next screen. This configuration can be changed later at GISIZE Toolbox.

Development installation will create a new folder 'GISIZE GEOSpatial' as Program Files path, containing all necessary binaries of the product, such as:

- Library (.DLL) with the GISIZE GEOSpatial control to use in ArchestrA graphics.
- Library (.DLL) needed to integrate the product with ArchestrA IDE.
- Library (.DLL) with the GISIZE GEOSpatial ArchestrA App for InTouch OMI.
- ArchestrA package with the GISIZE GEOSpatial Toolset.
- Installer and uninstaller auxiliary binaries

- License Utility application.
- Additional settings and files for InTouch OMI.

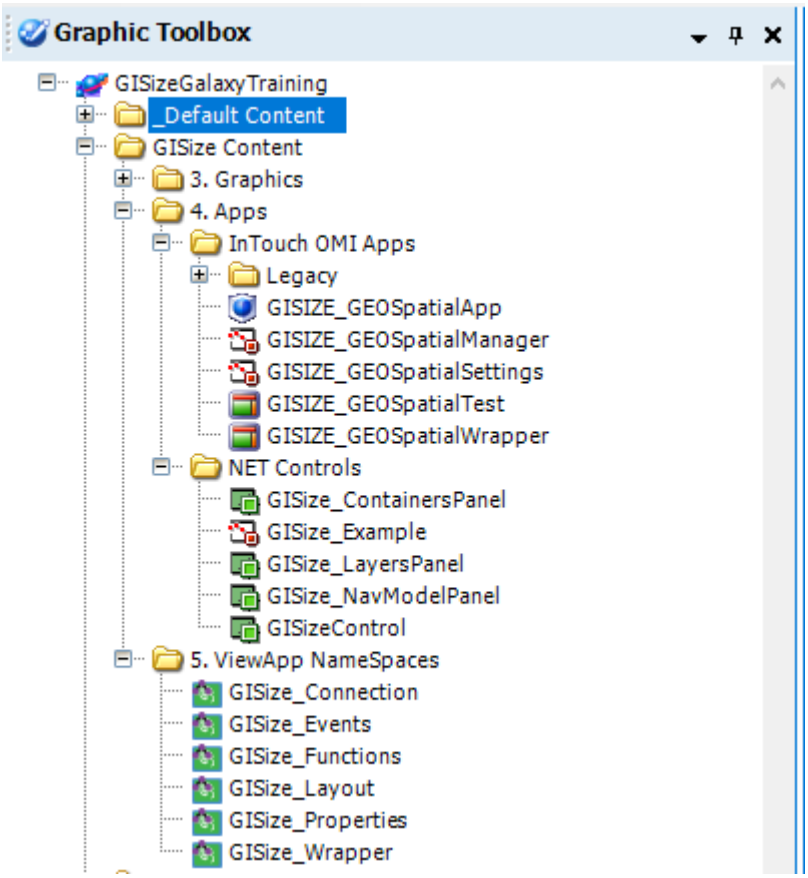
The installation will also add automatically a new tab page named 'GISIZE Toolbox' inside the ArchestrA IDE from where the user can configure and develop.

1.4.2. GISIZE GEOSpatial Graphics

Once the installation is finished, in order to use the GISIZE GEOSpatial control for ArchestrA graphics and/or the GISIZE GEOSpatial ArchestrA App, they must be imported into an ArchestrA Galaxy.

The easiest way to accomplish this is by importing the supplied ArchestrA packages that contains the GISIZE GEOSpatial toolset.

The ArchestrA packages file **Apps\NET Controls\GISIZE_NET_Controls_Example.aaPKG** and **Apps\InTouch OMI Apps\ GISIZE InTouch OMI Apps.aaPKG** can be found in the 'Program Files/GISIZE' path of the computer, and must be imported from ArchestrA IDE with the option Galaxy -> Import -> Object(s)



A new folder named 'GISIZE Content' will appear at Graphic Toolbox tab page. This folder contains the objects to be used in:

- InTouch OMI
  - The GISIZE GEOSpatial ArchestrA APP named **GISIZE_GEOSpatialApp**.
  - ViewApp Namespaces **GISize_Connection, GISize_Events, GISize_Functions, GISize_Layout, GISize_Properties, GISize_Wrapper** to interface with the GISize ArchestrA App.
  - A layout named **GISIZE_GEOSpatialWrapper** that contains the ArchestrA App with its properties linked to ViewApp attributes.
  - An ArchestrA Graphic named **GISIZE_GEOSpatialSettings** that contains the properties available in the GISize ViewApp Namespace to test the behavior of the ArchestrA App.

- An ArchestrA Graphic named **GISIZE_GEOSpatialManager** that contains an example initializing the ViewApp Attributes and monitoring their values to show graphics according the user interactions.
- A layout named **GISIZE_GEOSpatialTest** that contains a center empty pane and the **GEOSpatialSettings** to create a test ViewApp quickly.

- NET Controls (InTouch and InTouch for System Platform).
  - The GISIZE GEOSpatial controls named **GISizeControl**, **GISize_ContainersPanel**, **GISize_LayersPanel** and **GISize_NavModelPanel**.
  - An ArchestrA Graphic named **GISize_Example** that contains an example of usability of the GISIZE GEOSpatial control, ready to be used at an InTouch application.

1.4.2.1. Important note about importing GISizeControl

Please delete an existing GISizeControl object from the Graphic Toolset before attempting to import an aaPKG containing a GISizeControl.

If not done, in some occasions, existing ArchestrA objects may be incorrectly marked as "software upgrade needed" and will be required to redeploy.

1.5. Lab 01 - Installing the product

1.5.1. Introduction

In this lab, you will install the product.

1.5.2. Objectives

Upon completion of this lab, you will be able to:

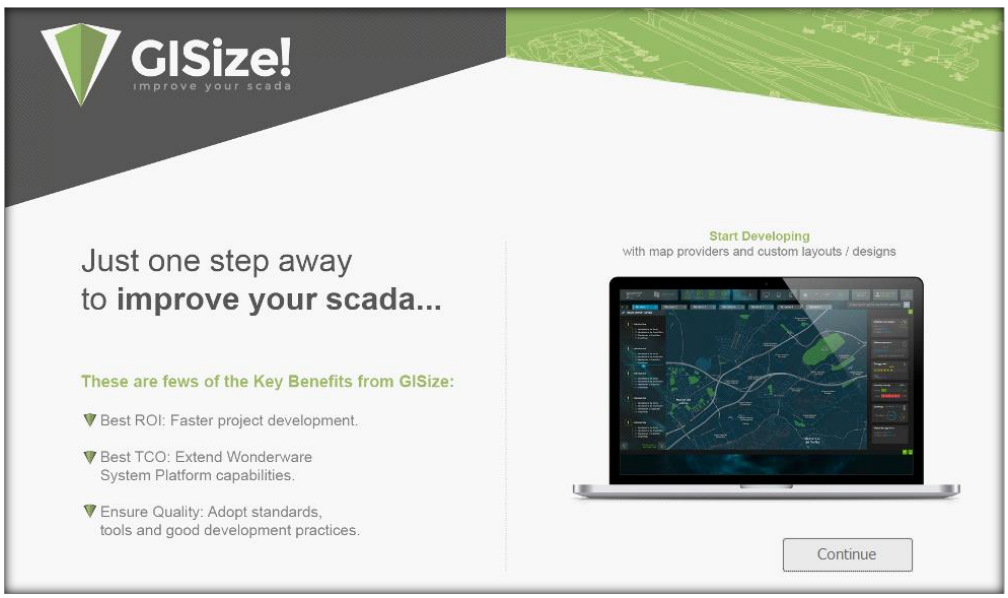
- Install GISIZE GEOSpatial

1.5.3. Install the product

In the following steps, you will install the product:

1. Close the Wonderware utilities (ArchestrA IDE, System Platform Management Console, InTouch View) if open.
2. Execute the installer **GISIZE_GEOSpatial_Setup_2019.y.z.vvvv.msi** with a user with administrator rights on the machine.

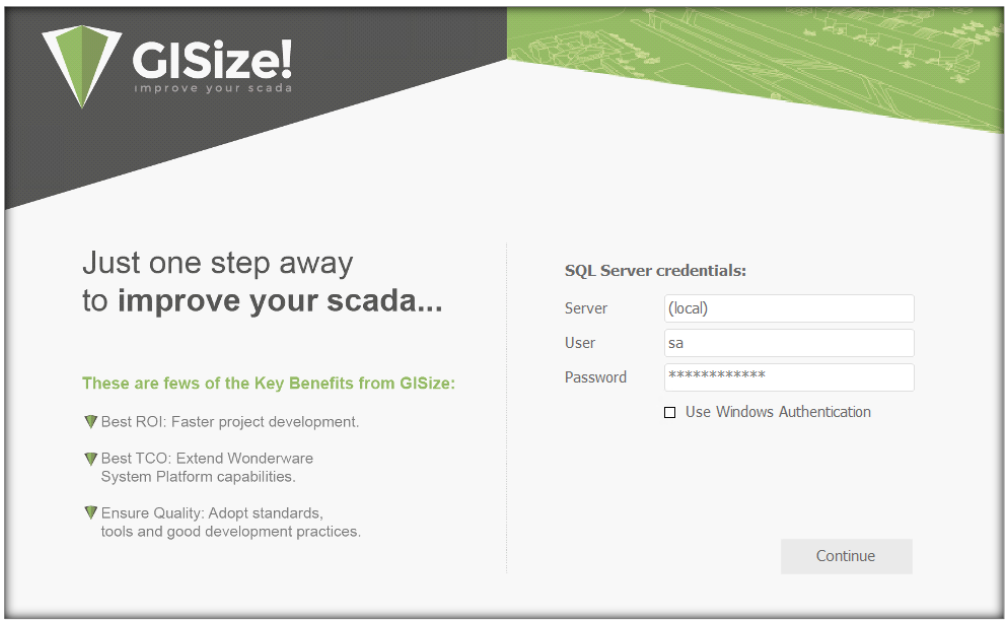
   The installation process starts copying the required files to a local folder and after this the **Installation** dialog box appears.
3. Click the **Continue** button to continue the process.



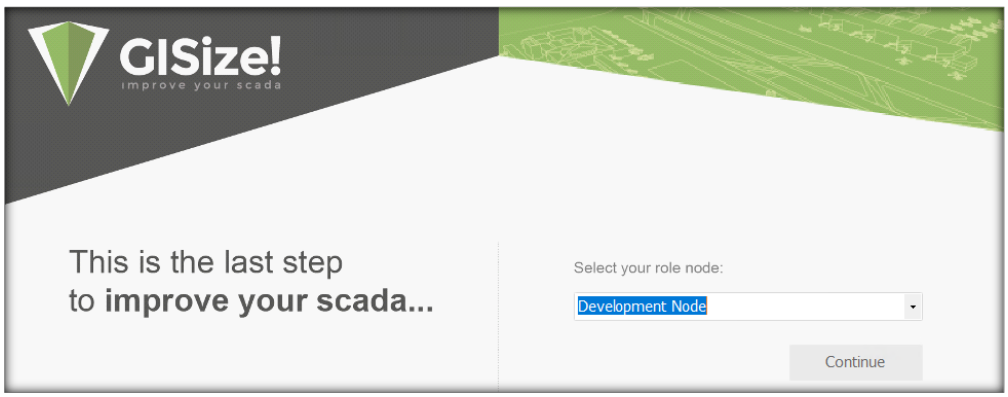
4. Read the **End User License Agreement** and click the **I Agree** button to continue the process.

5. **Please note that if you are migrating from a previous version to 2019, the names of Wonderware objects for InTouch OMI have changed.**

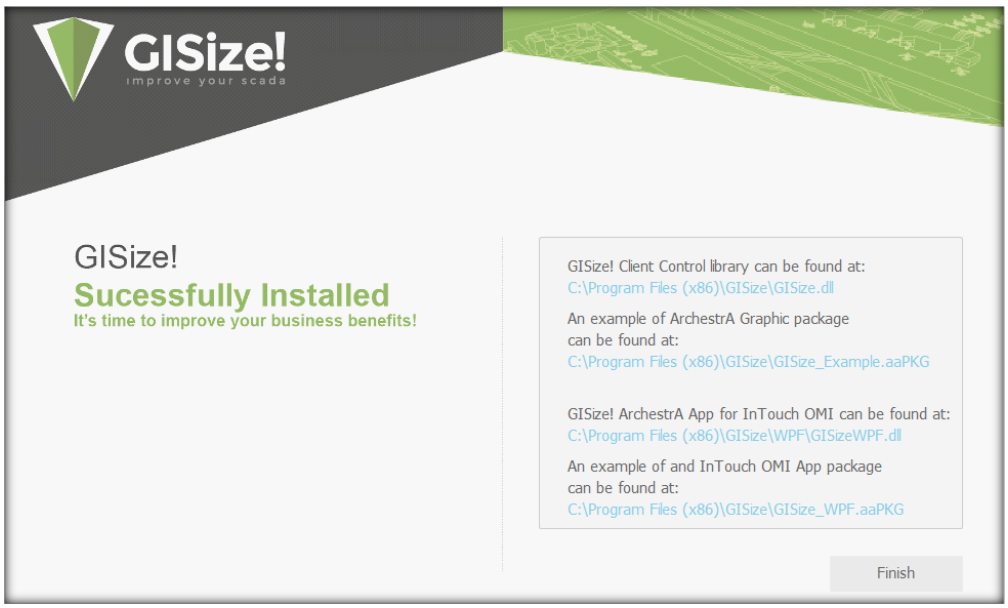
6. Configure the **SQL Server credentials** or click the **Use Windows Authentication** button if you will use the logged user credentials.



7. Select the option **Development Node** and click the **Continue** button to continue the process. The installation process starts. This will take a few moments.



8. When complete, the **Successfully Installed** message appears. Click the **Finish** button to complete the process.



1.6. The GISIZE Toolbox

This section provides an introduction to the GISIZE Toolbox and its options.

1.6.1. Overview

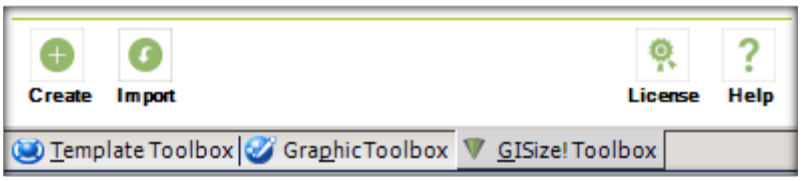
A new tab page called 'GISIZE Toolbox' will automatically appear in ArchestrA IDE after the installation of the product.

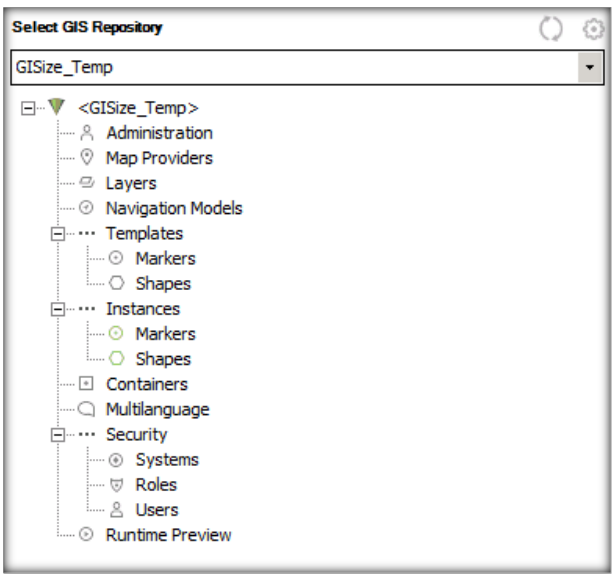
This toolbox allows complete configuration and development with the product within the ArchestrA IDE through GISIZE GEOSpatial repositories.

As its name suggests the GISIZE Toolbox is a new toolbox inside ArchestrA IDE like Graphic or Template Toolbox.

At the bottom of the GISIZE Toolbox lays four buttons of direct actions. From left to right they are:

- Create: To create a new GISIZE GEOSpatial repository.
- Import: To import a previously exported repository.
- License: To open the licensing utility.
- Help: To show a CHM help file about GISIZE GEOSpatial.





The combo box at the top allows the selection of the GISIZE GEOSpatial repositories that exists in the configured repository server.

Upon selection of a GISIZE GEOSpatial repository in the combo box, a check of the repository version will be made. If the system detects that the version of the repository is lower than the current version of GISIZE GEOSpatial 2019 the user will be asked to confirm a version upgrade of the repository. If the user declines the version upgrade, the repository will remain inoperative both in design and in runtime.

After selecting a GISIZE GEOSpatial repository, a tree view is shown with the different functionality groups to operate with the repositories. Double clicking on each node of the tree view opens the form of the following functionalities.

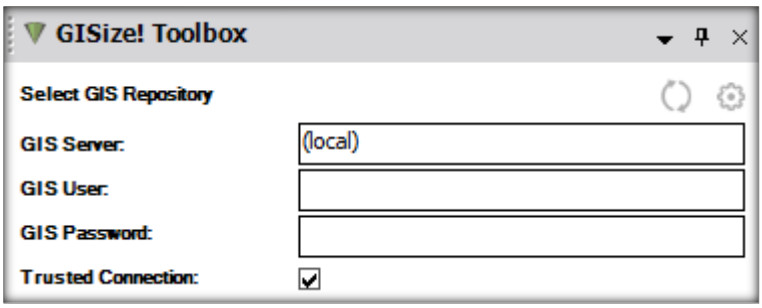
- **Administration**: Operations that affect the whole repository
- **Map Providers**: Management of the different map providers.
- **Layers**: Management of the layers containing markers and shapes instances.
- **Navigation Models**: Management of predefined navigation configurations.
- **Templates**: Management of Marker and Shape templates (classes).
- **Instances**: Management of the template's derived instances.
- **Multilanguage**: Management of multi-language visible texts.
- **Systems**: Management of groups of layers for security.
- **Roles**: Management of security roles for users.
- **Users**: Management of ArchestrA users with GISIZE GEOSpatial security.

1.6.2. Configuration

The GISIZE GEOSpatial repositories available for the GISIZE Toolbox rely on the repository server configured. A repository server is an SQL server where the repositories are stored.

By default, the GISIZE Toolbox its configured to use the local machine as repository server, though it generally uses the Galaxy Repository (GR) platform.

To configure the repository server where the GISIZE Toolbox works, the cogwheel icon located in the upper right corner of the toolbox must be clicked.



Upon clicking the cogwheel, a textbox with the current repository server used will appear. It can be replaced by typing the name (or IP direction) of the SQL server that acts as the repository server.

Clicking the refresh icon next to the cogwheel icon will update the content of the combo box with the GISIZE GEOSpatial repositories available in the configured repository server.

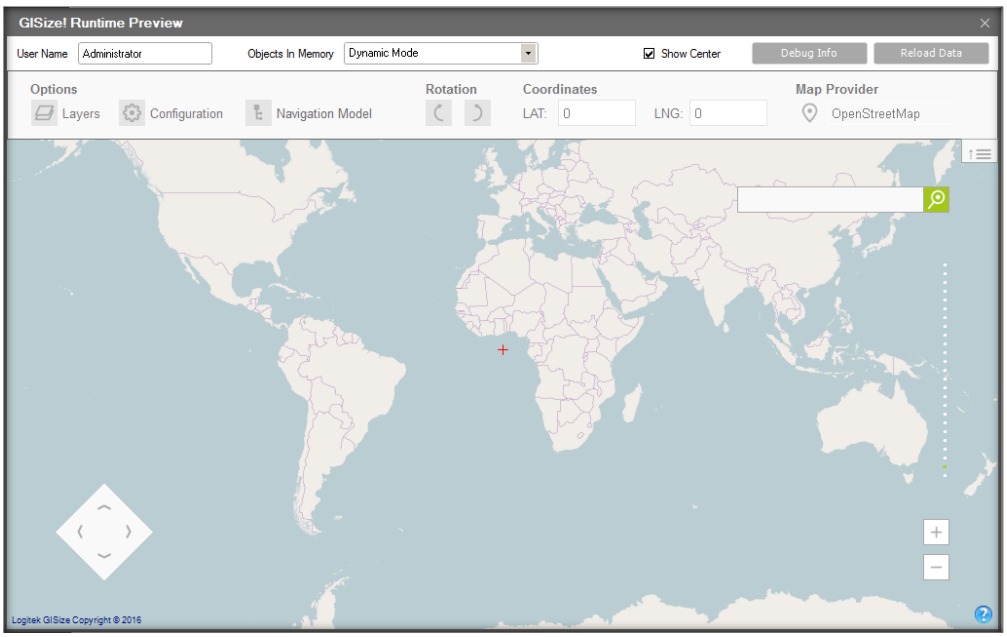
The configuration of the GISIZE Toolbox repository server is made by the Window's user logged in the system. That means that every user will have its own configuration.

Note that the connection to SQL server that acts as repository server is always through Windows Authentication, meaning that the currently logged user must have permissions to log in the SQL server.

1.6.3. Runtime Preview

The Runtime Preview window allows visualizing the current repository configuration of a GISIZE GEOSpatial control. This is a fast way to verify repository modifications without the need to run a GISIZE GEOSpatial control inside an InTouch application.

The window can be maximized by double clicking the top gray bar.



The Runtime Preview window features a GISIZE GEOSpatial control with the default user interface plus some exclusive controls:

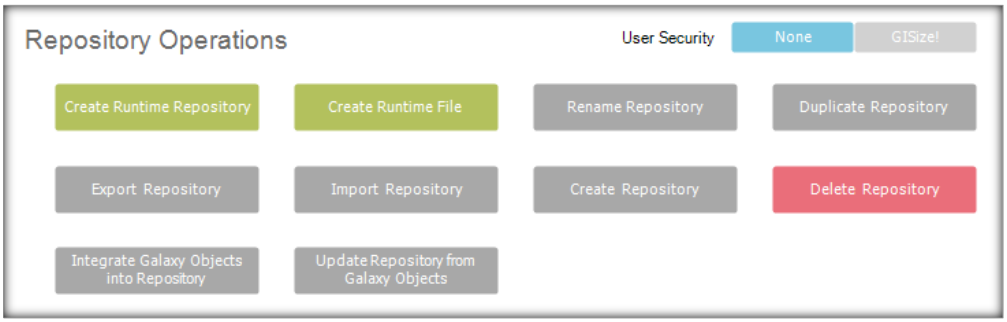
- **User Name**: Allows the simulated log of a security user. This is a way to test the behavior of the security configuration.
- **Objects in Memory**: A GISIZE GEOSpatial control can manage the memory in 3 different ways with the configuration of a parameter. This control allows the user to test the behavior of the memory management with the current repository. The 3 options are:
  - **Dynamic Mode**: The GISIZE GEOSpatial control only loads the required objects (layers, markers, shapes, etc…) whenever they are visible on the screen, and unloads them when they are not on the screen. This uses less memory than the other options but it is the slowest option to visualize objects. This is the default option.
  - **Static Mode**: The GISIZE GEOSpatial control loads all the required objects (layers, markers, shapes, etc…). The objects are unloaded when no longer needed. With this option the visualization of the objects is faster but the use of memory is bigger.
  - **Static Never Unload Mode:** The GISIZE GEOSpatial control loads all the required objects (layers, markers, shapes, etc…). Once loaded the object are never unloaded from memory. This is the fastest option to visualize the

objects but it uses more memory than other two modes. Only recommended for small repositories.

- **Show Center**: Shows/hides the central cross.
- **Debug Info**: This button opens a small window with debug information regarding object management and memory/CPU use.
- **Reload Data**: Unloads and reloads again the GISIZE GEOSpatial control with the repository configuration. Useful to verify changes.

1.7. Administration Panel

The Administration panel, also known as Repository Operations panel, is where repository operations related to the repository server are performed. Also, here it is indicated if the repository has the GISIZE GEOSpatial user security activated or not.



At the top right corner of the form, there are two buttons to enable or disable the GISIZE GEOSpatial User Security within the current repository. When the security is activated, the access to the operations of the repository will only be available to the ArchestrA Users with the right permissions.

By default, only the ArchestrA user 'Administrator' will have full rights to access all the functionalities.

There are several buttons in the form that perform different operations related to the repository server or the repository itself. From left to right and top to bottom, these buttons are:

Create Runtime Repository

This button creates a duplicate of the current repository but marking it as a runtime repository. Only runtime repositories can be used at runtime, unless a GISIZE GEOSpatial control is configured otherwise. An existing runtime repository with the same name will be overwritten.

Other than this operation, no other operations can be performed against runtime repositories, as they are intended to be closed repositories for runtime execution.

ArchestrA Network Account identification will be required to perform this operation if the configured GIS Repository Server is not (local).

Create Runtime File

Creates a standalone repository file independent from any repository server, which can be used as runtime repository for a GISIZE GEOSpatial control. This is useful for environments without access to an SQL server.

The resulting GISIZE GEOSpatial Runtime Repository (.GRR) file must be configured properly in a GISIZE GEOSpatial control as the source repository.

ArchestrA Network Account identification will be required to perform this operation if the configured GIS Repository Server is not (local).

Rename Repository

This button allows the user to rename the current repository. Note that this action will break any existing GISIZE GEOSpatial control configuration, which will still be pointing to the old named repository. The configuration of any GISIZE GEOSpatial control must be changed manually.

A repository cannot be renamed as an existing repository name. Renaming a repository does not require further identification.

Duplicate Repository

It creates a duplicate of the current repository with a new chosen name. The new repository cannot be renamed as an existing repository name.

ArchestrA Network Account identification will be required to perform this operation if the configured GIS Repository Server is not (local).

Export Repository

It allows the user to export the current repository to a GISIZE GEOSpatial Repository File (.GRF) file. The purpose of this file is to create a backup of the repository that can be imported later.

Unlike a GISIZE GEOSpatial Runtime Repository (.GRR) file, a GISIZE GEOSpatial Repository File (.GRF) file cannot be used as a source repository in GISIZE GEOSpatial controls.

ArchestrA Network Account identification will be required to perform this operation if the configured GIS Repository Server is not (local).

Import Repository

This button allows the user to import a previously exported repository as GISIZE GEOSpatial Repository File (.GRF) file. The user will be asked to type a new name for the repository being imported. The current repository cannot be overwritten.

Also, the imported repository cannot be renamed as an existing repository name.

ArchestrA Network Account identification will be required to perform this operation if the configured GIS Repository Server is not (local).

Create Repository

In the same way as the functionality located in the GISIZE Toolbox, this button creates a new empty GISIZE GEOSpatial repository.

The newly created repository cannot be named as an existing repository name.

Contrary to the functionality located at the GISIZE Toolbox, this button doesn't require permissions if Galaxy Security is configured, although can be secured with GISIZE GEOSpatial User Security.

ArchestrA Network Account identification will be required to perform this operation if the configured GIS Repository Server is not (local).

Delete Repository

This button allows deleting the current GISIZE GEOSpatial repository. It will ask for user confirmation, as this operation cannot be undone. Deleting a repository does not require further identification.

The only way to recover a deleted repository is to import a previously existing backup (.GRF) file of it.

Integrate Galaxy Objects into Repository

This button opens a wizard window that allows the creation of GISIZE GEOSpatial layers, templates, and instances based on existing areas, templates, and instances in the current ArchestrA Galaxy.

Update Repository from Galaxy Objects

This button opens a wizard window that allows the update of GISIZE GEOSpatial layers, templates, and instances previously associated with existing areas, templates, and instances in the current ArchestrA Galaxy.

1.8. Lab 02 - Getting Started

1.8.1. Introduction

In this lab, you will license the product and create the first GISIZE GEOSpatial project using the ArchestrA IDE.

1.8.2. Objectives

Upon completion of this lab, you will be able to:

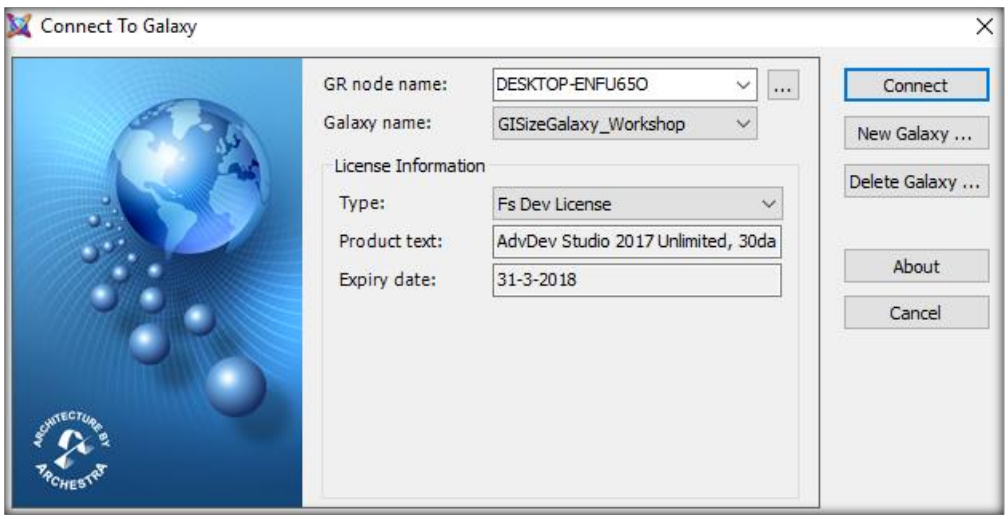
- Create a Galaxy.
- License GISIZE GEOSpatial.
- Create a GISIZE GEOSpatial Repository.
- Create a new Layer.
- Configure a Map Provider.
- Create a basic Maker Template and Instance.
- Create a basic Shape Template and Instance.
- Preview runtime repository.
- Integrate GISize in InTouch OMI
- Use ArchestrA Graphics as Markers

1.8.3. Create the Galaxy

In the following steps, you will create a Galaxy and connect to it.

1. Open the ArchestrA IDE

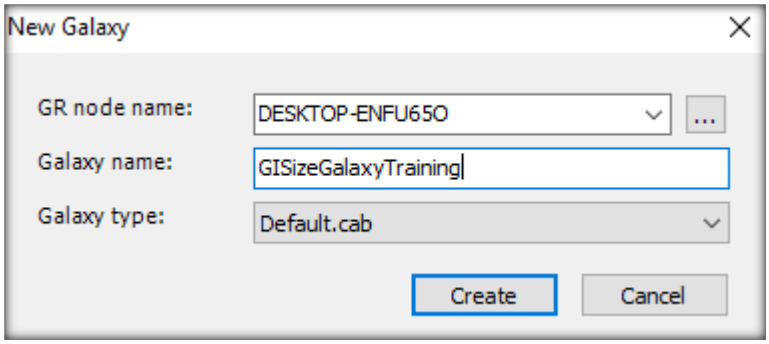
   The **Connect To Galaxy** dialog box appears with the local node name displayed in the **GR node name** drop-down list. Once a Galaxy has been created and accessed, the last GR node name connected to will be shown by default.



2. Click the **New Galaxy** Button to create a new Galaxy

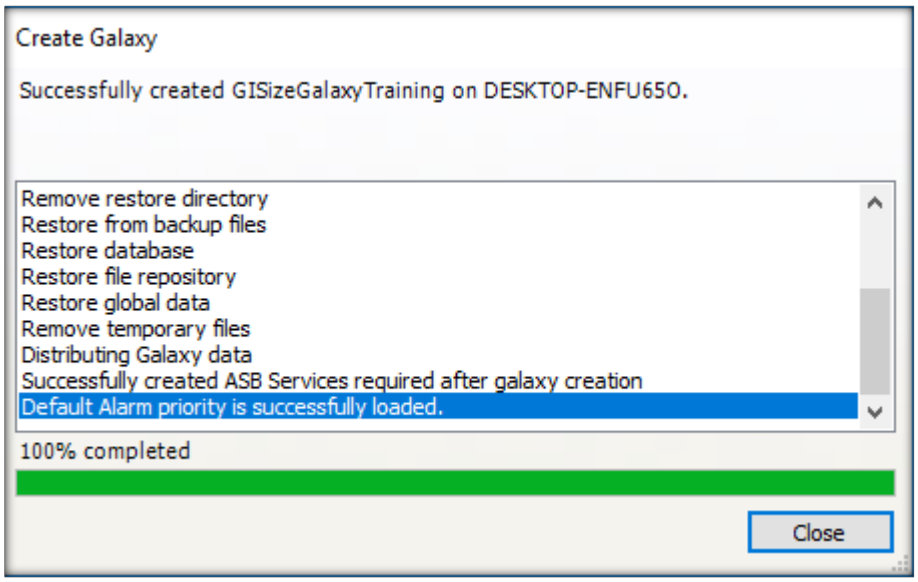
The **New Galaxy** dialog box appears.

3. In the **Galaxy name** field, enter **GISizeGalaxyTraining.**



4. In the **Galaxy type** field, confirm **Default.cab** is selected.
5. Click **Create**.

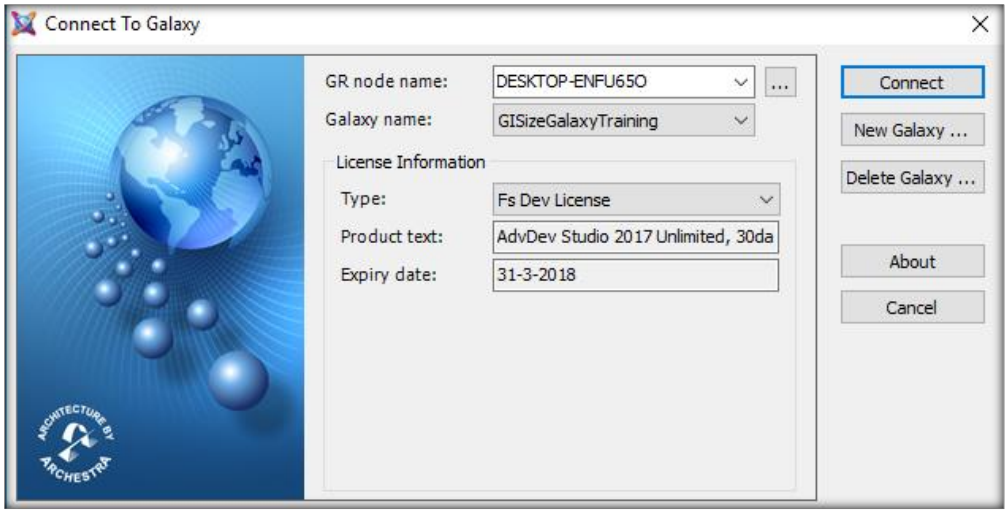
   The **Create Galaxy** dialog box appears and shows the Galaxy creation progress. This will take a few moments.



6. When complete, the **Create Galaxy** progress displays **100% completed**.
7. Click **Close.**

   The newly created **GISizeGalaxyTraining** you entered in the previous steps now appears in the **Galaxy name** drop-down list.

8. Click **Connect.**

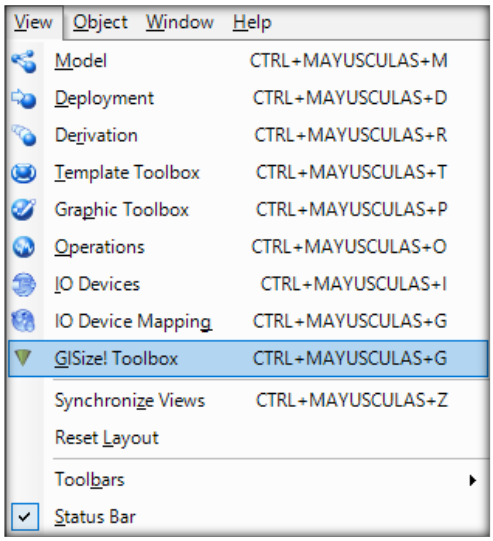


The Connect to Galaxy dialog box closes and, after a few seconds, the ArchestrA IDE opens.

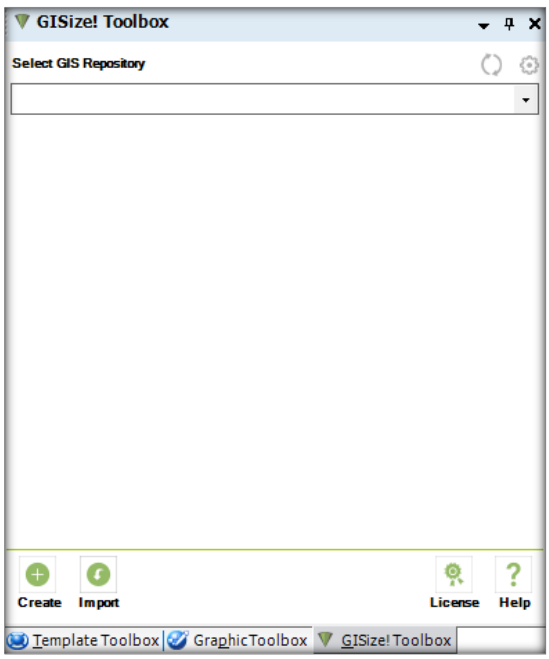
1.8.4. License the product

In the following steps, you will license the product:

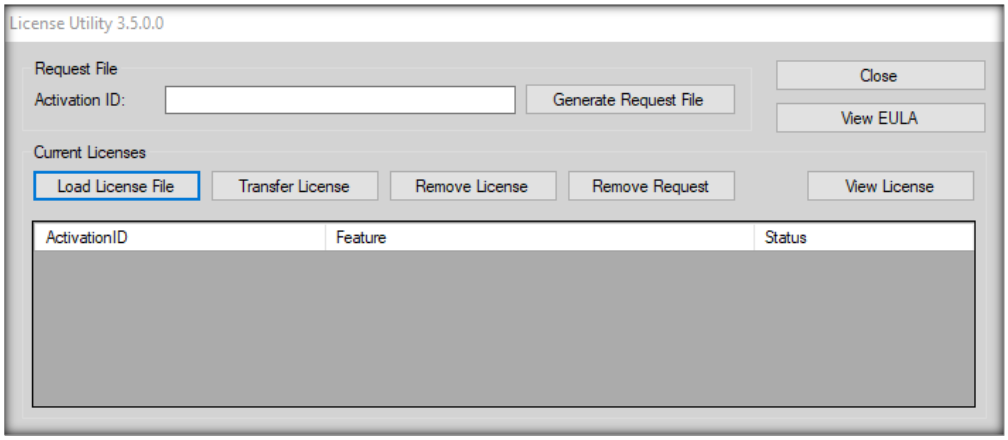
1. Click the **View** menu and select **GISIZE Toolbox**.



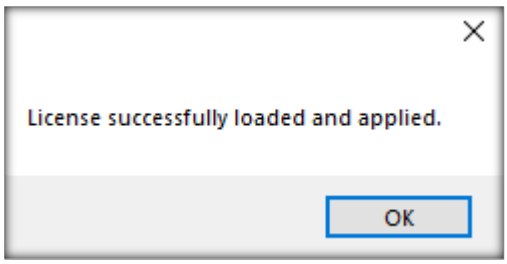
2. Click the **License** button to open the License Utility.



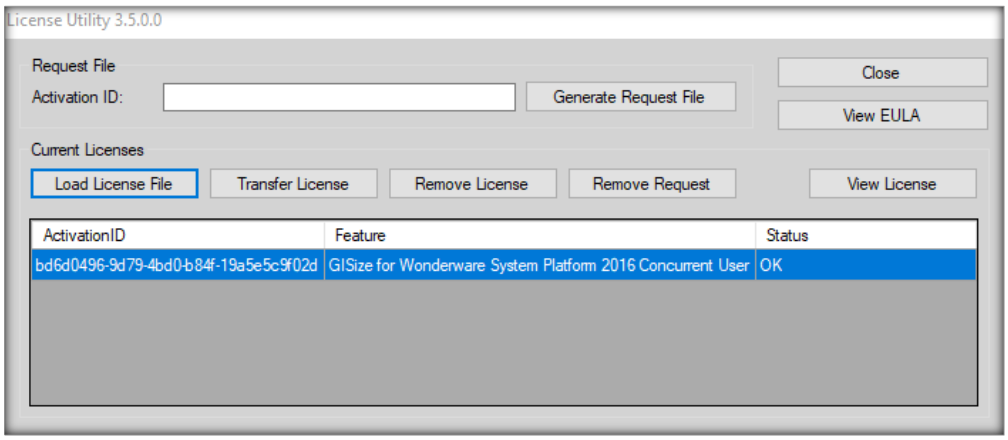
3. Click **the Load License File** button and select the **License** file.



4. When complete, the **License successfully loaded and applied** will appear. Click **OK**.



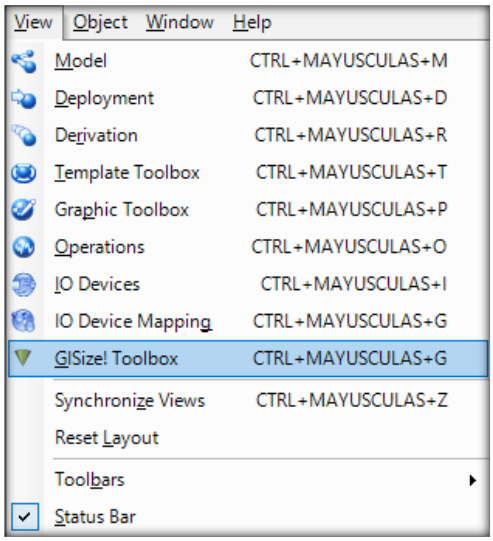
5. Check if the **Status** column is OK and click **Close**.



1.8.5. Create a GISIZE GEOSpatial Repository.

In the following steps, you will create a new GISIZE GEOSpatial Repository:

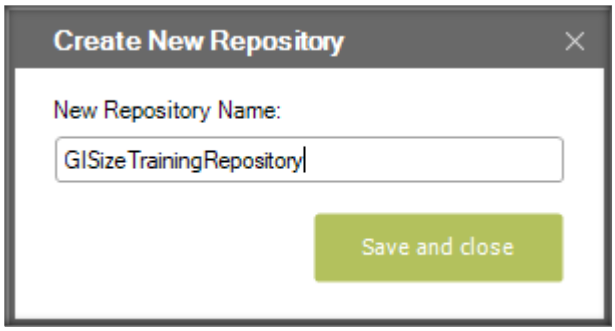
1. Click the **View** menu and select **GISIZE Toolbox**.



2. Click the **Create** button to create a new repository.

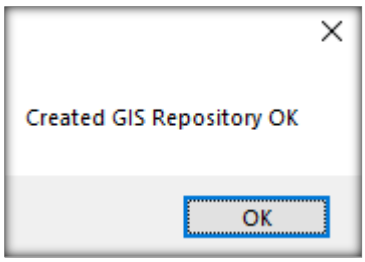
   The Create New Repository dialog box appears.

3. In the **New Repository Name** field, enter **GISizeTrainingRepository**.



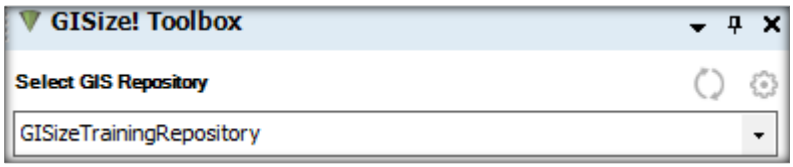
4. Click **Save and close**.

   This process will take a few moments. When complete, the **Created GIS Repository OK** message appears.



5. Click **OK**.

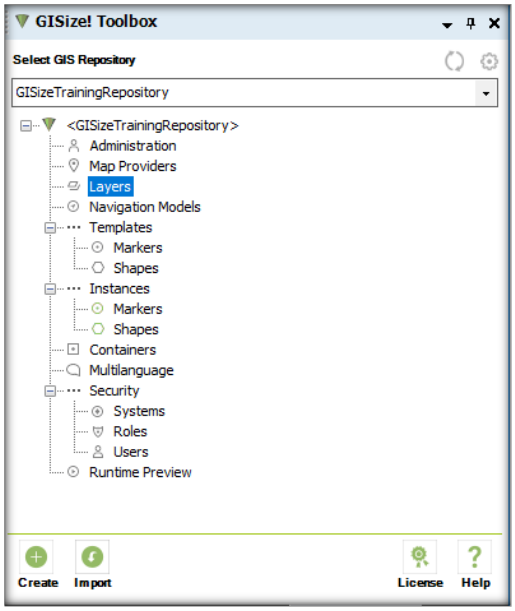
The newly created **GISizeTrainingRepository** you entered in the previous steps now appears in the **Select GIS Repository** drop-down list.



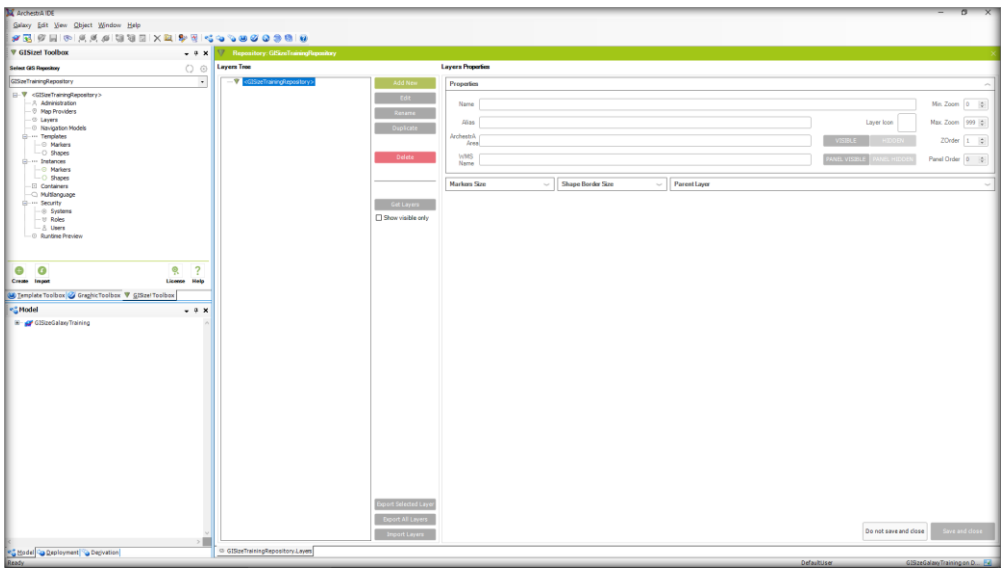
1.8.6. Create a new Layer and Configure a Map Provider

In the following steps, you will create a new layer and configure its availability for a Map Provider.

1. Select the **GISizeTrainingRepository** and double click in **Layers** option.



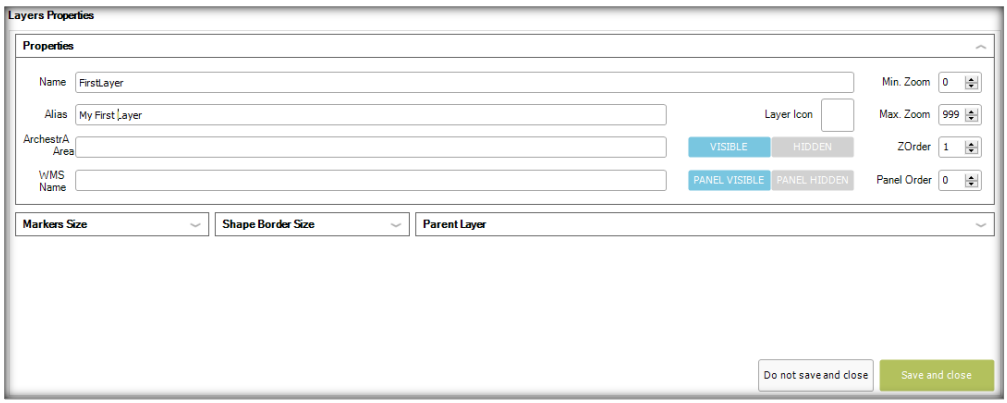
The **Layers** configuration window appears within the ArchestrA IDE.



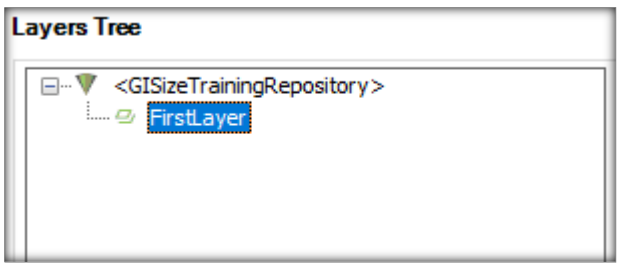
2. Click the **Add New** button to create a new layer.

The **Layer Properties** pane is now enabled to configure the new layer.

3. In the **Name** field, enter **FirstLayer**
4. In the **Alias** field, enter **My First Layer**



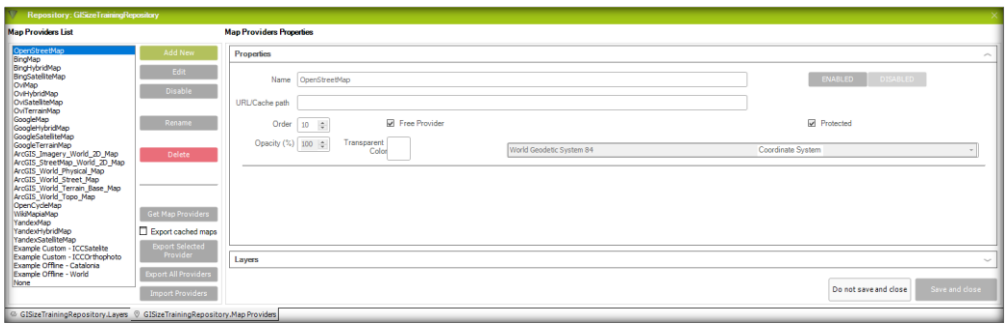
6. Click **Save and Close**.
7. The newly created layer you entered in step 3 now appears in the **Layers Tree.**



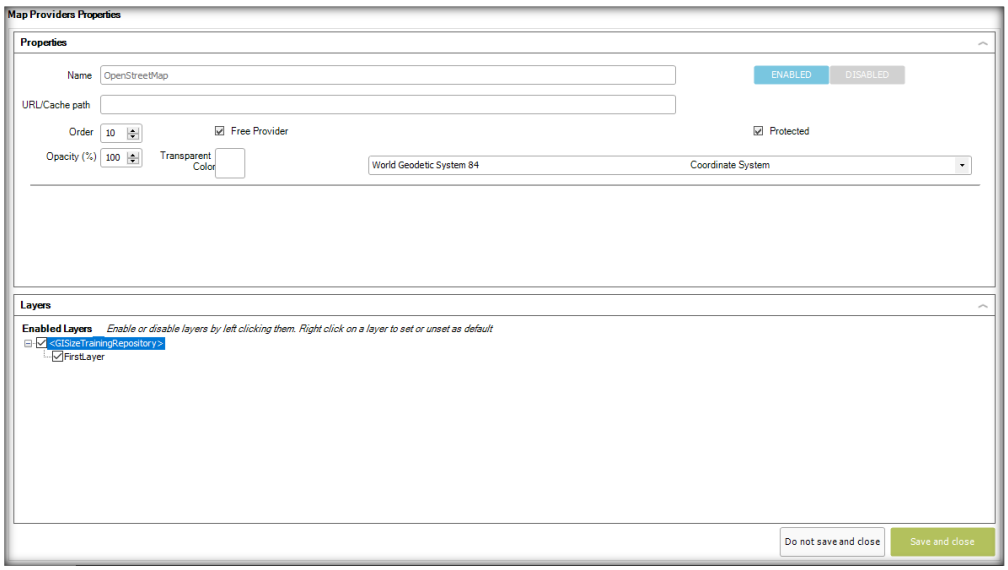
8.

9. Double click in **Map Providers** option.

   The **Map Provider** configuration window appears within the ArchestrA IDE.



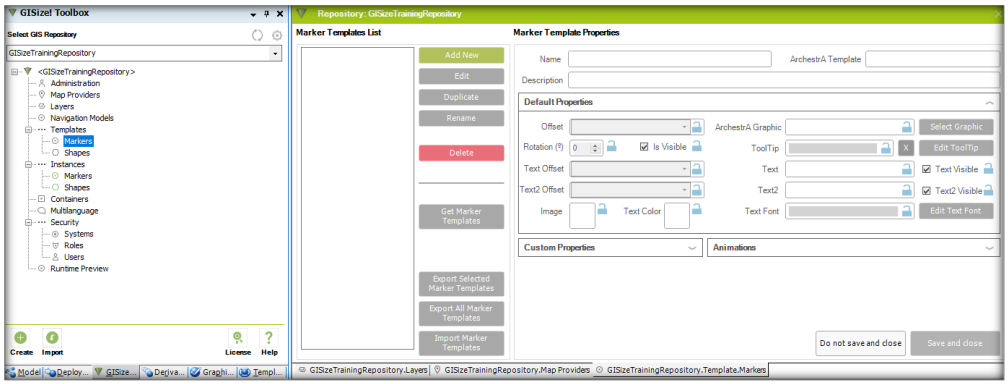
10. Select **OpenStreetMap**, click in **Edit** button and expand the **Layers** pane.
11. Click **FirstLayer** to enable it for this Map Provider and click **Save and close** to commit the changes.



1.8.7. Create a basic Marker Template and Instance.

In the following steps, you will:

1. Expand **Templates** and double click the **Markers.**

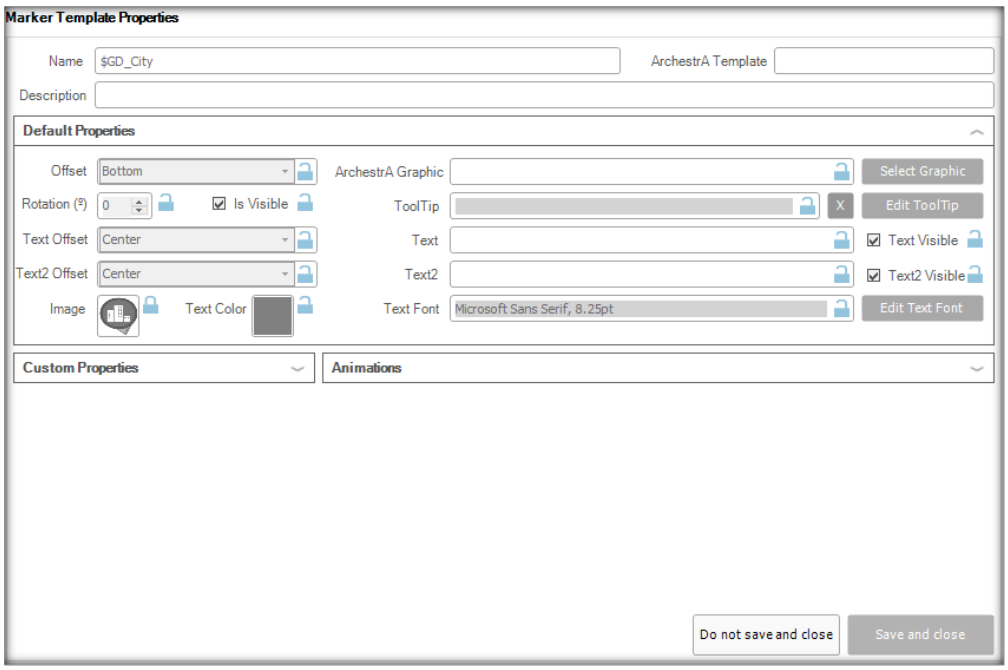


The **Marker Templates** configuration window will appear within the ArchestrA IDE.

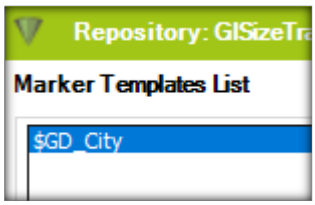
5. Click the **Add New** button to create a new marker template.

   The **Marker Template Properties** pane is now enabled to configure the new marker.

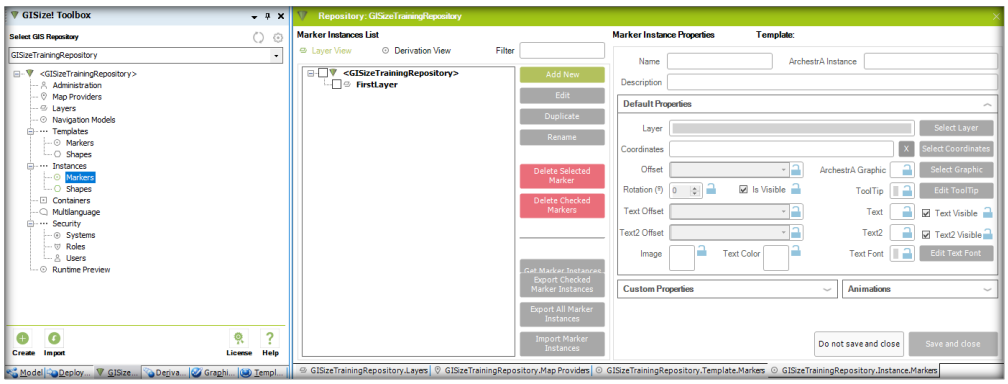
6. In the **Name** field, enter **GD_City**
7. Double click in the **Image** button and select the **Miscellaneous\Markers\City.png** file located in your training files.
8. For the **Image** attribute, **click** the lock icon.



The newly created template marker you entered in the previous steps now appears in the Marker Templates Tree.



9. Expand Instances and double click the **Markers**.



The **Marker Instances** configuration window appears within the ArchestrA IDE.

10. Click the **Add New** button to create a new marker instance.

    The **Marker Template Selection** dialog box appears.

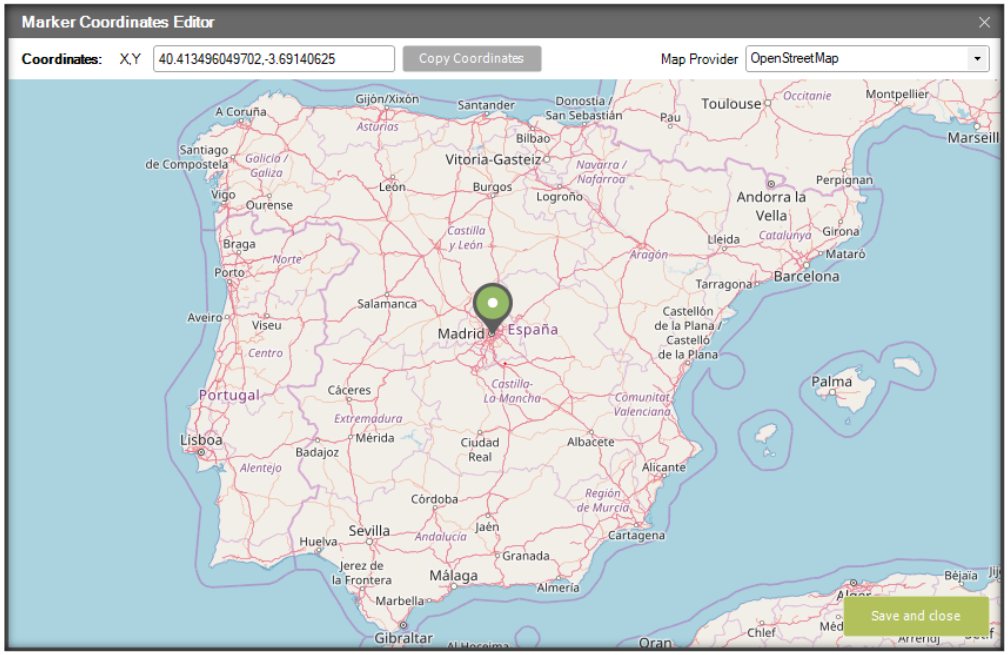
11. Select **GD_City** and click **Save and close** button.

    The **Marker Instance Properties** pane is now enabled to configure the new instance marker.

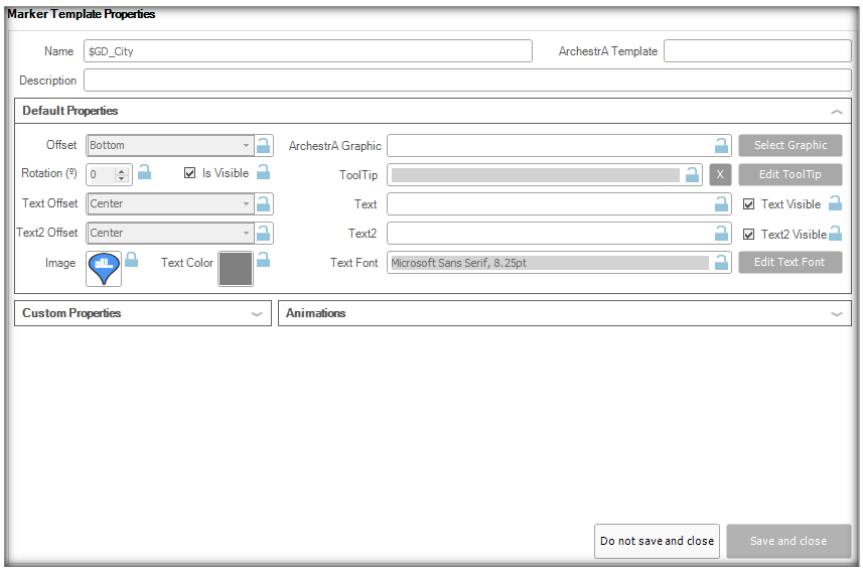
12. In the **Name** field, enter **GD_City_001**
13. Click **Select Layer** button and select **FirstLayer** layer.

14. Click **Select Coordinates** button.

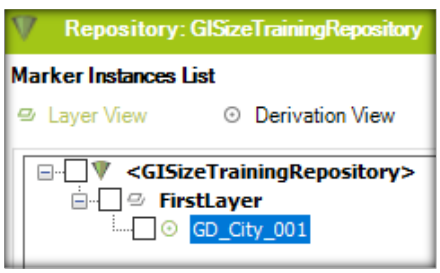
    The **Marker Coordinates Editor** appears.



15. Navigate on the map, select the coordinates for **Madrid City** and click **Save and close** button.



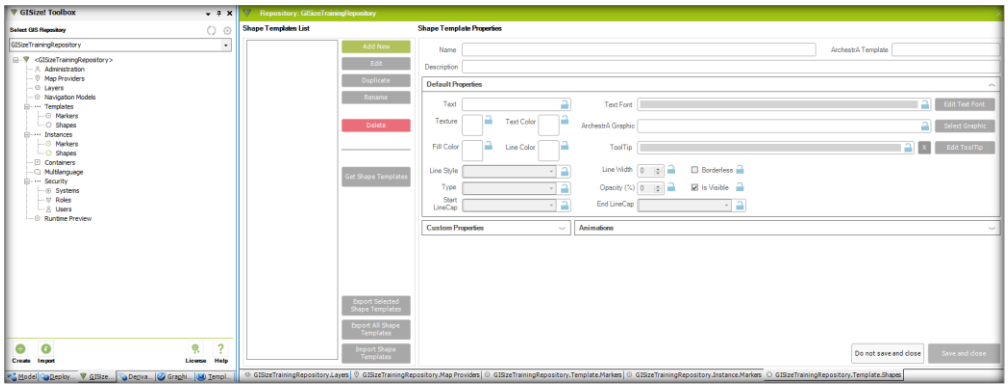
    The newly created template instance you entered in the previous steps now appears in the Marker Instances Tree.



1.8.8. Create a basic Shape Template and Instance.

In the following steps, you will:

1. Expand **Templates** and double click the **Shapes.**

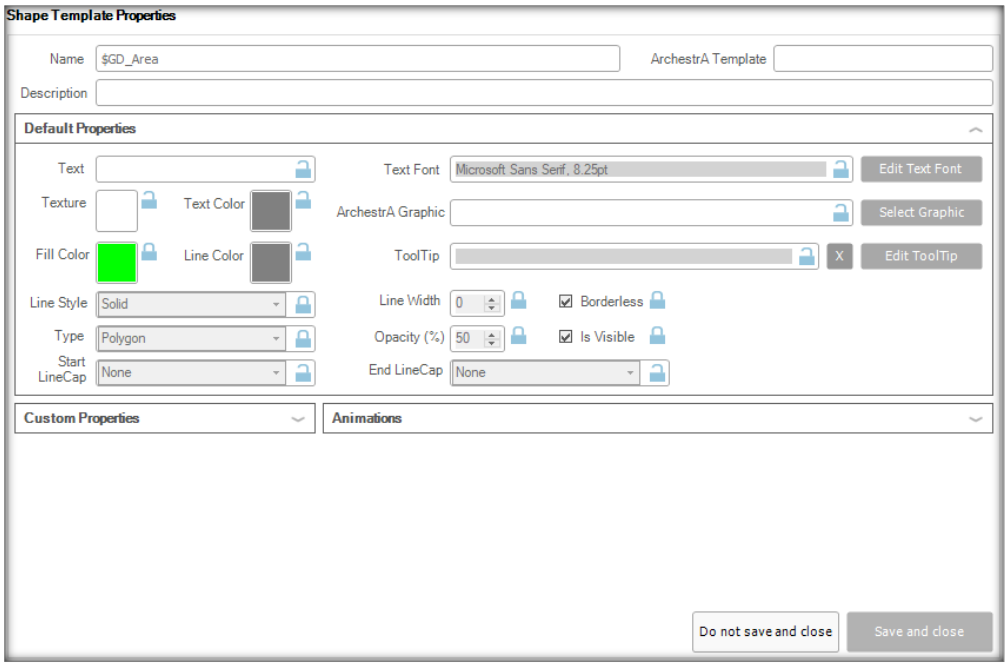


The **Shapes Templates** configuration window appears within the ArchestrA IDE.

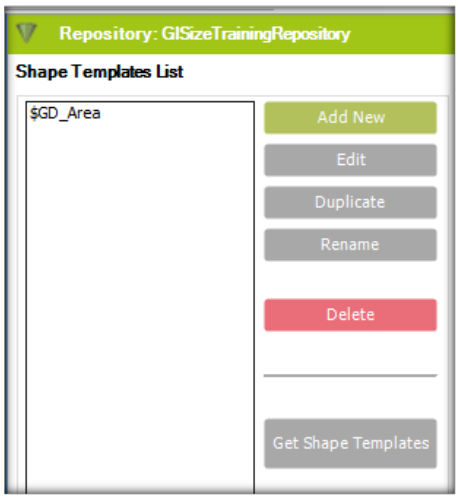
2. Click the **Add New** button to create a new shape template.

   The **Shape Template Properties** pane is now enabled to configure the new shape.

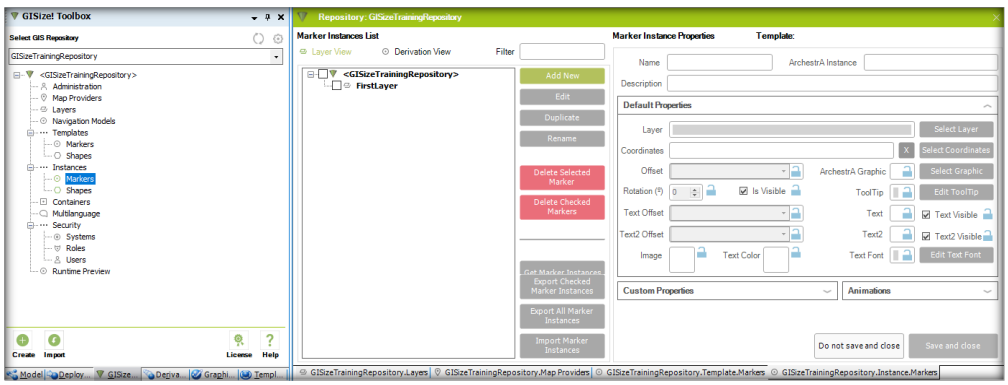
3. In the **Name** field, enter **GD_Area.**
4. Double click the **Fill Color** button and select the green color.
5. In the **Opacity** field, enter **50** %.
6. For the **Fill Color**, **Line Style**, **Type**, **Line Width**, **Opacity**, **Borderless** and **Is Visible** attributes, click the **Lock** icon.



The newly created template shape you entered in the previous steps now appears in the **Shape Templates Tree**.



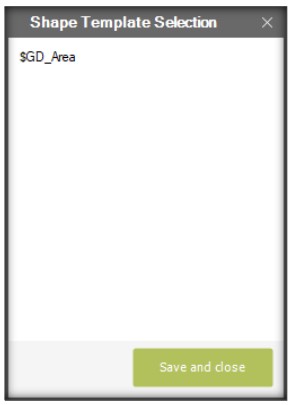
7. Expand **Instances** and double click the **Shapes**.



The **Shape Instances** configuration window appears within the ArchestrA IDE.

8. Click the **Add New** button to create a new shape instance.

   The **Shape Template Selection** dialog box appears.

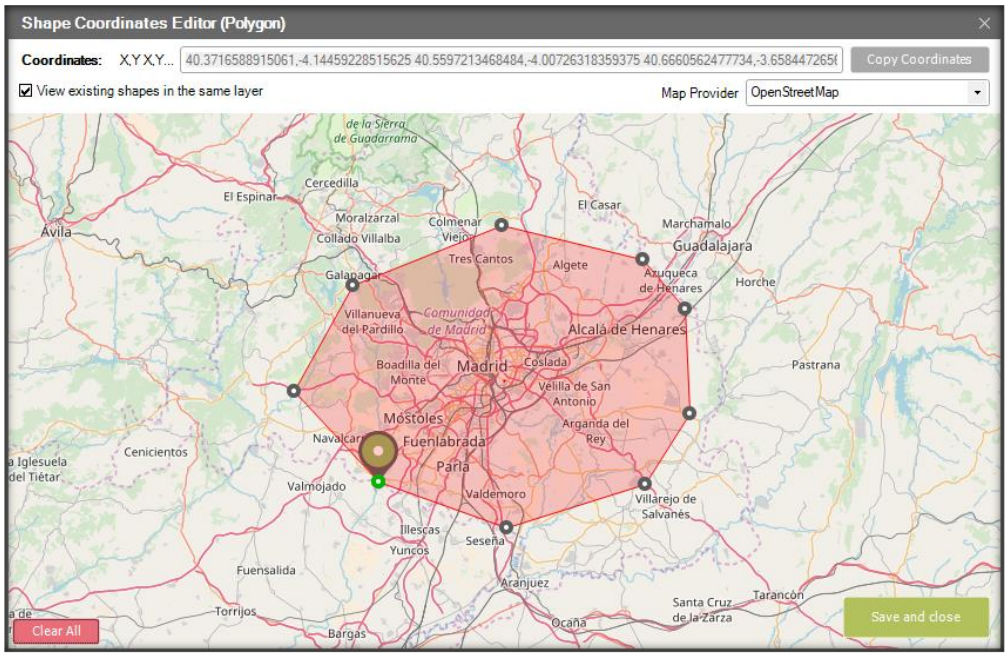


9. Select **GD_Area** and click the **Save and close button.**

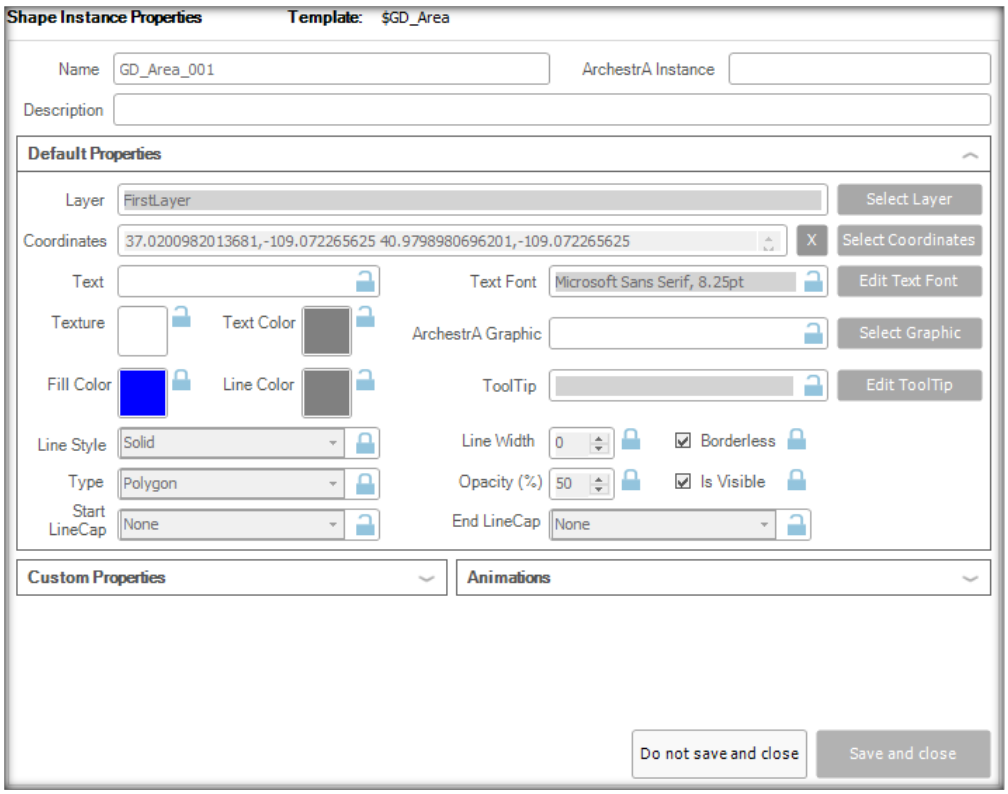
   The **Shape Instance Properties** pane is now enabled to configure the new shape instance.

10. In the **Name** field, enter **GD_Area_001**
11. Click Select Layer button and select **FirstLayer** layer.
12. Click Select Coordinates button.

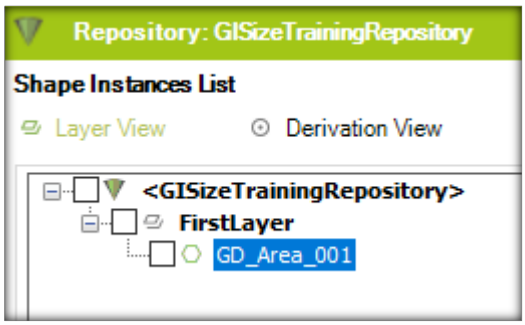
    The **Shape Coordinates Editor** appears.



13. Navigate on the map, select the coordinates for **Madrid City** (left double click to create a point and click right button to move the map) and click **Save and close** button.



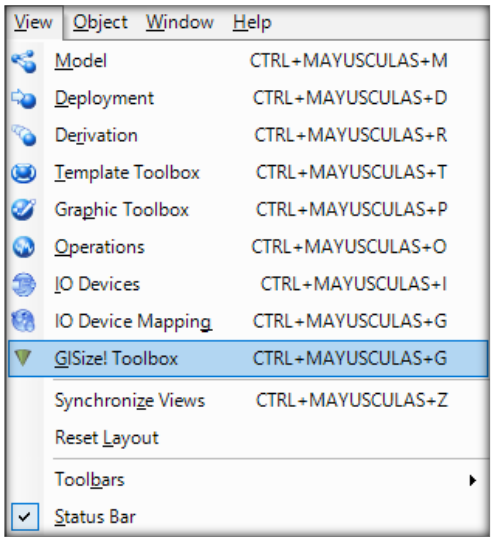
The newly created instance you entered in the previous steps now appears in the **Shape Instances Tree**.



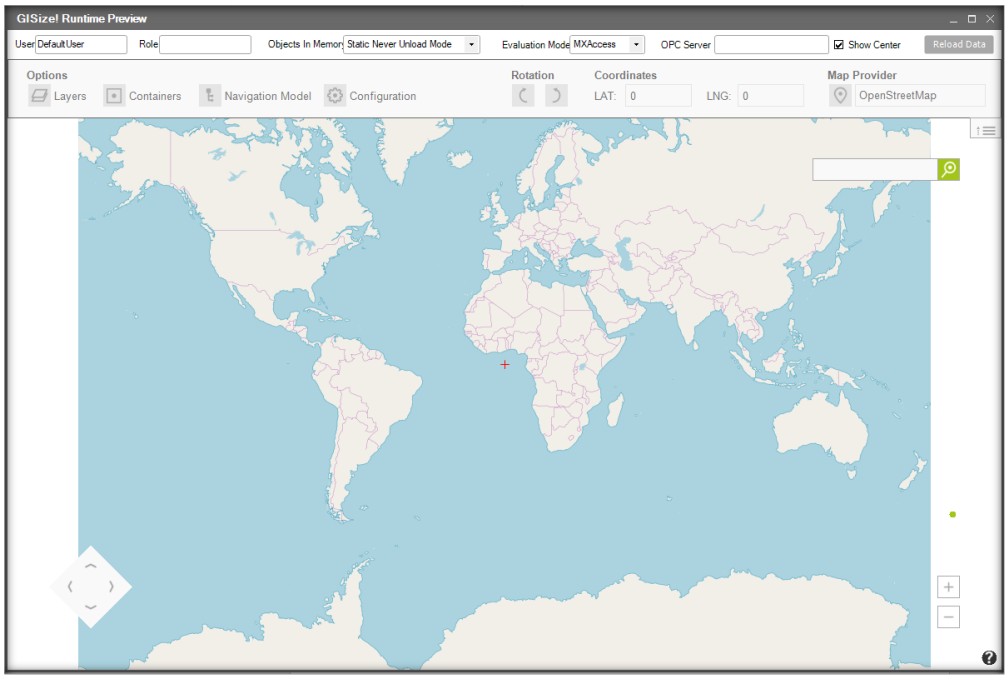
1.8.9. Preview Runtime.

In the following steps, you will:

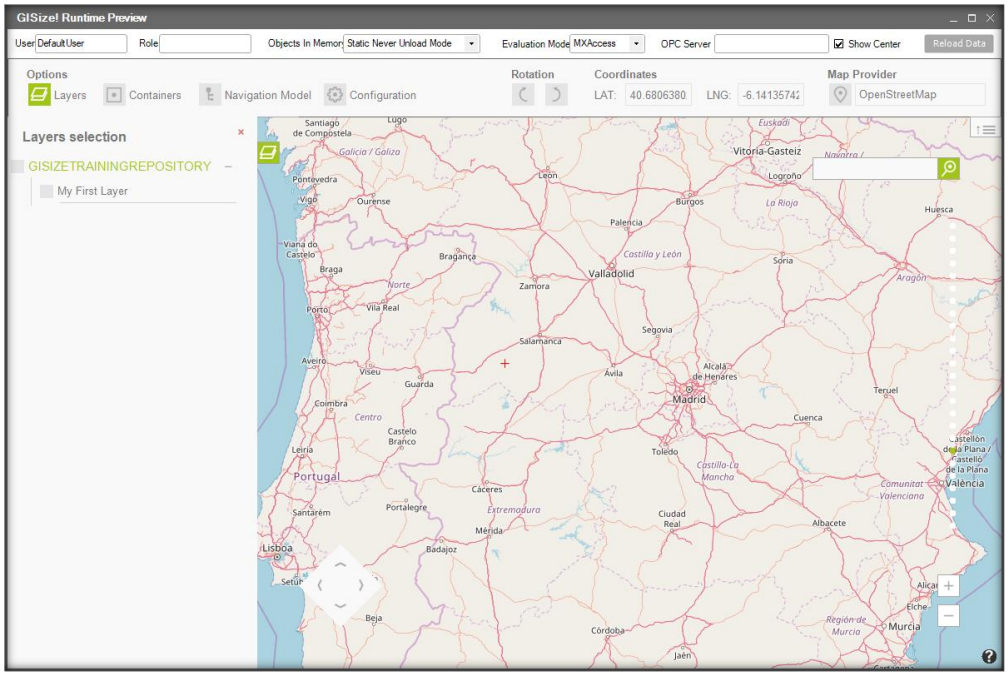
1. Click the **View** menu and select **GISIZE Toolbox**.



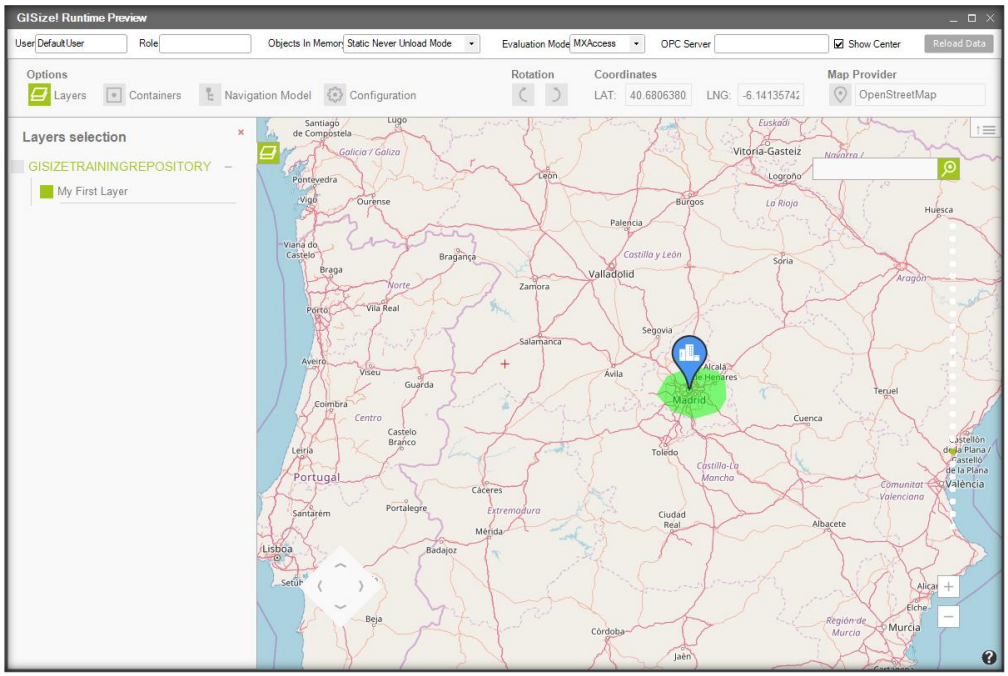
2. Double Click the **Runtime Preview** button to open the **GISIZE Runtime Preview** tool.



3. Click the **Layers** button to open the **Layers** selection pane.



4. Click the **My First Layer** checkbox to enable the visualization of the layers and its objects.

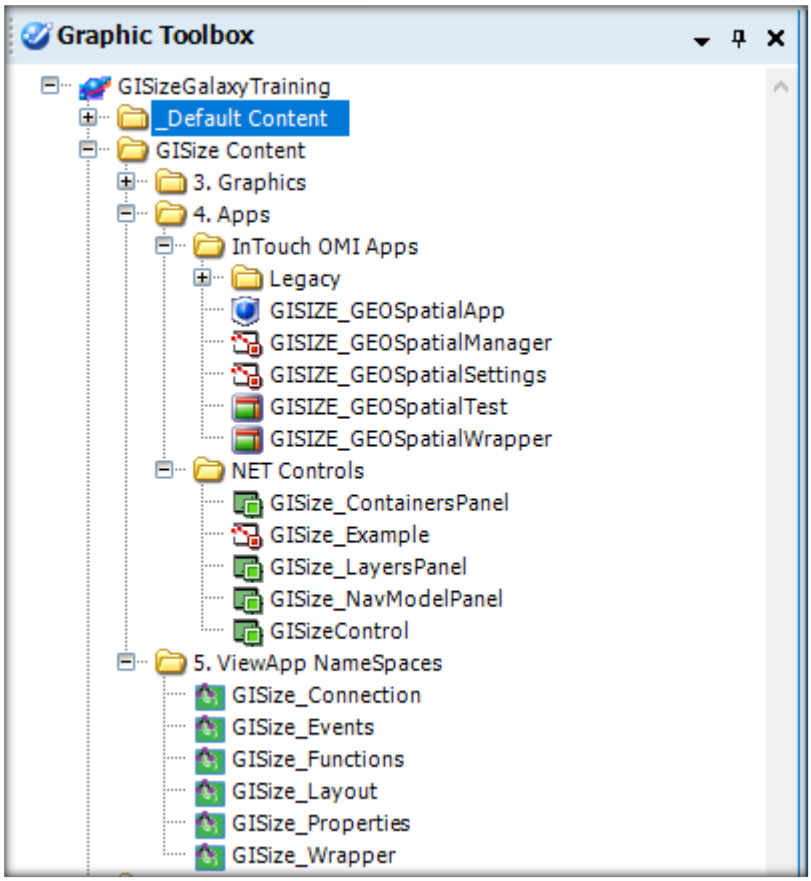


1.8.10. Import and test the GISIZE GEOSpatial objects for InTouch OMI

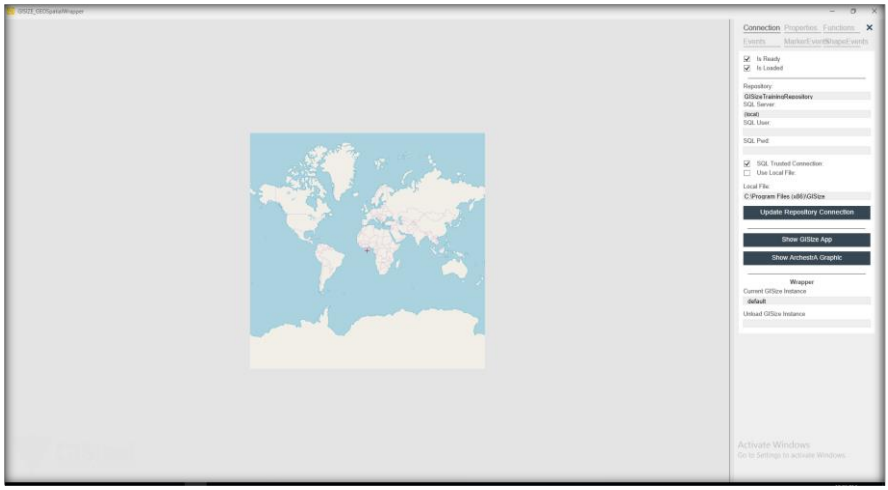
Please note that the following steps can only be done with Wonderware System Platform 2017 Update 3 or later.

In the following steps, you will import and test the GISIZE GEOSpatial objects for InTouch OMI.

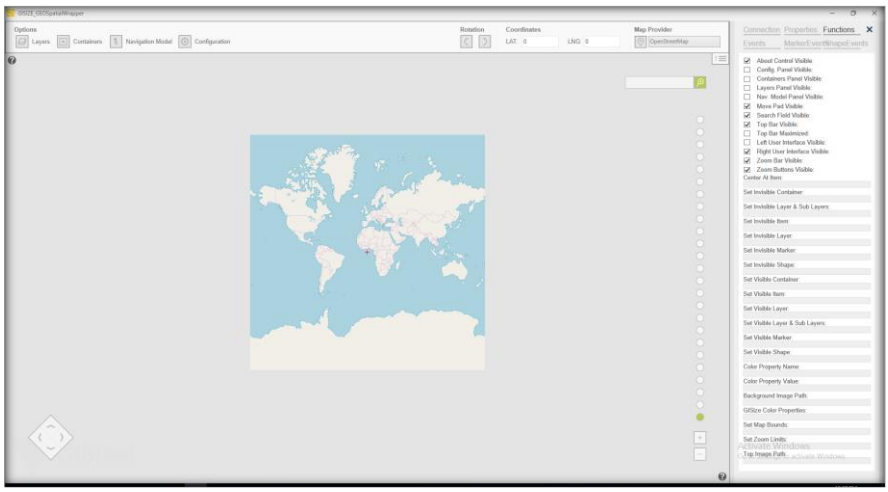
1. Import the ArchestrA package file **Apps\InTouch OMI Apps\ GISIZE InTouch OMI Apps.aaPKG** that can be found in the **Program Files/GISIZE** path of the computer, and must be imported from ArchestrA IDE with the option Galaxy -> Import -> Object(s).



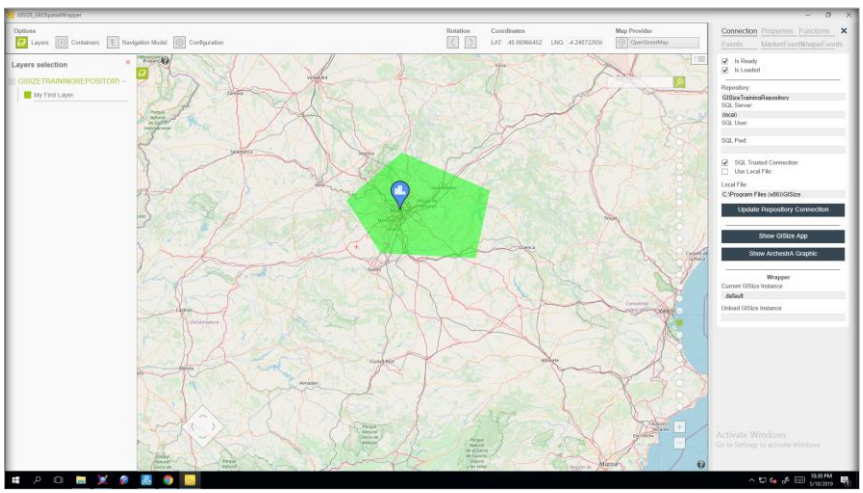
2. Create a new ViewApp application and open it.
3. Select the default screen profile and click **Next** button.
4. Select the **GISIZE_GEOSpatialTest** Layout and click **Finish** button.
5. Click the **Preview** button to test the GISIZE GEOSpatial App in InTouch OMI.
6. Introduce **GISizeTrainingRepository** in the **Repository** field and click **Show GISize App** button.



7. Go **Functions** tab and enable the **Right User Interface Visible** checkbox to show the User Interface.



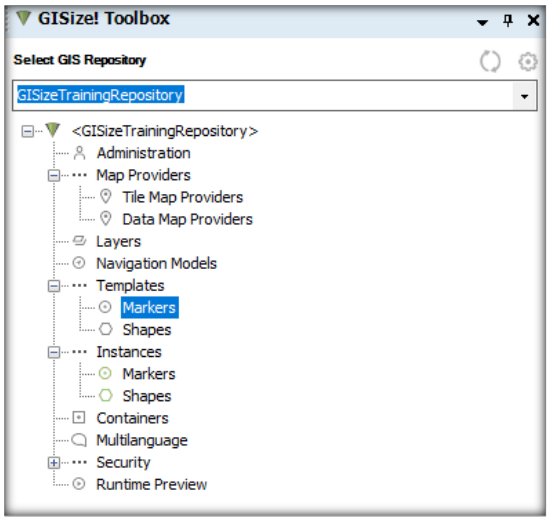
8. Click the **Layers** button to open the **Layers** selection pane.
9. Click the **My First Layer** checkbox to enable the visualization of the layers and its objects.



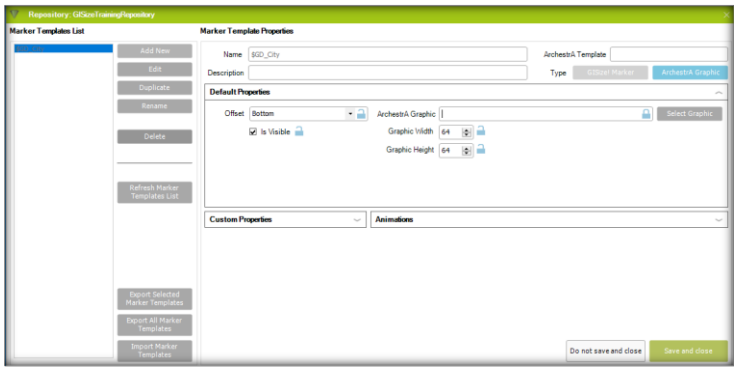
1.8.11. Use ArchestrA Graphics as GISize markers.

In the following steps, you will use ArchestrA Graphics as GISize markers.

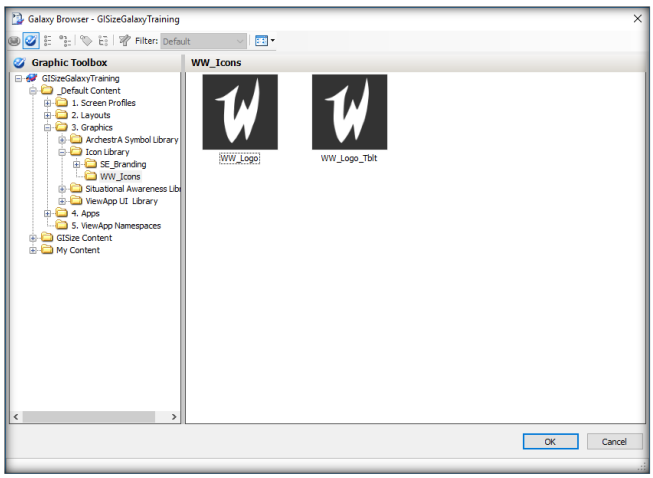
1. Double click the **Templates Markers** option.



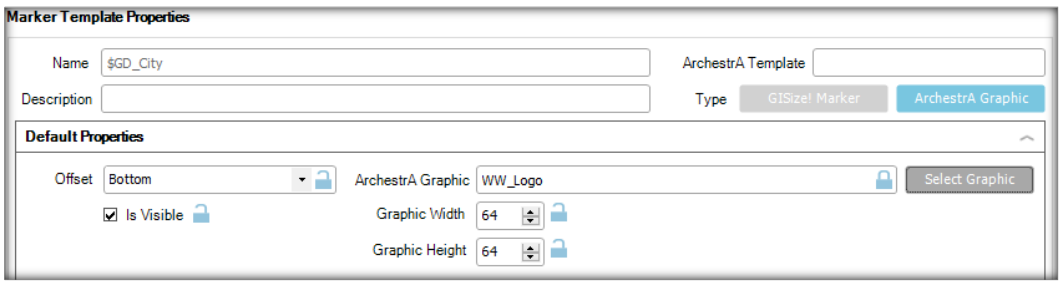
2. Edit the **$GD_City** marker template and click the **ArchestrA Graphic** button.



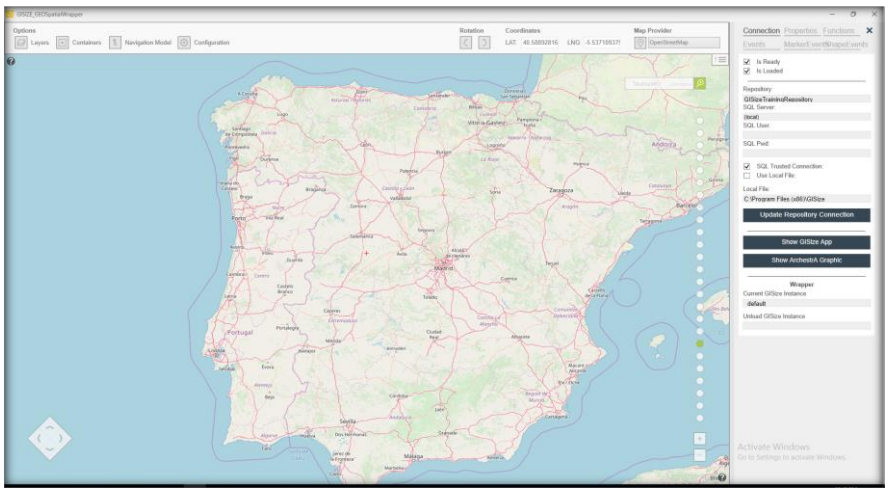
3. Click **Select Graphic** button and select the ww_logo graphic.



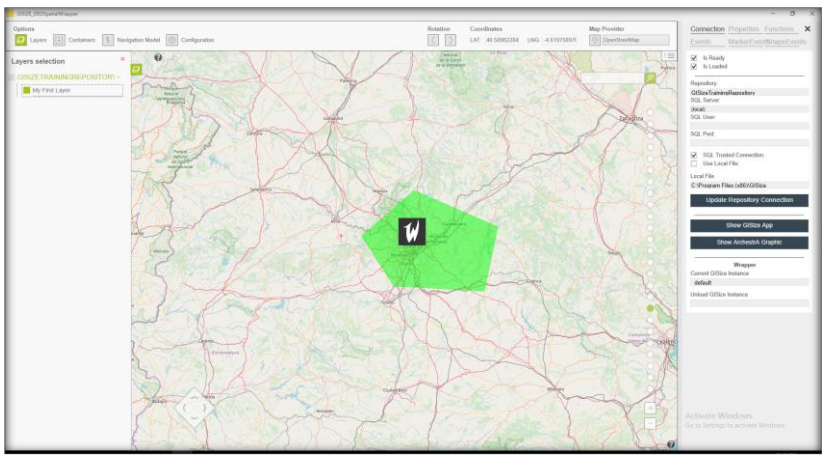
4. Click the **lock** icon for the **ArchestrA Graphic** field and click the **Save And Close** button to propagate the changes to the instances.



5. Return to the view app running, go the **Connection** Tab, click the **Update Repository Connection** button.



10. Click the **Layers** button to open the **Layers** selection pane.
11. Click the **My First Layer** checkbox to enable the visualization of the layers and its objects.



12. Click the **Wonderware logo** ArchestrA Graphic to verity the **Read&Writte** capabilities of the integration.
13. Edit the **$GD_City** marker template again and click the **GISize Marker** button to continue with the following labs.

2. Map Providers

The objectives of this module are:

- Explain the different types of maps and how to configure them.
- Create online map providers and explain how to connect to a WMS Server that uses the OGC Standards.
- Create vectorial map providers based on a SVG file.
- Create offline map providers.
- Merge offline and vectorial map providers.

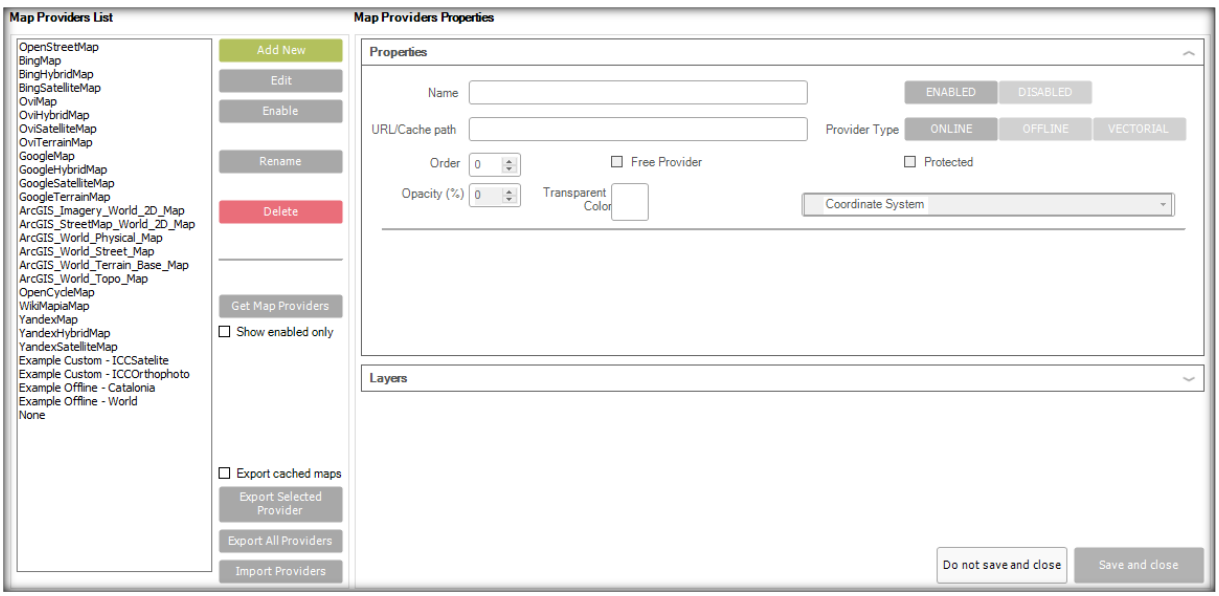
2.1. Introduction

The Map Providers panel allows the definition of new map providers and the configuration of the existing ones.

Map providers can be classified in 3 types: Online, Offline, and Vectorial.

- **Online map providers** are map providers that require access to internet to provide mapping. Examples of online map providers are Google Maps or OpenStreet Maps.
- **Offline map providers** are GISIZE GEOSpatial map providers that behave like cached online map providers. The purpose of the offline map providers is to provide map navigation at environments without internet access.
- **Vectorial map providers** are GISIZE GEOSpatial map providers based on SVG vector image files. They provide the capability of navigate through a vector image as if it were a map. Vectorial map providers can be optionally georeferenced allowing the use of latitude and longitude coordinates.

By default, GISIZE GEOSpatial comes with a set of online map providers and an empty especial map provider. All these map providers are marked as 'protected' and cannot be deleted from the repository because there are is no way to reproduce them if they are deleted, but they can be disabled.



The panel is divided into left and right sides. On the left side of the panel there is a list of available map providers and a set of buttons to perform operations with them. On the right side of the panel there is a collection of controls to modify the properties of a selected map provider

2.1.1. Map Providers List

The map providers list shows the current available map providers sorted by priority. Upon the selection of a map provider of the list, a set of operations can be performed through buttons. From top to bottom these buttons are:

- **Add New**

  This button allows creating a new map provider. Clicking on it disables the left side of the panel and enables the right side where the properties edition controls are located.

  The minimum requirement to add a new map provider is a unique name.

- **Edit**

  This button allows editing the properties of a previously existing map provider. Clicking on it disables the left side of the panel and enables the right side where the properties are located.

  The name of map provider being edited cannot be changed with this functionality. Also, most properties may not be available for editing if the map provider being edited is protected.

- **Disable / Enable**

  The text of this button changes between 'Enable' and 'Disabled' depending on the selected map provider. Clicking a disabled map provider will enable it, and an enabled map provider will disable it.

  A disabled map provider will not be available at runtime nor in the most of the functionalities at design.

- **Rename**

  This button allows changing the name of a selected map provider. When you click it a popup window will appear to type the new desired name. Duplicated names are not allowed. Also protected map providers cannot be renamed.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the final name.

- **Delete**

  This button allows deleting selected map providers. Before deleting a map provider, the user will be asked for confirmation. This operation cannot be undone.

  Protected map providers cannot be deleted.

- **Get Map Providers**

  When you click this button the list of available map providers will be refreshed. Usually there is no need to use this button because after performing any operation locally, the list is refreshed automatically.

  There is a checkbox under the button to filter the list to show only the enabled map providers.

- **Export Selected Provider**

  This button allows exporting a single selected map provider with all its properties to a XML file. This is a way to back up a map provider to be imported later. The file can be edited externally with any text editor.

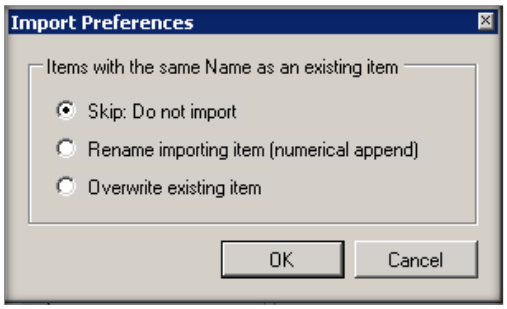
- **Export All Providers**

  This button allows exporting all the existing map providers with all its properties into a single XML file.

  This is a way to back up all the map providers of a repository to be imported later. The file can be edited externally with any text editor.

- **Import Providers**

  This button allows importing an XML file with previously exported map providers. When clicked, the user will be asked to select an XML file within an explorer window. After selecting an XML file, the user will have to choose the way to import the file in case that a map provider with the same name is found.



  The import preferences are:

  - **Skip – Do not import**: If the existing map provider has the same name as the imported one, the imported one will be ignored and skipped from the import operation. The existing map provider will remain unaffected.
  - **Rename importing item**: If the existing map provider has the same name as the imported one, the one being imported will be renamed adding an incremental number at the end of its name. The existing map provider will remain unaffected.
  - **Overwrite existing item**: If the existing map provider has the same name as the imported one, the existing one will be overwritten with the one being imported. Configuration of the previously existing map provider will be lost.

2.1.2. Map Providers Properties

The right side of the Map Providers panel contains the property controls of the map providers. Upon the selection of a map provider, the property controls will be updated with its values. These controls will remain disabled unless a map provider is being edited or created.

To accept any modification made during the editing or the creation of a map provider, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a map provider, user must click the **Do not save and close** button. Clicking this button meanwhile a new map provider is being created will abort the process. In both cases the ongoing process will come to an end.

The available map provider's properties can be classified depending on their type. The main properties are those that are common to any type of map provider. Online, offline, and vectorial properties are specific of their respective map provider's type. Finally, the Enabled Layers are separate properties.

2.1.2.1. Main properties

These properties are common to all the map providers, and its controls are located in the top-right part of the panel.

From top to bottom and left to right, the properties are:

- **Name**: The unique identifier name of the map provider. There cannot be 2 map providers with the same name. This control is only enabled during the process of creating a new map provider. Name cannot be left empty.
- **Enabled/Disabled**: A pair of buttons that shows the current state of the map providers and allows enabling or disabling it
- **URL/Cache path**:
  - For offline and vectorial map providers, it means the local path where the map cache will be stored. Offline and vectorial map providers must have this property informed to function correctly.
  - For online map providers, it means the URL to access to an OGC standard map provider. Typical map providers (OpenStreetMap, Yahoo, Bing, ArcGIS, Google, etc.) don't make use of this.
- **Provider Type**: these 3 buttons allow the definition of the type of the map provider (online, offline, vectorial).
  - **Online**: map providers that require access to internet to provide mapping.
  - **Offline**: map providers with offline cached data. Original data is an online provider. The data must be cached previously in design.
  - **Vectorial**: providers with offline vectorial map data. Origin data is an SVG file. The data must be cached previously in design.
- **Order**: indicates the order in which the object is displayed at runtime.
- **Protected**: this checkbox indicates if the map provider is protected or not. Protected map providers cannot be deleted and most of their properties are not editable.
- **Free Provider**: Editable checkbox that indicates if the map provider is free to use or not.
- **Opacity**: indicates the opacity of the tile images rendered on the map.
- **Transparent Color:** indicates the color of the tile images that will be transparent.
- **Coordinate System:** allows users to select between the World Geodetic Coordinate System (EPSG:4326 WGS 84) and the Qatar Coordinate System (EPSG:2932 QND95 / Qatar National Grid).

2.1.2.2. Enabled layers

At the right side half bottom part of the panel there is a tree view with all the currently defined layers in the repository, each node being a layer.

The purpose of this control is to define which layers are available for the selected map provider, as typically a map provider has its own layers.

By left clicking a node, the layer that represents will be enabled or disabled. An enabled layer will appear with its checkbox checked and will be selectable at runtime when running with the selected map provider.

By right clicking a node, the layer will be defined as default layer. This means that if the layer is enabled, it will be loaded automatically once the map provider is loaded. Enabled layers that are not defined as default will not be loaded automatically.

Note that by default newly created map providers does not have any layer enabled nor set as default. Also, any newly created layer will not be enabled for any map provider by default.

2.2. Online map provider's properties

This section explains how to connect to a WMS Server that uses the OGC Standards.

2.2.1. URL Syntax

The OpenGIS® Web Map Service Interface Standard (WMS) provides a simple HTTP interface for requesting geo-registered map images from one or more distributed GEOSpatial databases. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc.) that can be displayed in a browser application. You can use URL parameters to communicate with a WMS service in GISIZE GEOSpatial. *For more information check the link http://www.openGEOSpatial.org/standards/wms*

The online resource of each operation supported by a compliant WMS server is an HTTP uniform resource locator (URL), so a WMS service can be considered to be a Representational State Transfer (REST) service. Unlike a standard web service, a SOAP client is not necessary for consuming a WMS service, and a web browser is the simplest client. You can get a WMS service's map image by sending a URL request to the server and viewing the corresponding responses in the browser as either an XML document or an image.

When editing the URL of a custom online Map Provider the next wild cards are supported:

- **{0}**: Origin latitude
- **{1}**: Origin longitude
- **{2}**: Ending latitude
- **{3}**: Ending longitude
- **{4}**: Tile Width in pixels
- **{5}**: Tile Height in pixels
- **{6}**: Layers list
- **{7}**: Zoom level
- **{8}**: X position
- **{9}**: Y position

2.2.2. Custom Process

This attribute is intended to indicate a special functionality of the map provider.

Currently supported commands are:

- **ReverseCoords**: Indicates to GISIZE GEOSpatial that the coordinates supplied by the online map provider are switched.
- **Zoom0**: Indicates to GISIZE GEOSpatial that the map provider starts at zoom 0. By default, most map providers start at zoom 1.
- **CenterOffset**: Indicates to GISIZE GEOSpatial that the map provider builds its tiles with the offset at the center. By default, most map providers build its tiles with the offset at the top-left corner.

2.2.3. Layer Parameter – WMS Name

This parameter allows users to associate a GISIZE GEOSpatial Layer to a WMS Layer in an Online Map Provider (parameter {6} in the URL definition).

2.3. Lab 03 - Creating an online map provider

2.3.1. Introduction

In this lab, you will create an online map connected to an ArcGIS sample.

2.3.2. Objectives

Upon completion of this lab, you will be able to:

- Get Capabilities for a WMS Map Provider.
- Test WMS URL using a web browser.
- Create an online map provider.
- Connect to an ArcGIS sample provider.
- Associate a GISIZE GEOSpatial Layer to a WMS Layer.
- Create and configure custom online map providers.

2.3.3. Create and test an URL request to an ArcGIS sample

In the following steps, you will get the capabilities of a WMS Server and test it using a web browser.

1. Open a web browser and go to the link **http://sampleserver1.arcgisonline.com/ArcGIS/rest/services**
2. Navigate to **Specialty** and then to **ESRI_StatesCitiesRivers_USA**
3. Navigate to **Supported Interfaces: WMS**

   The XML file with the WMS capabilities should open in the web browser

http://sampleserver1.arcgisonline.com/ArcGIS/services/Specialty/ESRI_StatesCitiesRivers_USA/MapServer/WMSServer?request=GetCapabilities&service=WMS

In this file you will find the implemented parameters in the WMS service and the available values to use (For more information check the Implementation Specifications that can be found at the link http://portal.openGEOSpatial.org/files/?artifact_id=14416)

Regarding GISIZE GEOSpatial, the most important parameters are:

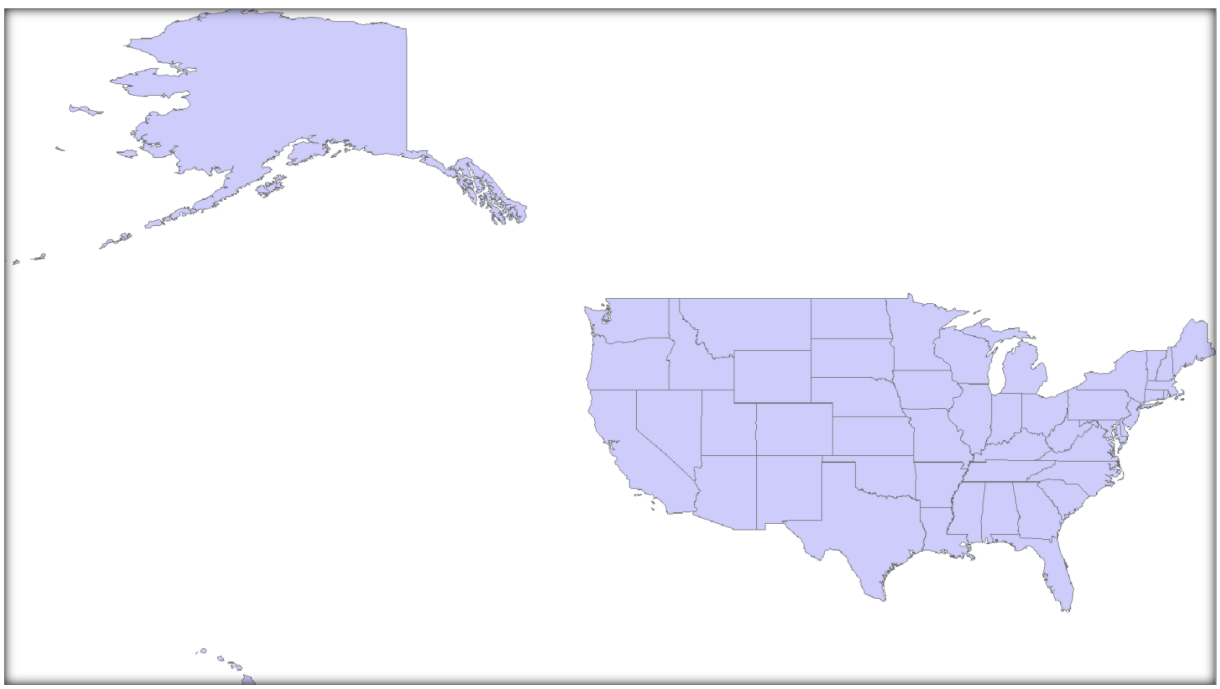
- Format=image/png
- CRS=CRS:84
- BBOX=-178.217598,18.924782,-66.969271,71.406235
- Layers: 0 for States, 1 for Rivers

The URL to test the WMS Service should be created with the following parts:

- http://sampleserver1.arcgisonline.com/ArcGIS/services/Specialty/ESRI_StatesCitiesRivers_USA/MapServer/WMSServer?version=1.3.0&request=GetMap&styles=
- &format=image/png&CRS=CRS:84
- &bbox=-178.217598,18.924782,-66.969271,71.406235
- &width=1920&height=1080
- &layers=0

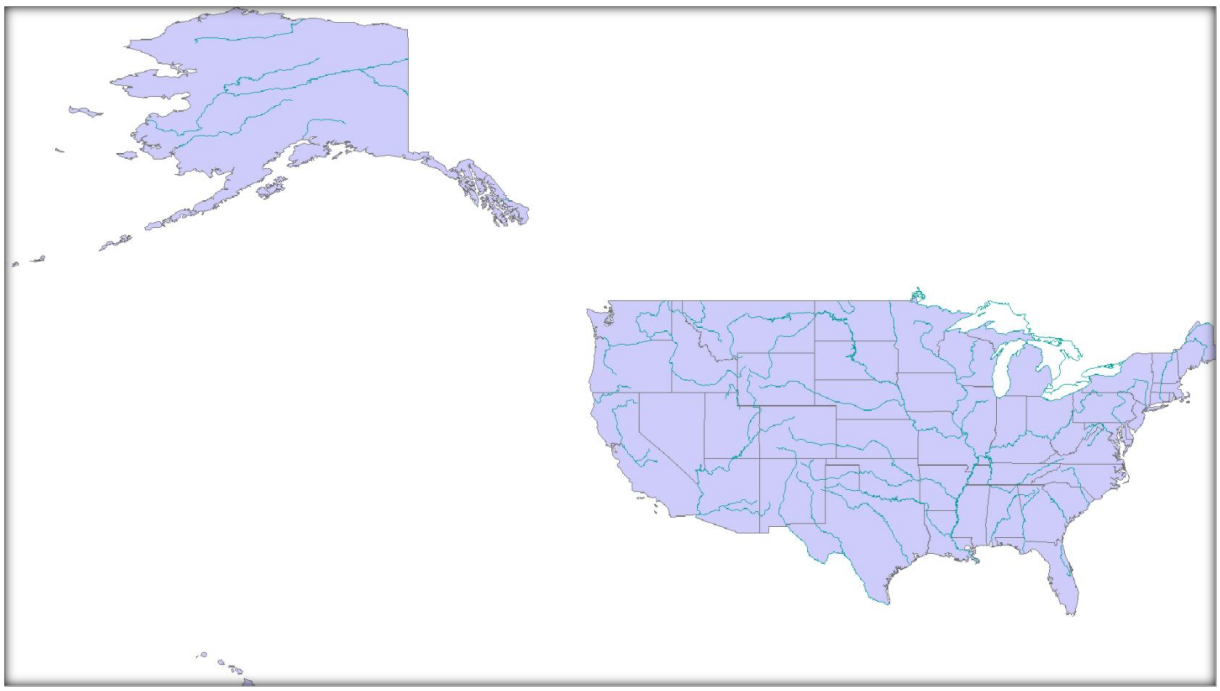
4. Navigate to the below link to test the WMS Service

   http://sampleserver1.arcgisonline.com/ArcGIS/services/Specialty/ESRI_StatesCitiesRivers_USA/MapServer/WMSServer?version=1.3.0&request=GetMap&styles=&format=image/png&CRS=CRS:84&bbox=-178.217598,18.924782,-66.969271,71.406235&width=1920&height=1080&layers=0



5. Modify the URL to see the **States and Rivers** layers (layers=0,1)

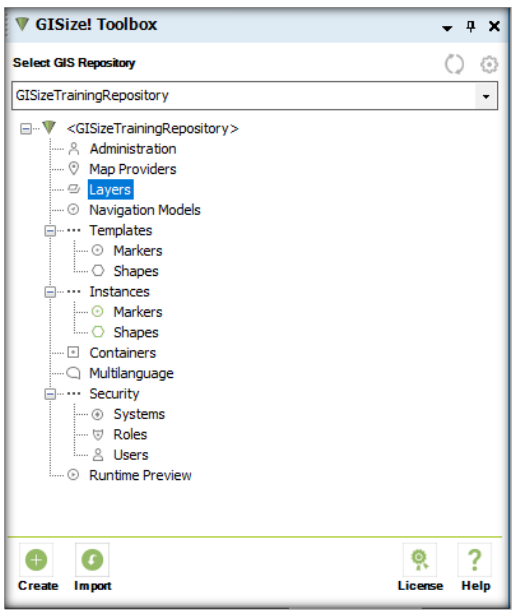
   http://sampleserver1.arcgisonline.com/ArcGIS/services/Specialty/ESRI_StatesCitiesRivers_USA/MapServer/WMSServer?version=1.3.0&request=GetMap&styles=&format=image/png&CRS=CRS:84&bbox=-178.217598,18.924782,-66.969271,71.406235&width=1920&height=1080&layers=0,1



2.3.4. Create an online map provider connected to an ArcGIS sample

In the following steps, you will create an online map provider with an URL connected to an ArcGIS sample provider.

1. Double click the **Map Providers** option.

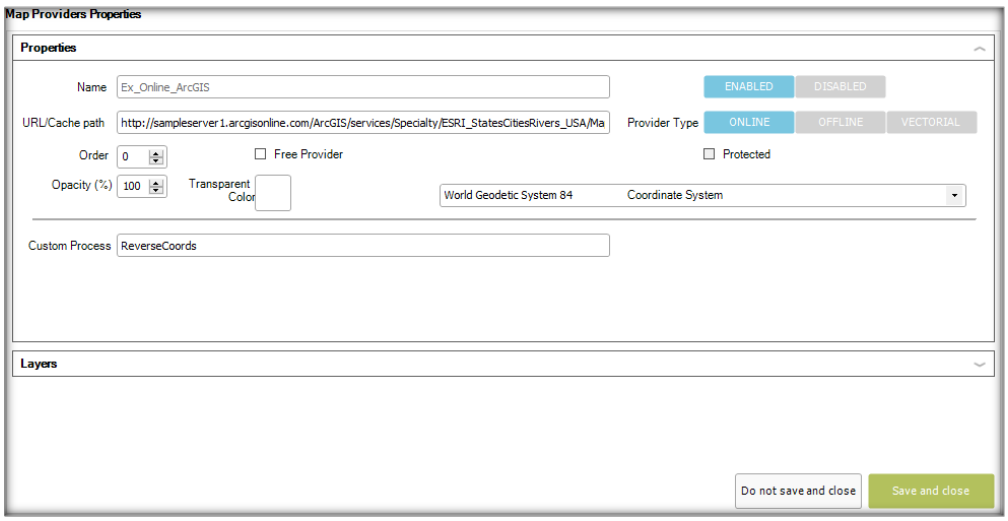


   The **Map Providers** configuration window appears within the ArchestrA IDE.

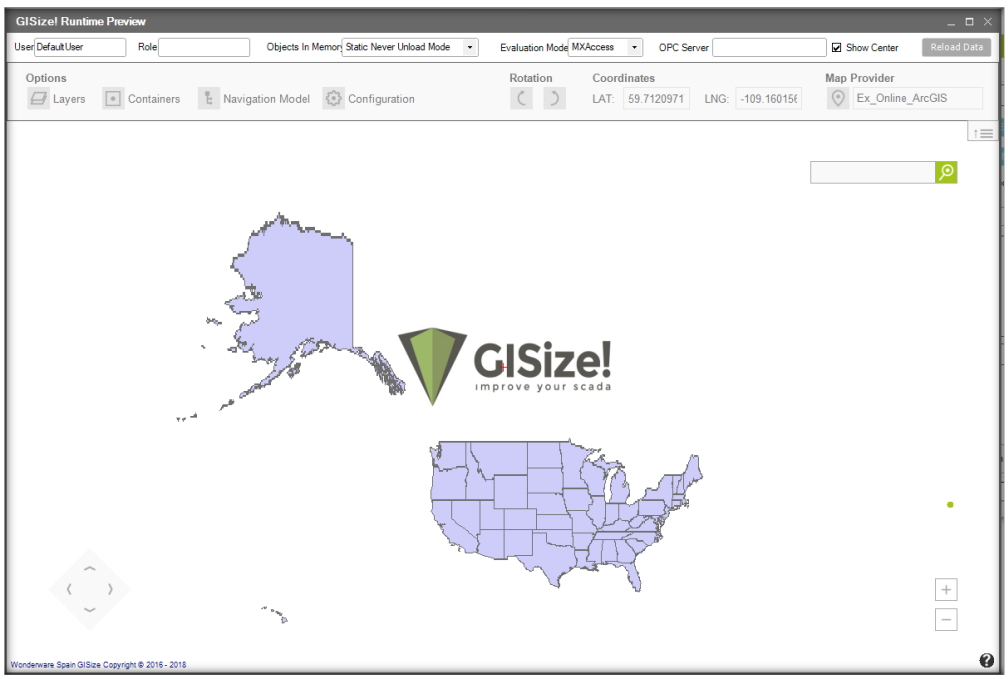
2. Click the **Add New** button to create a new Map Provider.
3. Enter **Ex_On_ArcGIS** in the field **Name.**
4. Click the **Enabled** button.
5. Click the **Online** button.
6. Enter **15** in the **Order** Field.
7. Enter the below URL in the field **URL/Cache path**.

   http://sampleserver1.arcgisonline.com/ArcGIS/services/Specialty/ESRI_StatesCitiesRivers_USA/MapServer/WMSServer?version=1.3.0&request=GetMap&styles=&format=image/png&CRS=CRS:84&bbox={0},{1},{2},{3}&width={4}&height={5}&layers=0,{6}

8. Enter **ReverseCoords** in the field **Custom Process**.

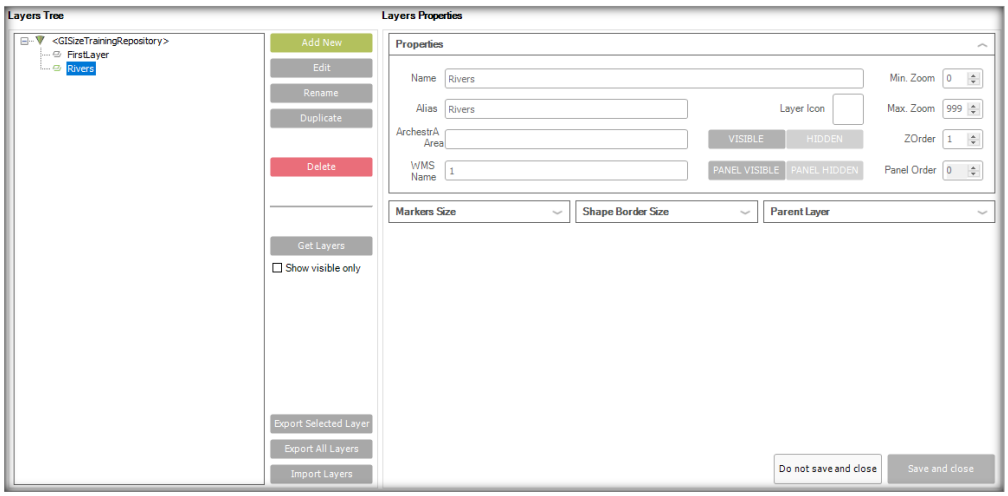


9. Open the **Runtime Preview** tool to see the new map provider.

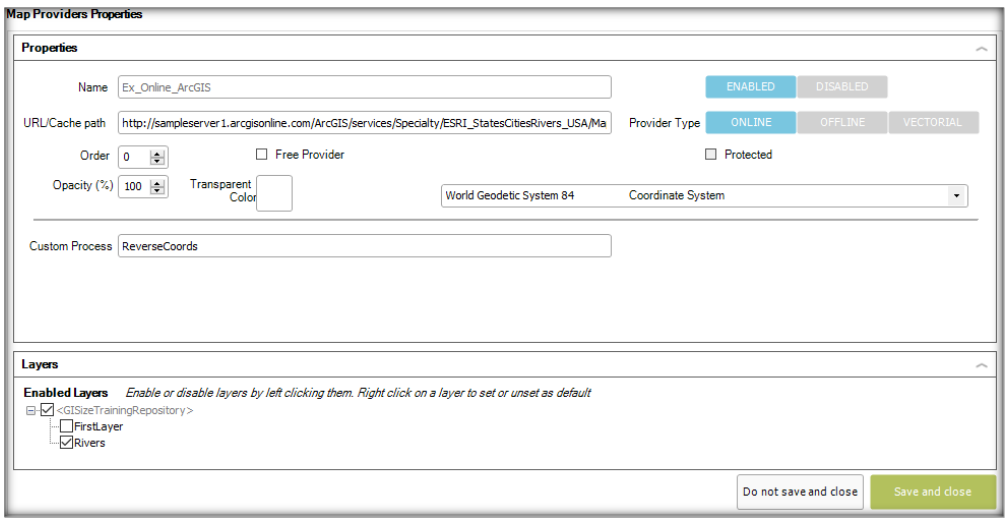


10. Double click the **Layers** option to open the **Layers Configuration Pane**.

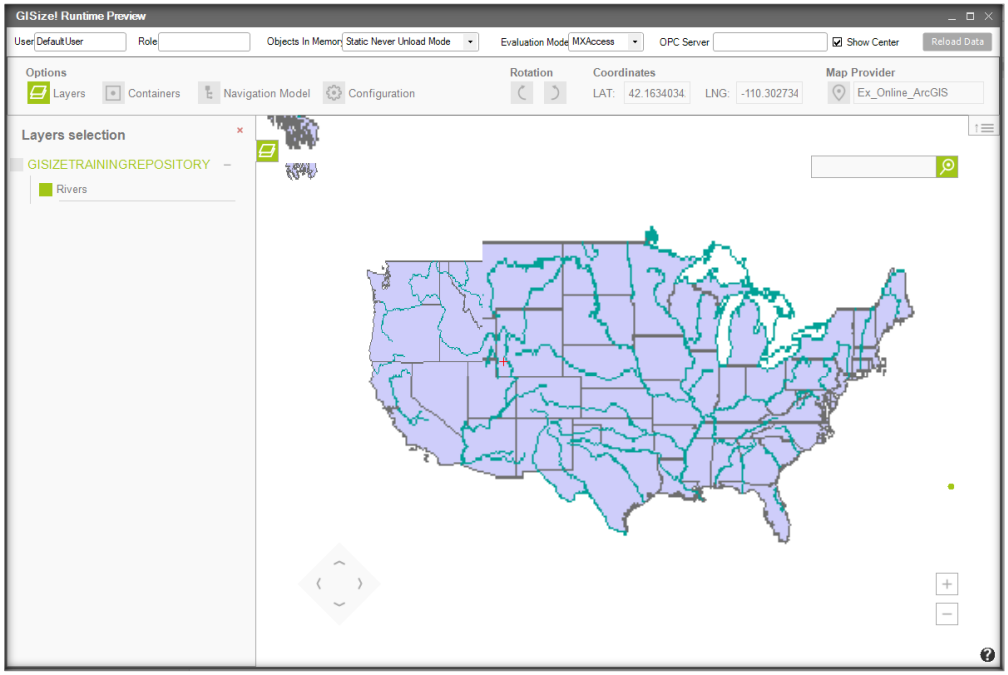
11. Click the **Add New** button to create a new Layer.
12. Enter **Rivers** in the field Name.
13. Enter **1** in the field **WMS Name**.



14. Edit the **Ex_On_ArcGIS** map provider and enable the **Rivers** layer.



15. Go to **Runtime Preview** tool, click the **Reload** button, click the **Layers** button and enable the **Rivers** layer.



2.4. Lab 04 - Creating an online map provider with MapBox

2.4.1. Introduction

In this lab, you will create an online map connected to a MapBox provider.

2.4.2. Objectives

Upon completion of this lab, you will be able to:

- Create and configure custom online map providers.

2.4.3. Create and test an URL request to a Mapbox sample

According the Mapbox API documentation (https://www.mapbox.com/api-documentation/#static) the URL to a Mapbox map should have the below format:

- /styles/v1/{MapName}/{style_id}/static/{lon},{lat},{zoom},/{width}x{height}?&access_token=<your_access_token>

In the following steps, you will create an URL and test it using a web browser.

1. Open a web browser and go to the link **https://www.mapbox.com/**
2. Navigate to **Products – Maps**, select **MapBox_Light** and then go to **View Mapbox Light Live**.
3. Copy the URL to see the MapName and the Access Token to test it.

   https://api.mapbox.com/styles/v1/**mapbox/light-v9**.html?title=true&**access_token=pk.eyJ1IjoibWFwYm94IiwiYSI6ImNpejY4M29iazA2Z2gycXA4N2pmbDZmangifQ.-g_vE53SD2WrJ6tFX7QHmA#1.07/0/0**
4. Navigate to the below link to test the map provider:

   https://api.mapbox.com/styles/v1/**mapbox/light-v9**/static/0,0,0/800x600?access_token=**pk.eyJ1IjoibWFwYm94IiwiYSI6ImNpejY4M29iazA2Z2gycXA4N2pmbDZmangifQ.-g_vE53SD2WrJ6tFX7QHmA**

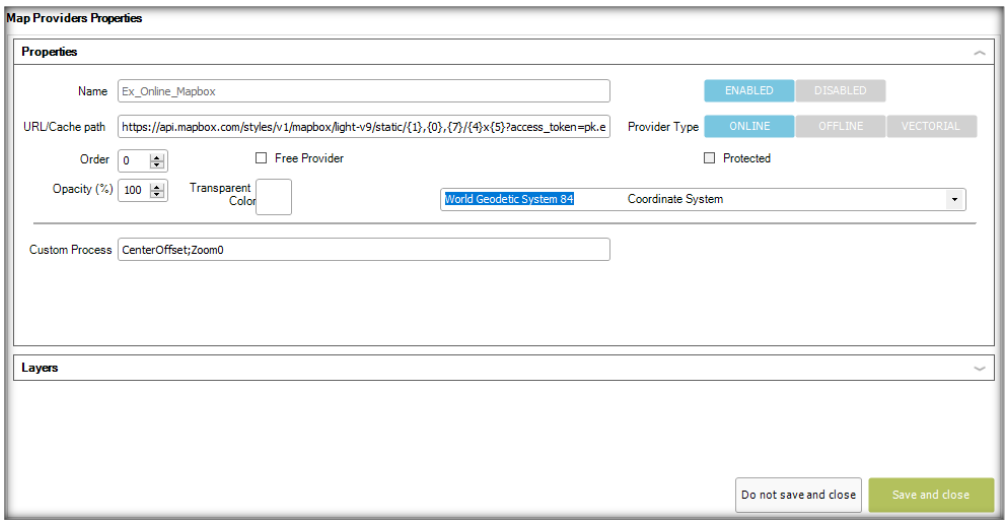
2.4.4. Create an online map provider connected to a Mapbox sample

In the following steps, you will create an online map provider with an URL connected to a Mapbox sample provider.

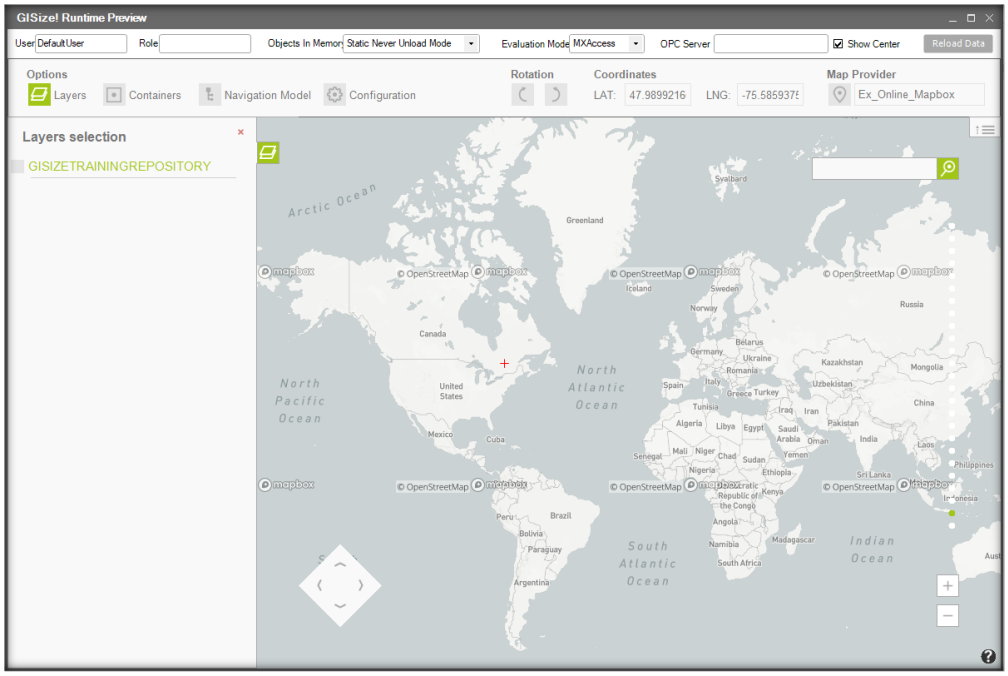
1. Open **Map Providers** configuration pane.
2. Click the **Add New** button to create a new Map Provider.
3. Enter **Ex_On_MapBox** in the field **Name.**
4. Click the **Enabled** button.
5. Click the **Online** button.
6. Enter **15** in the **Order** Field.
7. Enter the below URL in the field **URL/Cache path**.

   https://api.mapbox.com/styles/v1/mapbox/light-v9/static/{1},{0},{7}/{4}x{5}?access_token=pk.eyJ1IjoibWFwYm94IiwiYSI6ImNpejY4M29iazA2Z2gycXA4N2pmbDZmangifQ.-g_vE53SD2WrJ6tFX7QHmA

8. Enter **CenterOffset;Zoom0** in the field **Custom Process**.



9. Open the **Runtime Preview** tool to see the new map provider.



2.5. Vectorial map provider's properties

This section explains how to create a custom Map Provider based on a SVG file.

2.5.1. Configuration

These properties are exclusive and only to be shown by vectorial map providers.

From top to bottom and left to right the control properties are:

- **Zoom Levels**: Number of zoom levels that will be generated from the SVG file. Above this zoom level, the map will lose visual quality.
- **SVG File**: Clicking inside this panel the user will be able to select the Scalable Vector Graphic (SVG) file that will act as the map source for the map provider. If there is a SVG file already configured, the panel will show a miniature version of it. The content of this control will only be loaded while editing or creating a map provider.
- **Georeferenced**: This check box indicates whether the coordinates of the map provider will be georeferenced or not. The coordinates of a non-georeferenced map provider will not have relation with a real location on earth and the SVG graphic will occupy the entire world.
- **Area**: This control only applies if the map provider is defined as georeferenced. It represents the georeferenced area that the vectorial map will occupy. To select the area easily, the user can click the 'Select Area' button that will open a popup window. This window will allow drawing the rectangular area over an online map. Use the right mouse button to drag the map. Use the mouse wheel to zoom in/out. Use the left mouse button meanwhile the Alt key is pressed to generate the rectangular area to be cached. The SVG image will be drawn inside the rectangular area, so the user can overlap the vectorial image with the online map.

Once the properties are filled, it's time to generate the cached map. To do that, the green **Generate cache and accept** button must be clicked.

Upon clicking the button, a popup window will appear showing the current process of cache generation. The process can be stopped anytime by clicking the **Abort** button.

The time required to finish the cache varies greatly depending on the size of the area and the maximum zoom indicated. Note that number of map tile to generate increases exponentially by each additional zoom level.

When the process finishes, the editing or creation process of the map provider will be accepted.

2.6. Lab 05 - Creating a vectorial map provider

2.6.1. Introduction

In this lab, you will create several samples of vectorial map providers.

2.6.2. Objectives

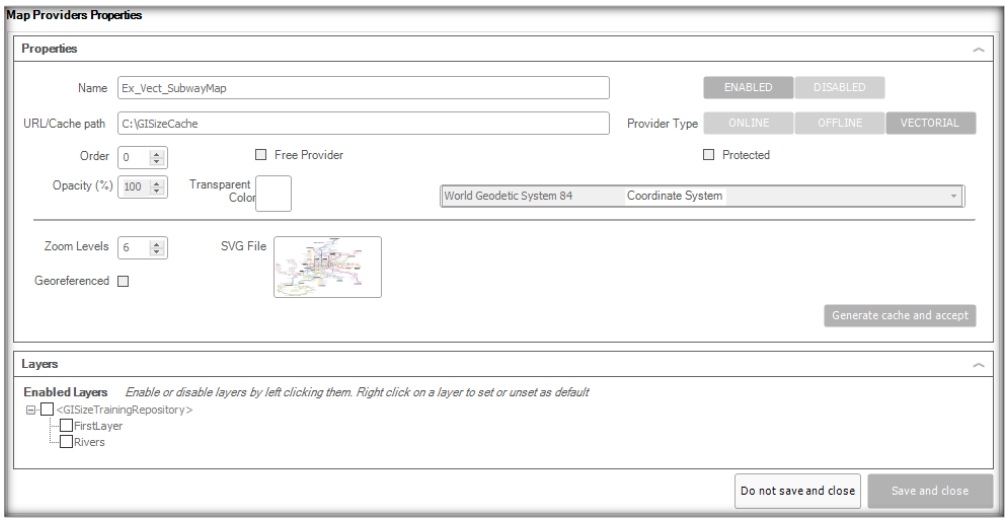
Upon completion of this lab, you will be able to:

- Create a vectorial map provider.
- Geo-reference a vectorial map provider.
- Set a Map Provider as a Layer.

2.6.3. Create a vectorial map provider

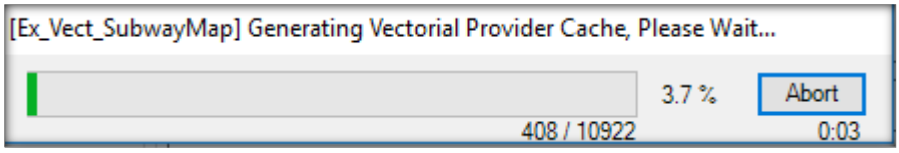
In the following steps, you will create a vectorial map provider.

1. Double click the **Map Providers** option.
2. Click the **Add New** button to create a new Map Provider.
3. Enter **Ex_Vect_SubwayMap** in the field **Name.**
4. Click the **Enabled** button.
5. Click the **Vectorial** button.
6. Enter **15** in the **Order** Field.
7. Enter **C:\GISIZECache** in the field **URL/Cache path**.
8. Enter **3** in the field **Zoom Levels**.
9. Double click the **SVG File attribute** and select the file **Miscellaneous\Maps SVG\Ex_ Ex_Vect_SubwayMap.svg** file located in your training files.

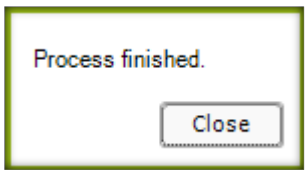


10. Click the **Generate cache and accept** button.

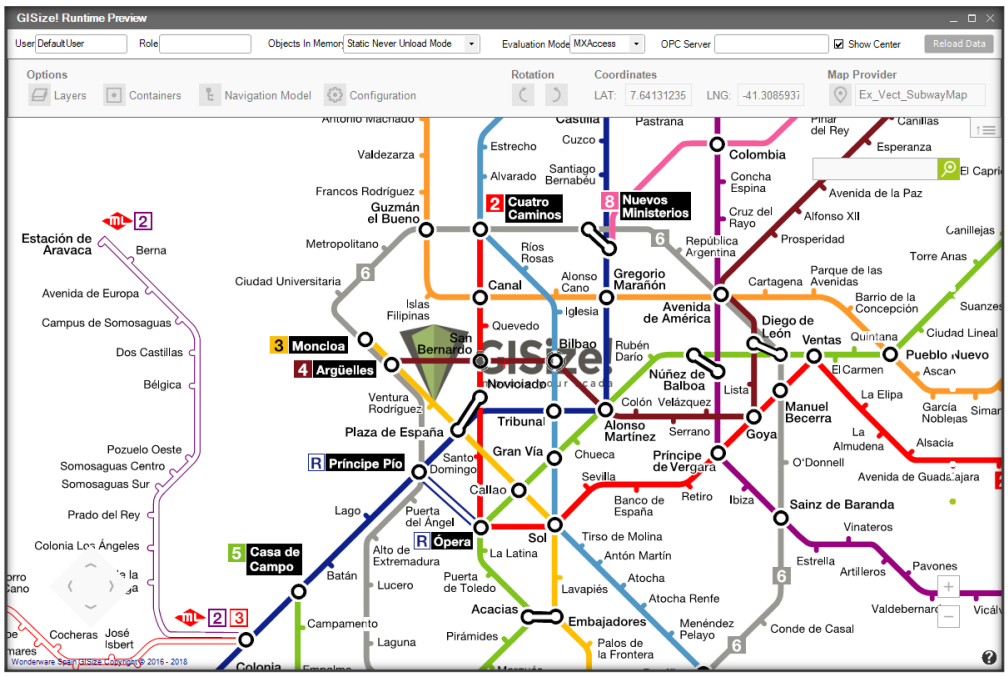
    The **Generating Vectorial Provider Cache** dialog box appears and shows the creation progress. This will take a few minutes.



11. Click the **Close** button when the process finished.



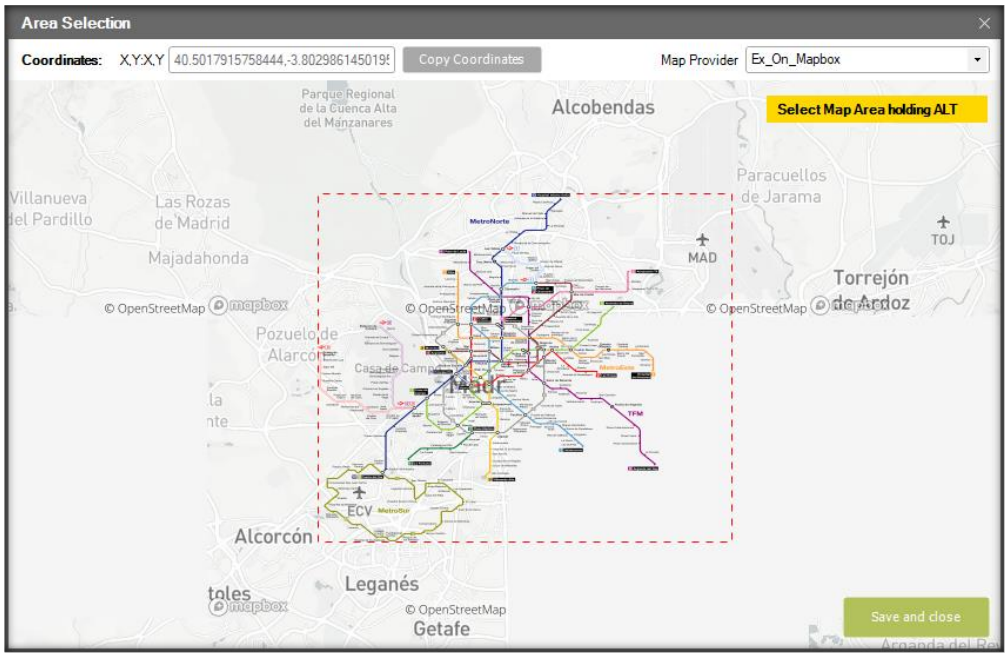
12. Open the **Runtime Preview** tool to see the new map provider.



2.6.4. Geo-reference a vectorial map provider

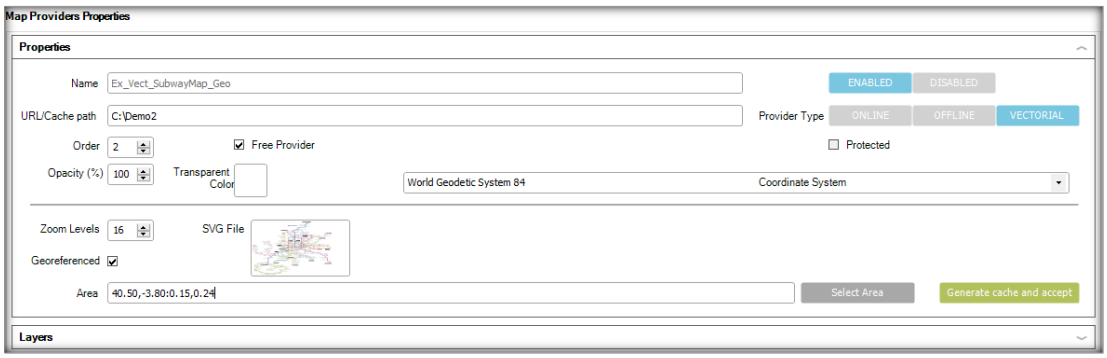
In the following steps, you will geo-reference a vectorial map provider.

1. Select the **Ex_Vect_SubwayMap** map provider.
2. Click the **Edit** button to edit the Map Provider.
3. **Enable** the Georeferenced checkbox.
4. Enter **40.50,-3.80:0.15,0.24** in the field area.
5. Click the **Select Area** button to see this area in the **Area Selection** dialog box.



You can modify or select a new area holding the key ALT.

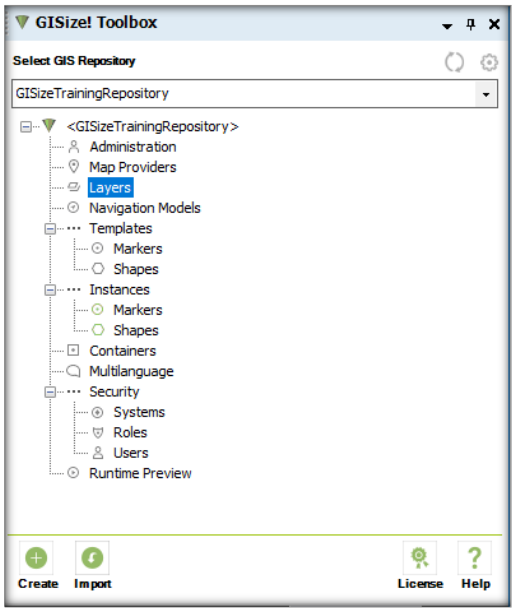
6. **Close** the previous windows.
7. Enter **12** in the field **Zoom Levels.**



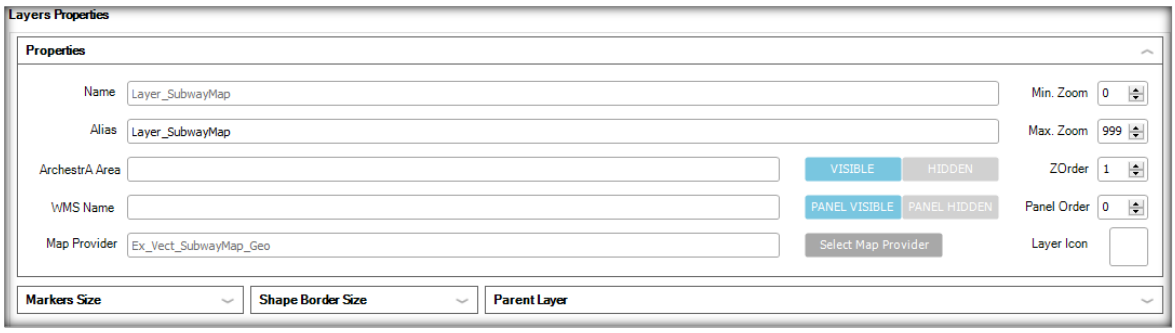
8. Click the **Generate** button.
9. Open the **Runtime Preview** tool to see the new map provider.
10. You can import the map providers created in the previous lab with more zoom levels cache (folder **Lab 05 - Creating a vectorial map provider)**.

2.6.5. Set a Map Provider as a Layer

1. Double click the **Layers** option.

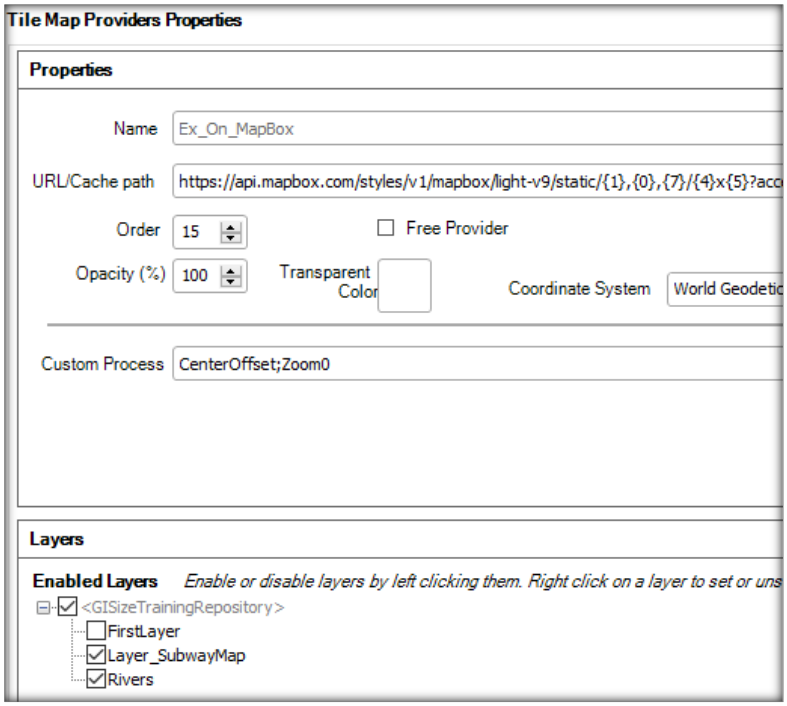


2. Click the **Add New** button to create a new Layer.
3. Enter **Layer_SubwayMap** in the field **Name.**
4. Click the **Select Map Provicer** button and select the **Ex_Vect_SubwayMap_Geo** Map Provider.

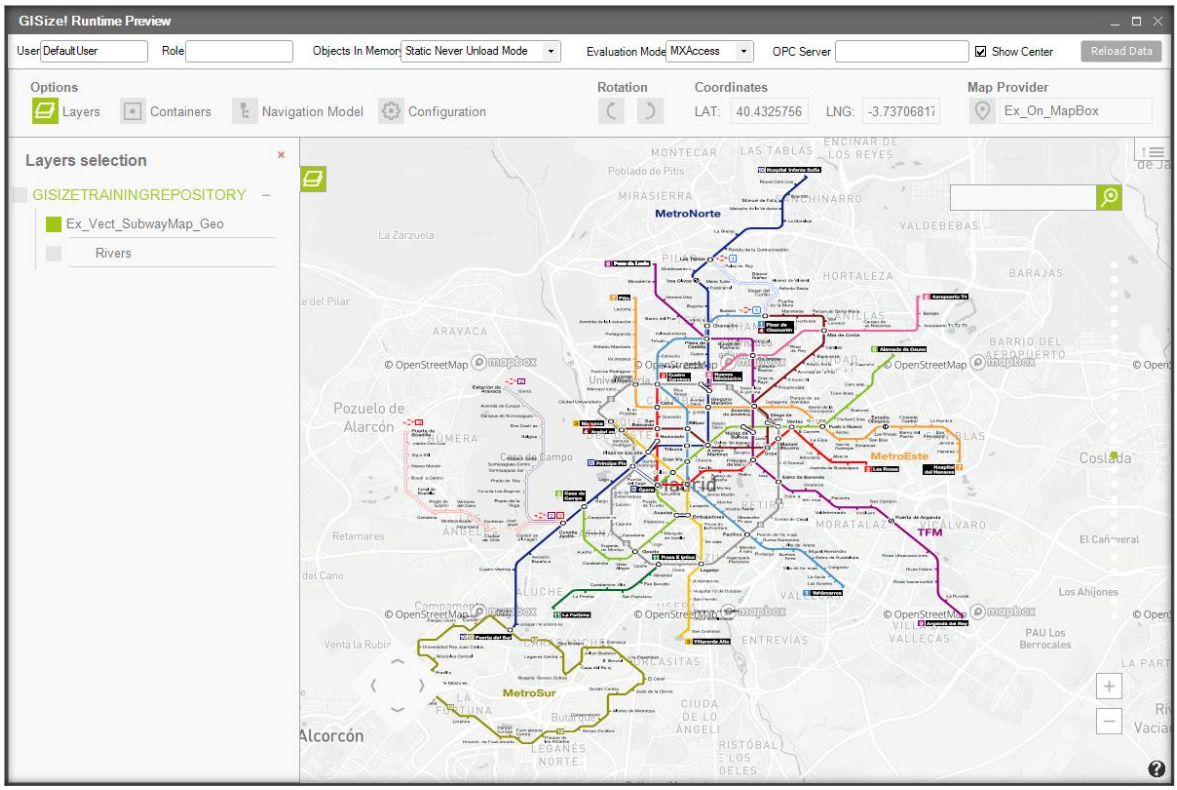


5. Double click the **Map Providers** option.
6. Edit the **Ex_On_MapBox** Map Provider.

7. Enable the **Layer_SubwayMap** layer.



8. Open the **Runtime Preview** tool, select the **Ex_On_MapBox** Map Provider and enable the **Layer_SubwayMap** to superimpose one map provider as a layer.



2.7. Lab 06 - Show examples of different vectorial map providers

2.7.1. Introduction

In this lab, you will import vectorial map providers to see different examples using vectorial files.

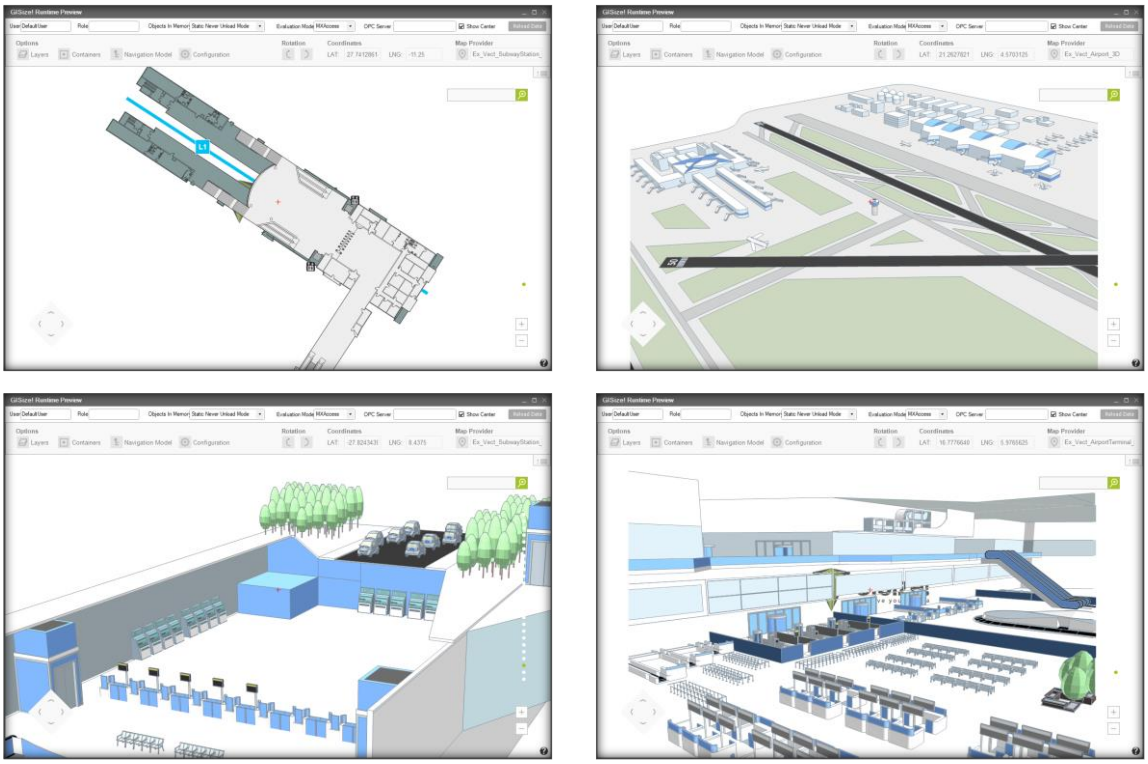
2.7.2. Objectives

Upon completion of this lab, you will be able to:

- Import a vectorial map provider.

2.7.3. Import vectorial map providers

1. Import the Vectorial Map Providers located in your training files folder **Lab 06 - Show examples of different vectorial map providers** to see different examples using vectorial files (also you will find the map providers created in the previous lab with more zoom levels cached).

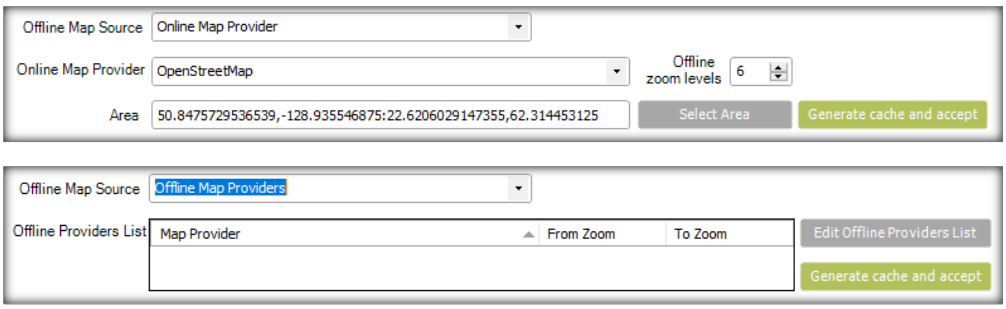


2.8. Offline map provider's properties

This section explains how to cached online map providers.

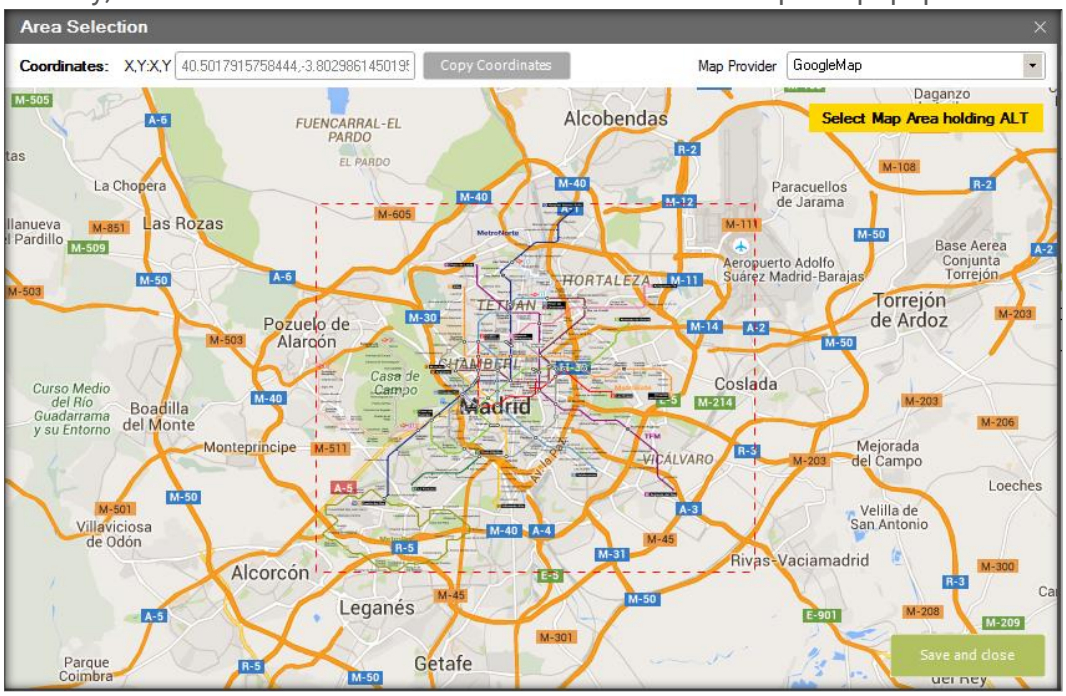
2.8.1. Configuration

These properties are exclusive and only to be shown by offline map providers.



From top to bottom the control properties are:

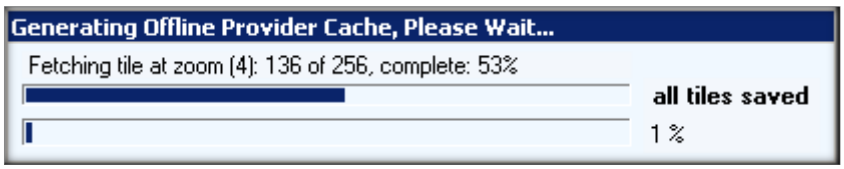
- **Offline Map Source**: Combo box to select the source of the offline map. It can be an online map provider or a combination of offline and vectorial map providers.
  - **Online Map Provider**: Combo box to select the online map provider that will act as map source during the caching of the map at design.
  - **Offline Map Provider List:** Grid to combine different offline map providers
- **Offline zoom levels**: Maximum level of zoom that will be cached from the online provider. Above this zoom level the map will lose visual quality.
- **Area**: This textbox control shows the list of 4 coordinates that make up the rectangular area that will be cached from the online map. To select the area easily, the user can click the **Select Area** button that will open a popup window.



This window will allow drawing the rectangular area over an online map. Use the right mouse button to drag the map. Use the mouse wheel to zoom the map. Use the left mouse button meanwhile the Alt key is pressed to generate the rectangular area to be cached.

Once the online map provider, the zoom level, and the area to be cached are indicated, it's time to generate the cached map. To do that the green **Generate cache and accept** button must be clicked.

Upon clicking the button, a popup window will appear showing the current process of cache generation. The process cannot be stopped.



The time required to finish the cache varies greatly depending on size of the area and the maximum zoom level indicated.

When the process finishes, the editing or creation process of the map provider will be accepted.

2.9. Lab 07 - Creating an offline map provider

2.9.1. Introduction

In this lab, you will create an offline map provider based on an online map provider.

2.9.2. Objectives

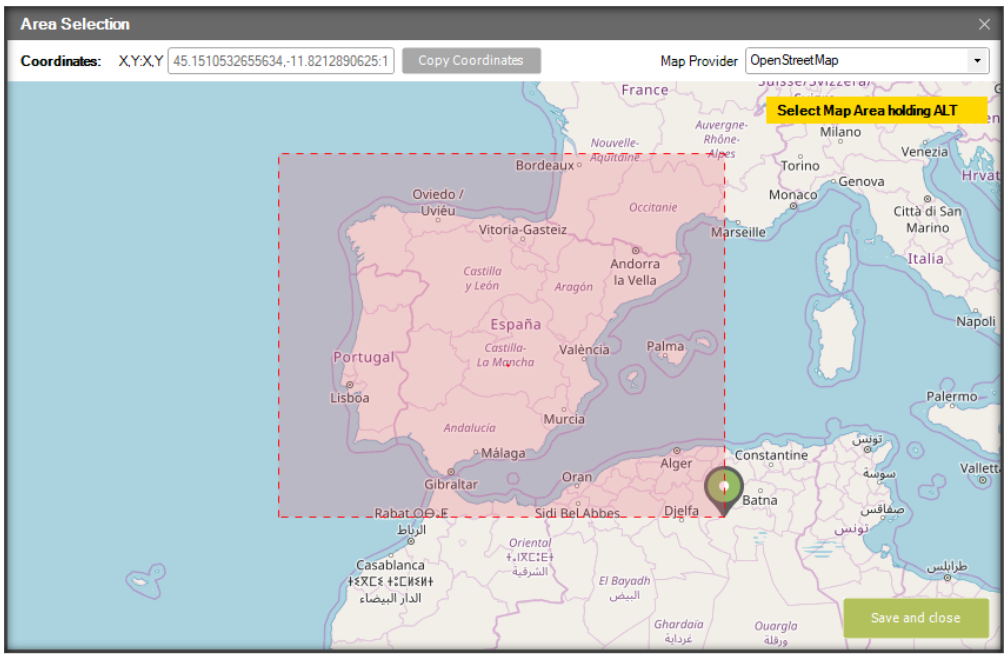
Upon completion of this lab, you will be able to:

- Create an offline map provider.

2.9.3. Create an offline map provider

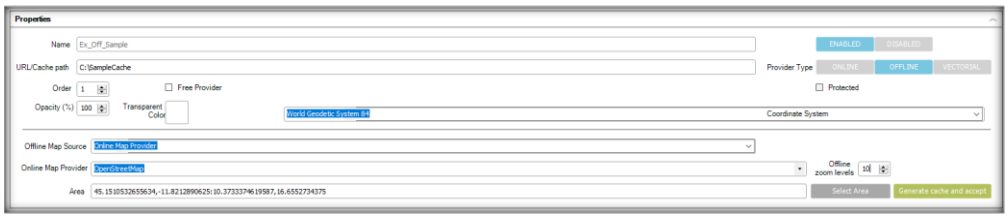
In the following steps, you will create an offline map provider based on an online map provider.

1. Double click the **Map Providers** button.
2. Click the **Add New** button to create a new Map Provider.
3. Enter **Ex_Off_Sample** in the field **Name.**
4. Click the **Enabled** button.
5. Click the **Offline** button.
6. Enter **15** in the **Order** Field.
7. Enter **C:\GISIZECache** in the field **URL/Cache path**.
8. Select **Online Map Provider** in the **Offline Map Source** field.
9. Select **OpenStreetMap** in the **Online Map Provider** field.
10. Enter **6** in the **Offline zoom levels** field.
11. Click the **Select Area** button to open the **Area Selection** dialog box.
12. Select an area for the Spain country holding the ALT key.

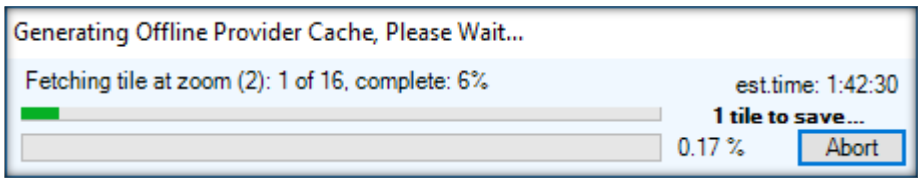


13. Click **Save and close** button.

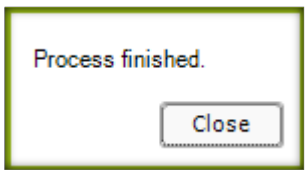
14. Click **Generate cache and accept** to save the configuration and create the cache for this map provider.



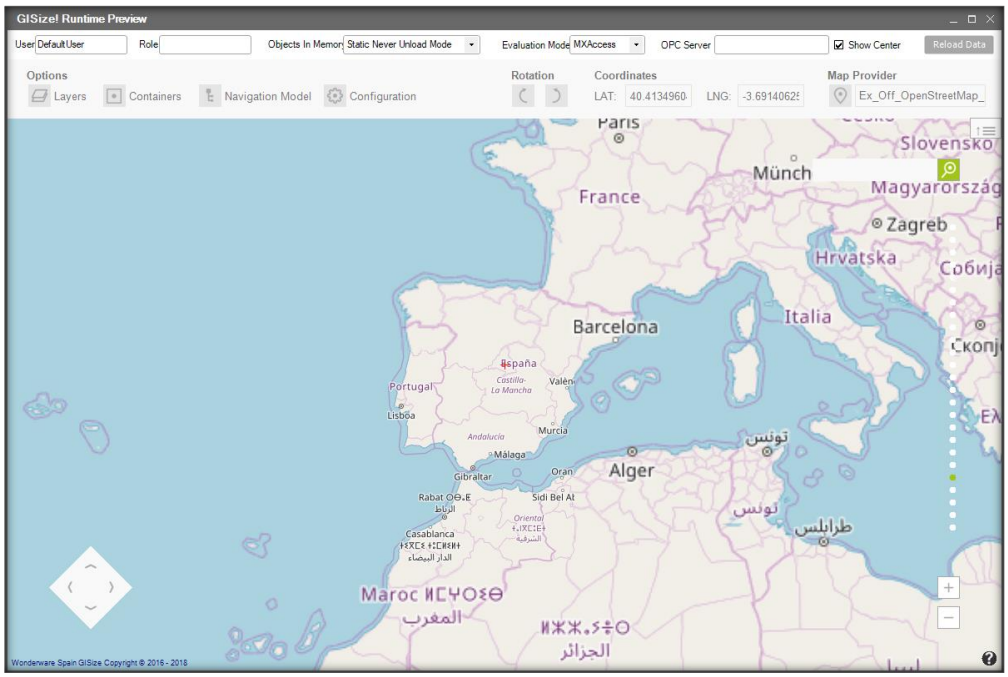
The **Generating Vectorial Provider Cache** dialog box appears and shows the creation progress. This will take a few minutes.



13. Click the **Close** button when the process finished.



14. Open the **Runtime Preview** tool to see the new map provider.



2. Import the Offline Map Provider **Ex_Off_OpenStreetMap_Madrid** located in your training files folder **Lab 07 - Creating an offline map provider** to see an offline map provider with more zoom levels cached.

2.10. Lab 08 - Combine several map providers

2.10.1. Introduction

In this lab, you will combine several offline and vectorial map providers.

2.10.2. Objectives

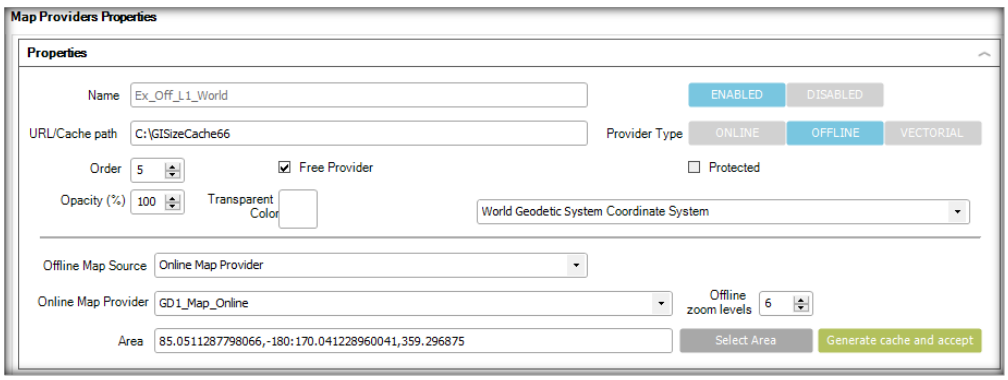
Upon completion of this lab, you will be able to:

- Combine several offline and vectorial map providers.

2.10.3. Combine several offline map providers.

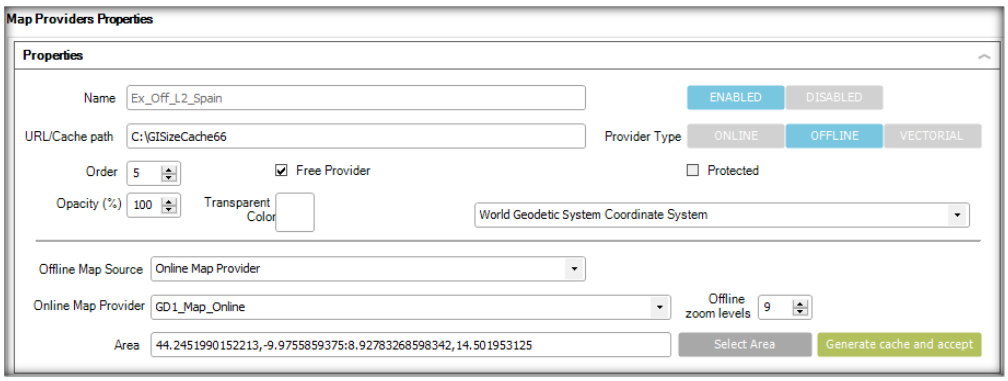
In the following steps, you will combine several offline map providers creating a new offline provider.

1. Import the Offline Map Provider **Ex_Off_L1_World** located in your training files folder **Lab 08 - Combine several map providers.**



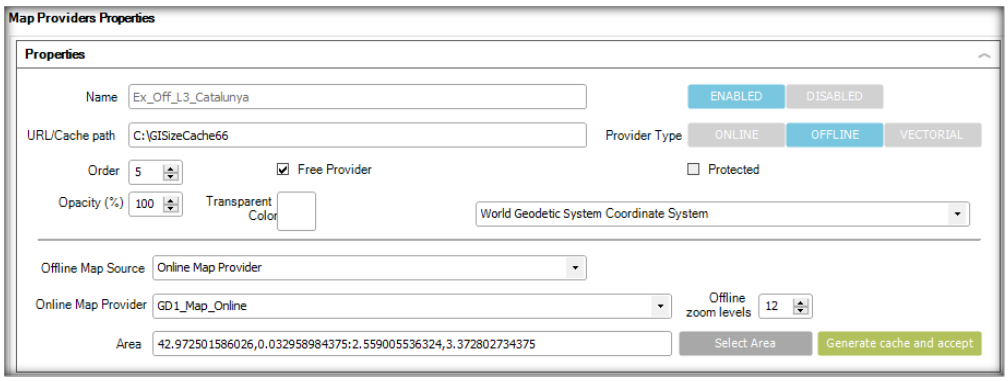
This map provider, with the cache already created, is configured for the entire world area with 6 offline zoom levels.

2. Import the Offline Map Provider **Ex_Off_L2_Spain** located in your training files folder **Lab 08 - Combine several map providers.**



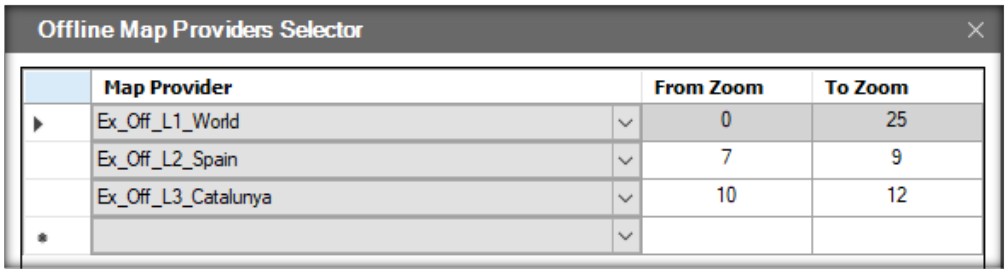
This map provider, with the cache already created, is configured for Spain with 9 offline zoom levels.

3. Import the Offline Map Provider **Ex_Off_L3_Catalunya** located in your training files folder **Lab 08 - Combine several map providers.**

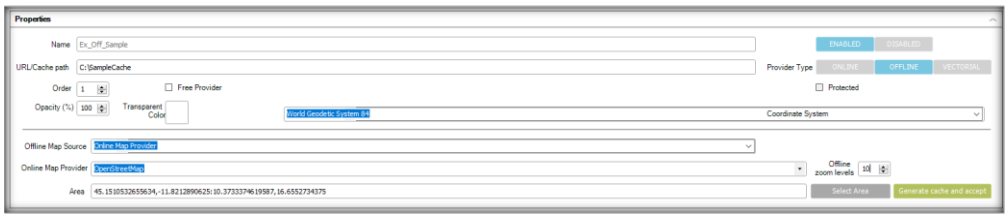


This map provider, with the cache already created, is configured for the Catalunya region (area inside Spain), with 12 offline zoom levels.

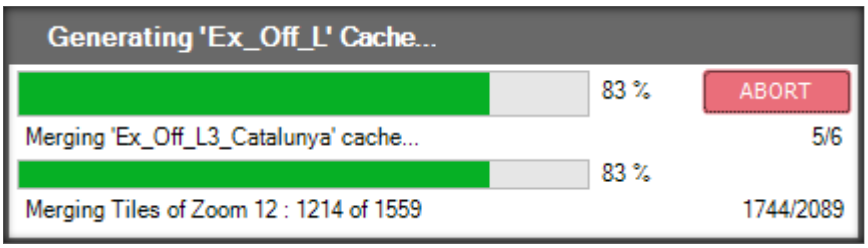
4. Open the **Runtime Preview** tool to check the tiles created for each map provider.
5. Double click the **Map Providers**.
6. Click the **Add New** button to create a new Map Provider.
7. Enter **Ex_Off_L123** in the field **Name.**
8. Enter **15** in the **Order** field.
9. Click the **Enabled** button.
10. Click the **Offline** button.
11. Enter **C:\GISIZECache** in the field **URL/Cache path**.
12. Select **Offline Map Provider** in the **Offline Map Source** field.
13. Click the **Edit Offline Providers List** button to configure the offline map providers used to create the new one.
14. Configure the **Ex_Off_L1_World** offline map as the first map.
15. Configure the **Ex_Off_L2_Spain** offline map as the second map configuring the zoom from **7** to **9**.
16. Configure the **Ex_Off_L3_Catalunya** offline map as the third map configuring the zoom from **10** to **12**.



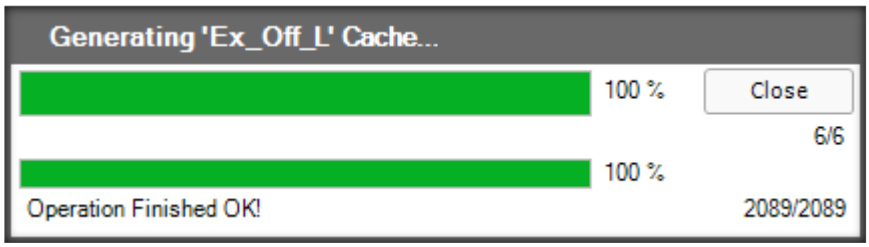
15. Click the **Generate cache and accept** button to save the configuration and create the cache for this map provider.



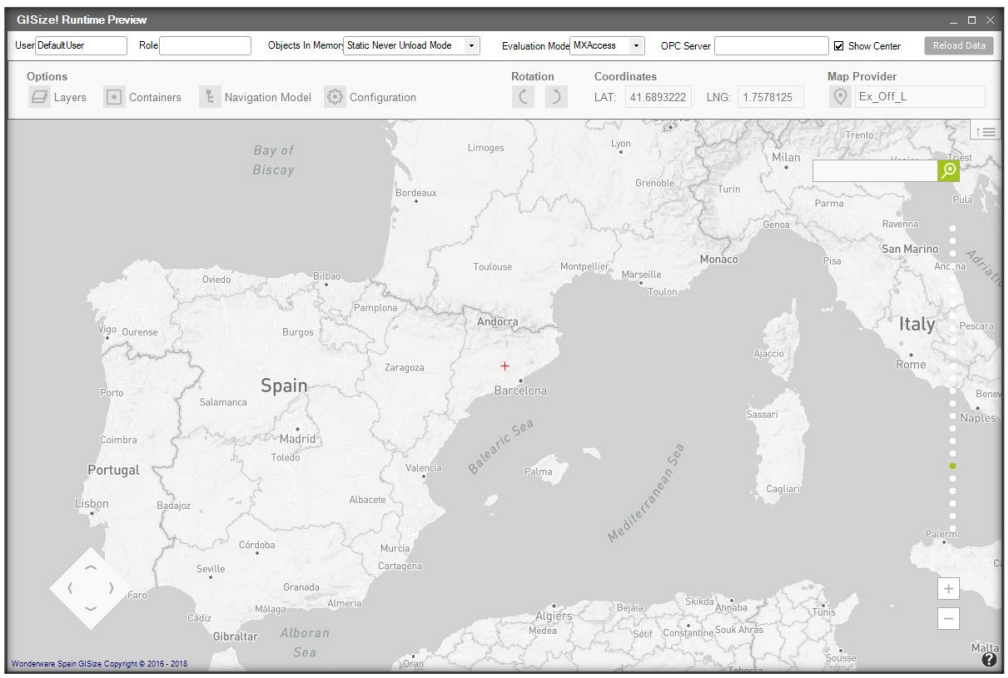
The **Generating Cache** dialog box appears and shows the creation progress. This will take a few minutes.



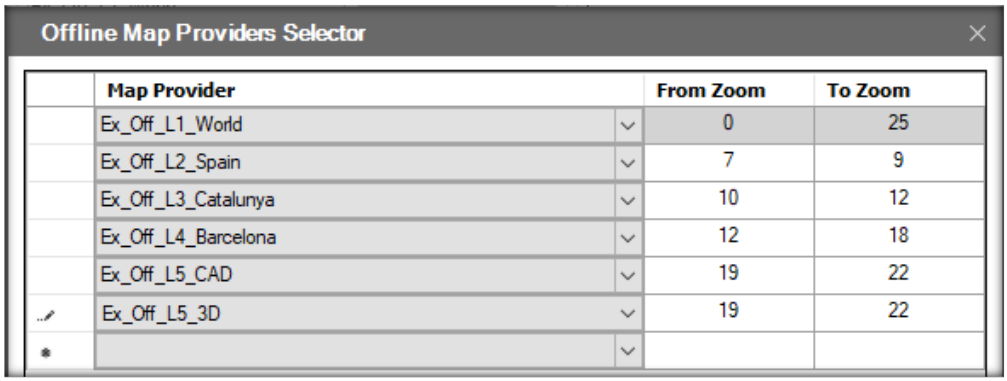
15. Click the **Close** button when the process is finished.



16. Open the **Runtime Preview** tool to see the new map provider.



17. Import the Offline Map Provider **Ex_Off_L4_Barcelona, Ex_Off_L5_CAD, Ex_L5_3D** and **Ex_Off_AllLevels** located in your training files folder **Lab 08 - Combine several map providers**.



The **Ex_Off_AllLevels** combines offline and vectorial map providers.

18. Open the **Runtime Preview** tool to see the new offline zoom levels existing in the **Ex_Off_AllLevels** map provider.

3. Layers and Navigation Models

The objectives of this module are:

- Explain the layers and how to configure their properties for different behaviors.
- Explain the navigation model concept and how to configure it.

3.1. The Layer Model

This section explains the layers and how to configure their properties for different behaviors.

3.1.1. Introduction

The Layers panel allows the definition of new layers and the configuration of the existing ones.

A GISIZE GEOSpatial layer can be conceptually interpreted as a transparent level where markers and shapes are attached. All marker and shape instances belong to a layer.

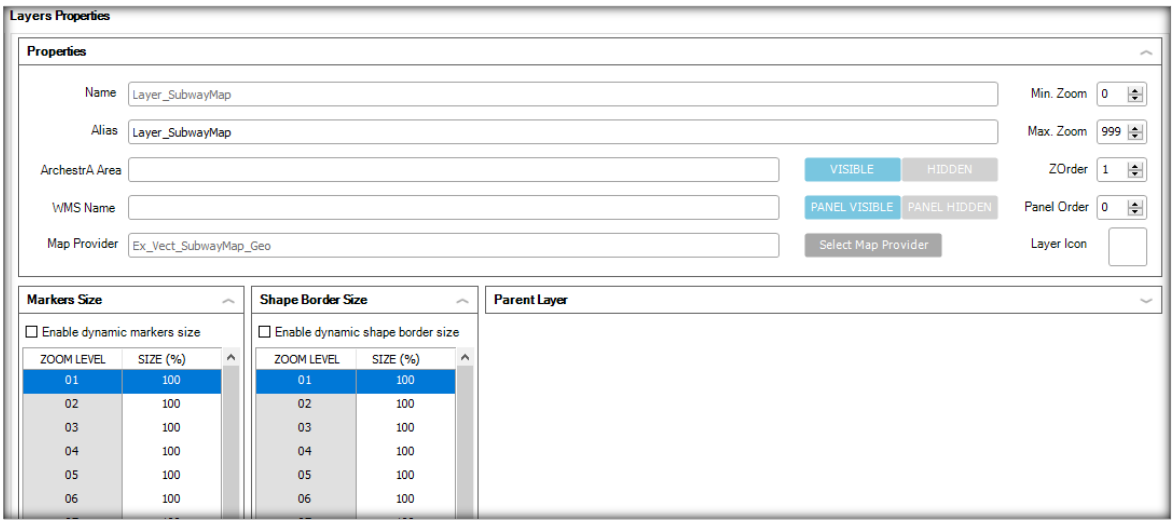
The layer concept allows grouping markers and shapes by user discretion. A visible layer will show its markers and shapes, in the same way that a disabled layer will show none.

The layers have hierarchy, meaning that a layer may depend on another top layer.

The visibility of a layer is restricted by four factors that must take effect at same time: configured visibility, map providers, zoom range, and user permissions:

- **Configured visibility**: A layer can be defined as not visible. This overrides any other conditions.
- **Map Provider**: A layer may be visible only if it's enabled by the map provider visualized at a given time.
- **Zoom Range**: A layer may be visible only if the active zoom at a given time is inside the range of zooms defined for the layer.
- **User Permissions**: If GISIZE GEOSpatial security is enabled, the layer will be visible only if the ArchestrA user logged has rights to see it.

Exceptionally markers and shapes belonging to a disabled layer can be shown if it's indicated at security's system definition.



The panel is divided into left and right sides. On the left side of the panel there is a tree view of available layers and a set of buttons to perform operations with them. On the right side of the panel, there is a collection of controls to modify the properties of a selected layer.

Layers tree

The layer's tree shows the current available layers sorted by name and grouped hierarchically. The hierarchy of a layer can be changed as easily as a drag and drop of the layer over the tree view. Note that the top node of the tree is not a layer but the repository they belong to.

Upon the selection of a layer of the tree view, a set of operations can be performed through buttons. From top to bottom these buttons are:

- **Add New**: This button allows creating a new layer. Clicking on it disables the left side of the panel and enables the right side where the property editing controls are located.

  The minimum requirement to add a new layer is a unique name.

- **Edit:** This button allows editing the properties of a previously existing layer. Clicking on it disables the left side of the panel and enables the right side where the property editing controls are located.

  The name of the layer being edited cannot be changed with this functionality.

- **Rename:** This button allows changing the name of a selected layer. Upon clicking it a popup window will appear to type the new desired name. Duplicated names are not allowed

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

- **Duplicate:** Clicking this button will create a copy of the selected layer. The resulting layer will be named as the original layer name adding the suffix '_Copy'. If a layer already exists with that name, an extra incremental numeric suffix will be added. The hierarchy will be respected.

- **Delete:** Clicking this button will delete the selected layer and all its children layers. Before deleting a layer, the user will be asked for confirmation. This operation cannot be undone.

- **Get Layers:** Clicking this button will refresh the tree of available layers. Usually there is no need to use this button because after performing any operation locally, the tree is refreshed automatically.

  There is a checkbox under the button to filter the tree view to show only the layers defined as visible.

- **Export Selected Layer:** This button allows exporting a single selected layer with all its properties to a XML file. This is a way to back up a layer to be imported later. The file can be edited externally with any text editor.

- **Export All Layers:** This button allows exporting all the existing layers with all its properties into a single XML file.

  This is a way to back up all the layers of a repository to be imported later. The file can be edited externally with any text editor.

- **Import Layers:** This button allows importing an XML file with previously exported layers. Clicking it, the user will be asked to select an XML file within an explorer window. After selecting an XML file, the user will have to choose the way to import the file in case that a layer with the same name if found.

  The import preferences are:

  - **Skip – Do not import**: If an existing layer has the same name as an imported one, the imported one will be ignored and skipped from the import operation. The existing layer will remain unaffected.
  - **Rename importing item (numeric append)**: If an existing layer has the same name as an imported one, the one being imported will be renamed appending an incremental number at the end of its name. The existing layer will remain unaffected
  - **Overwrite existing item**: If an existing layer has the same name as an imported one, the existing one will be overwritten by the one being imported. Configuration of the previously existing layer will be lost.

  Note that the importing process of a layer may fail if an imported layer belongs to a parent layer that doesn't exists.

3.1.2. Layer Properties

The right side of the Layers panel contains the property controls of the layers. Upon the selection of a layer, the property controls will be updated with its values. These controls will remain disabled unless a layer is being edited or created.

To accept any modification made during the editing or the creation of a layer, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a layer, user must click the button 'Do not save and close'. Clicking this button meanwhile a new layer is being created will abort the process. In both cases the ongoing process will come to an end.

The available layer's properties can be classified in three groups: Main properties, Markers size, and Parent Layer.

3.1.2.1. Main properties

These are the common properties of the layers, and its controls are located in the top-right part of the panel.

From top to bottom and left to right, the properties are:

- **Name:** The unique identifier name of the layer. There cannot be 2 layers with the same name. This control only is enabled during the process of creating a new layer. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Name cannot be left empty.
- **Alias:** This is the visible name of the layer at runtime. Unlike the name property, alias property can be duplicated and most characters are allowed. If Alias is left empty, it will automatically take the same value as Name property.
- **ArchestrA Area:** This is the name of an existing Area in the ArchestrA Galaxy associated to the layer. It is used by the integration update wizard process to know from witch Area the layer must be updated.

- **WMS Name:** List of Web Map Service layers associated to the GISIZE GEOSpatial layer. It's used for custom online map providers that contain the parameter {6} in the URL definition.
- **Map Provider.** A Map Provider can be linked to a Layer to superimpose it the current Map Provider when the layer is enabled.
- **Layer Icon:** This control allows associating a small image to the layer. The image will be shown at the Layers Tab at runtime. Clicking in the control will open an explorer window to select an image. If the layer has a previously associated image, it will be shown inside the control.
- **Visible / Hidden:** These buttons define the visibility of the layer. A hidden layer will never be shown. The main purpose of this functionality is to hide a layer temporarily, either because it is in development or because it will be used in the future.
- **Panel Visible / Panel Hidden:** These buttons define the visibility of the layer in the Layers Pane Selector at Runtime. However, its operation will be the same as the other layers, that is, they can be enable/disable by scripting and it will be enabled/disabled when enabling/disabling their parents in the selection panel
- **Min. Zoom:** The minimum zoom which the layer will be visible. At lower zooms the layer will not be visible. Default value is 0 meaning there is no minimum zoom.
- **Max. Zoom:** The maximum zoom which the layer will be visible. At higher zooms the layer will not be visible. Default value is 999 meaning there is no maximum zoom.
- **ZOrder:** Indicates the order in which the layers are stacked. It affects the visibility of the markers and shapes of the layers. The items of a layer with lower ZOrder will be displayed over the items of a layer with higher ZOrder.
- **Panel Order:** Indicates the order in the Layers Pane Selector at Runtime.

3.1.2.2. Marker Size

This control is a table located in bottom-left of the panel. The purpose of this table is to define the relative size of the markers that belong to the layer depending on the displayed zoom.

By default, markers are drawn with the same size as the image associated to them. This means that the size of a marker is the same for any zoom displayed. Sometimes this may be an undesired behavior and what is expected is that the marker changes its size accordingly with the zoom.

To activate the dynamic resizing of markers in a layer, the checkbox labeled **Enable dynamic markers size** must be checked.

The table has 2 columns, the left one indicates the level of the zoom and the right one indicates the relative size as a percentage of the original image size of the marker. Each row of the table relates the zoom to the relative size of the marker.

3.1.2.3. Shape Border Size

This control is a table located at bottom-center of the panel. The purpose of this table is to define the relative thickness of the shape borders that belong to the layer depending on the displayed zoom.

By default, the thickness of the shape borders is drawn proportionally to the zoom level. Sometimes this may be an undesired behavior and what is expected is that thickness doesn't change or changes at a different rate.

To activate the dynamic resizing of shape borders in a layer, the checkbox labeled **Enable dynamic shape border size** must be checked.

The table has 2 columns, the left one indicates the level of the zoom and the right one indicates the relative size as a percentage of the original thickness of the shape border. Each row of the table relates the zoom to the relative size of the shape border.

3.1.2.4. Parent Layer

Located in the bottom-right side of the panel, there is a tree view that contains all the current defined layers. Each node of the tree (except the very top one) represents a layer.

This control allows indicating which layer is the parent layer of the layer currently being edited.

3.2. Lab 09 - Creating the Layer Model

3.2.1. Introduction

In this lab, you will create a hierarchy model for layers and configure its main properties.

3.2.2. Objectives

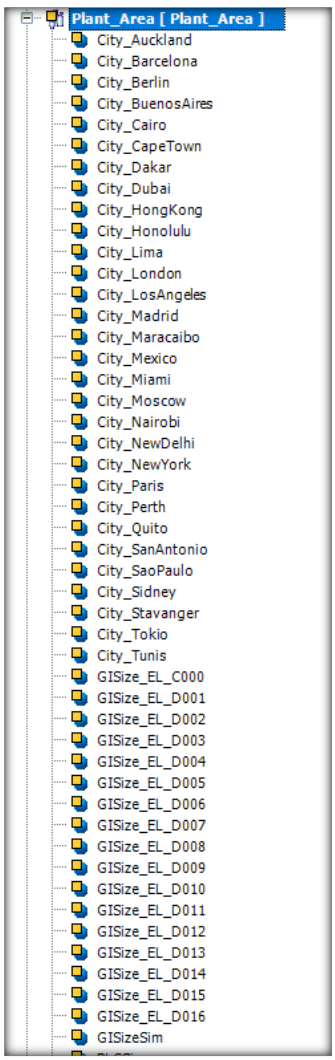
Upon completion of this lab, you will be able to:

- Create a layer.
- Configure the minimum and maximum zoom for a layer.
- Hide a layer in the Layer Panel Selector
- Enable and configure dynamic marker size.

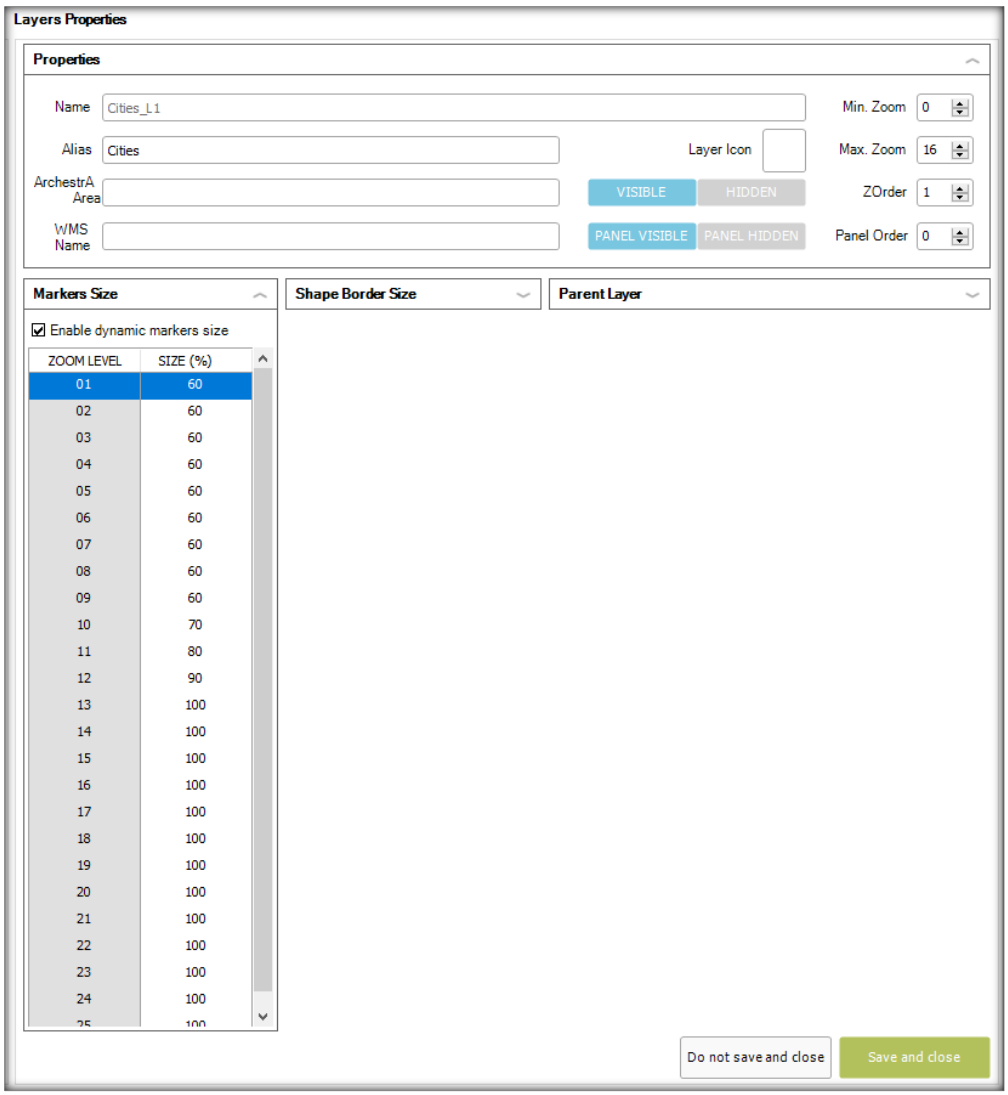
3.2.3. Create the layer model

In the following steps, you will create a layer model.

1. Import the Script Function Library **OpenWeather.aaSLIB** located in your training files (folder **Lab 09 - Creating the Layer Model**).
2. **Import** the Wonderware objects **Cities_Energy_Sim_Demos.aaPKG** located in your training files (folder **Lab 09 - Creating the Layer Model**).

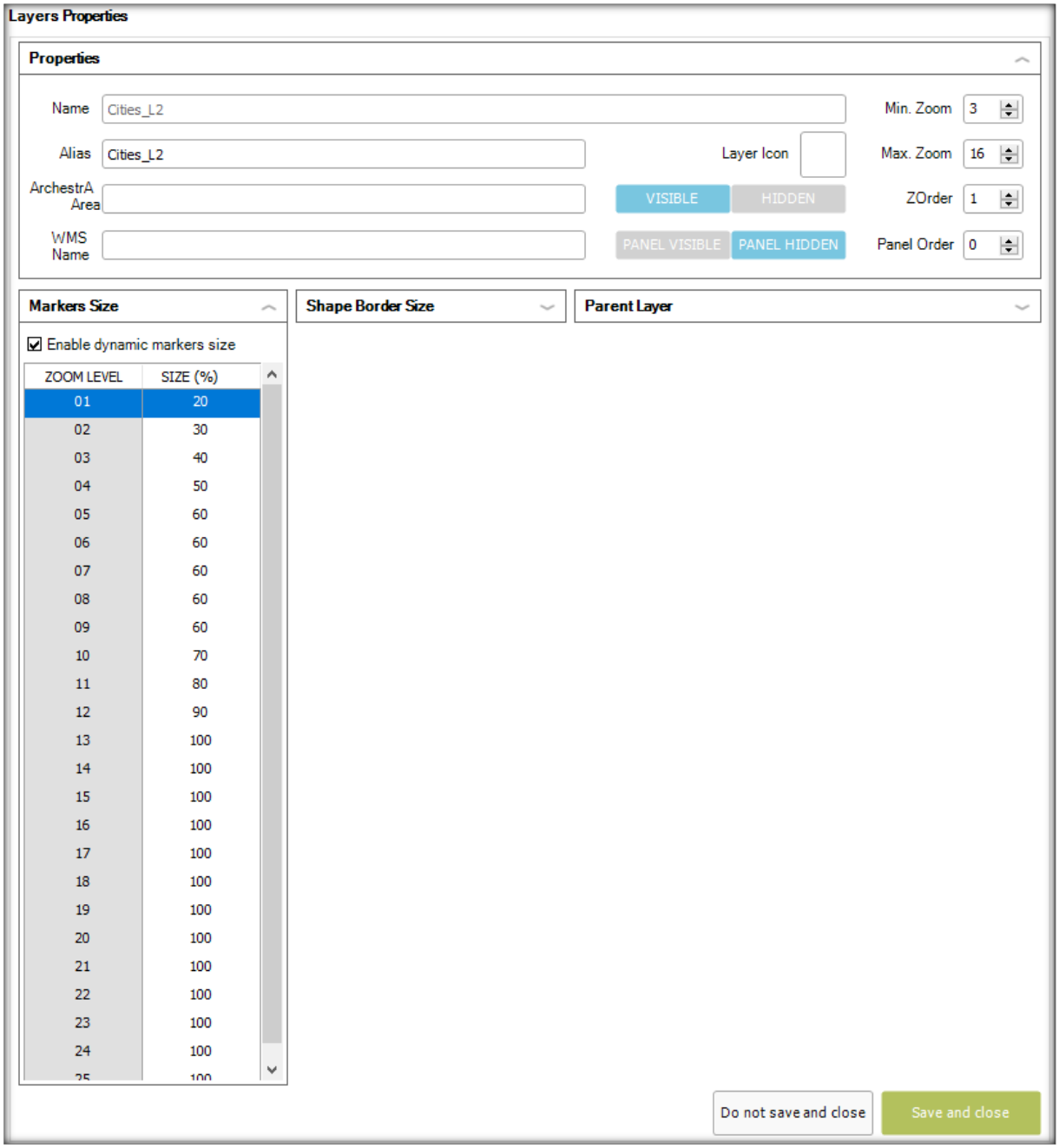


3. **Deploy** these new Wonderware objects located in the **Plant_Area**.
4. Double click the **Layers** button.
5. Click the **Add New** button to create a new Layer.
6. Enter **Cities_L1** in the **Name** field.
7. Enter **Cities** in the **Alias** field.
8. Enter **16** in the **Max. Zoom** field.
9. Expand **Markers Size** and check the **Enable dynamic markers size** option.
10. Enter **60** in the **size (%)** column for zoom levels **01 to 09**.
11. Enter **70, 80 and 90** in the **size (%)** column for zoom levels **10, 11 and 12**.



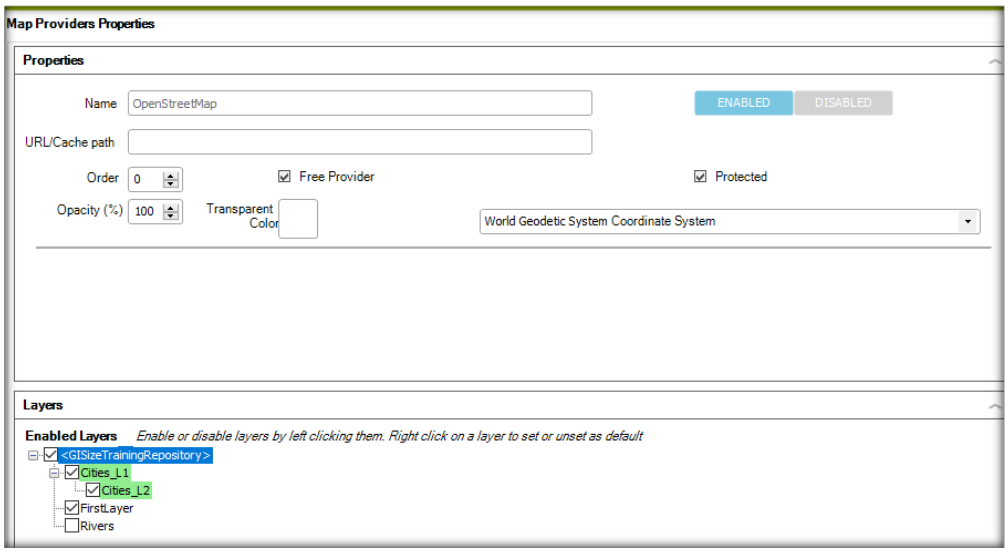
12. Click **Save and Close** to create the new layer.
13. Click the **Add New** button to create a new Layer.
14. Enter **Cities_L2** in the **Name** field.
15. Enter **Cities_L2** in the **Alias** field.
16. Enter 3 in the **Min. Zoom** field.

17. Enter **16** in the **Max. Zoom**.
18. Expand **Markers Size** and check the **Enable dynamic markers size** option.
19. Click the **Panel Hidden** button to hide the layer in **Layer Panel Selector**.
20. Enter **20, 30, 40, and 50** in the **size (%)** column for zoom levels **01, 02, 03 and 04**.
21. Enter **60** in the **size (%)** column for zoom levels **05 to 09**.
22. Enter **70, 80 and 90** in the **size (%)** column for zoom levels **10, 11 and 12**.

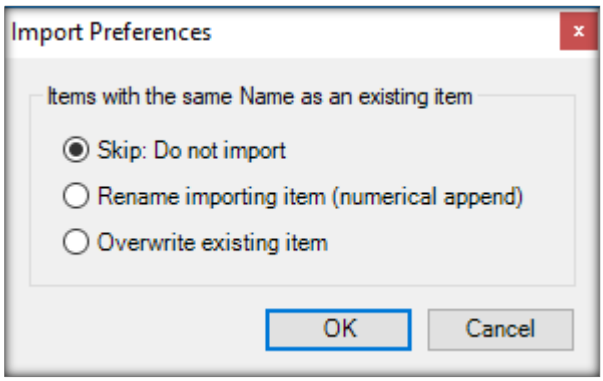


23. Click **Save and Close** to create the new layer.
24. Double click the **Map Providers** button.
25. Select **OpenStreenMap** and click the **Edit** button to edit the map provider.

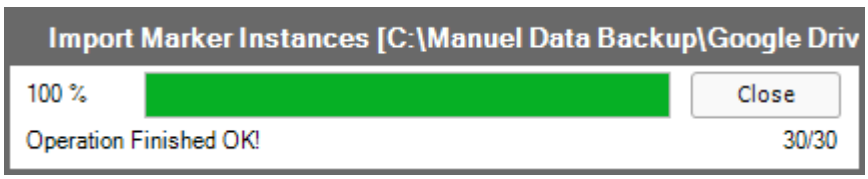
26. Enable the **Cities_L1** and **Cities_L2** layers, and right click to set as **default** layers for the map.



27. Double click the **Marker Instances** button.
28. Click the **Import Marker Instances** button to import new cities.
29. Select **Skip: Do not import** in the Import Preferences dialog box.

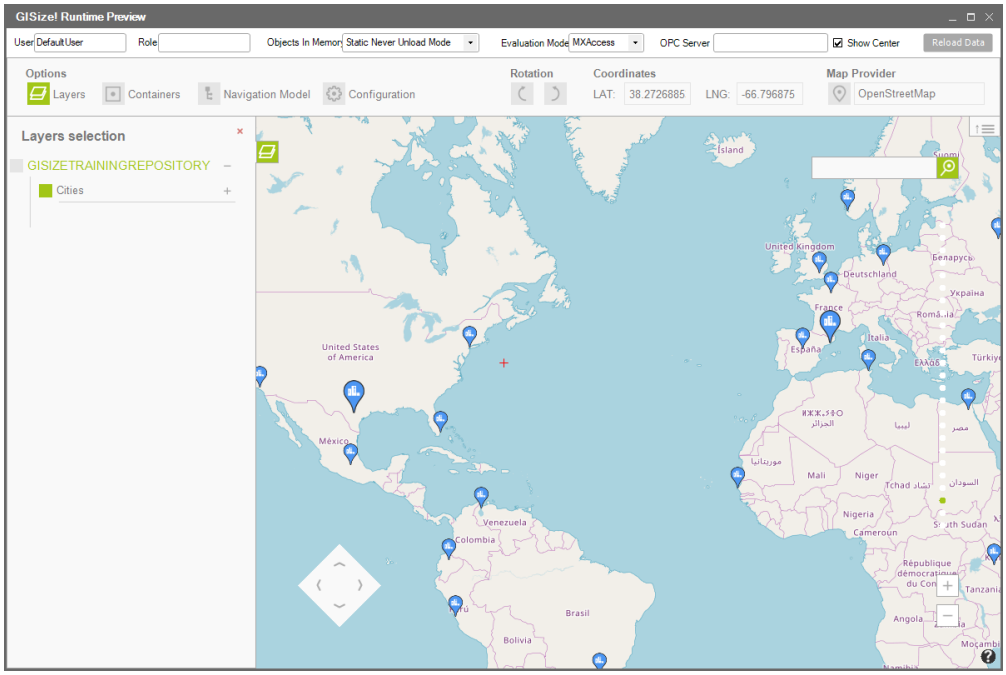


30. Click the **Ok** button and select the file **MarkerInstances_GD_Cities.xml** located in your training files (folder **Lab 09 - Creating the Layer Model**).



31. Click the **Close** button when the process finish.

32. Open the Runtime Preview tool to see the behavior of the objects enabling the layers and zooming in/out in the map.



3.3. Navigation Models

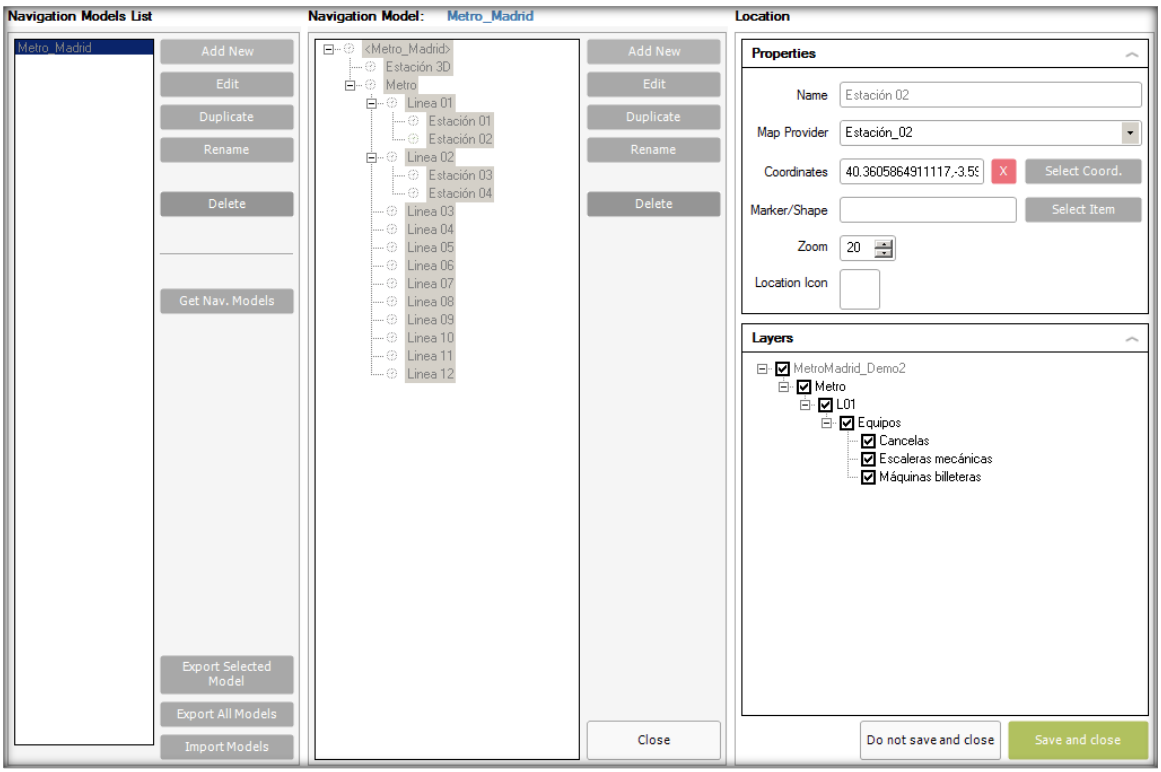
This section explains the navigation model concept and how to configure it.

3.3.1. Introduction

Conceptually a navigation model is a hierarchically predefined set of map visualizations (Locations) composed of: a map provider, a set of active layers, a default zoom, and a configured location.

In itself, a navigation model is just a collection of locations organized hierarchically.

At the runtime, selecting a location of a navigation model allows navigating to a defined position at a defined zoom in a defined map with a defined set of layers active.



The panel is divided into three parts. On the left side of the panel there is a list of currently defined navigation models and a set of buttons to perform operations with them. At the center part of the panel there is a tree view that shows the hierarchic distribution of the locations that define a selected navigation model. Finally, on the right side of the panel there is a collection of controls to modify the properties of a selected location of the current navigation model.

3.3.2. Navigation Models List

The navigation models list shows the current available navigation models sorted alphabetically. Upon the selection of a navigation model of the list, a set of operations can be performed through the buttons next to it. From top to bottom these buttons are:

- **Add New:** This button allows creating a new navigation model. Upon clicking it a popup window will appear asking the user a name for the new navigation model. Duplicated names are not allowed.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

After introducing a name, a new empty navigation model will be created, the left side of the panel will be disabled and the central part where the tree view of its locations is located will became enabled. The newly created navigation model will switch to editing mode.

- **Edit:** This button allows editing of a selected navigation model.
- **Duplicate:** This button will create a copy of the selected navigation model. The resulting navigation model will be named as the original layer name adding the suffix '_Copy'. If a navigation model already exists with that name, an extra incremental numeric suffix will be added.

  In addition, as the names of the locations must be unique, the names of the locations of a navigation model being duplicated will be renamed adding an incremental numeric suffix to them. The hierarchy of the locations will be respected.
- **Rename:** This button allows changing the name of a selected navigation model. Upon clicking it a popup window will appear to type the new desired name. Duplicated names are not allowed. The names of the locations belonging to the navigation model will not be renamed.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.
- **Delete:** Clicking this button will delete the selected navigation model and all the locations belonging to it. Before deleting a navigation model the user will be asked for confirmation. This operation cannot be undone.
- **Get Nav. Models:** Clicking this button will refresh the list of available navigation models. Usually there is no need to use this button because after performing any operation locally, the list is refreshed automatically.
- **Export Selected Model:** This button allows exporting a single selected navigation model with all its locations and properties to a XML file. This is a way to back up a navigation model to be imported later. The file can be edited externally with any text editor.
- **Export All Models:** This button allows exporting all the existing navigation models with all its locations and properties into a single XML file.

  This is a way to back up all the navigation models of a repository to be imported later. The file can be edited externally with any text editor.
- **Import Models:** This button allows importing an XML file with previously exported navigation models. Clicking it, the user will be asked to select an XML file within an explorer window. After selecting an XML file, the user will have to choose the way to import the file if a navigation model with the same name is found.

  The import preferences are:

  - **Skip – Do not import**: If an existing navigation model has the same name as an imported one, the imported one will be ignored and skipped during the importing operation. The existing navigation model will remain unaffected.
  - **Rename importing item (numeric append)**: If an existing navigation model has the same name of an imported one, the one being imported and its locations will be renamed appending an incremental number at the end of its name. The existing navigation model will remain unaffected

- **Overwrite existing item**: If an existing navigation model has the same name of an imported one, the existing one will be overwritten with the one being imported. Configuration of the previously existing navigation model will be lost.

3.3.3. Navigation Model Tree

The navigation model tree shows a hierarchical tree view of the locations defined in the navigation model being edited. Except the top one, every node of the tree represents a location.

The hierarchical organization of the locations is mostly esthetical, as it's the way they will be presented at the runtime. The hierarchy of a location from the navigation model can be changed as easily as a drag and drop of the location over the tree view.

Upon the selection of a navigation model of the list, a set of operations can be performed through the buttons next to it. From top to bottom these buttons are:

- **Add New:** This button allows creating a new location for the current navigation model. Clicking it disables the central part of the panel and enables the right side.

  The minimum requirement to add a new location is a unique name and the selection of a map provider.

- **Edit:** This button allows editing the properties of a previously existing location from a navigation model. Clicking it disables the central part of the panel and enables the right side.

  The name of the location being edited cannot be changed with this functionality.

- **Duplicate:** This button will create a copy of the selected location from a navigation model.

  The resulting location will be named as the original location name adding the suffix '_Copy'. If a location already exists with that name, an extra incremental numeric suffix will be added. The hierarchy of the locations will be respected.

- **Rename:** This button allows changing the name of a selected location from a navigation model. Upon clicking it a popup window will appear to type the new desired name. Duplicated names are not allowed.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

- **Delete:** Clicking this button will delete the selected location from a navigation model and all its children locations. Before deleting a location, the user will be asked for confirmation. This operation cannot be undone.

3.3.3.1. Location Properties

The right side of the Navigation Models panel contains the property controls of the locations belonging to navigation models. Upon the selection of a location at the central part of the panel, the property controls will be updated. These controls will remain disabled unless a location is being edited or created.

To accept any modification made during the editing or the creation of a location, user must click the green **Save and close** button. It will also finish the editing or creation process.

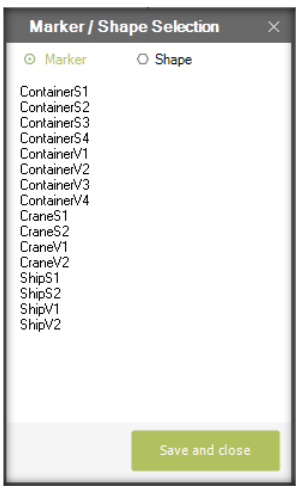
To cancel any modification made during the editing of a location, user must click the button **Do not save and close**. Clicking this button while a new location is being created will abort the process. In both cases the ongoing process will come to an end.

The available location properties can be classified into two groups: Main properties and Layers.

- **Main properties:** These are the common properties of the locations, and their controls are located on the right part of the panel.

  From top to bottom, the properties are:

  - **Name**: The unique identifier name of the location. There cannot be 2 locations with the same name, even if they belong to different navigation models. This control is only enabled during the process of creating a new location. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Name cannot be left empty.
  - **Map Provider**: This combo box allows the selection of the map provider that will be used at the runtime when navigating to the selected location. Note that only enabled map providers can be selected. Map provider cannot be left empty.
  - **Coordinates**: This textbox shows the coordinates (latitude and longitude) of the map that will be centered at the runtime when navigating to the selected position. The coordinates can be easily selected by clicking the **Select Coord.** button. It will open a new window with a GISIZE GEOSpatial control. Left clicking the map to set the coordinates. The default value is 0,0.
  - **Marker/Shape**: This is an alternative method to the coordinates. User can type the name of an existing marker or shape instance and the map will be centered at the runtime over the item when navigating to the selected location. The item can be easily selected by clicking the **Select Item** button. It will open a new window with a list of the currently existing marker and shape instances. The default value is empty.

  - **Zoom**: Indicates the zoom of the map that will be applied at the runtime when navigating to the selected location.

- **Location Icon**: This control allows the association of an image to the location. This image will be shown at the runtime in the Navigation Models Tab. Clicking the **Control** button will open an explorer window to select an image. If the location has a previously associated image, it will be shown inside the control.

- **Layers**

  Located at the right-bottom side of the panel, there is a tree view that contains the current layers accessible by the previously selected map provider. Each node of the tree (except the very top one) represents a layer.

  Note that even if a children layer is marked as enabled by a certain map provider, if the parent layer is not enabled, the children node will not appear in the tree view, thus making it not selectable.

3.4. Lab 10 - Creating a navigation model

3.4.1. Introduction

In this lab, you will create some examples of navigation models.

3.4.2. Objectives

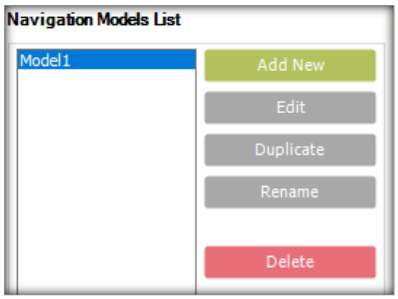
Upon completion of this lab, you will be able to:

- Create a new Navigation Model List.
- Configure a location scenario with a specific map provider, zoom, coordinates and layers.
- Configure a location scenario with a specific map provider, zoom and item.

3.4.3. Create the navigation model

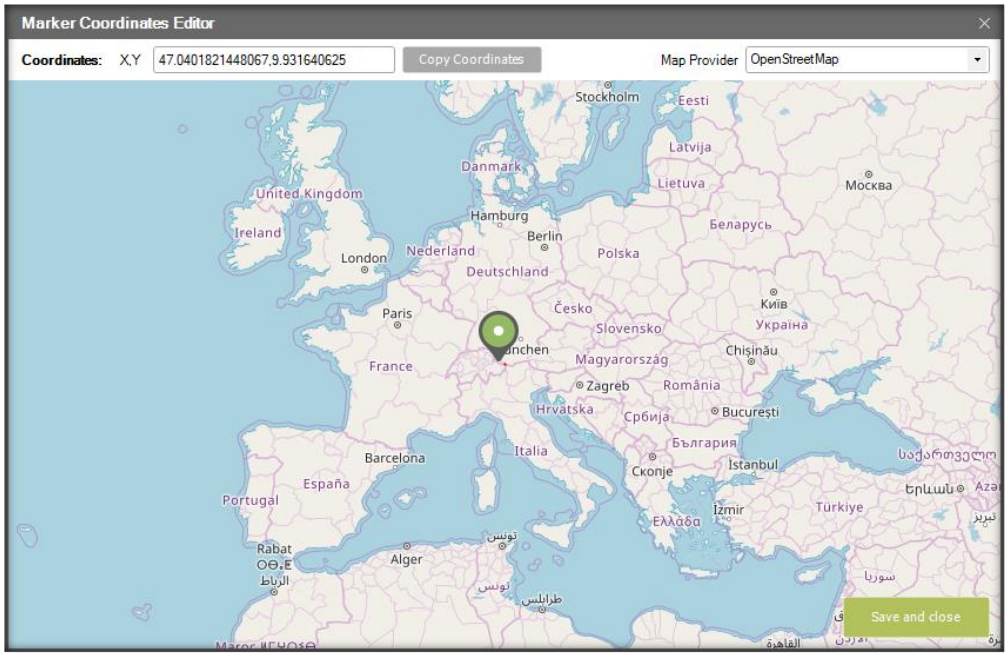
In the following steps, you will create a navigation model.

1. Double click the **Navigation Models** button.
2. Click the **Add New** button to create a new Navigation Model.
3. Enter **Model1** in the **New Navigation Model** dialog box.

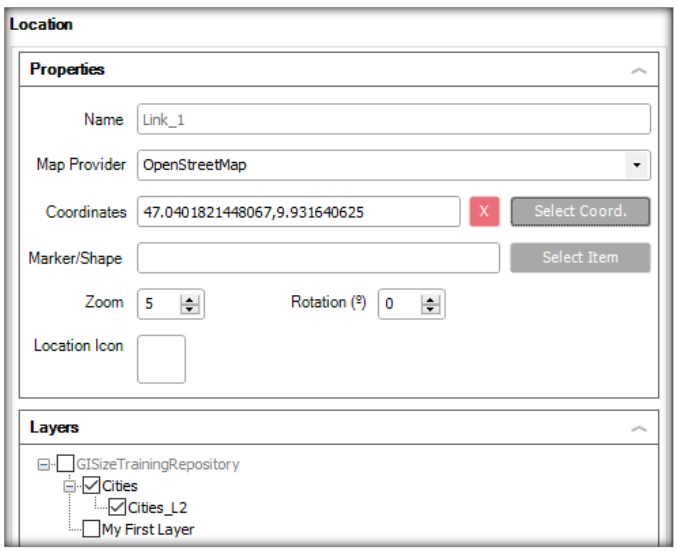


4. **Edit** the **Model1** and click the **Add New** button to create a new location.
5. Enter **Link_1** in the Name field.
6. Select **OpenStreetMap** in the **Map Provider** combo box.

7. Click the **Select Coord.** button to select a coordinate of **some place in Europe**.

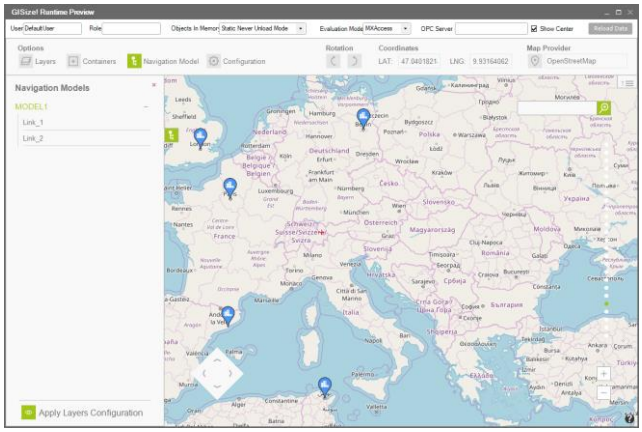


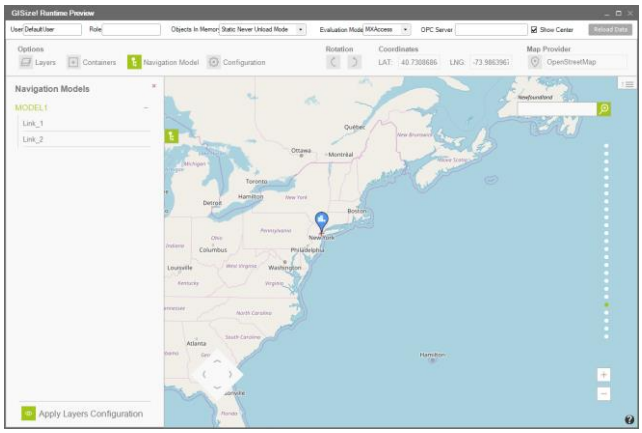
8. Enter **5** in the Zoom field.
9. Expand **Layers** and enable the **Cities_L1 and Cities_L2** layers.
10. Click the **Save and close** button to create the new location.



11. Select the **Link_1** location and click the **Duplicate** button to create a new location using the same configuration.
12. Click the **Edit** button to edit the location.
13. Click the **Select Item** button and select **City_NewYork** in the **Marker/Shape Selection** dialog box.
14. Click the **Save and close** button to save the configuration.

15. Open **the Runtime Preview** tool, open the **Navigation Models Panel**, and click the links to go to each location.





4. GISIZE GEOSpatial Objects

The objectives of this module are:

- Explain the markers and shapes as the centerpiece of the object-oriented framework of GISIZE GEOSpatial.
- Describe the properties of the markers and shapes and how to create and configure Custom Properties and Animations.
- Create and configure tooltips.
- Create GISIZE GEOSpatial instances automatically using the Import and Export features.
- Describe the utilities to create and update GISIZE GEOSpatial objects automatically using the Wonderware model as a reference.

4.1. Introduction to Markers and Shapes

This section explains the markers and shapes as the centerpiece of the object-oriented framework of GISIZE GEOSpatial.

4.1.1. Introduction

The Markers and Shapes are the centerpiece of the object-oriented framework of GISIZE GEOSpatial. Through these objects, you can model virtually any asset that you want to represent as spatial or geographical data on a Geographic Information System (GIS).

- Markers

  A pointer to a specific coordinate on a map that is represented as an image

- Shapes

  A collection of specific coordinates on a map that are represented as a polyline if they are opened or a polygon if they are closed

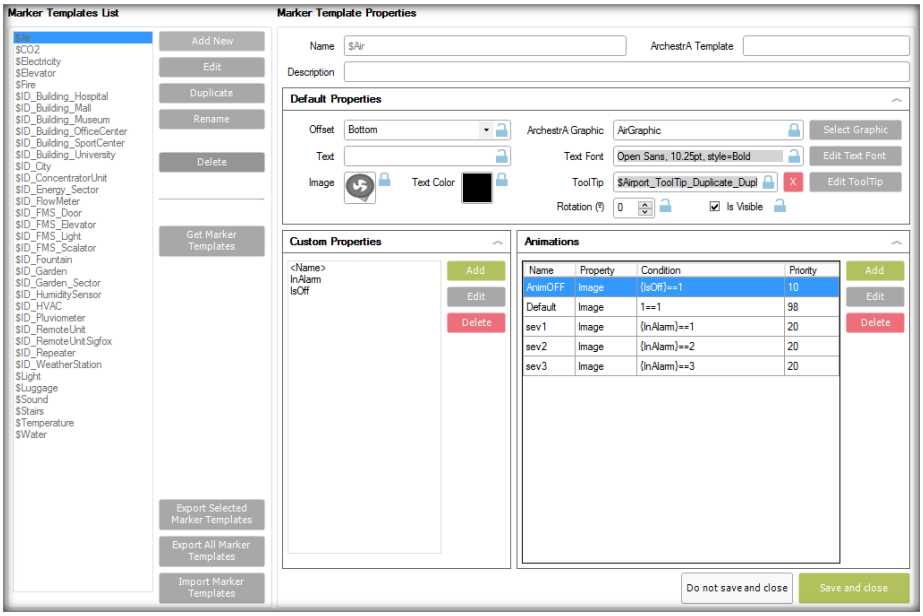
The GISIZE GEOSpatial Objects are provided in the form of templates and instances. Templates allow the configuration of reusable standards while instances implement the standards for each individual asset. For example, all common configuration for a flowmeter in the application can be modeled as a $FlowMeter Marker Template and then create instances configuring only the coordinates of each asset.

4.1.2. Marker Templates

The Marker Template panel allows the definition of new marker templates and the configuration of the existing ones.

As its name indicates, a marker template is the abstract definition of a map marker class that may be used to create derivate marker instances that inherit its properties. Modifying a marker template may potentially modify all its derived marker instances.

Technically a marker is a pointer to a specific coordinate on a map that is represented as an image. Marker templates are not associated to any layer as they don't have physical representation. For the same reason, marker templates don't have defined coordinates.



The panel is divided into left and right sides. On the left side of the panel there is a list of currently available marker templates and a set of buttons to perform operations. On the right side of the panel there is a collection of controls to modify the properties of a selected marker template.

4.1.2.1. Marker Templates List

The marker templates list shows the current available marker templates sorted alphabetically. Note that all the template names start with the dollar symbol '$', similarly to ArchestrA nomenclature. Upon the selection of a marker template of the list, a set of operations can be performed through the buttons next to it. From top to bottom these buttons are:

- **Add New:** This button allows creating a new marker template. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The minimum requirement to add a new marker template is a unique name.

- **Edit:** This button allows editing the properties of a previously existing marker template. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The name of the marker template being edited cannot be changed with this functionality.

- **Duplicate:** This button allows creating a copy of the selected marker template. The resulting marker template will be named as the original marker template name adding the suffix '_Copy'. If a marker template already exists with that name, an extra incremental numeric suffix will be added.

- **Rename:** This button allows changing the name of a selected marker template. Upon clicking it a popup window will appear to type the new desired name. Duplicated names are not allowed.

  If the maker template has any derived marker instances, these will be automatically updated to make reference to the newly named maker template.

Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

- **Delete:** Clicking this button will delete the selected marker template with all its properties. Before deleting a marker template, the user will be asked for confirmation. This operation cannot be undone.

  Be aware that deleting a marker template will also delete all its derived marker instances, thus leading to a possible undesired information loss.

- **Get Marker Templates:** Clicking this button will refresh the list of available marker templates. Usually there is no need to use this button because after performing any operation locally, the list is refreshed automatically.

- **Export Selected Marker Templates:** This button allows exporting the currently selected marker templates with all its properties to an XML file. This is a way to backup marker templates to be imported later. The file can be edited externally with any text editor.

- **Export All Marker Templates:** This button allows exporting all the existing marker templates in a repository with all its properties into a single XML file.

  This is a way to back up all the marker templates of a repository to be imported later. The file can be edited externally with any text editor.

- **Import Marker Templates:** This button allows importing an XML file with previously exported marker templates. Upon clicking it, the user will be asked to select an XML file within an explorer window. After selecting an XML file, the user will have to choose the way to import the file if a marker template with the same name is found.

  The import preferences are:

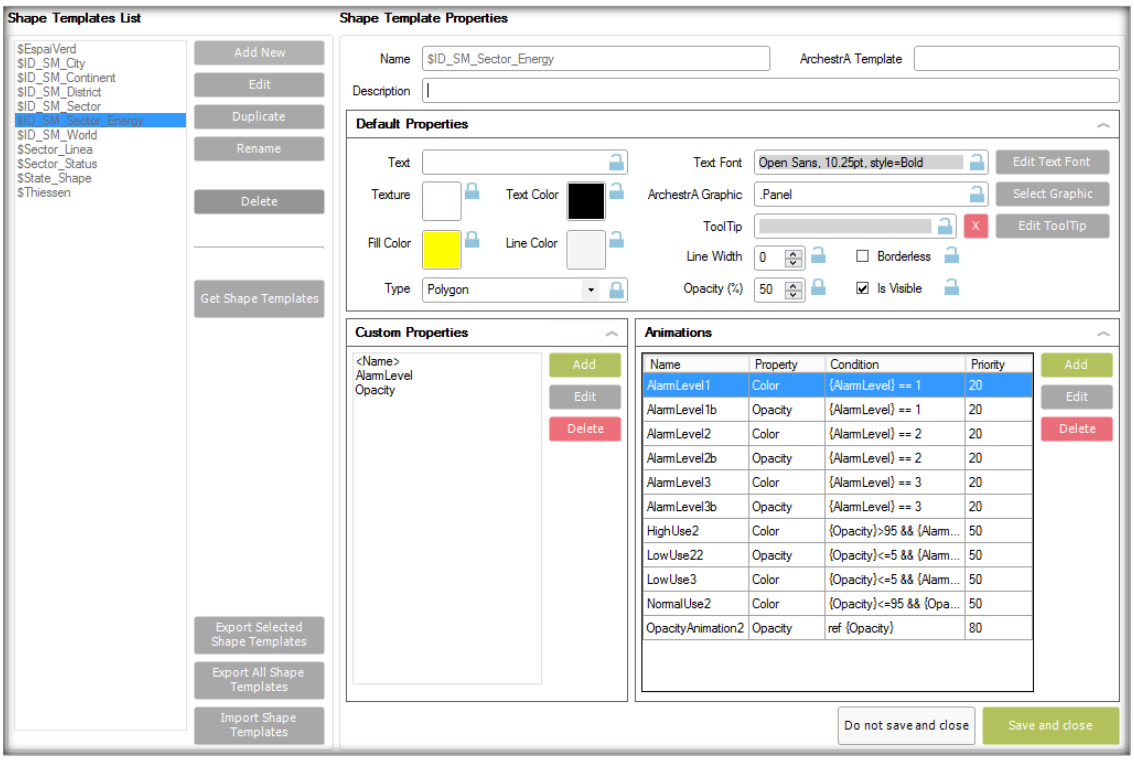
  - **Skip – Do not import**: If an existing marker template has the same name as an imported one, the imported one will be ignored and skipped from the importing operation. The existing marker template and its derived marker instances will remain unaffected.
  - **Rename importing item (numeric append)**: If an existing marker template has the same name as an imported one, the one being imported will be renamed appending an incremental number at the end of its name. The existing marker template and its derived marker instances will remain unaffected.
  - **Overwrite existing item**: If an existing marker template has the same name as an imported one, the existing one will be overwritten with the one being imported. Configuration of the previously existing marker template will be lost. Derived marker instances will be updated according to the new imported marker template.

4.1.3. Shape Templates

The Shape Template panel allows the definition of new shape templates and the configuration of the existing ones.

As its name indicates, a shape template is the abstract definition of a map shape class (polygon or polyline) which may be used to create derivate shape instances that inherit its properties. Modifying a shape template may potentially modify all its derived shape instances.

Technically a shape is a collection of specific coordinates on a map that are represented as a line if they are open or a polygon if they are closed. Shape templates are not associated to any layer as they don't have physical representation. For the same reason, shape templates don't have defined coordinates.



The panel is divided into left and right sides. On the left side of the panel there is a list of currently available shape templates and a set of buttons to perform operations with them. On the right side of the panel there are a collection of controls to modify the properties of a selected shape template.

4.1.3.1. Shape Templates List

The shape template list shows the current available shape templates sorted alphabetically. Note that all the template names start with the dollar symbol '$', similarly to ArchestrA nomenclature. Upon the selection of a shape template of the list, a set of operations can be performed through the buttons next to it. From top to bottom these buttons are:

- **Add New:** This button allows creating a new shape template. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The minimum requirement to add a new shape template is a unique name.

- **Edit:** This button allows editing the properties of a previously existing shape template. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The name of the shape template being edited cannot be changed with this functionality.

- **Duplicate:** This button allows creating a copy of the selected shape template. The resulting shape template will be named as the original shape template name adding the suffix '_Copy'. If a shape template already exists with that name, an extra incremental numeric suffix will be added.

- **Rename:** This button allows changing the name of a selected shape template. Upon clicking it a popup window will appear to type the new desired name. Duplicated names are not allowed.

  If the shape template has any derived shape instances, these will be automatically updated to make reference to the newly named shape template.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

- **Delete**

  Clicking this button will delete the selected shape template with all its properties. Before deleting a shape template, the user will be asked for confirmation. This operation cannot be undone.

  Be aware that deleting a shape template will also delete all its derived shape instances, thus leading to a possible undesired information loss.

- **Get Shape Templates**

  Clicking this button will refresh the list of available shape templates. Usually there is no need to use this button because after performing any operation locally, the list is refreshed automatically.

- **Export Selected Shape Templates**

  This button allows exporting the currently selected shape templates with all its properties to an XML file. This is a way to backup shape templates to be imported later. The file can be edited externally with any text editor.

- **Export All Shape Templates**

  This button allows exporting all the existing shape templates in a repository with all its properties into a single XML file.

  This is a way to back up all the shape templates of a repository to be imported later. The file can be edited externally with any text editor.

- **Import Shape Templates**

  This button allows importing an XML file with previously exported shape templates. Upon clicking it, the user will be asked to select an XML file within an explorer window. After selecting an XML file, the user will have to choose the way to import the file if a shape template with the same name is found.

  The import preferences are:

  - **Skip – Do not import**: If an existing shape template has the same name as an imported one, the imported one will be ignored and skipped from the

importing operation. The existing shape template and its derived marker instances will remain unaffected.

- **Rename importing item (numeric append)**: If an existing shape template has the same name as an imported one, the one being imported will be renamed appending an incremental number at the end of its name. The existing shape template and its derived marker instances will remain unaffected.
- **Overwrite existing item**: If an existing shape template has the same name as an imported one, the existing one will be overwritten with the one being imported. Configuration of the previously existing shape template will be lost. Derived shape instances will be updated according to the new imported shape template.

4.2. Enhancing Objects

This section describes the marker and shape properties and how to create and configure Custom Properties and Animations.

4.2.1. Marker Template Properties

The right side of the Marker Template panel contains the property controls of the marker templates. Upon the selection of a marker template, the property controls will be updated. These controls will remain disabled unless a marker template is being edited or created.

To accept any modification made during the editing or the creation of a marker template, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a marker template, user must click **Do not save and close**. Clicking this button while a new marker template is being created, will abort the process. In both cases the ongoing process will come to an end.

The available marker template properties can be classified in three groups: Main properties, Custom Properties, and Animations.

4.2.1.1. Marker Template Main Properties

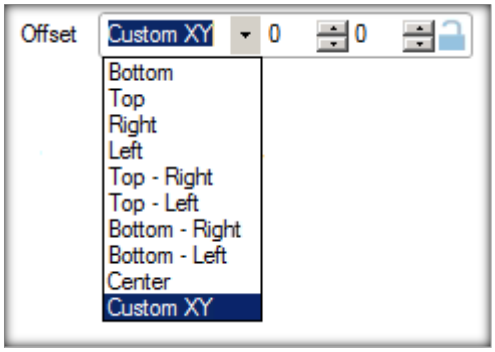
These are the common properties of the marker properties, and its controls are located in the top-right side of the panel.

Similar to ArchestrA properties, all the property controls that can be inherited by marker instances have a lock to indicate whether the control value is inherited or not. Only those that are locked will be inherited by derived marker instances.

Properties inherited at derived marker instances are protected and cannot be edited at the instances.

From top to bottom and left to right, the properties are:

- **Name**: The unique identifier name of the marker template. There cannot be 2 marker templates with the same name. This control only is enabled during the process of creating a new marker template. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Also, the name of a template will always start with the dollar symbol '$' and it will be added automatically if it is necessary. Name cannot be left empty. Derived marker instances never inherit this property.
- **ArchestrA Template**: This is the name of an existing Template in the ArchestrA Galaxy associated to the marker. It is used by the "Integrate Galaxy Objects into Repository" process to know from which Template the marker is to be updated.
- **Description**: A textbox field to type a description of the marker template. This property plays no role at the runtime, but as any other property, it is accessible by scripting for integrators. It can be left empty. Derived marker instances never inherit this property.
- **Offset**: This combo box indicates the location of the point of the coordinates of the marker with respect to the image that it represents. By default, the origin of the coordinates is situated at the bottom of the image.



If the 'Custom XY' offset is selected, it will be available to configure the custom offset in pixels (X, Y).

- **ArchestrA Graphic**: This textbox allows the association of an ArchestrA Graphic with the marker. This allows the use of the wildcard word '<Name>' that will be substituted by the marker instance name at runtime. This property can be accessed by scripting at the runtime to open an ArchestrA Graphic by name.
- **Text**: Along with an image, at the runtime a marker can be represented as a short text. This control allows the user to define this text. Default value is empty. The value of this control can be inherited by derived marker instances.
- **Text Font**: This control allows the definition of the font of the visible text (if any) of the marker. Clicking in the **Edit Text Font** button will open a font selection window.
- **Image**: Allows the definition of the image that will represent the marker in the map. Clicking the control will open an explorer window to select an image. If the location has a previously associated image, it will be shown inside the control. Default value is empty. The value of this property can be inherited by derived marker instances.
- **Text Color**: This control allows the definition of the color of the visible text (if any) of the marker. Clicking the control will open a color selection window. Canceling the color selection, the selected color will become empty, thus rendering the text invisible.
- **ToolTip**: Markers and shapes can have tooltips associated to them. These tooltips can appear at the runtime leaving the cursor over the items a configurable number of seconds. By clicking the **Edit ToolTip** button a new window will appear allowing the definition of the tooltip for the selected marker template. The textbox only shows a read only auto generated tooltip name that hints the existence of a tooltip for the marker template. By default, no tooltip is generated. Refer to chapter 5.1.17 for more information about creating and editing tooltips.
- **Rotation**: This control indicates in degrees the rotation of the marker image. By default, its 0, meaning, the image is not rotated.
- **Is Visible**: This control indicates whatever the marker will be visible at the runtime or not.

4.2.1.2. Marker Template Custom Properties

These controls allow the editing and creation of marker instance custom properties. This kind of properties behaves like any other predefined property but don't have visual representation as they are intended to be used by scripting or by the animations.

Custom properties can have fixed values or make reference to an ArchestrA object property.

A list at the bottom-center part of the panel shows the currently defined custom properties for the marker template. Note that by default a custom property named '<Name>' always exists. This special property makes reference to the name the derived marker instance's name, and cannot be deleted or edited.

Next to the list there are three buttons to operate with the custom properties. These buttons are:

- **Add**: Allows the creation of a new custom property. Upon clicking it the custom property editor window will appear. Refer to chapter 5.1.15 for more information about creating and editing custom properties.
- **Edit**: Allows the editing of the selected custom property of the list. Note that the custom property <Name> it's not editable. Upon clicking it the custom property editor window will appear. Refer to chapter 5.1.15 for more information about creating and editing custom properties.
- **Delete**: Allows deleting the selected custom property of the list. User will be asked for confirmation. This operation cannot be undone. Note that the custom property <Name> cannot be deleted.

4.2.1.3. Marker Template Animations

These controls allow the editing and creation of marker instance animations. Animations allow the modification of the marker properties at the runtime.

Next to the list there are three buttons to operate with the custom properties. These buttons are:

- **Add**: Allows the creation of a new animation. Upon clicking it the animation editor window will appear. Refer to chapter 5.1.16 for more information about creating and editing animations.
- **Edit**: Allows the editing of the selected animation of the list. Upon clicking it animation editor window will appear. Refer to chapter 5.1.15 for more information about creating and editing animations.
- **Delete**: Allows deleting the animation property of the list. User will be asked for confirmation. This operation cannot be undone.

4.2.2. Shape Template Properties

The right side of the Shape Template panel contains the property controls of the shape templates. Upon the selection of a shape template, the property controls will be updated. These controls will remain disabled unless a shape template is being edited or created.

To accept any modification made during the editing or the creation of a shape template, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a shape template, user must click the **Do not save and close** button. Clicking this button while a new shape template is being created will abort the process. In both cases the ongoing process will come to an end.

The available shape template properties can be classified in three groups: Main properties, Custom Properties, and Animations.

4.2.2.1. Shape Templates Main Properties

These are the common properties of the shape properties, and their controls are located in the top-right side of the panel.

Similar to ArchestrA properties, all the property controls that can be inherited by shape instances have a lock to indicate whether the control value is inherited or not. Only those whose locks are closed will he inherited by derived shape instances.

Properties inherited by the derived shape instances are protected and cannot be edited at the instances.

From top to bottom and left to right, the properties are:

- **Name**: The unique identifier name of the shape template. There cannot be 2 shape templates with the same name. This control is only enabled during the process of creating a new shape template. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Also, the name of a template will always start with the dollar symbol '$' and it will be added automatically if it is necessary. Name cannot be left empty. The derived shape instances never inherit this property.
- **ArchestrA Template**: This is the name of an existing Template in the ArchestrA Galaxy associated with the shape. It is used by the integration update wizard process to identify from which Template the shape must be updated.
- **Description**: A textbox field to type a description of the shape template. This property plays no role at the runtime, but like any other property, it is accessible by scripting for integrators. It can be left empty. The derived shape instances never inherit this property.
- **Text**: Along with a polygon or route, at the runtime a shape can be represented by a short text. This control allows the definition of that text. Default value is empty. The value of this control can be inherited by derived shape instances.
- **Text Font**: This control allows the definition of the font of the visible text (if any) of the shape. Clicking the control will open a font selection window.
- **Texture**: This control allows the definition of the image that will represent the shape texture on the map. This parameter only will have effect if the shape is a polygon. Clicking the control will open an explorer window to select an image. If

the property has a previously associated image, it will be shown inside the control. Default value is empty. The value of this control can be inherited by derived shape instances.

- **Text Color**: This control allows the definition of the color of the visible text (if any) of the shape. Clicking the control will open a color selection window. Canceling the color selection, the selected color will become empty, thus rendering the text invisible.
- **ArchestrA Graphic**: This textbox allows the association of an ArchestrA Graphic with the shape. This allows the use of the wildcard word '<Name>' that will be substituted by the shape instance name at the runtime. This property can be accessed by scripting at the runtime to open an ArchestrA Graphic by name.
- **ToolTip**: Markers and shapes can have tooltips associated to them. These tooltips can appear at the runtime leaving the cursor over the items a configurable number of seconds. Clicking the **Edit ToolTip** button will open a new window allowing the definition of the tooltip for the selected shape template. The textbox only shows a read only auto generated tooltip name that hints the existence of a tooltip for the marker template. By default, no tooltip is generated. Refer to chapter 5.1.17 for more information about creating and editing tooltips.
- **Fill Color**: This control allows the definition of the fill color of the shape. This parameter only will have an effect if the shape is a polygon. Clicking the control will open a color selection window. If the color selection is cancelled, the selected color will become empty.
- **Line Color**: This control allows the definition of the line color of the shape if it's a polyline or the border color if the shape is a polygon. Clicking the control will open a color selection window. If the color selection is cancelled, the selected color will become empty.
- **Line Width**: This control allows the definition of the line/border width of the shape. It represents the width in pixels that will appear at the top visible zoom. Line width will dynamically change at the runtime with the zoom. A width less than 1 will represented at the runtime with a width of 1.
- **Borderless**: This check box is to decide, if the polygon shape will have border or not.
- **Type**: This control allows the definition of the type of the shape. It can be a polyline or a polygon. The first one will represent the shape as an open route while the second will represent the shape as a close polygon.
- **Opacity (%)**: This control allows the definition of the opacity of the shape. A value of 100 will render the shape completely opaque and a value of 0 will render the shape invisible.
- **Is Visible**: This control indicates whatever the shape will be visible at the runtime or not.

4.2.2.2. Shape Templates Custom Properties

These controls allow the editing and creation of shape instance custom properties. This kind of properties behaves like any other predefined property but doesn't have visual representation as they are intended to be used by scripting or by animations.

Custom properties can have fixed values or make reference to an ArchestrA object property.

A list at the bottom-center part of the panel shows the currently defined custom properties for the shape template. Note that by default a custom property named '<Name>' always exists. This special property makes reference to the name of the derived shape instance's name, and cannot be deleted or edited.

Next to the list there are three buttons to operate with the custom properties. These buttons are:

- **Add**: This control allows the creation of a new custom property. Upon clicking it the custom property editor window will appear. Refer to chapter 5.1.15 for more information about creating and editing custom properties.
- **Edit**: This control allows the editing of the selected custom property of the list. Note that the custom property <Name> it's not editable. Upon clicking it the custom property editor window will appear. Refer to chapter 5.1.15 for more information about creating and editing custom properties.
- **Delete**: This control allows deleting the selected custom property of the list. User will be asked for confirmation. This operation cannot be undone. Note that the custom property <Name> cannot be deleted.

4.2.2.3. Shape Templates Animations

These controls allow the editing and creation of shape instances animations. Animations allow the modification of the shape properties values at runtime.

Next to the list there are three buttons to operate with the custom properties. These buttons are:

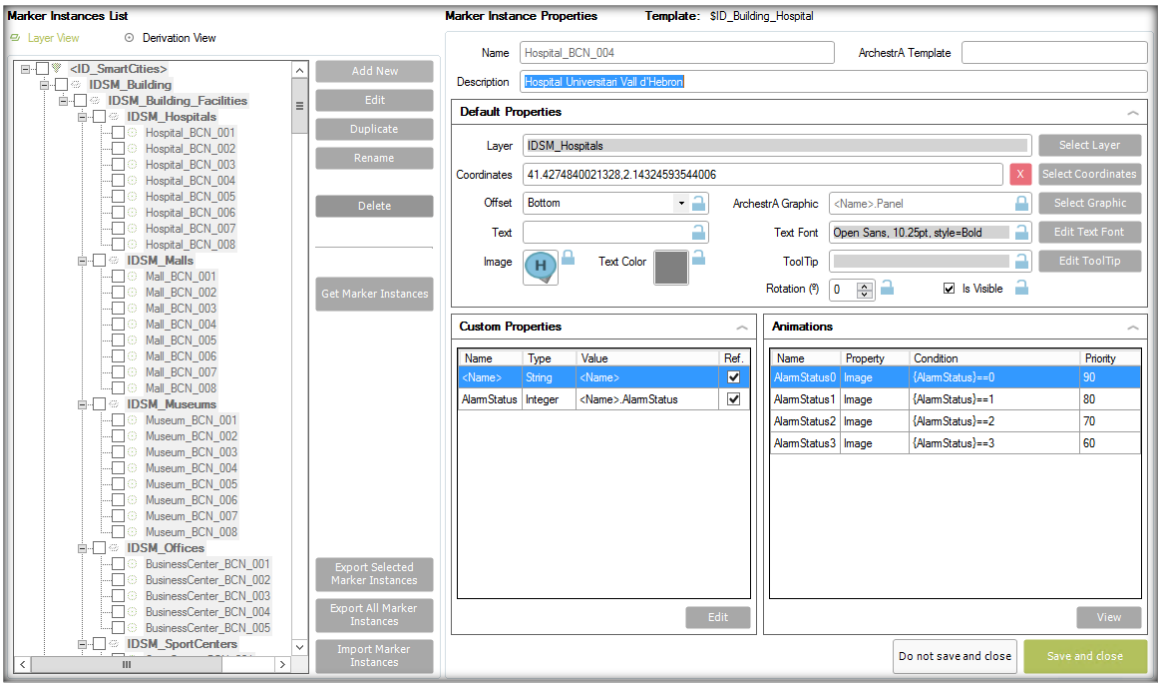
- **Add**: This control allows the creation of a new animation. Upon clicking it the animation editor window will appear. Refer to chapter 5.1.16 for more information about creating and editing animations.
- **Edit**: This control allows the editing of the selected animation of the list. Upon clicking it animation editor window will appear. Refer to chapter 5.1.15 for more information about creating and editing animations.
- **Delete**: This control allows deleting the selected animation of the list. User will be asked for confirmation. This operation cannot be undone.

4.2.3. Marker Instances

The Marker Instance panel allows the definition of new marker instances and the configuration of the existing ones.

As its name indicates, a marker instance represents the runtime implementation of a map marker instance which is derived from a marker template and inherit its properties. Modifying a marker instance will not modify its parent marker template.

Technically a marker is a pointer at a specific coordinate on a map that is represented as an image. Marker instances are always associated to a layer where they will be physically displayed. For the same reason, marker instances will always have defined coordinates.



The panel is divided into left and right sides. On the left side of the panel there is a list tree of currently available marker instances organized by layer or template derivation, and a set of buttons to perform operations. On the right side of the panel there is a collection of controls to modify the properties of a selected marker instance.

4.2.3.1. Marker Instances Tree

The marker instances tree shows the current available marker instances ordered by the layers to which they belong, or by the templates to which they are derived. Upon the selection of a marker instance of the tree, a set of operations can be performed through the buttons next to it. From top to bottom these buttons are:

- **Add New:** This button allows creating a new marker instance. Clicking it will open a pop up window to select a marker template from which it will be derived. This selection cannot be changed after. After that the left side of the panel will be disabled and the right side where the property controls for editing are located will be enabled.

  The minimum requirement to add a new marker template is a unique name, a layer that it belongs to and the coordinates where it will be displayed.

- **Edit:** This button allows editing the properties of a previously existing marker instance. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The name of the marker instance being edited cannot be changed with this functionality.

- **Duplicate:** This button will create a copy of the selected marker instance. The resulting marker instance will be named as the original marker instance name adding the suffix '_Copy'. If a marker instance already exists with that name, an extra incremental numeric suffix will be added.

- **Rename:** This button allows changing the name of a selected marker instance. Clicking it will open a popup window to type the new desired name. Duplicated names are not allowed.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

- **Delete:** Clicking this button will delete the selected marker instance with all its properties. Before deleting a marker instance the user will be asked for confirmation. This operation cannot be undone.

- **Get Marker Instances:** Clicking this button will refresh the list of available marker instances. Usually there is no need to use this button because after performing any operation locally, the list is refreshed automatically.

- **Export Selected Marker Instances:** This button allows exporting the currently selected marker instances with all their properties to a XML file. This is a way to backup marker instances to be imported later. The file can be edited externally with any text editor.

  Note that the selection of the marker instances is done by checking the checkboxes of the tree, not by clicking over the instance name.

- **Export All Marker Instances:** This button allows exporting all the existing marker instances in a repository with all their properties into a single XML file.

  This is a way to back up all the marker instances of a repository to be imported later. The file can be edited externally with any text editor.

- **Import Marker Instances:** This button allows importing an XML file with previously exported marker instances. Upon clicking it, the user will be asked to select an XML file within an explorer window. After selecting an XML file, the user will have to choose the way to import the file if a marker template with the same name is found.

- Note that the importing of a marker instance will fail if the layer which it belongs to or the marker template that it is derived doesn't exist.

  The import preferences are:

  - **Skip – Do not import**: If an existing marker instance has the same name as an imported one, the imported one will be ignored and skipped from the importing operation. The existing marker instance will remain unaffected.
  - **Rename importing item (numeric append)**: If an existing marker instance has the same name as an imported one, the one being imported will be renamed appending an incremental number at the end of its name. The existing marker instance will remain unaffected.

- **Overwrite existing item**: If an existing marker instance has the same name as an imported one, the existing one will be overwritten with the one being imported. Configuration of the previously existing marker instance will be lost.

4.2.3.2. Marker Instance Properties

The right side of the Marker Instance panel contains the property controls of the marker instances. Upon the selection of a marker instance, the property controls will be updated. These controls will remain disabled unless a marker instance is being edited or created.

To accept any modification made during the editing or the creation of a marker instance, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a marker instance, user must click the **Do not save and close** button. Clicking this button while a new marker instance is being created, will abort the process. In both cases, the ongoing process will come to an end.

The available marker instance's properties can be classified in three groups: Main properties, Custom Properties, and Animations.

4.2.3.2.1.1. Marker Instances Main Properties

These are the common properties of the marker properties, and its controls are located at the top-right side of the panel.

Similar to ArchestrA properties, all the property controls that can be inherited by marker instances have a lock to indicate whether the control value is inherited or not. Those whose locks are closed are inherited from the template and cannot be edited.

From top to bottom and left to right, the properties are:

- **Name**: The unique identifier name of the marker instance. There cannot be 2 marker instances with the same name. This control is only enabled during the process of creating a new marker instance. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Name cannot be left empty. This property is never inherited from marker templates.
- **ArchestrA Instance**: This is the name of an existing Instance in the ArchestrA Galaxy associated with the marker. It is used by the integration update wizard process to know from witch Instance the marker must be updated.
- **Description**: A textbox field to type a description of the marker instance. This property plays no role at the runtime, but as any other property, it is accessible by scripting for integrators. It can be left empty. This property is never inherited from marker templates.
- **Layer**: Indicates the layer that the marker instance belongs to. This property cannot be empty. The marker instance will be displayed within this layer.
- **Coordinates**: Indicates the coordinates where the marker instance will be displayed. This property cannot be empty.
- **Offset**: This combo box indicates where the point of the coordinates of the marker is located with respect to the image that represents. By default, the origin of the coordinates is situated at the bottom of the image. If the 'Custom XY' offset is selected, it will be available to configure the custom offset in pixels (X, Y).

- **ArchestrA Graphic**: This textbox allows the association of an ArchestrA Graphic with the marker. This allows the use of the wildcard word '<Name>' that will be substituted by the marker instance name at the runtime. This property can be accessed by scripting at runtime to open an ArchestrA Graphic by name. The value of this control can be inherited from marker template.
- **Text**: Along with an image, at the runtime a marker can be represented with a small text. This control allows the definition of that text. Default value is empty. The value of this control can be inherited from marker template.
- **Text Font**: This control allows the selection of the font of the visible text (if any) of the marker. Clicking the control's button will open a font selection window. This property can be inherited from marker template.
- **Image**: Allows the selection of the image that will represent the marker on the map. Clicking the control will open an explorer window to select an image. If the marker has a previously associated image, it will be shown inside the control. Default value is empty. The value of this control can be inherited from marker template.
- **Text Color**: This control allows the selection of the color of the visible text (if any) of the marker. Clicking the control will open a color selection window. Canceling the color selection, the selected color will become empty, thus rendering the text invisible. The value of this control can be inherited from marker template.
- **ToolTip**: Markers and shapes can have tooltips associated to them. These tooltips can appear at the runtime leaving the cursor over the items a configurable number of seconds. Upon clicking the **Edit ToolTip** button a new window will appear allowing the definition of the tooltip for the selected marker template. The textbox only shows a read only auto generated tooltip name that hints the existence of a tooltip for the marker template. By default, no tooltip is generated. Refer to chapter 5.1.17 for more information about creating and editing tooltips.
- **Rotation**: This control indicates in degrees the rotation of the marker image. By default, it is 0, meaning, the image is not rotated. This property can be inherited from marker template.
- **Is Visible**: This control indicates whatever the marker will be visible at the runtime or not. This property can be inherited from marker template.

4.2.3.2.1.2. Marker Instances Custom Properties

These controls allow the editing and creation of custom properties for marker instances. This kind of properties behaves like any other predefined property but doesn't have visual representation as they are intended to be used by scripting or by animations.

Custom properties can have fixed values or make reference to an ArchestrA object property.

A list at the bottom-center part of the panel shows the currently defined custom properties for the marker instance. Note that by default a custom property named '<Name>' always exists. This special property makes reference to the name the derived marker instance's name, and cannot be deleted or edited.

Custom properties at the instances cannot be created or deleted as they are inherited from the template.

Next to the list, there is a button to operate with the custom properties:

- **Edit**: Allows the editing of the selected custom property of the list. Note that the custom property <Name> is not editable. Upon clicking it the custom property editor window will appear. Refer to chapter 5.1.15 for more information about creating and editing custom properties. A custom property locked by the template will not be editable at the derived instance.

4.2.3.2.1.3. Marker Instances Animations

These controls allow the visualization of marker instance animations. Animations cannot be edited from the instances.

Next to the list there is a button to operate with the custom properties:

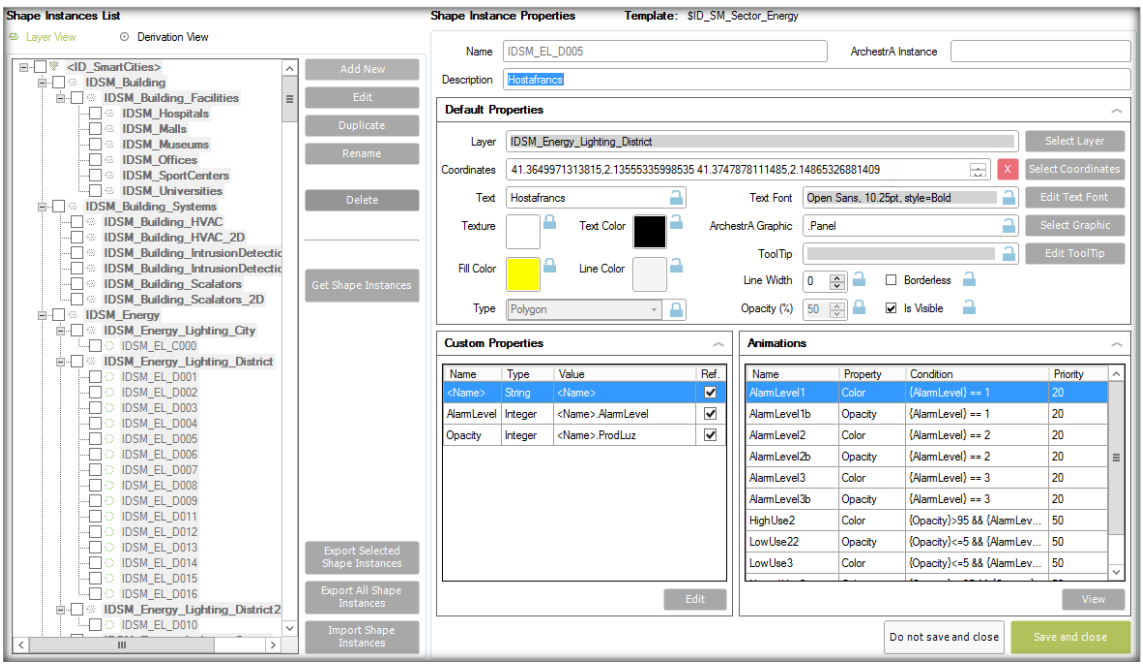
- **View**: Allows the visualization of the selected animation of the list. Upon clicking it, animation editor window will appear in view mode. Refer to chapter 5.1.15 for more information about visualizing animations.

4.2.4. Shape Instances

The Shape Instance panel allows the definition of new shape instances and the configuration of the existing ones.

As its name indicates, a shape instance represents the runtime implementation of a map shape instance (polygon or polyline) which is derived from a shape template and inherit its properties. Modifying a shape instance will not modify its parent marker template.

Technically a shape is a collection of specific coordinates on a map that are represented as a route if they are open or a polygon if they are closed. Shape instances are always linked to a layer where they will be physically displayed. For the same reason, shape instances will always have a defined collection of coordinates.



The panel is divided into left and right sides. On the left side of the panel there is a tree of currently available shape instances organized by layers or template derivation, and a set of buttons to perform operations. On the right side of the panel there is a collection of controls to modify the properties of a selected shape instance.

4.2.4.1.1.1. Shape Instances List:

The shape instances tree shows the current available shape instances ordered by the layers to which they belong to, or by the templates to which they are derived. Upon the selection of a shape instance of the tree, a set of operations can be performed through the buttons next to it. From top to bottom these buttons are:

- **Add New:** This button allows creating a new shape instance. Upon clicking it a pop up window will appear to select a shape template from which the instance will derive. This selection cannot be changed before. After that, the left side of the panel will be disabled and the right side where the property controls for editing are located will be enabled.

  The minimum requirement to add a new marker template is a unique name, a layer that it belongs to, and the coordinates where it will be displayed.

- **Edit:** This button allows editing the properties of a previously existing shape instance. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

The name of the shape instance being edited cannot be changed with this functionality.

- **Duplicate:** This button will create a copy of the selected shape instance. The resulting shape instance will be named as the original shape instance name adding the suffix '_Copy'. If a shape instance already exists with that name, an extra incremental numeric suffix will be added.

- **Rename:** This button allows changing the name of a selected shape instance. Upon clicking it a popup window will appear to type the new desired name. Duplicated names are not allowed.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

- **Delete:** Clicking this button will delete the selected shape instance with all its properties. Before deleting a shape instance the user will be asked for confirmation. This operation cannot be undone.

- **Get Shape Instances:** Upon clicking this button the list of available shape instances will be refreshed. Usually there is no need to use this button because after performing any operation locally, the list is refreshed automatically.

- **Export Selected Shape Instance:** This button allows exporting the currently selected shape instances with all its properties to a XML file. This is a way to backup shape instances to be imported later. The file can be edited externally with any text editor.

  Note that the selection of the shape instances is done by checking the checkboxes of the tree, not by clicking over the instance name.

- **Export All Shape Instances:** This button allows exporting all the existing shape instances in a repository with all their properties into a single XML file.

  This is a way to back up all the shape instances of a repository to be imported later. The file can be edited externally with any text editor.

- **Import Shape Instances:** This button allows importing an XML file with previously exported shape instances. Upon clicking it, the user will be asked to select an XML file within an explorer window. After selecting an XML file, the user will have to choose the way to import the file if a shape template with the same name is found.

  Note that the importing of a shape instance will fail if the layer that it belongs or the shape template that it is derived from doesn't exist.

  The import preferences are:

  - **Skip – Do not import**: If an existing shape instance has the same name as an imported one, the imported one will be ignored and skipped from the importing operation. The existing shape instance will remain unaffected.
  - **Rename importing item (numeric append)**: If an existing shape instance has the same name as an imported one, the one being imported will be renamed appending an incremental number at the end of its name. The existing shape instance will remain unaffected.
  - **Overwrite existing item**: If an existing shape instance has the same name as an imported one, the existing one will be overwritten with the one being imported. Configuration of the previously existing shape instance will be lost.

4.2.4.1.1.2. Shape Instance Properties

The right side of the Shape Instance panel contains the property controls of the shape instances. Upon the selection of a shape instance, the property controls will be updated. These controls will remain disabled unless a shape instance is being edited or created.

To accept any modification made during the editing or the creation of a shape instance, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a shape instance, user must click the **Do not save and close** button. Clicking this button while a new shape instance is being created will abort the process. In both cases the ongoing process will come to an end.

The available shape instance properties can be classified in three groups: Main properties, Custom Properties, and Animations.

4.2.4.1.1.3. Shape Instance Main Properties

These are the common properties of the shape instances, and their controls are located at the top-right side of the panel.

Similar to ArchestrA properties, all the property controls that can be inherited by shape instances have a lock to indicate whether the control value is inherited or not. Those whose locks are closed are inherited from the template and cannot be edited.

From top to bottom and left to right, the properties are:

- **Name**: The unique identifier name of the shape instance. There cannot be two shape instances with the same name. This control is only enabled during the process of creating a new shape instance. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Name cannot be left empty. This property is never inherited from shape templates.

- **ArchestrA Instance**: This is the name of an existing Instance in the ArchestrA Galaxy associated with the shape. It is used by the integration update wizard process to know from witch Instance the shape must be updated.
- **Description**: A textbox field to type a description of the shape instance. This property plays no role at the runtime, but as any other property, it is accessible by scripting for integrators. It can be left empty. This property is never inherited from shape templates.
- **Layer**: Indicates the layer which the shape instance belongs to. This property cannot be empty. The shape instance will be displayed within this layer.
- **Coordinates**: Indicates the collection of coordinates that make up the polyline or polygon that represents the shape. This property cannot be empty. Using the **Select Coordinates** button, a window that greatly simplifies the generation of these coordinates, can be opened.
- **Text**: Along with a polygon or route, at the runtime a shape can be represented by a short text. This control allows the definition of that text. Default value is empty. This property can be inherited from shape templates.
- **Text Font**: This control allows the selection of the font of the visible text (if any) of the shape. Clicking in the control's button will open a font selection window. This property can be inherited from shape templates.
- **Texture**: Allows the selection of the image that will represent the shape texture on the map. This parameter will only have an effect if the shape it's a polygon. Clicking the control will open an explorer window to select an image. If the property has a previously associated image, it will be shown inside the control. Default value is empty. This property can be inherited from shape templates.
- **Text Color**: This control allows the selection of the color of the visible text (if any) of the shape. Clicking the control will open a color selection window. If the color selection is canceled, the selected color will become empty, thus rendering the text invisible. This property can be inherited from shape templates.
- **ArchestrA Graphic**: This textbox allows the association of an ArchestrA Graphic with the shape. This allows the use of the wildcard word '<Name>' that will be substituted by the shape instance's name at the runtime. This property can be accessed by scripting at the runtime to open an ArchestrA Graphic by name. This property can be inherited from shape templates.
- **ToolTip**: Markers and shapes can have tooltips associated to them. These tooltips can appear at the runtime leaving the cursor over the items a configurable number of seconds. Upon clicking the **Edit ToolTip** button a new window will appear allowing the selection of the tooltip for the selected shape template. The textbox only shows a read only auto generated tooltip name that hints the existence of a tooltip for the marker template. By default, no tooltip is generated. Refer to chapter 5.1.17 for more information about creating and editing tooltips.
- **Fill Color**: This control allows the selection of the fill color of the shape. This parameter will only have an effect if the shape is a polygon. Clicking the control will open a color selection window. Canceling the color selection, the selected color will become empty. This property can be inherited from shape templates.
- **Line Color**: This control allows the definition of the line color of the shape if it's a polyline or the border color of the shape if it's a polygon. Clicking in the control will

open a color selection window. If the color selection is canceled, the selected color will become empty. This property can be inherited from shape templates.

- **Line Width**: This control allows the selection of the line/border width of the shape. It represents the width in pixels that will appear at the top visible zoom level. Line width will dynamically change at the runtime with the zoom. A width less than 1 will represented at the runtime with a width of 1. This property can be inherited from shape templates.
- **Borderless**: This check box is to define, if the polygon shape will have a border or not. This property can be inherited from shape templates.
- **Type**: This control allows the selection of the type of the shape. It can be a polyline or a polygon. The first one will represent the shape as an open route meanwhile the second will represent de shape as a close polygon. This property can be inherited from shape templates.
- **Opacity (%)**: This control allows the selection of the opacity of the shape. A value of 100 will render the shape completely opaque and a value of 0 will render the shape invisible. This property can be inherited from shape templates.
- **Is Visible**: This control indicates whatever the shape will be visible at the runtime or not. This property can be inherited from shape template.

4.2.4.1.1.4. Shape Instance Custom Properties

These controls allow the editing of and creation of custom properties for shape instances. This kind of properties behaves like any other predefined property but doesn't have visual representation as they are intended to be used by scripting or by animations.

Custom properties can have fixed values or make reference to an ArchestrA object property.

A list at the bottom-center part of the panel shows the currently defined custom properties for the shape instance. Note that by default a custom property named '<Name>' always exists. This special property makes reference to the name the derived shape instance's name, and cannot be deleted or edited.

Custom properties at the instances cannot be created or deleted as they are inherited from the template.

Next to the list, there is a button to operate with the custom properties:

- **Edit**: Allows the editing of the selected custom property of the list. Note that the custom property <Name> is not editable. Upon clicking it the custom property editor window will appear. Refer to chapter 5.1.15 for more information about creating and editing custom properties. A custom property locked by the template will not be editable at the derived instance.

4.2.4.1.1.5. Shape Instance Animations

These controls allow the visualization of shape instance animations. Animations cannot be edited from the instances.

Next to the list, there is a button to operate with the custom properties:

- **View**: Allows the visualization of the selected animation of the list. Upon clicking it, animation editor window will appear in view mode. Refer to chapter 5.1.15 for more information about visualizing animations.

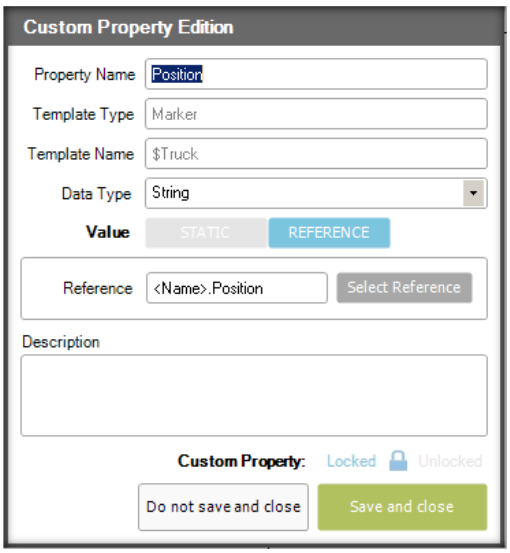
4.2.5. Custom Properties Editing

This popup window allows visualizing and editing the custom properties of templates and instances.

Custom properties behave like any other predefined property but don't have visual representation as they are intended to be used by scripting or by animations. They can have fixed values or make reference to an ArchestrA object property.

To accept any modification made during the editing or the creation of a custom property, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a custom property, user must click the **Do not save and close** button. Clicking this button while a new custom property is being created will abort the process. In both cases, the ongoing process will come to an end.



This window contains the property controls necessary to define a custom property. From top to bottom these controls are:

- **Property Name**: The unique identifier name of the custom property. There cannot be two custom properties with the same name in the same template/instance. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Name cannot be left empty.
- **Template Type**: Read only field. Indicates the custom property belongs to a marker, a shape, or a tooltip.
- **Template Name**: Read only field. Indicates the name of the template to which the custom property belongs. If the custom property belongs to an instance, its parent template name will be shown.
- **Data type**: indicates the type of the data the custom property contains. Allowed types are:
  - Boolean
  - Integer
  - Decimal

    - String
- **Value**: These buttons indicates if the value of the custom property will be a static predefined one or a reference to an ArchestrA object's attribute. If the first option is chosen, a default value can be indicated. If the second option is chosen, the full name of the ArchestrA object's attribute must be indicated. In the second case, the integrated ArchestrA Galaxy Browser can be used to get the reference name.
- **Description**: A textbox field to type a description of the custom property. This property plays no role at the runtime, but as any other property, it is accessible by scripting for integrators. It can be left empty.
- **Custom Property Lock**: Protect/unprotect the custom property. This property is only editable at the template's custom properties.

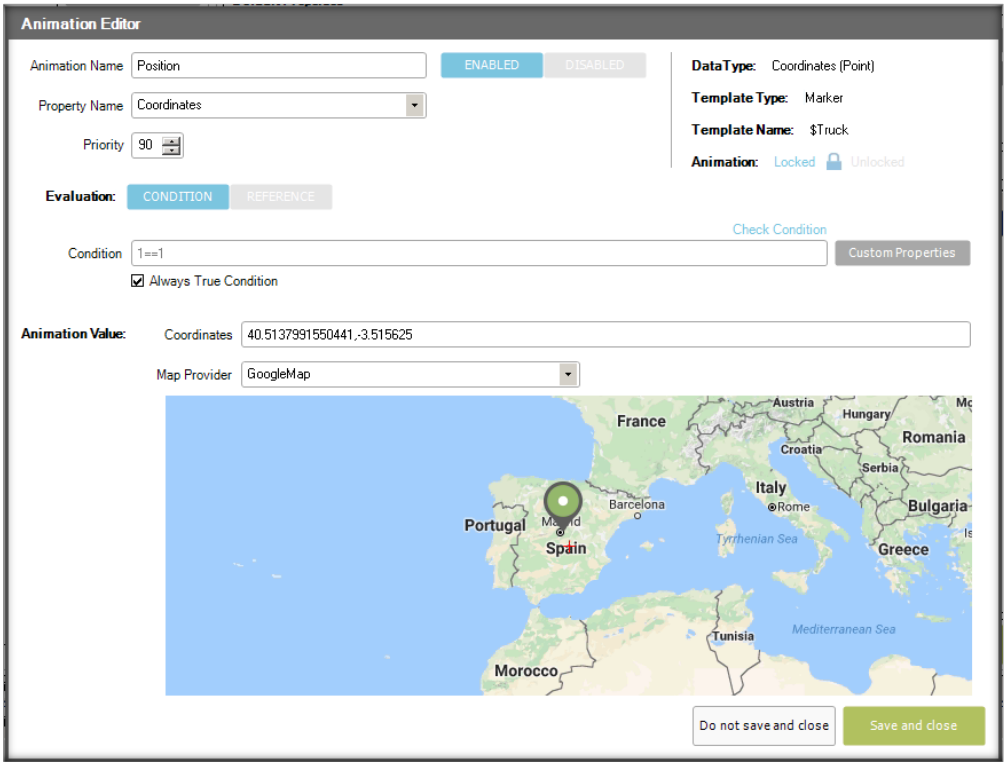
4.2.6. Animations

This popup window allows visualizing and editing the animations of templates and instances.

Animations allow the modification of the instance property values at the runtime. The definition of an animation includes a condition to evaluate and value to apply to a property if the condition is met.

To accept any modification made during the editing or the creation of an animation, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of an animation, user must click the **Do not save and close** button. Clicking this button while a new animation is being created, will abort the process. In both cases, the ongoing process will come to an end.



This window contains the property controls necessary to define a custom property. From top to bottom these controls are:

- **Animation Name**: The unique identifier name of the animation. There cannot be two animations with the same name in the same template/instance. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Name cannot be left empty.
- **Data Type**: Read only field. Indicates the type of the data of the property that affects this animation.
- **Property Name**: Indicates the name of the main property of template/instance that affects this animation. Selectable properties depend on the type of template/instance being edited.
- **Template Type**: Read only field. Indicates if the animation belongs to a marker, a shape, or a tooltip.

- **Priority**: Indicates the priority of the animation versus other animations. If two or more animations apply against the same property value at same time, the animation with higher priority will be the one to apply the value. Value of 1 have the highest priority and value 99 is the lowest priority. The default values of the main properties have priority of 99. Animations minimum priority is 98.
- **Template Name**: Read only field. Indicates the name of the template to which the animation belongs. If the custom property belongs to an instance, its parent template's name will be shown.
- **Animation Lock**: Protect/unprotect the animation.
- **Evaluation**: Indicates whatever the animation value will be applied if a certain condition is met or the animation value is the value of a reference
    - **Condition**: a condition evaluation requires indicating a condition (in C# style) and an animation value to apply if the evaluation of the condition becomes true. Previously defined custom properties can be used in the condition between brackets **{ }**. The evaluation of the condition must always be a Boolean value. The animation value will depend on the property name indicated previously. It is necessary to indicate a property name to indicate an animation value.
    - **Reference**: a reference evaluation requires a reference to a previously defined custom property between brackets **{ }**. At the runtime, the selected property will take the value of the referred custom property.
    - **Event**: the animation will occur if the selected mouse event is activated. This is the only evaluation mode supported for Container animations. The available events are:
        - **Click**: Occurs when the instance is clicked.
        - **DoubleClick**: Occurs when the instance is double clicked.
        - **Enter**: Occurs when the cursor enters the instance.
        - **Leave**: Occurs when the cursor leaves the instance.
        - **Select**: Occurs when the instance is selected. Previously selected instances are automatically unselected.
        - **Toggle Click**: Occurs when a previously clicked instance is clicked again.
        - **Toggle DoubleClick**: Occurs when a previously double clicked instance is double clicked again.

4.2.7. ToolTips

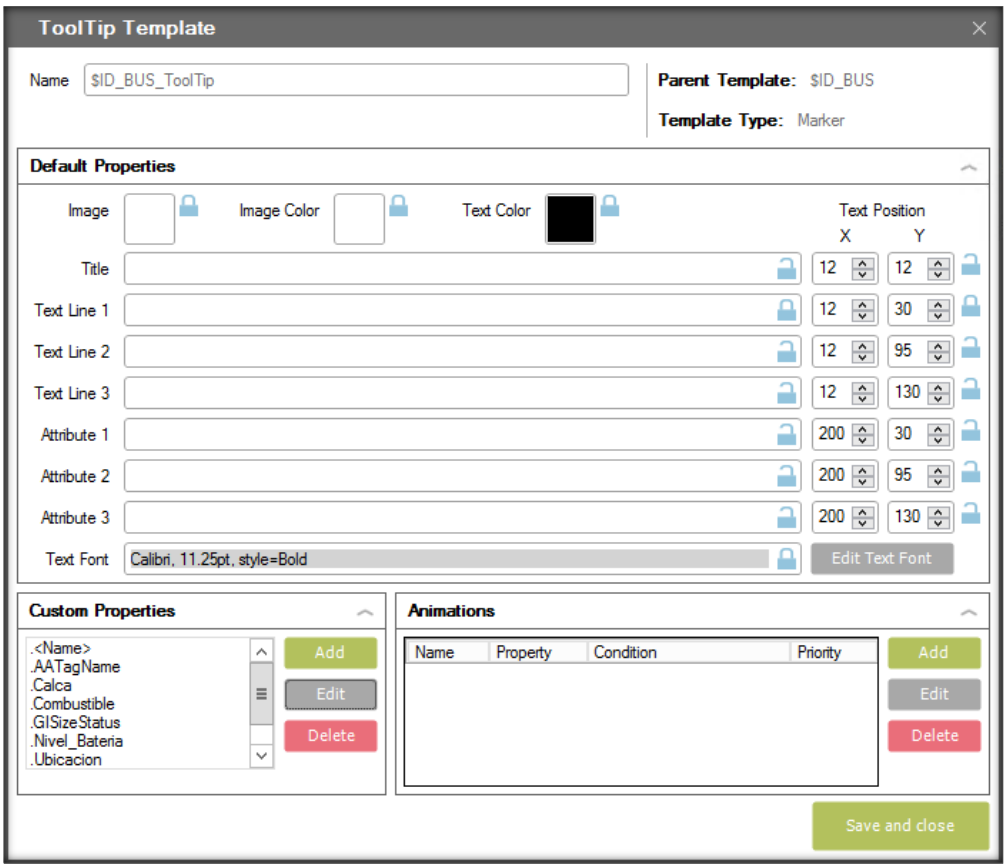
Markers and shapes can have tooltips associated to them. These tooltips can appear at the runtime leaving the cursor over the items a configurable number of seconds. This popup window allows visualizing and editing the tooltip of templates and instances.

A tooltip is visualized at the runtime with an image and up to 3 lines of text that can represent dynamic values.

Tooltips must be created in the templates.

To accept any modification made during the editing or the creation of a tooltip, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a tooltip, user must click the **X** button located at the top-right corner. Clicking this button while a new tooltip is being created will abort the process. In both cases the ongoing process will come to an end.



This window contains the property controls necessary to define a tooltip. From top to bottom these controls are:

- **Name**: Read only field. The unique identifier name of the tooltip. Auto generated.
- **Parent Template**: Available only at template tooltips. Read only field. Indicates the name of the template to which the tooltip belongs.
- **ToolTip Template**: Available only at instance tooltips. Read only field. Indicates the name of the template of the instance to which the tooltip belongs.
- **Layer**: Available only at instance tooltips. Read only field. Indicates the layer where the tooltip will be visualized. It's always the same layer of the instance to which the tooltip belongs.

- **Parent Instance**: Available only at instance tooltips. Read only field. Indicates the name of the instance to which the tooltip belongs.
- **Template Type**: Available only at template tooltips. Read only field. Indicates the type of the template (shape or marker) to which the tooltip belongs.
- **Instance Type**: Available only at instance tooltips. Read only field. Indicates the type of the instance (shape or marker) to which the tooltip belongs.
- **Image**: Allows the selection of the image that will represent the background of the tooltip. Size of the tooltip is defined by the size of the image. Clicking the control will open an explorer window to select an image. If the tooltip has a previously associated image, it will be shown inside the control. Default value is empty. The value of this control can be inherited.
- **Image Color**: This control allows the selection of the back color of the tooltip. The color will be applied over the background image of the tooltip. Clicking the control will open a color selection window. If the color selection is cancelled, the selected color will become empty. The value of this control can be inherited.
- **Text Color**: This control allows the selection of the color of the text in the tooltip. Clicking the control will open a color selection window. If the color selection is cancelled, the selected color will become empty. The value of this control can be inherited.
- **Title**: Text to be visualized at the top-left side of the tooltip. It is two points bigger than the rest of the texts. This property can be inherited.
- **Text Line 1**: Text to be visualized on the left side of the first line of the tooltip. This property can be inherited.
- **Text Line 2**: Text to be visualized on the left side of the second line of the tooltip. This property can be inherited.
- **Text Line 3**: Text to be visualized on the left side of the third line of the tooltip. This property can be inherited.
- **Attribute 1**: Text to be visualized on the right side of the first line of the tooltip. This property can be inherited.
- **Attribute 2**: Text to be visualized on the right side of the second line of the tooltip. This property can be inherited.
- **Attribute 3**: Text to be visualized on the right side of the third line of the tooltip. This property can be inherited.
- **Text Positions**: The position (in pixels) of the title, the text lines, and the attributes can be configured independently with their own controls. These properties can be inherited.
- **Text Font**: This control allows the selection of the font of the visible text (if any) of the tooltip. Clicking the control will open a font selection window. This property can be inherited.
- **Custom Properties**: In the same way to marker and shapes, tooltips can have their own custom properties. They are operated in the same way. The only difference is that tooltips automatically inherit the custom properties of their parent template/shape. These special custom properties cannot be edited or deleted and have a dot (.) as prefix in their names.

- **Animations**: In the same way to marker and shapes, tooltips can have their own animations. They are operated in the same way.

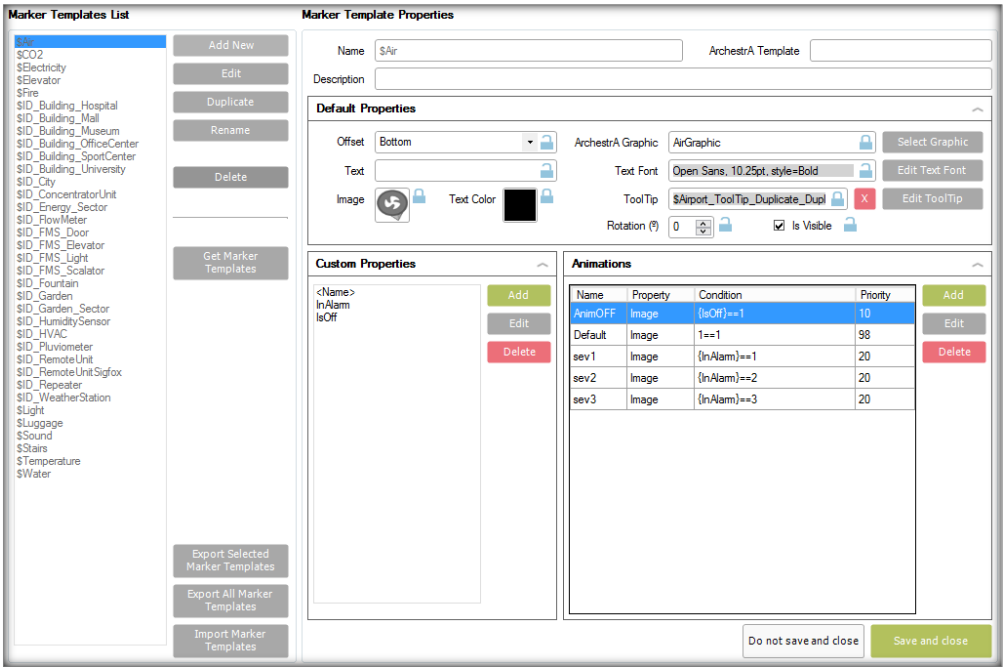
4.2.8. Containers

The Containers panel allows the definition of containers and the configuration of the existing ones.

A container is the definition of a group of marker and/or shape instances. As a group, these instances share visibility, and can also share Event type animations.

Containers are not associated to any layer or map provider. That means the instances can belong to different layers.

At the runtime, the visibility of the containers can be controlled in a similar way to visibility of the layers. The visibility of the containers and the visibility of layers can be used as a matrix of visibility of elements.



The panel is divided into left and right sides. On the left side of the panel there is a list of currently available containers and a set of buttons to perform operations. On the right side of the panel there is a collection of controls to modify the properties of a selected container.

4.2.8.1. Containers List

The marker template list shows the current available marker templates sorted alphabetically. Note that all the template names start with the dollar symbol '$', similar to ArchestrA nomenclature. Upon the selection of a marker template of the list, a set of operations can be performed through the buttons next to it. From top to bottom these buttons are:

- **Add New:** This button allows creating a new marker template. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The minimum requirement to add a new marker template is a unique name.

- **Edit:** This button allows editing the properties of a previously existing marker template. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The name of the marker template being edited cannot be changed with this functionality.

- **Duplicate:** This button will create a copy of the selected marker template. The resulting marker template will be named as the original marker template name adding the suffix '_Copy'. If a marker template already exists with that name, an extra incremental numeric suffix will be added.

- **Rename:** This button allows changing the name of a selected marker template. Clicking it will open a popup window to type the new desired name. Duplicated names are not allowed.

  If the maker template has any derived marker instances, these will be automatically updated to make reference to the newly named marker template.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

- **Delete:** Clicking this button will delete the selected marker template with all its properties. Before deleting a marker template, the user will be asked for confirmation. This operation cannot be undone.

  Be aware that deleting a marker template will also delete all its derived markers instances, thus leading to a possible undesired information loss.

- **Get Containers:** Clicking this button will refresh the list of available marker templates. Usually there is no need to use this button because after performing any operation locally, the list is refreshed automatically.

- **Export Selected Containers:** This button allows exporting the currently selected marker templates with all their properties to a XML file. This is a way to back up marker templates to be imported later. The file can be edited externally with any text editor.

- **Export All Containers:** This button allows exporting all the existing marker templates in a repository with all their properties into a single XML file.

  This is a way to back up all the marker templates of a repository to be imported later. The file can be edited externally with any text editor.

- **Import Containers:** This button allows importing an XML file with previously exported marker templates. Upon clicking it, the user will be asked to select an XML file within an explorer window. After selecting an XML file, the user will have to choose the way to import the file if a marker template with the same name is found.

  The import preferences are:

  - **Skip – Do not import**: If an existing marker template has the same name as an imported one, the imported one will be ignored and skipped from the importing operation. The existing marker template and its derived marker instances will remain unaffected.
  - **Rename importing item (numeric append)**: If an existing marker template has the same name as an imported one, the one being imported will be renamed appending an incremental number at the end of its name. The

existing marker template and its derived marker instances will remain unaffected.

- **Overwrite existing item**: If an existing marker template has the same name as an imported one, the existing one will be overwritten with the one being imported. Configuration of the previously existing marker template will be lost. Derived marker instances will be updated accordingly with the new imported marker template.

4.2.8.2. Container Properties

The right side of the Marker Template panel contains the property controls of the marker templates. Upon the selection of a marker template, the property controls will be updated. These controls will remain disabled unless a marker template is being edited or created.

To accept any modification made during the editing or the creation of a marker template, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a marker template, user must click the **Do not save and close** button. Clicking this button while a new marker template is being created will abort the process. In both cases the ongoing process will come to an end.

The available marker template properties can be classified in three groups: Main properties, Custom Properties, and Animations.

4.2.8.2.1.1. Container Main Properties

These are the common properties of the marker properties, and their controls are located at the top-right side of the panel.

Similar to ArchestrA properties, all the property controls that can be inherited by marker instances have a lock to indicate whether the control value is inherited or not. Only those whose locks are closed will be inherited by derived marker instances.

Properties inherited by derived marker instances are protected and cannot be edited at the instances.

From top to bottom and left to right, the properties are:

- **Name**: The unique identifier name of the marker template. There cannot be two marker templates with the same name. This control is only enabled during the process of creating a new marker template. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Also, the name of a template will always start with the dollar symbol '$' and will be added automatically if it is necessary. Name cannot be left empty. This property is never inherited by derived marker instances.
- **ArchestrA Template**: This is the name of an existing Template in the ArchestrA Galaxy associated to the marker. It is used by the integration update wizard process to know from which Template the marker must be updated.
- **Description**: A textbox field to type a description of the marker template. This property plays no role at the runtime, but as any other property, it is accessible by scripting for integrators. It can be left empty. This property is never inherited by derived marker instances.

- **Offset**: This combo box indicates where the point of the coordinates of the marker is located with respect to the image that represents. By default, the origin of the coordinates is situated at the bottom of the image.
- If the 'Custom XY' offset is selected, it will be available to configure the custom offset in pixels (X, Y).
- **ArchestrA Graphic**: This textbox allows the association of an ArchestrA Graphic with the marker. This allows the use of the wildcard word '<Name>' that will be substituted by the marker instance name at the runtime. This property can be accessed by scripting at the runtime to open an ArchestrA Graphic by name.
- **Text**: Along with an image, at the runtime a marker can be represented by a short text. This control allows the definition of that text. Default value is empty. The value of this control can be inherited by derived marker instances.
- **Text Font**: This control allows the selection of the font of the visible text (if any) of the marker. Clicking the control button will open a font selection window.
- **Image**: This control allows the selection of the image that will represent the marker on the map. Clicking the control will open an explorer window to select an image. If the location has a previously associated image, it will be shown inside the control. Default value is empty. The value of this control can be inherited by derived marker instances.
- **Text Color**: This control allows the selection of the color of the visible text (if any) of the marker. Clicking the control will open a color selection window. If the color selection is cancelled, the selected color will become empty, thus rendering the text invisible.
- **ToolTip**: Markers and shapes can have tooltips associated to them. These tooltips can appear at the runtime leaving the cursor over the items a configurable number of seconds. Clicking the **Edit ToolTip** button will open a new window allowing the definition of the tooltip for the selected marker template. The textbox only shows a read only auto generated tooltip name that hints the existence of a tooltip for the marker template. By default, no tooltip is generated. Refer to chapter 5.1.17 for more information about creating and editing tooltips.
- **Rotation**: This control indicates in degrees the rotation of the marker image. By default, its 0, meaning the image is not rotated.
- **Is Visible**: This control indicates whatever the marker will be visible at the runtime or not.

4.2.8.2.1.2. Containers Instances List

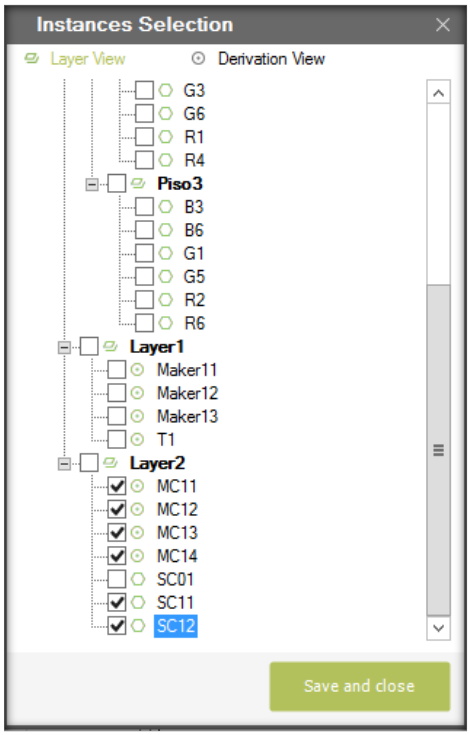
These controls allow the editing and creation of custom properties for marker instances. This kind of properties behaves like any other predefined property but doesn't have visual representation as they are intended to be used by scripting or by animations.

Custom properties can have fixed values or make reference to an ArchestrA object property.

A list at the bottom-center part of the panel shows the currently defined custom properties for the marker template. Note that by default a custom property named '<Name>' always exists. This special property makes reference to the derived marker instance's name, and cannot be deleted or edited.

Next to the list there are three buttons to operate with the custom properties. These buttons are:

- **Add**: Allows the creation of a new custom property. Upon clicking it the custom property editor window will appear. Refer to chapter 5.1.15 for more information about creating and editing custom properties.
- **Edit**: Allows the editing of the selected custom property of the list. Note that the custom property <Name> is not editable. Upon clicking it the custom property editor window will appear. Refer to chapter 5.1.15 for more information about creating and editing custom properties.
- **Delete**: Allows deleting the selected custom property of the list. User will be asked for confirmation. This operation cannot be undone. Note that the custom property <Name> cannot be deleted.



Animations

These controls allow the editing and creation of container animations. Animations allow the modification of the container's instance properties at the runtime.

Container animations only support Event type evaluations.

Next to the list, there are three buttons to operate with the custom properties. These buttons are:

- **Add**: Allows the creation of a new animation. Upon clicking it the animation editor window will appear. Refer to chapter 5.1.16 for more information about creating and editing animations.
- **Edit**: Allows the editing of the selected animation of the list. Upon clicking it animation editor window will appear. Refer to chapter 5.1.15 for more information about creating and editing animations.

- **Delete**: Allows deleting the animation property of the list. User will be asked for confirmation. This operation cannot be undone.

4.3. Lab 11 - Modeling a Marker

4.3.1. Introduction

In this lab, you will model a marker using Custom Properties and Animations.

4.3.2. Objectives

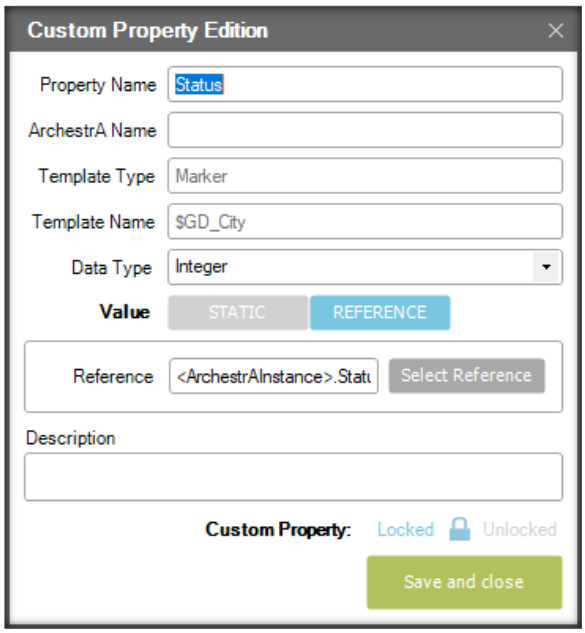
Upon completion of this lab, you will be able to:

- Create and configure markers.
- Create Custom Properties and link them with Application Server objects.
- Create Animations to modify the behavior of objects.

4.3.3. Create Custom Properties

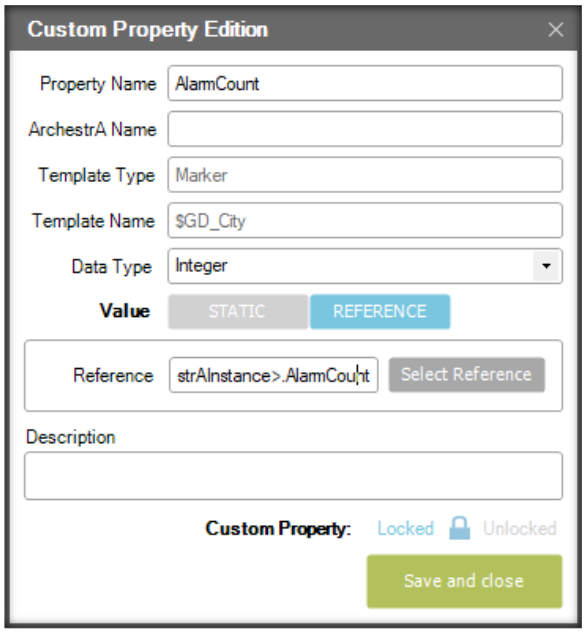
In the following steps, you will create custom properties.

1. Double click the **Marker Template** option.
2. Double click **$GD_City** template to edit it.
3. Expand **Custom Properties** section and click the **Add** button to create a new custom property.
4. Enter **Status** in the field **Property Name**.
5. Select **Integer** in the combo box **Data Type**.
6. Click the **Reference** button.
7. Enter **<ArchestrAInstance>.Status** in the field **Reference**.
8. Click the **padlock icon** to lock this custom property.
9. Click the **Save and close** button.



10. Click the **Add** button to create a new custom property.
11. Enter **AlarmCount** in the field **Property Name**.
12. Select **Integer** in the combo box **Data Type**.

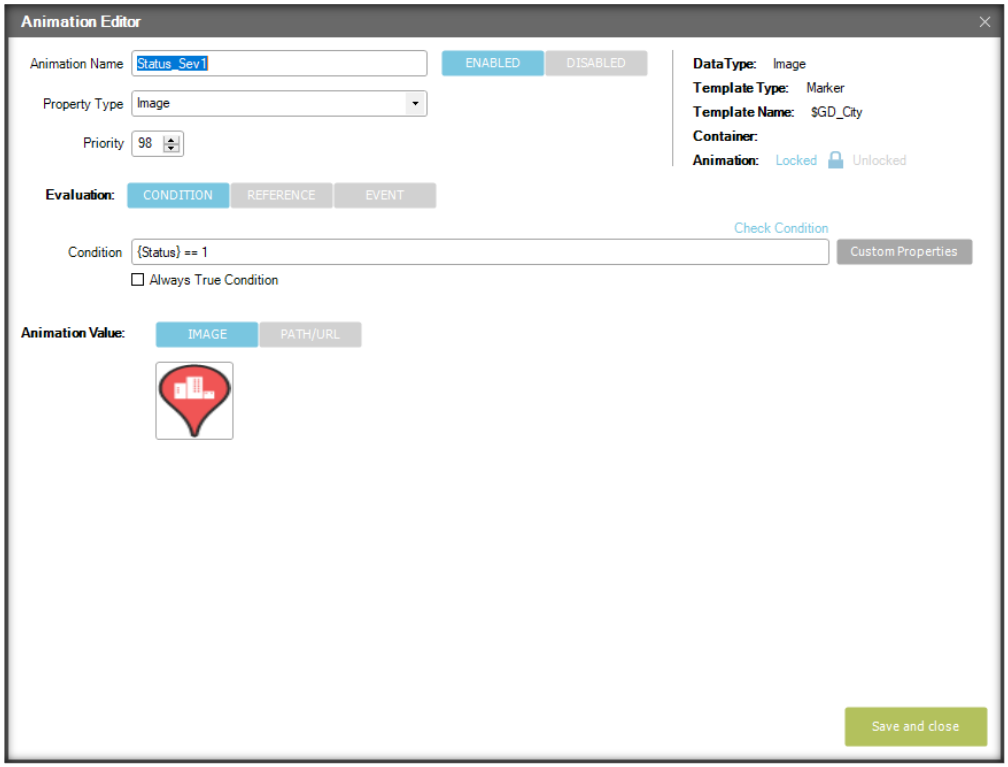
13. Click the **Reference** button.
14. Enter **<ArchestrAInstance>.AlarmCount** in the field **Reference**.
15. Click the **padlock icon** to lock this custom property.
16. Click the **Save and close** button.



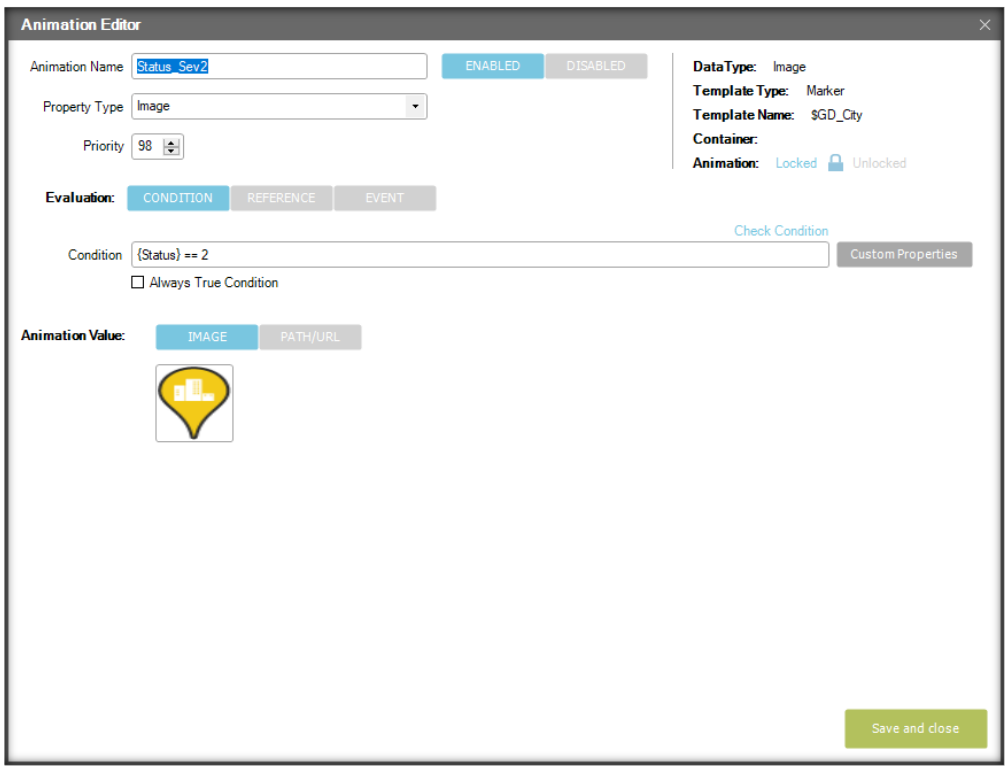
4.3.4. Create Animations

In the following steps, you will create animations.

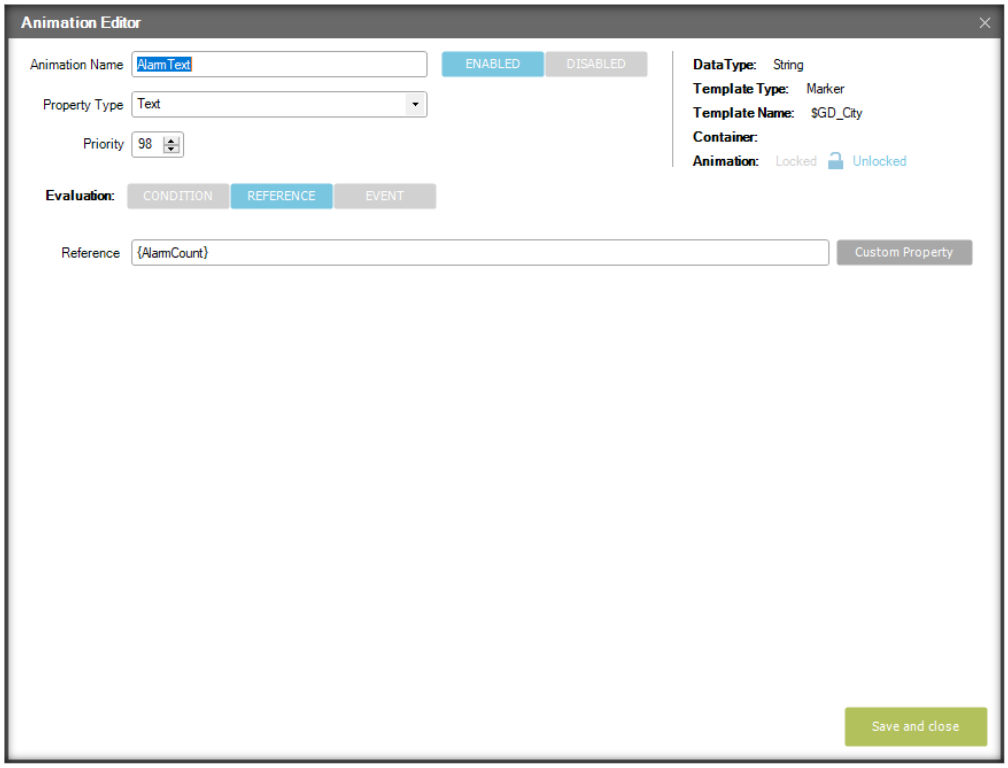
1. Expand **Animations** section and click the **Add** button to create a new animation.
2. Enter **Status_Sev1** in the field **Animation Name**.
3. Select **Image** in the combo box **Property Type**.
4. Click the **Condition** button.
5. Disable **Always True Condition** checkbox.
6. Delete the content of the **Condition** Field. Click the **Custom Properties** button and select **Status**. Add **==1** to the field in order to have **{Status}==1.**
7. Double click the **Image** property and select the file **Miscellaneous\Markers\City_Sev1.png** located in the training files.
8. Click the **padlock icon** to lock this animation.
9. Click the **Save and close** button.



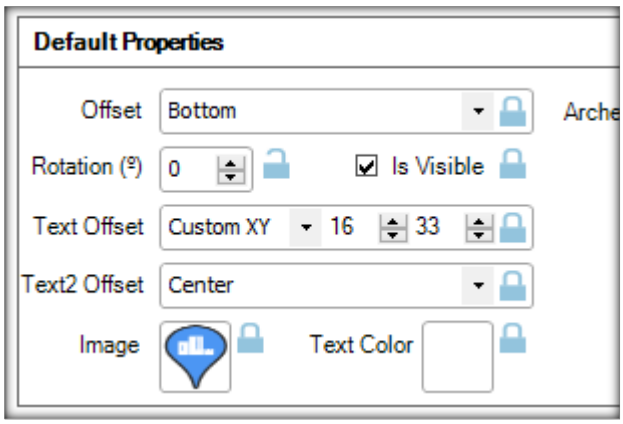
10. Create a new Animation to change the image when the Severity is 2 **({Status==2)**



11. Create a new Animation to represent a text linked to the **AlarmCount** property.



12. Configure the **Text Offset** property with **Custom XY** and **16x33** values.

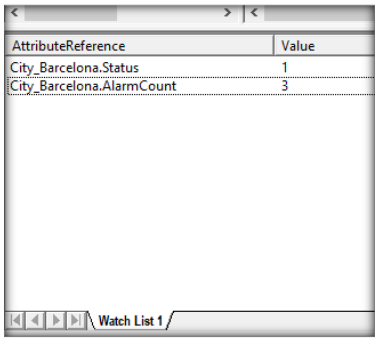


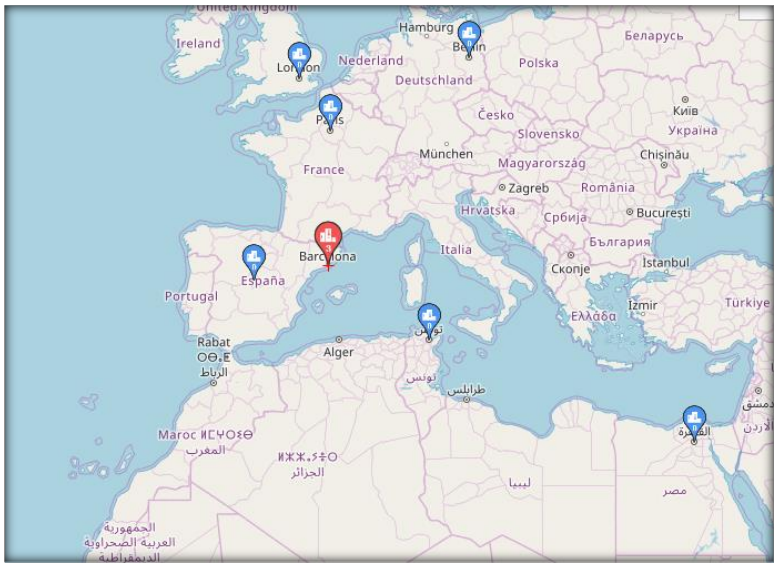
13. Click the **Save and Close** button in the marker template to save the template and update the instances with the new configuration

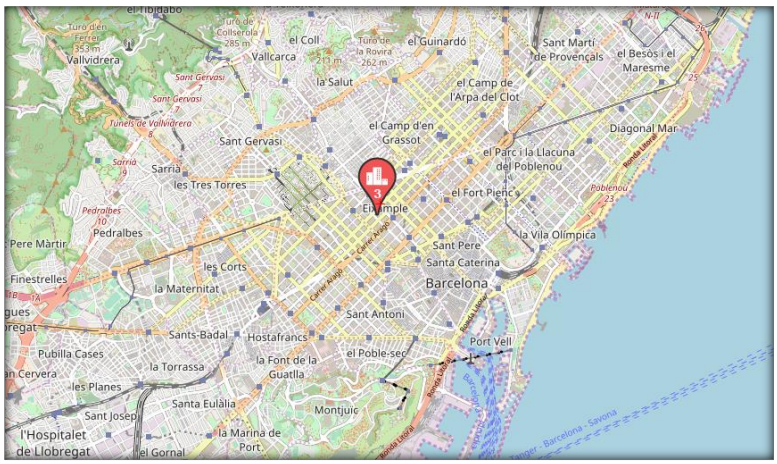
4.3.5. Test at the Runtime

In the following steps, you will test the marker instances animations at the runtime.

1. Double Click the **Runtime Preview** button to open the **GISIZE GEOSpatial Runtime Preview** tool.
2. Open the **Object Viewer** and modify the **Status** and **AlarmCount** attributes to see the behavior of the marker instances.







4.4. Lab 12 - Modeling a Shape

4.4.1. Introduction

In this lab, you will import a collection of shapes already created as an elaborate example.

4.4.2. Objectives

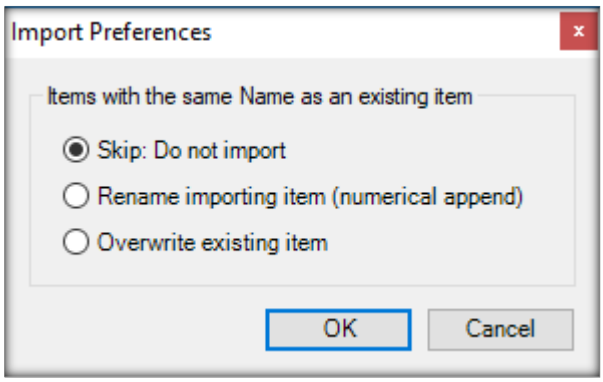
Upon completion of this lab, you will be able to:

- Create and configure shapes.
- Create Custom Properties and link them with Application Server objects.
- Create Animations to modify the behavior of objects.

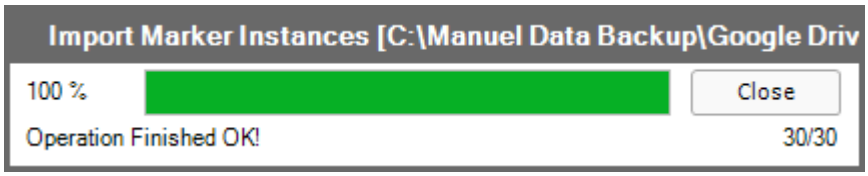
4.4.3. Import Shapes

In the following steps, you will import a shape template and instances already created.

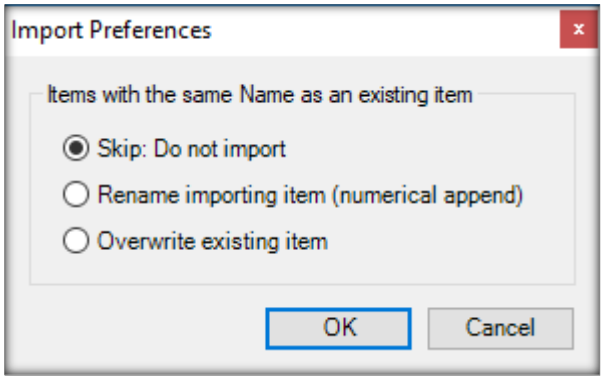
1. Double click the **Layers** button.
2. Click **Import Layers** to import the **Energy** layers.
3. Select **Skip: Do not import** at the Import Preferences dialog box.



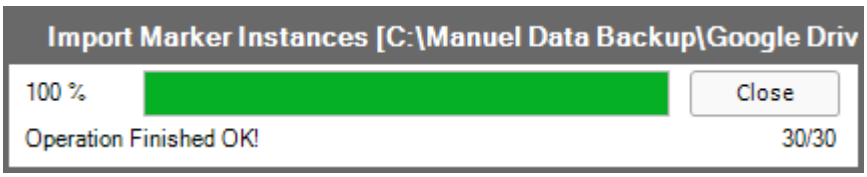
4. Click **Ok** button and select the file **Layers\Layers_Energy.xml** located in your training files.



5. Click the **Close** button when the process finish.
6. Double click the **Shape Template** option.
7. Click **Import Shape Templates** to import the **GD_Energy** shape template.
8. Select **Skip: Do not import** at the Import Preferences dialog box.



9. Click **Ok** button and select the file **Shapes\Shape_Template_GD_Energy.xml** located in your training files.

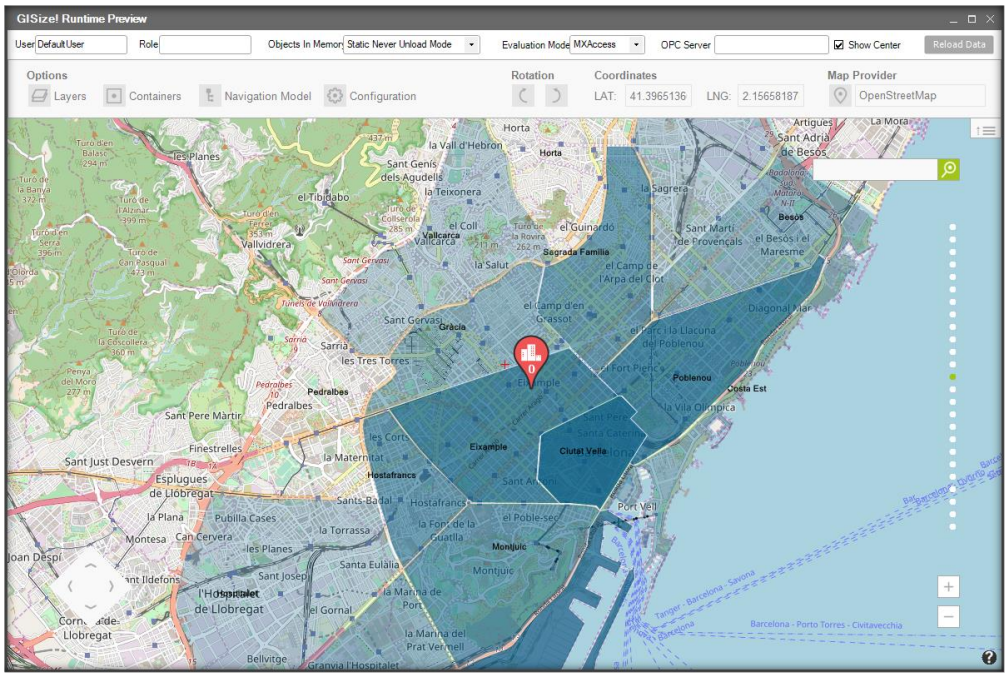


10. Click the **Close** button when the process finishes.
11. Open the **Shape Instances** option and **Import** the shapes instances from the file **Shapes\Shape_Instances_GD_Energy.xml** located in your training files.
12. **Enable** the **Energy Layers** at the **OpenStreetMap** provider.

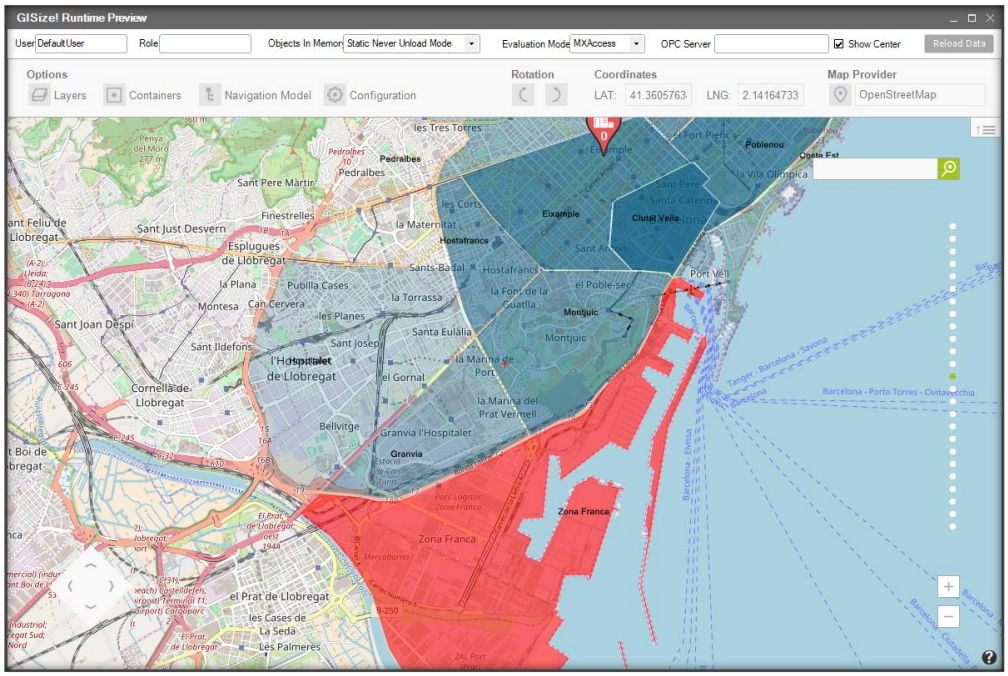
4.4.4. Test at the Runtime

In the following steps, you will test the shape instances animations at the runtime.

1. Analyze the configuration of the shape template and open the **GISIZE GEOSpatial Runtime Preview** tool to see the behavior of the shape instances.



You can modify the attribute **AlarmLevel** with a value between 0 and 2 to check how the shape color changes. For example, if you enter **2** for the attribute **EL_D001.AlarmLevel**, the shape color changes into red.



4.5. Lab 13 - Modeling a Tooltip

4.5.1. Introduction

In this lab, you will model a tooltip to show additional information related to the GISIZE GEOSpatial object.

4.5.2. Objectives

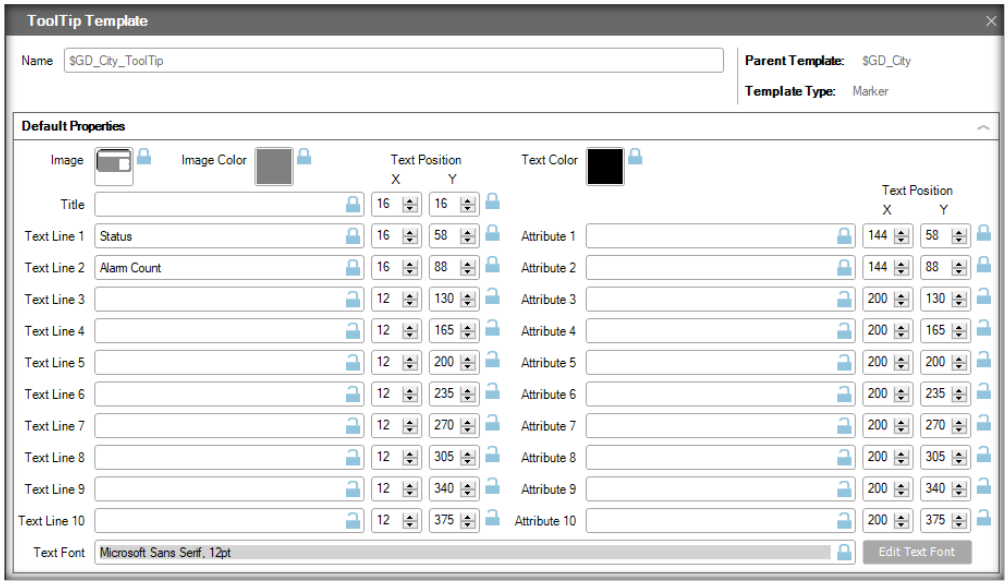
Upon completion of this lab, you will be able to:

- Create and configure tooltips.
- Modify the content of the tooltips with custom properties.

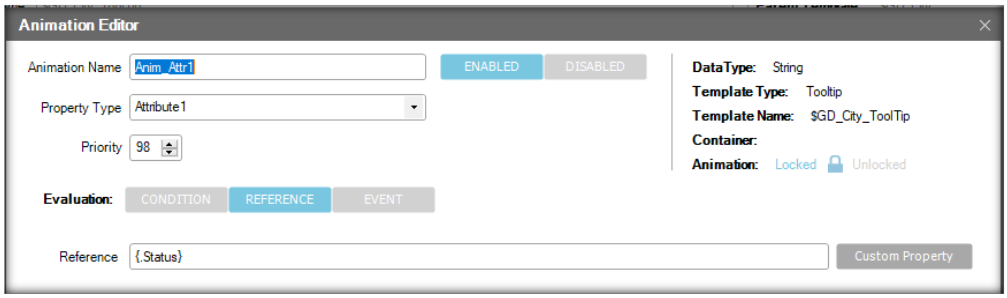
4.5.3. Create a tooltip

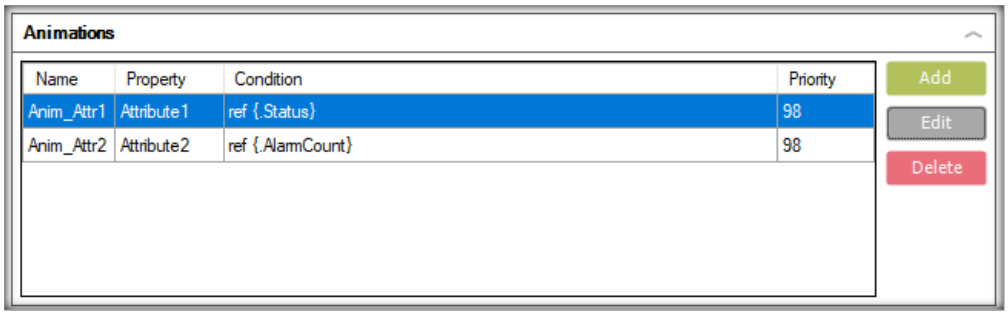
In the following steps, you will create a tooltip.

1. Double click the **Marker Template** button.
2. Double click the **$GD_City** template to edit it.
3. Click the **Edit Tooltip** button to create a tooltip object.
4. Double click the **Image** property and select the file **Miscellaneous\Tooltips\City_Tooltip.png** located in your training files.
5. Double click the **Image Color** property and select **gray** color.
6. Double click the **Text Color** property and select **black** color.
7. Enter **Status** in the field **Text Line 1** and enter **Alarm Count** in the field **Text Line 2.**
8. Configure the **XY** columns for the following lines according to:
    a. Title: X=16, Y=16
    b. Text Line 1: X=16, Y=58
    c. Text Line 2: X=16, Y=88
    d. Attribute 1: X=144, Y=58
    e. Attribute 2: X=144, Y=88
9. Change the **Font size** to **12pt**.



10. Expand Animations and create animations for Attribute 1 and Attribute 2 linked to Status and AlarmCount custom properties.



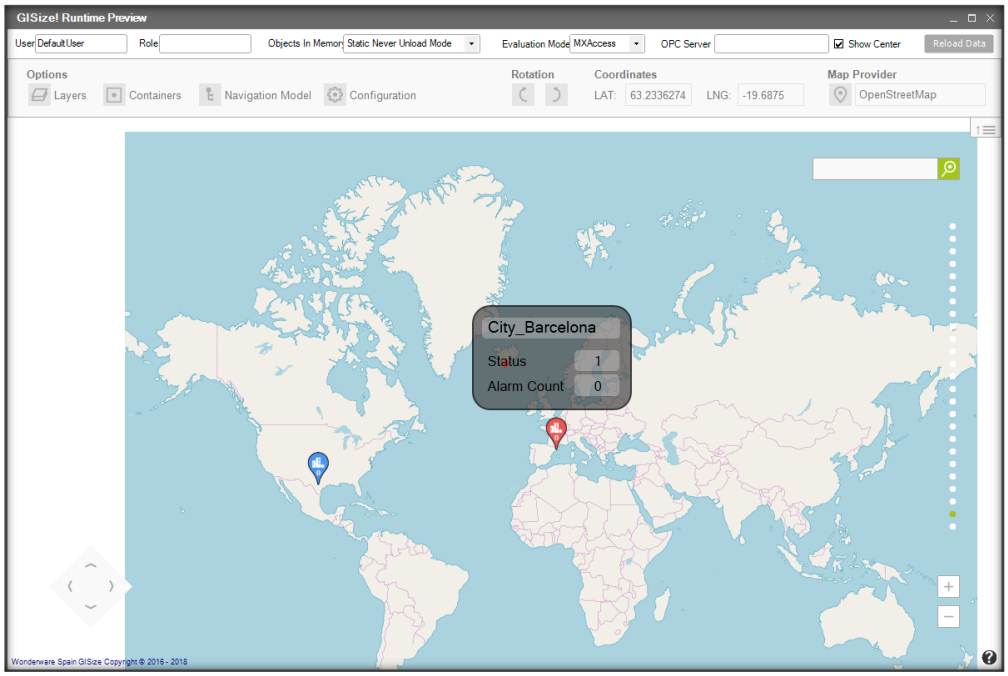


11. Lock all the properties and animations configured previously.
12. Click the **Save and close** button.

4.5.4. Test at the Runtime

In the following steps, you will test the shape instances animations at the runtime.

1. Open the **GISIZE GEOSpatial Runtime Preview**, move the mouse over a City Marker and wait 5 seconds to see the new tooltip (in the InTouch application it is possible to configure the delay time to open a tooltip automatically, but in the runtime preview tool is pre-defined to 5 seconds).



4.6. Lab 14 - Exporting and importing objects

4.6.1. Introduction

In this lab, you will create GISIZE GEOSpatial instances automatically using the Import and Export features.

4.6.2. Objectives

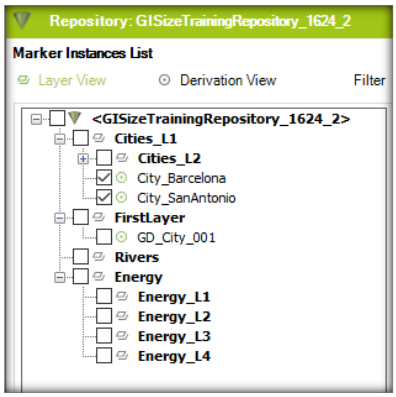
Upon completion of this lab, you will be able to:

- Export GISIZE GEOSpatial Instances.
- Modify the XML files and modify their content.
- Import or modify GISIZE GEOSpatial instances automatically.

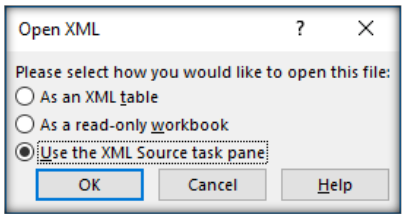
4.6.3. Export and import objects

In the following steps, you will export and import instances, create a new one using Microsoft Excel and import it.

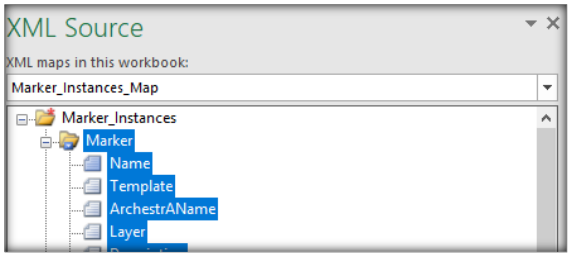
1. Double click the **Marker Instances** button.
2. Select the Instances **City_Barcelona** and **City_SanAntonio**.

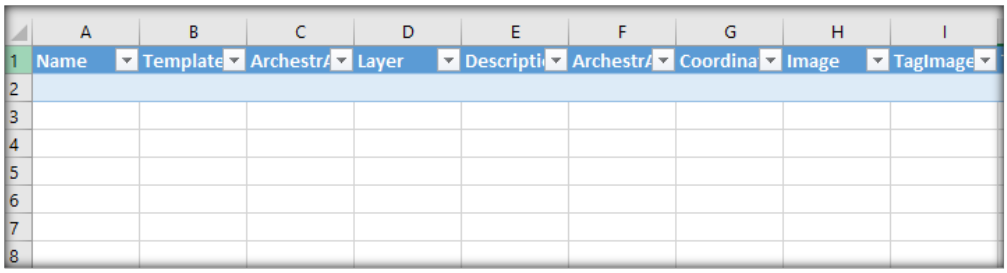


3. Click **Export Checked Marker Instances** and save the file using the filename **Cities_Export.xml.**
4. Open the file using **Microsoft Excel** and choose the option **Use the XML Source pane.**

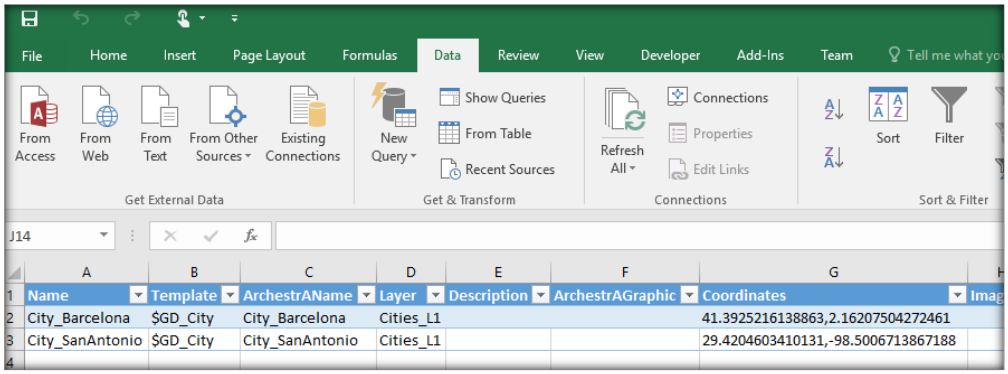


5. Click the **Marker** folder in the **XML Source tree** and drag and drop it to the first cell of the sheet.

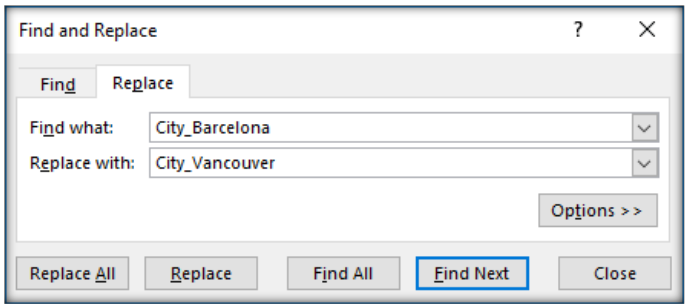

XML Source
XML maps in this workbook:
Marker_Instances_Map
Marker_Instances
Marker
Name
Template
ArchestrAName
Layer


A
B
C
D
E
F
G
H
I
Name
Template
ArchestrA
Layer
Descripti
ArchestrA
Coordina
Image
TagImage

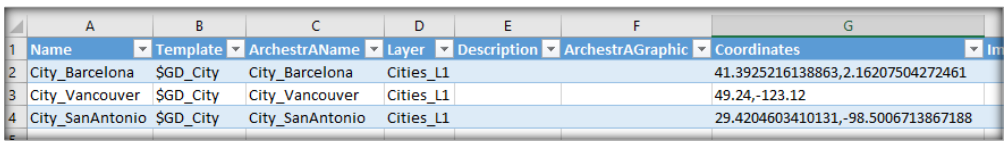
6. Select the **Data** tab and click the **Refresh All** button.



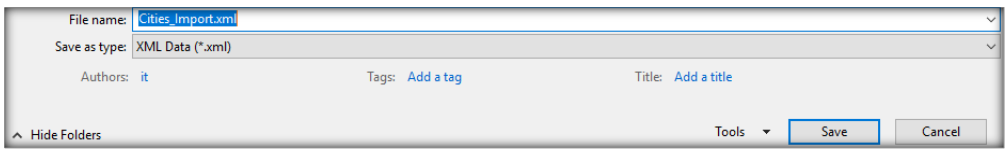
7. Copy the **City_Barcelona** row and insert it between the second and third row.
8. Select the second **City_Barcelona** row and **replace** (On the Edit menu, click Replace) **Barcelona** with **Vancouver**.



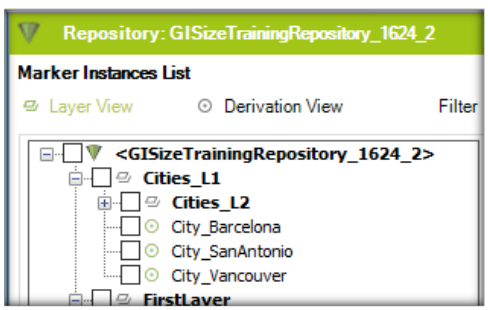
9. Enter **49.24,-123-12** in the Coordinates cell for **City_Vancouver**.



10. On the **File** menu, click **Save As** and Save as type **XML Data (*.xml)** using the filename **Cities_Import.xml.**



11. Return to the **Marker Instances** panel in the **ArchestrA IDE,** click the **Import Marker Instances** button and select the file created previously.

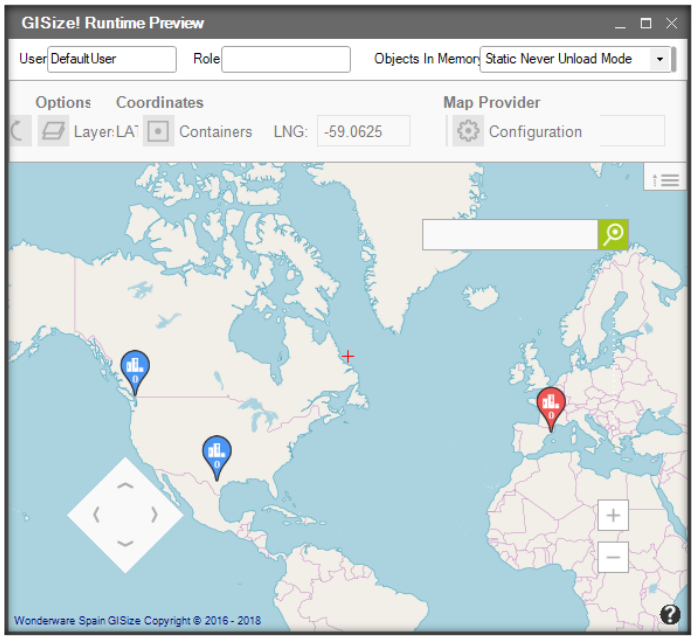


12. Return to the **Marker Instances** panel in the **ArchestrA IDE,** click the **Import Marker Instances** button and select the file created previously.
13. Go to **Derivation View** in ArchestrA IDE, select the Wonderware template **$GD_City**, create a new Instance for **City_Vancouver** and deploy it.

4.6.4. Test at the Runtime

In the following steps, you will test the shape instances animations at the runtime.

1. Double Click **the Runtime Preview** button to open the GISIZE GEOSpatial Runtime Preview tool and see the new Marker Instance for Vancouver City.

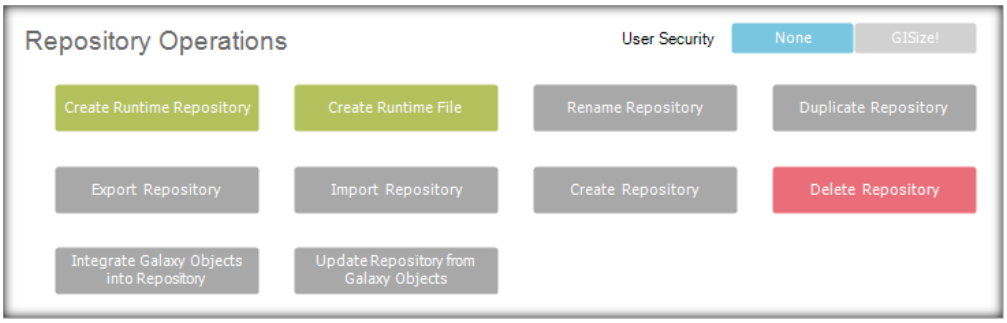


4.7. Automatic creation of objects

This section describes the utilities to create and update GISIZE GEOSpatial objects automatically using the Wonderware model as a reference.

4.7.1. Introduction

In the Administration panel there are two buttons to access the utilities to create or update the GISIZE GEOSpatial model using the Wonderware model as a reference.



4.7.2. Integrate Galaxy Objects into Repository

This button opens a wizard window that allows the creation of GISIZE GEOSpatial layers, templates, and instances based on existing areas, templates, and instances in the current ArchestrA Galaxy.

The process is pretty straightforward, allowing to choose which templates and instances are to be integrated.

Templates (and their custom properties) created this way will automatically be associated with their equivalent items in the Galaxy.

The integration process is design limited to some properties: Names, descriptions, parent layers, parent templates, custom properties, and few more.

ArchestrA Galaxy identification will be required to perform this operation if needed.

4.7.3. Update Repository from Galaxy Objects

This button opens a wizard window that allows the update of GISIZE GEOSpatial layers, templates, and instances previously associated with existing areas, templates, and instances in the current ArchestrA Galaxy.

The galaxy object can be associated with the integration wizard process or manually editing its corresponding property.

The process is pretty straightforward, allowing to choose which templates and instances are to be updated.

The updating process is design limited to some properties: Names, descriptions, parent layers, parent templates, custom properties, and few more.

ArchestrA Galaxy identification will be required to perform this operation if needed.

4.8. Lab 15 - Integrating Galaxy Objects into Repository

4.8.1. Introduction

In this lab, you will use the utilities to create and update GISIZE GEOSpatial objects automatically using the Wonderware model as a reference.

4.8.2. Objectives

Upon completion of this lab, you will be able to:

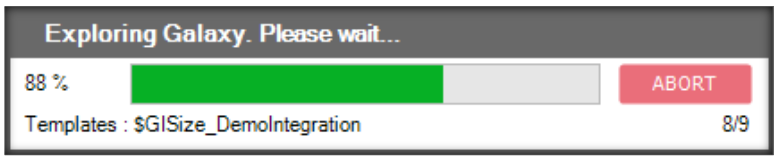
- Create GISIZE GEOSpatial objects using Wonderware model as a reference.
- Update GISIZE GEOSpatial objects previously associated with Wonderware objects.

4.8.3. Integrate Galaxy Objects in the Repository

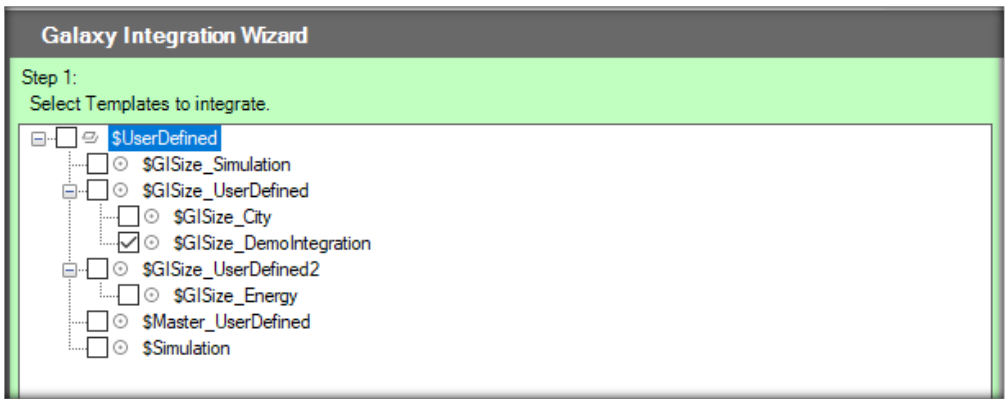
In the following steps, you will create GISIZE GEOSpatial Objects using the Wonderware model as a reference.

1. Import the ArchestrA package **Labs\Lab15\GISIZE_DemoIntegration.aaPKG** located in your training files.
2. Double click the **Administration** button.
3. Click the **Integrate Galaxy Objects in the Repository** button.

   The process to explore the galaxy will take a few moments.



2. In the Step 1, select the **GISIZE_DemoIntegration** Wonderware template and click the **Next** button.

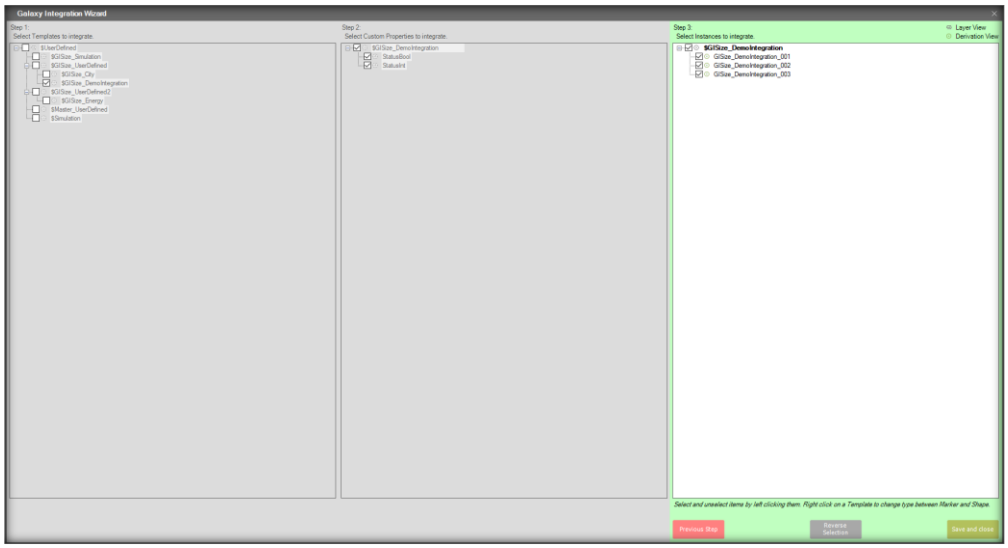


   The process to explore the galaxy will take a few moments.

3. In the Step 2, select the **StatusBool** and **StatusInt** Wonderware Attributes and click the **Next** button.

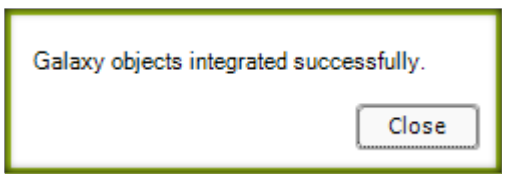
The process to explore the galaxy will take a few moments.

4. In the Step 3, select the **Wonderware Instances** and click the **Next** button.

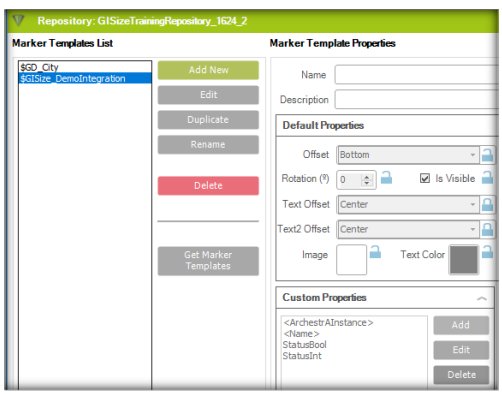


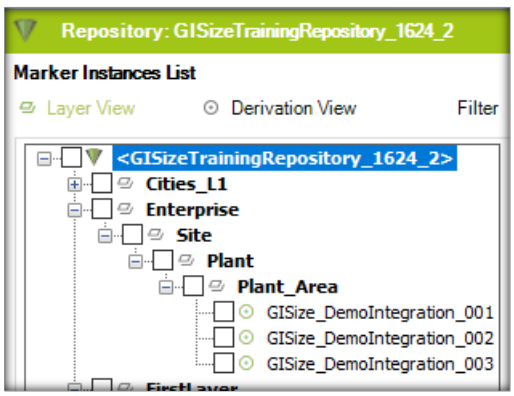
The process to create the GISIZE GEOSpatial objects will take a few moments.

5. Click the **Close** button when process finish.

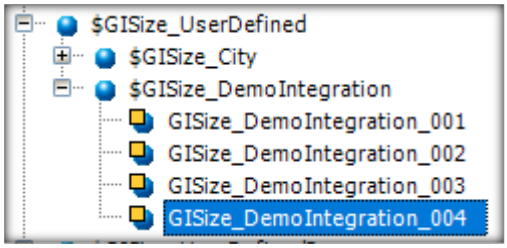


6. Analyze the new Layers, Marker Templates and Marker Instances created in the repository.





7. In ArchestrA IDE create a new instance derived from **GISIZE_DemoIntegration template**.

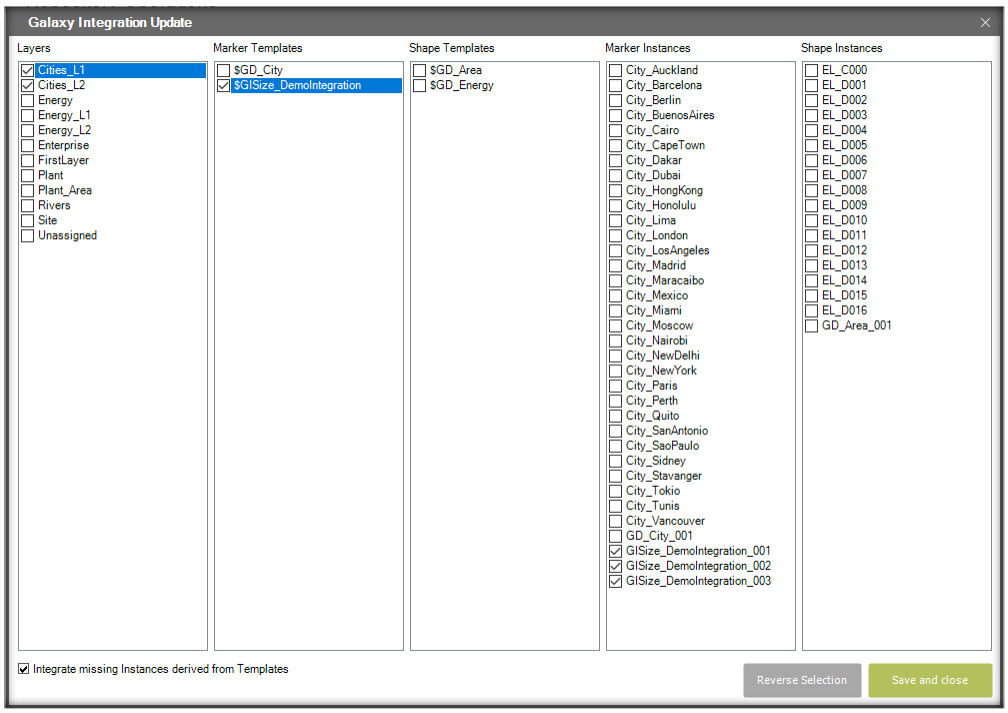

$GISize_UserDefined
$GISize_City
$GISize_DemoIntegration
GISize_DemoIntegration_001
GISize_DemoIntegration_002
GISize_DemoIntegration_003
GISize_DemoIntegration_004

4.8.4. Update Repository from Galaxy Objects

In the following steps, you will update the repository with the new instances created in the Wonderware model.

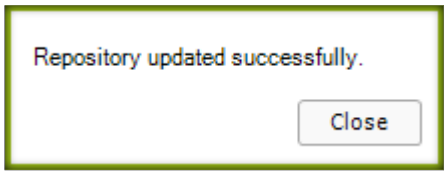
1. Double click the **Administration** button.
2. Click the **Update Repository from Galaxy Objects** button.

   The process to explore the galaxy will take a few moments.
3. Click the **Reverse Selection** button and select the **GISIZE_DemoIntegration** Template.

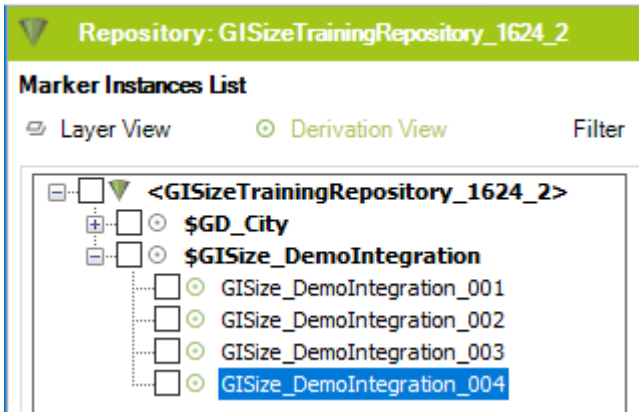


4. Click the **Save and Close** button.

   The process to create the GISIZE GEOSpatial objects will take a few moments.
5. Click the **Close** button when process finish.



6. Analyze the Marker Instance created in the repository.



5. Wonderware Integration

The objectives of this module are:

- Describe the graphical user interface provided by GISIZE GEOSpatial at the Runtime
- Describe the GISIZE GEOSpatial Control, its properties and events and how to use it in an ArchestrA Graphic to integrate GISIZE GEOSpatial in InTouch for System Platform and InTouch Standalone.
- Describe the GISIZE GEOSpatial ArchestrA App, its properties and how to use it in a Layout to integrate GISIZE GEOSpatial in InTouch OMI.

5.1. Run Time behavior

This section explains the graphical user interface provided by GISIZE GEOSpatial at the Runtime

5.1.1. Introduction

At the runtime, the GISIZE GEOSpatial control behaves like an interactive add-on for Wonderware System Platform that enhances the visualization and navigation of spatial or geographical data in a Geographic Information System (GIS).

The GISIZE GEOSpatial control allows representing all the elements in a huge installation as spatial or geographical data in a Geographic Information System (GIS). It's possible to use the standard map providers or create your own map or image provider based on vector files.

In the next chapters, how to navigate a map provider and the controls comprising the user interface will be explained

5.1.2. Navigation

The navigation though a map provider is very simple:

- **Movement:** It's done by dragging the map, keeping the left button of the mouse pressed and moving the cursor.
- **Zoom:** It's done with the mouse wheel. Moving forward to zoom in and moving backward to zoom out.
- **Element interaction:** Markers and shapes trigger events when they are clicked or double clicked with the left button of the mouse.

5.1.3. User Interface

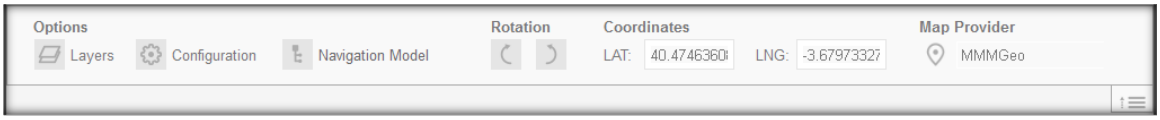
The GISIZE GEOSpatial control comes with a predefined user interface comprises a set of controls. Each of these controls performs functionalities that are also accessible by scripting.

The usage of this user interface is optional and its visibility can be configured by scripting.

In the next chapters the controls of the predefined user interface will be explained.

5.1.3.1. Main bar

The main bar is a bar located at the top side of the GISIZE GEOSpatial control. Its width is adjusted automatically to the width of the control.



This control allows the visualization of 3 option panels:

- **Layers:** Clicking this button shows/hides the panel of layer visualization.
- **Configuration:** Clicking this button shows/hides the panel of GISIZE GEOSpatial control configurations.
- **Navigation Model:** Clicking this button shows/hides the panel of navigation model selection.

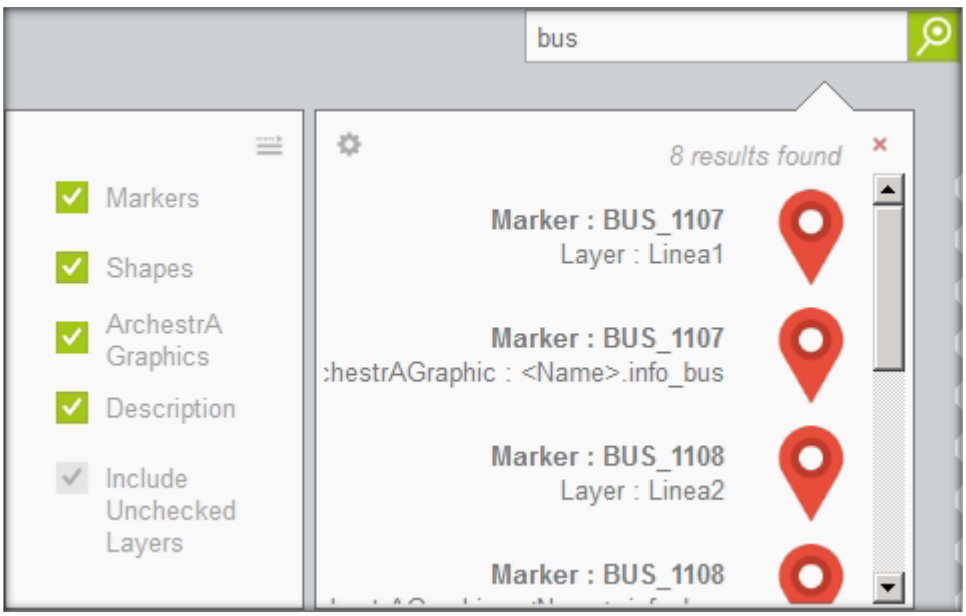
This control also allows the visualization and configuration of the following properties:

- **Rotation:** Clicking these buttons rotates the content of the GISIZE GEOSpatial control.
- **Coordinates:** This control shows the current Latitude and Longitude position on the map. The position can be edited to center the map to the desired location.
- **Map Provider:** This control is a drop down list that shows the current map provider visualized and allows selecting other ones.

The main bar can be minimized or restored by clicking the bottom-left button.

5.1.3.2. Search tool

The search tool looks like a text field with a magnifying glass. It's used to search for instances of markers and shapes by name and by associated ArchestrA graphic. The tool will search for any partial or full match.



The result of the search will be displayed in a list that will appear under the text field. This list will show the name of the markers or shapes, the associated ArchestrA graphic (if any) and the graphic representation of the instance. Clicking any result will center the map over the instance.
Upon clicking the clog icon, a small panel will appear allowing to filter the results by markers, shapes, ArchestrA graphics, and description. Clicking the red cross icon will close the result panel.

5.1.3.3. Movement Pad

The movement pad allows the navigation though the map step by step.

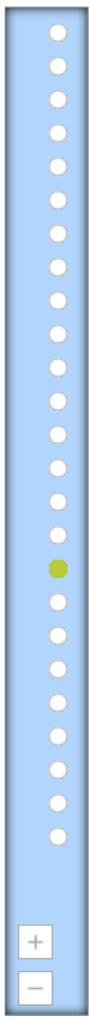
Clicking the arrow will move the location of the map to the indicated direction.

5.1.3.4. Zoom control

This control is represented by a set of circles, one for each of the 24 zoom levels that exist. The circle corresponding to the current zoom level appears colored.

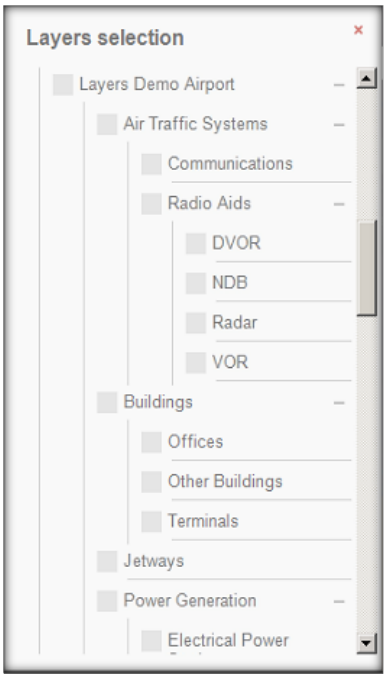
Clicking a circle, will zoom the map provider to the zoom of the clicked circle. This allows navigating directly to the desired zoom without passing through the intermediate zooms.

At the bottom, there are 2 buttons to zoom in/out step by step.



5.1.3.5. Layer Selection tab

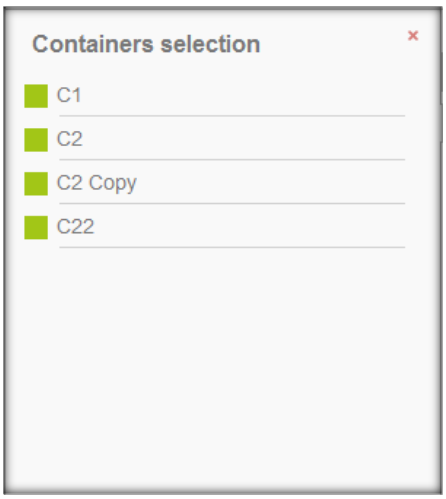
This panel shows all the layers allowed for the current map provider in a tree like view. The currently active layers are checked and the inactive ones are unchecked. Inactive layers will be always invisible, even if the conditions are correct.



Layers in the tree are clickable, allowing to set the layers active/inactive.

5.1.3.6. Containers Selection tab

This panel shows all the containers which contain at least one item potentially visible on screen in a check list view. The currently active containers are checked and the inactive ones are unchecked. By default, all the containers are marked as visible.

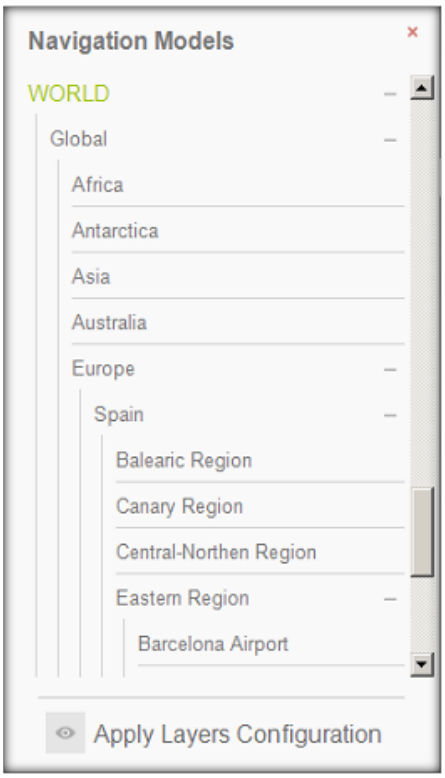


Containers in the list are clickable, allowing to set the containers active/inactive.

5.1.3.7. Navigation Models tab

This panel shows all the navigation models defined in the current repository in a tree like view. Upon clicking a navigation model, the selected navigation model will be applied to the

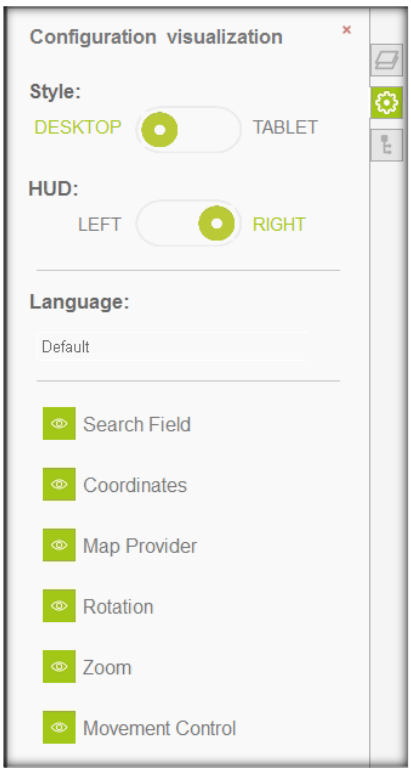
GISIZE GEOSpatial control, thus potentially changing the map provider, the location, and the zoom level.



If the button at the bottom is marked, the enabled layer configuration of the selected navigation model will be applied too.

5.1.3.8. Configuration tab

This panel allows the configuration of the visualization of the elements of the user interface.

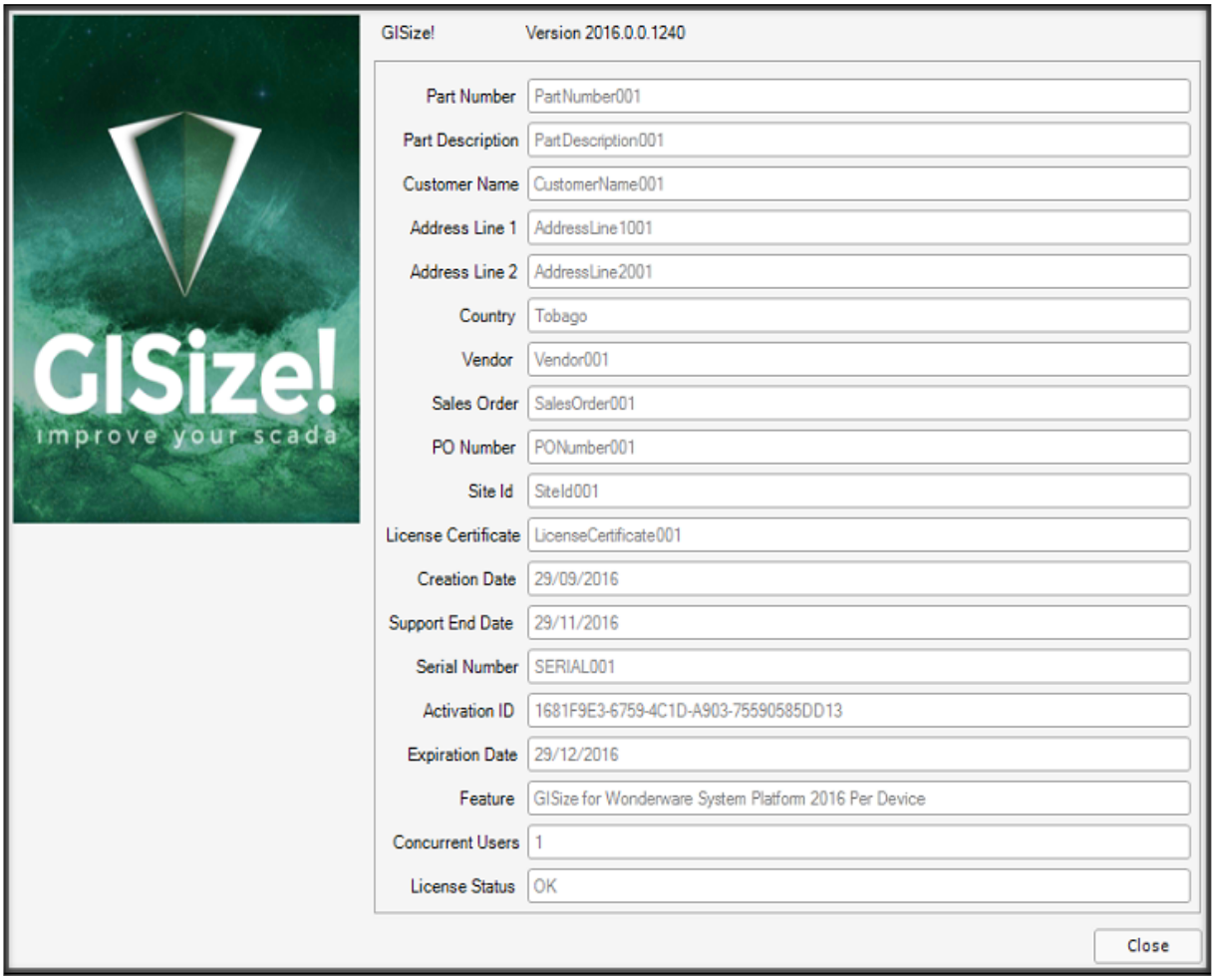


The configurable elements are:

- **Style:** To visualize the user interface in desktop or tablet mode.
- **HUD:** To arrange the user interface elements at the left or the right side of the control.
- **Language:** Allows changing the language of the GISIZE GEOSpatial control between the defined languages.
- **Search Field:** Shows/hides the search tool control.
- **Coordinates:** Shows/hides the coordinates at the main bar control.
- **Map Provider:** Show/hides the map provider selector at the main bar control.
- **Rotation:** Show/hides the rotation buttons at the main bar control.
- **Zoom:** Show/hides the lateral zoom control.
- **Movement control:** Show/hides the movement pad control.

5.1.3.9. About Info

Upon clicking the question mark icon at the GISIZE GEOSpatial control, the About Info pop up window will appear.



This informative window shows relative information about the current version of the GISIZE GEOSpatial control and the current active license.

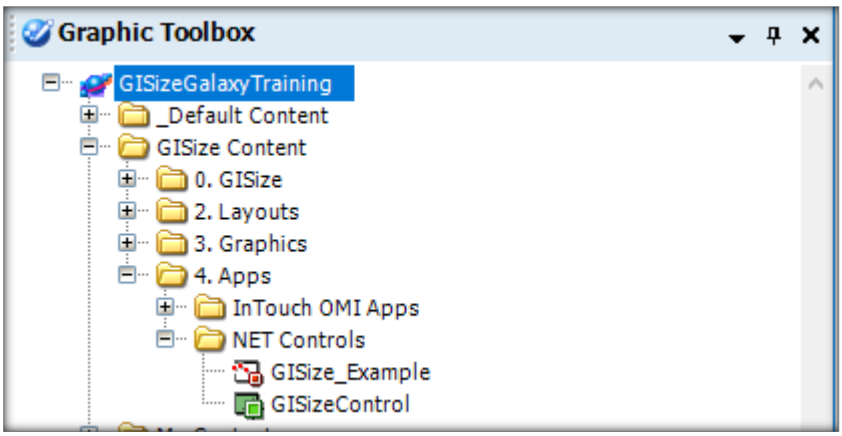
5.2. GISIZE GEOSpatial Control

This section describes the GISIZE GEOSpatial Control, its properties and events and how use it in an ArchestrA Graphic to integrate GISIZE GEOSpatial in InTouch for System Platform and InTouch Standalone.

5.2.1. Introduction

The GISIZE GEOSpatial control is a .NET control that can be integrated inside an ArchestrA graphic control. This control is the viewer of the maps, shapes, and markers defined in a GISIZE GEOSpatial repository. A GISIZE GEOSpatial control always works together with a GISIZE GEOSpatial repository, through an SQL database or a standalone repository file.

After importing the GISIZE GEOSpatial objects, there must be a folder named **GISIZE Content/4. Apps/Net Control** at the Graphic Toolbox in the ArchestrA IDE.



This folder contains the GISIZE GEOSpatial control itself named **GISIZEControl**, and an ArchestrA graphic demo named **GISIZE_Example**.

A GISIZE GEOSpatial control must be integrated inside an ArchestrA graphic to be used in an InTouch application. This is done through the 'Embed Graphic' option like with any other .Net control or symbol.

Once added inside an ArchestrA graphic, a GISIZE GEOSpatial control must be configured properly.

5.2.2. Configuration

The configuration of a GISIZE GEOSpatial control is done through the editing of its public properties. The easiest way to do this is by selecting the GISIZE GEOSpatial control inside an ArchestrA graphic. Doing that, the left panel of the symbol window will be updated.

GISize Connection	
LocalFile	
Repository	**Demo_GISize**
SQLPassword	
SQLServer	**(local)**
SQLTrustedConnection	**True**
SQLUser	
UseLocalFile	**False**
GISize Properties	
ControlVersion	2016.0.0.1281
DesignTime	**True**
EvaluationMode	**MXAccess**
Language	**Default**
LicenseStatus	OK
MapProvider	**GoogleMap**
ObjectsInMemory	**Dynamic**
Rotation	**0**
ShowCenter	True
StartLocation	**0,0**
ToolTipDelay	**5000**
UserName	
Zoom	**0**

Some of them are common to all .NET controls, but the specific GISIZE GEOSpatial control properties can be classified as connection and behavior properties.

5.2.2.1. GISIZE GEOSpatial Connection Properties

These properties are located under the 'GISIZE GEOSpatial Connection' group. They allow the GISIZE GEOSpatial control to connect and use a previously defined GISIZE GEOSpatial repository. These properties are (alphabetically ordered):

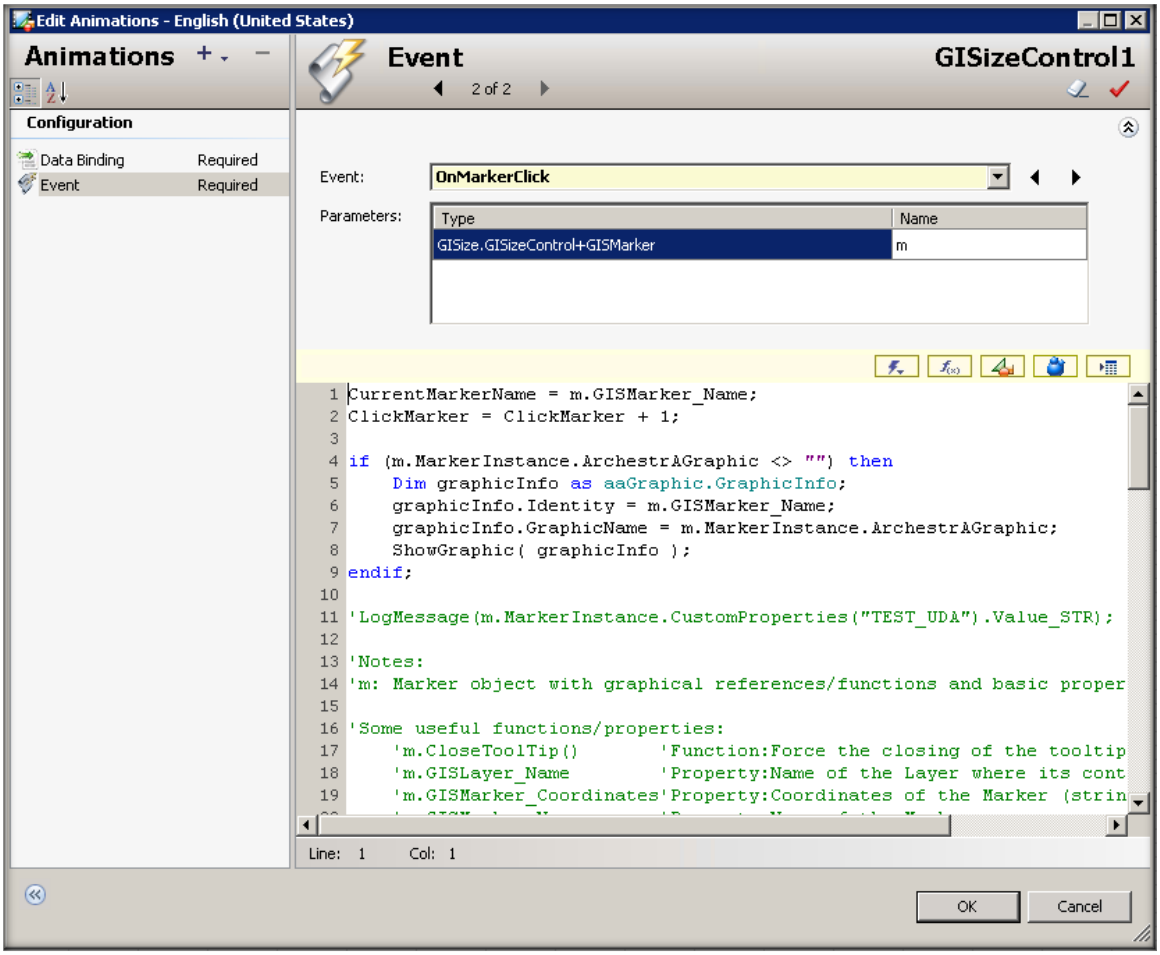
- **Local File**: Contains the full local path (file included) to the location of a GISIZE GEOSpatial Runtime Repository standalone file (.GRR). The control will use this file as its repository data. The property **'UseLocalFile'** must be true.
- **Repository**: Contains the name of the repository to be loaded.
- **SQLPassword**: Contains the SQL password to access an SQL server that contains GISIZE GEOSpatial repositories. Note that the password is stored in plain text. It's not used if the property **'SQLTrustedConnection'** is set to true.
- **SQLServer**: Name of the SQL server that contains the GISIZE GEOSpatial repository.
- **SQLTrustedConnection**: Boolean property that indicates if the connection to the SQL server is made with integrated OS security(true) or by using a native SQL user/password (false). To avoid security issues, it's recommended setting this property to true.
- **SQLUser**: Contains the SQL user to access an SQL server that contains GISIZE GEOSpatial repositories.
- **UseLocalFile**: Boolean property that indicates if the connection is made to an SQL server or an GISIZE GEOSpatial Runtime Repository standalone file (.GRR)

5.2.2.2. GISIZE GEOSpatial Behavior Properties

- **ControlVersion**: Read only property. Indicates the installed version of GISIZE GEOSpatial 2019.
- **DesignTime**. Boolean property that indicates if the GISIZE GEOSpatial control will accept Design Time Repositories or not. If the value is False, the GISIZE GEOSpatial control will work only with the Runtime Repositories. By default, it is False.
- **EvaluationMode**: Indicates the data source for the evaluation of custom properties and animations.
- **Language**: Indicates the name of the GISIZE GEOSpatial language to be used at the runtime. By default, it is 'Default'.
- **LicenseStatus**: Read only property. Indicates the current status of the license applied to GISIZE GEOSpatial 2019.
- **MapProvider**. String property that indicates the Map Repository to visualize at the runtime.
- **ObjectsInMemory**: A GISIZE GEOSpatial control can manage the memory of memory in 3 different ways with the configuration of a parameter. This control allows to test the behavior of the memory manage methods with the current repository. The 3 options are:
  - **Dynamic Mode**: The GISIZE GEOSpatial control only loads the required objects (layers, markers, shapes, etc…) that are visible on the screen, and unloads them when they are not on the screen any more. This uses less memory than the other options but also is the slowest option to visualize objects. This is the default option.
  - **Static Mode**: The GISIZE GEOSpatial control loads all the required objects (layers, markers, shapes, etc…), even if they are not on the screen any more. The objects are unloaded when no longer needed. With this option the visualization of the objects is faster but the use of memory is bigger too.
  - **Static Never Unload Mode:** The GISIZE GEOSpatial control loads all the required objects (layers, markers, shapes, etc…), even if they are not on the screen any more. Once loaded the object are never unloaded from memory. This is the fastest option to visualize the objects but also is the one that use most memory. Only recommended for small repositories.
- **Rotation:** Indicates in degrees the rotation of the content on the GISIZE GEOSpatial control. By default, it is 0.
- **ShowCenter:** Shows/hides the central cross.
- **StartLocation:** Indicates in Latitude and Longitude format the location where the map provider will draw at start. Default value is '0,0'.
- **ToolTipDelay:** Indicates in milliseconds the time that takes a tooltip to appear. Default value is 5000.
- **UserName:** Indicates the name of the starting GISIZE GEOSpatial user.
- **Zoom:** Indicates the initial zoom level of the map provider at start.

5.2.3. Scripting

The GISIZE GEOSpatial control, in addition to the commented properties, comes with many other properties and functions suitable to be used by scripting inside an ArchestrA graphic.

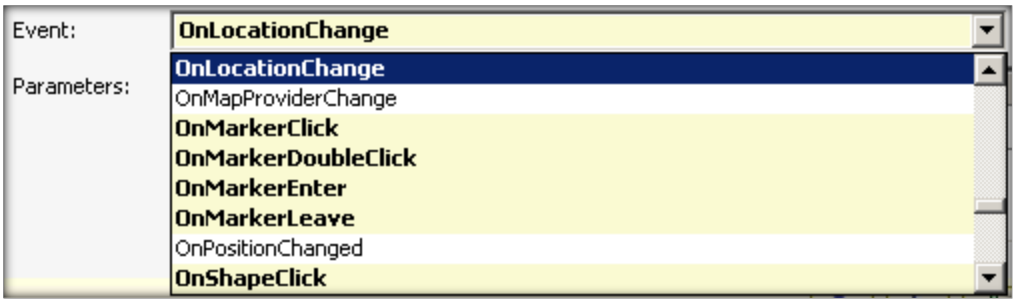


The **GISIZE_Example** ArchestrA graphic that comes with GISIZE GEOSpatial, contains examples and annotations of the use of some properties and functions.

For an exhaustive reference of GISIZE GEOSpatial properties and functions refer to the **GISIZE API User's Reference** document.

5.2.4. Events

Along with properties and functions, GISIZE GEOSpatial comes with a set of custom events that can be used by scripting too. Most of the events supply a parameter that indicates the value or object associated with the event triggered.



Some notable custom events are:

- **OnLocationChange:** Triggers when the location on the map provider is changed.
- **OnMapProviderChange:** Triggers when the map provider of the GISIZE GEOSpatial control is changed.
- **OnMarkerClick:** Triggers when a marker is clicked once.
- **OnMarkerDoubleClick:** Triggers when a marker is double-clicked.
- **OnMarkerEnter:** Triggers when the cursor is on a marker.
- **OnMarkerLeave:** Triggers when the cursor leaves a marker.
- **OnShapeClick:** Triggers when a shape is clicked once.
- **OnShapeDoubleClick:** Triggers when a shape is double-clicked.
- **OnShapeEnter:** Triggers when the cursor enters a shape.
- **OnShapeLeave:** Triggers when the cursor leaves a shape.
- **OnZoomChanged:** Triggers when the zoom of the control is changed.

Along with properties and functions, GISIZE GEOSpatial comes with a set of custom events that can be used by scripting too.

For an exhaustive reference of GISIZE GEOSpatial events refer to the **GISIZE API User's Reference** document.

5.3. Lab 16 - Integrating GISIZE in InTouch for SP

5.3.1. Introduction

In this lab, you will integrate GISIZE GEOSpatial in InTouch for SP.

5.3.2. Objectives

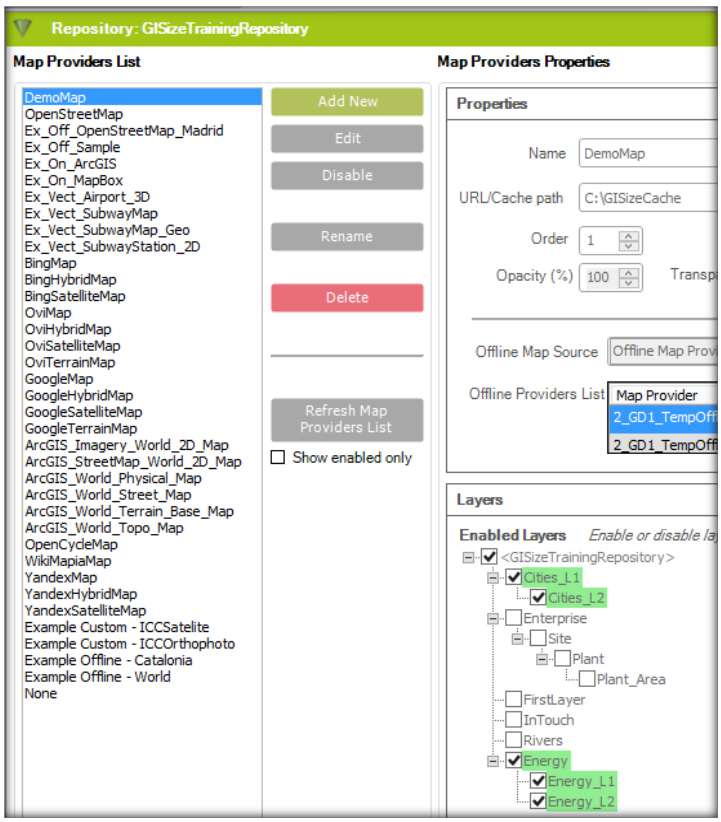
Upon completion of this lab, you will be able to:

- Import the GISIZE GEOSpatial control and the ArchestrA Graphic basic example.
- Configure the GISIZE GEOSpatial control to connect with a GISIZE GEOSpatial repository.
- Use basic GISIZE GEOSpatial Properties, Methods and Functions.

5.3.3. Import and configure ArchestrA Packages

In the following steps, you will integrate GISIZE GEOSpatial in InTouch for System Platform:

1. Rename the MapProvider **Ex_Off_AllLevels** to **DemoMap**, and introduce **1** in the field parameter **Order** (the **Ex_Off_AllLevels** map provider was previously imported during lab **Lab 08 - Combine several map providers**). Enable the layers for **Cities** and **Energy** and set them by default.

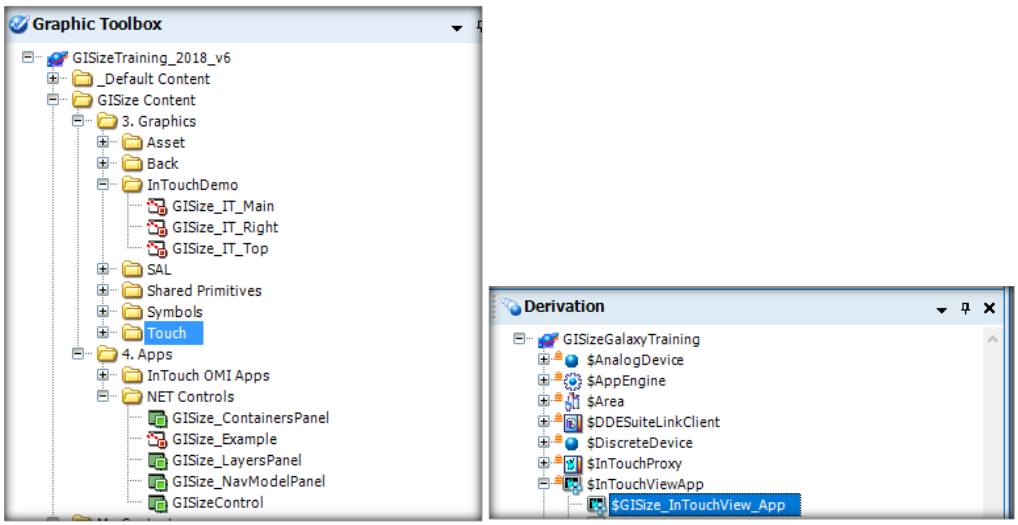


2. Import the ArchestrA package file **Apps\NET Controls\GISIZE_NET_Controls_Example.aaPKG** that can be found in the **Program Files/GISIZE** path of the computer, and must be imported from ArchestrA IDE with the option Galaxy -> Import -> Object(s).

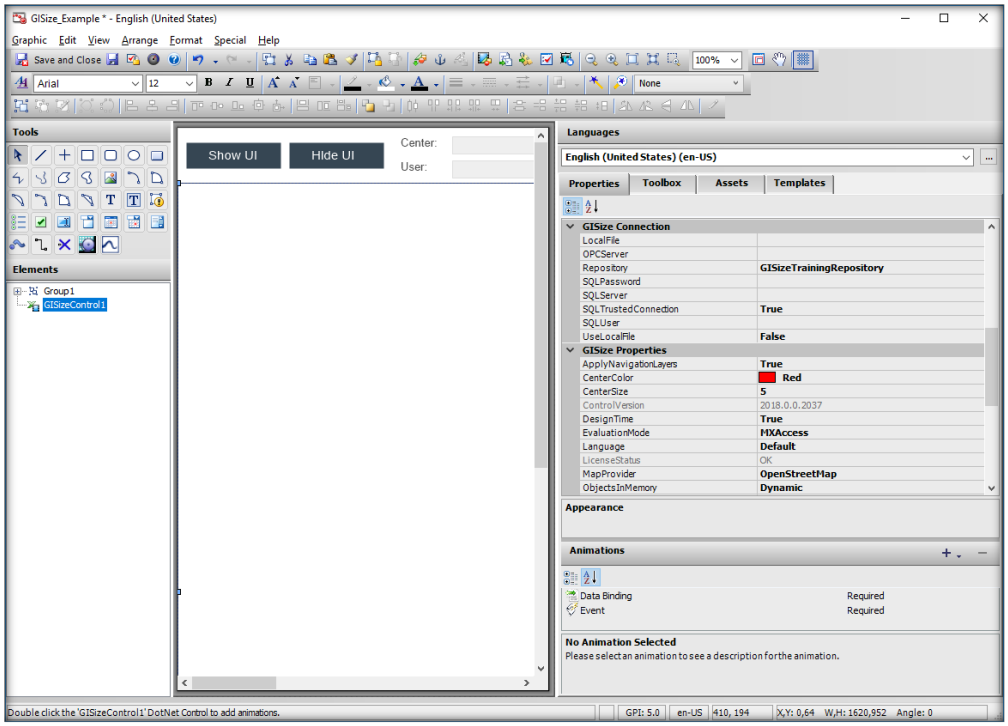
3. Import the **GalaxyStyles.xml** (ArchestrA IDE -> Galaxy -> Import -> Galaxy Style Library) located in your training files (folder **Lab 16 - Integrating GISIZE in InTouch for SP**).
4. Import the ArchestrA packages that can be found in the **Lab 16 - Integrating GISIZE in InTouch for SP** folder located in your training files.

   You will find the following objects:

   - Graphic Toolbox:
     - **GISIZE_IT_Main**
     - **GISIZE_IT_Right**
     - **GISIZE_IT_Top**
   - Derivation View
     - **$GISIZE_InTouchView_App**



5. Open the **GISIZE_Example** ArchestrA Graphic and verify at least the value of the following properties:
    - Repository: **GISIZETrainingRepository**
    - SQLTrustedConnection: **True**
    - DesignTime: **True**



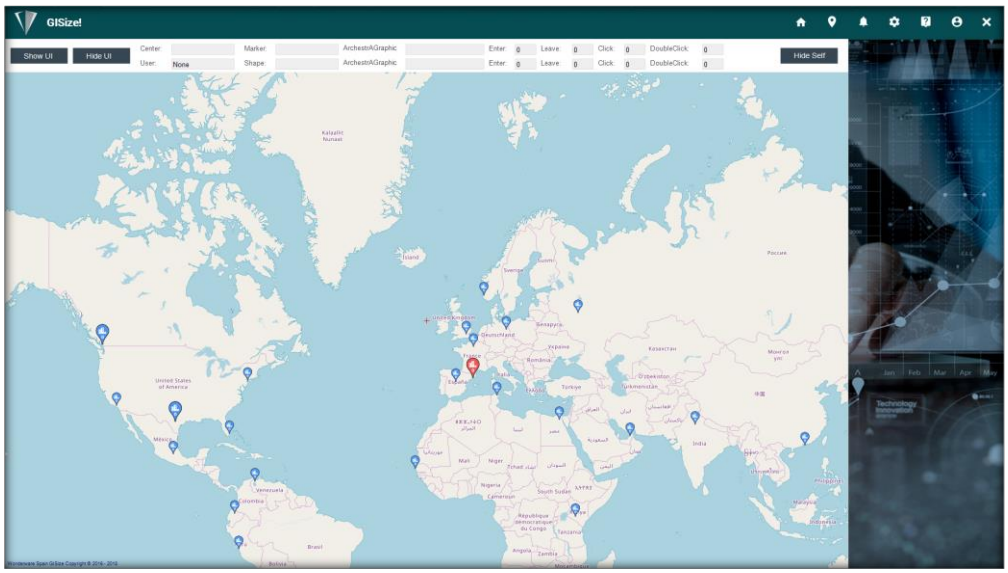
6. Edit the **$GD_City** Marker Template and enter **<ArchestrAInstance>.Faceplate** in the **ArchestrA Graphic** field.

5.3.4. Test InTouch View App

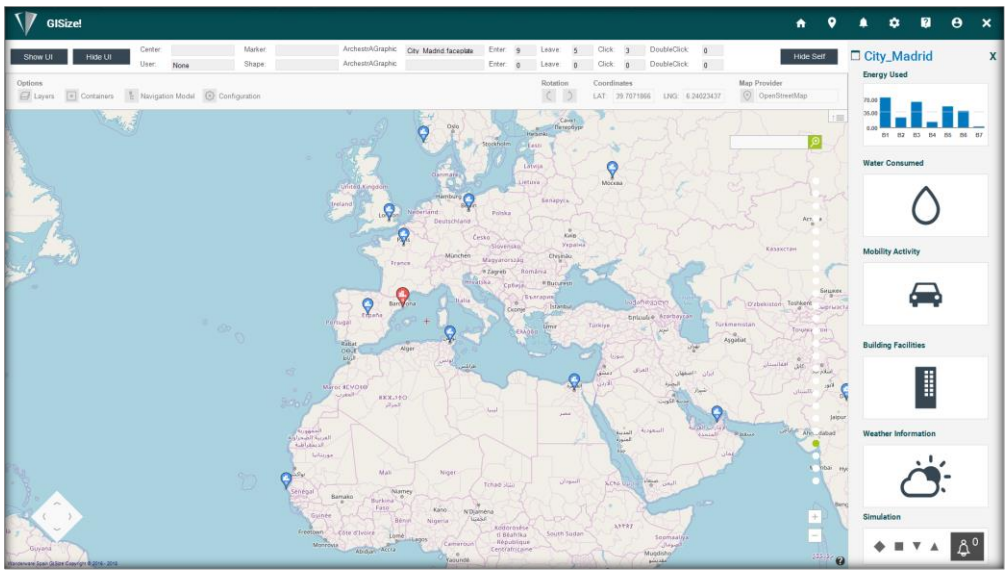
In the following steps, you will test GISIZE GEOSpatial embedded in an InTouch View App.

1. Edit **$GISIZE_InTouchView_App** InTouchViewApp.
2. In Window Maker open **Main Window**.
3. Click the **Runtime** button to open Window Viewer.

4. Click the **Map** button to open GISIZE GEOSpatial control embedded in **GISIZE_IT_AG** ArchestrA Graphic.



5. Test the behavior of the application clicking Show UI and Hide UI buttons, and clicking the Markers and Shapes.



5.4. Lab 17 - Communicate GISIZE with InTouch Standalone

5.4.1. Introduction

In this lab, you will communicate GISIZE GEOSpatial with InTouch Standalone in order to create InTouch Standalone solutions.

5.4.2. Objectives

Upon completion of this lab, you will be able to:

- Configure the OI.GATEWAY.2 driver to communicate with an InTouch Application.
- Configure references in the GISIZE GEOSpatial Custom Property to link using the OPC Group.
- Configure GISIZE GEOSpatial to communicate using the OPC Server protocol.

5.4.3. Create a new InTouch Standalone application

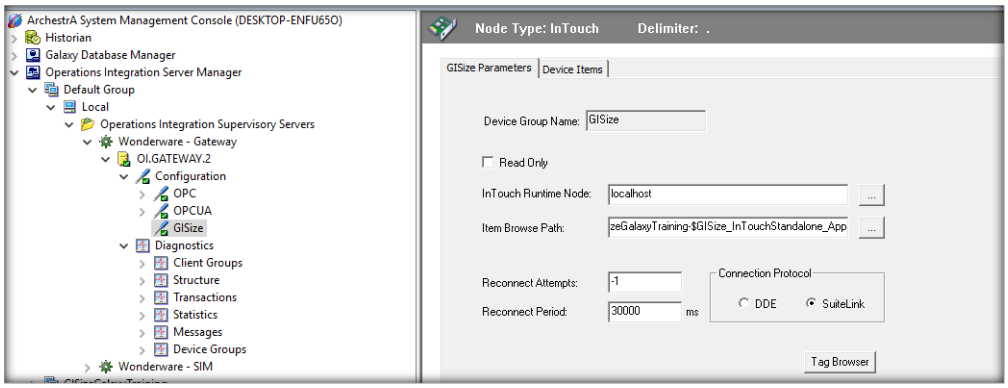
In the following steps, you will run an InTouch Standalone application.

1. Import the ArchestrA package file **$GISIZE_InTouchStandalone_App.aaPKG** located in your training files (folder **Lab 17 - Communicate GISIZE with InTouch Standalone**).
2. Open the **$GISIZE_InTouchStandalone_App.aaPKG** InTouch View App and put it at Runtime.

5.4.4. Configure the OI Gateway

In the following steps, you will configure the OIGateway to connect with an InTouch Standalone application.

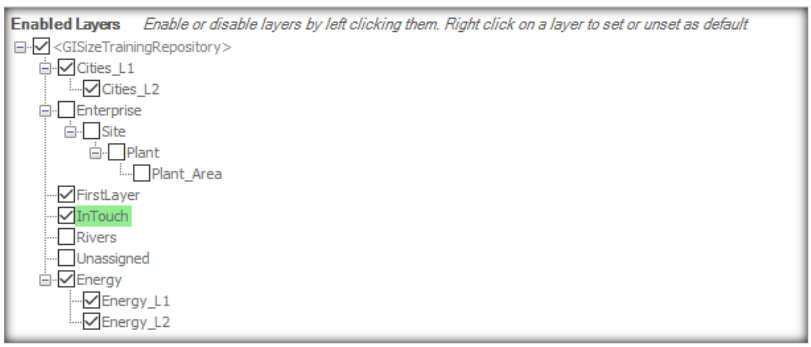
1. Open **System Platform Management Console** (SMC).
2. Expand **OI.GATEWAY.2**.
3. Right Click the **Configuration** folder and add a **New InTouch Connection**.
4. Enter **\\YOURMACHINENAME\GISIZEGalaxyTraining-$GISIZE_InTouchStandalone_App** in the **Item Browse Path** field.
5. Enter **-1** in the **Reconnect Attemps** field.



5.4.5. Configure GISIZE to connect with the OI.GATEWAY.2

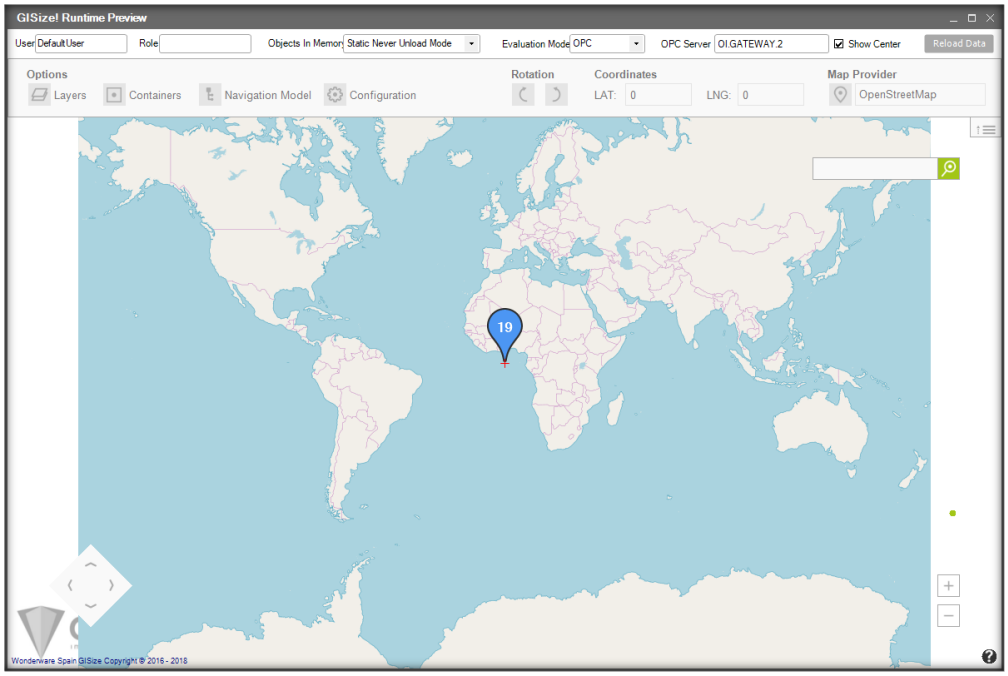
In the following steps, you will create the GISIZE GEOSpatial objects and configure the Runtime Preview to connect with the InTouch Application.

1. Create a new Layer named **InTouch.**
2. Enable this layer at **OpenStreetMapProvider** and make it the only one by default.



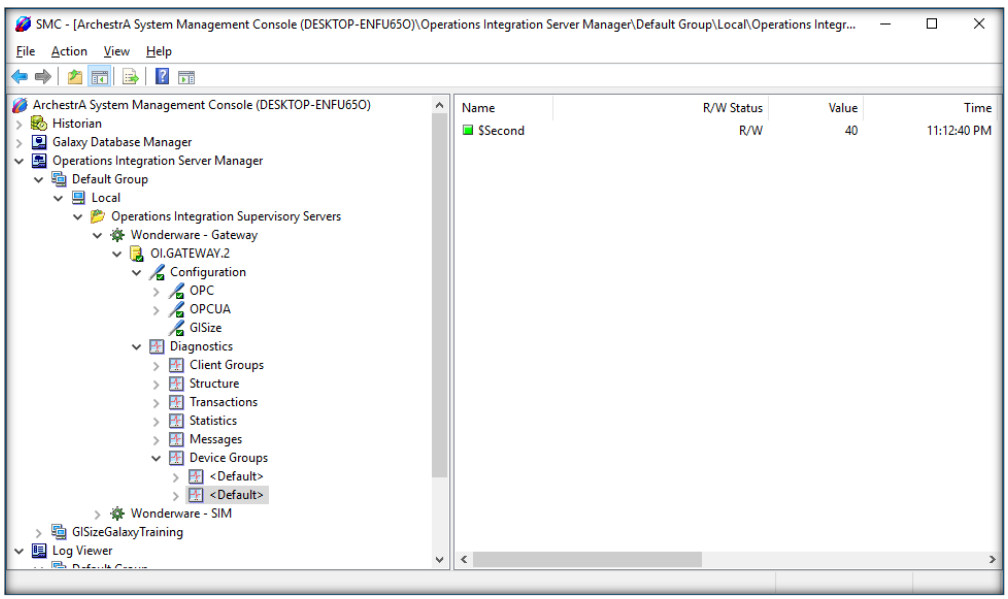
3. Import **MarkerTemplate_InTouchSeconds.xml** Marker Template (folder **Lab 17 - Communicate GISIZE with InTouch Standalone**).
4. Import **MarkerInstance_InTouchSeconds.xml** Marker Instance (folder **Lab 17 - Communicate GISIZE with InTouch Standalone**).
5. Open **GISIZE GEOSpatial Runtime Preview** tool.
6. Enter **OI.GATEWAY.2** in the **OPC Server** field.

7. Change the **Evaluation Mode** to OPC.



You should see how the value in the marker is linked to the tag **$Seconds** in the InTouch View Application.

You also can check the **Diagnostics** in the SMC.



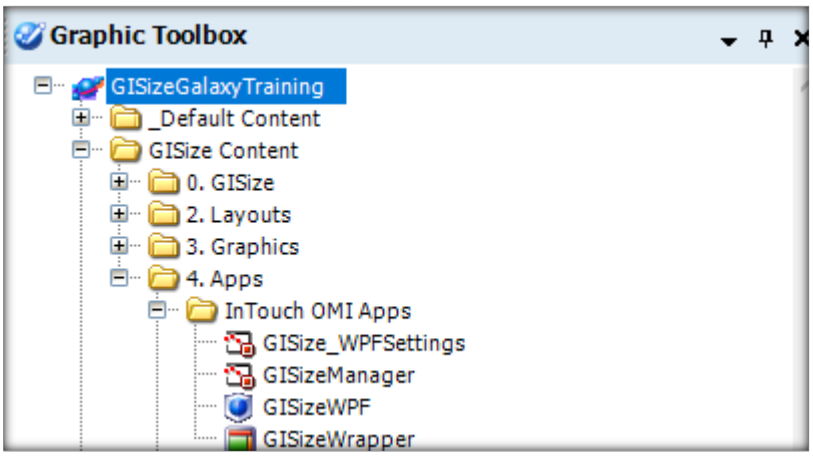
5.5. GISIZE GEOSpatial App

This section describes the GISIZE GEOSpatial ArchestrA App, its properties and how to use it in a Layout to integrate GISIZE GEOSpatial in InTouch OMI.

5.5.1. Introduction

The GISIZE GEOSpatial App is an ArchestrA App that can be integrated inside an ArchestrA layout to be visualized in InTouch OMI.

After importing of the GISIZE GEOSpatial objects, there must be a folder named **GISIZE Content/4. Apps/InTouch OMI Apps** at the Graphic Toolbox in the ArchestrA IDE.

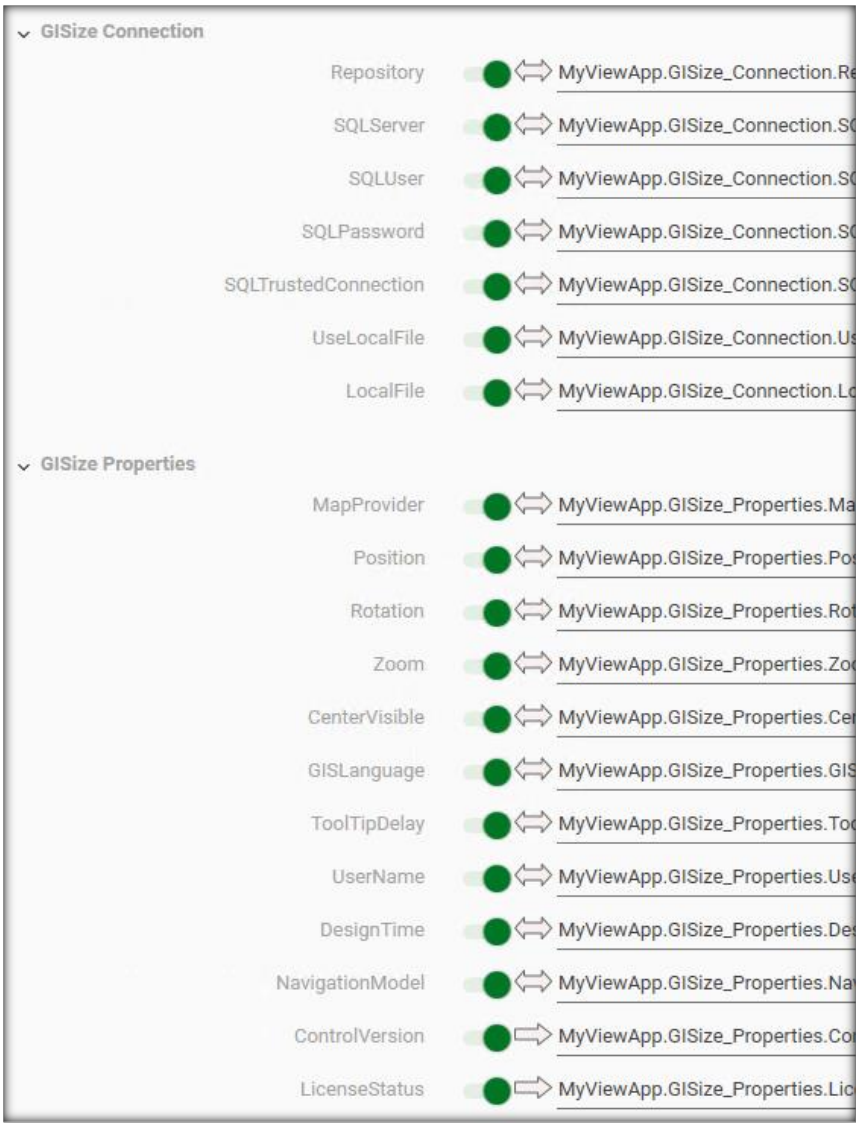


This folder contains the GISIZE GEOSpatial App named '**OI.GATEWAY.2**, an ArchestrA graphic demo named **GISIZE_GEOSpatial_Settings**, and an ArchestrA layout named **GISIZE_GEOSpatial_Wrapper** to be used in an InTouch OMI application.

The GISIZE GEOSpatial App must be configured properly through the editing of its properties at the ArchestrA layout.

5.5.2. Configuration

A GISIZE GEOSpatial App is configured through the editing of its public properties. The easiest way to do this, is by selecting the GISIZE GEOSpatial App inside an ArchestrA layout. Doing that will update the right panel of the layout window with the public properties.



The properties are organized in four groups:

- **GISIZE_Connection:** Read/write properties to connect to a GISIZE GEOSpatial Repository.
- **GISIZE_Properties**: Read/write properties to control the behavior at the runtime.
- **GISIZE_Functions**: Write only properties to execute GISIZE GEOSpatial functions.
- **GISIZE_Events:** Read only properties with the real time values of GISIZE GEOSpatial events.

These properties can be attached to System Tags and be accessible this way at the runtime from/to InTouch OMI.

5.5.2.1. GISIZE Connection Properties

These properties are located under the **GISIZE Connection** group. They allow the GISIZE GEOSpatial App to connect and use a previously defined GISIZE GEOSpatial repository. These properties are:

- **Repository**: Contains the name of the repository to be loaded. It's not used if the property UseLocalFile is set to true.
- **SQLServer**: Name of the SQL server that contains the GISIZE GEOSpatial repository. It's not used if the property UseLocalFile is set to true.
- **SQLUser**: Contains the SQL user to access an SQL server that contains GISIZE GEOSpatial repositories. It's not used used if the property UseLocalFile is set to true.or if the property 'SQLTrustedConnection' is set to true.
- **SQLPassword**: Contains the SQL password to access an SQL server that contains GISIZE GEOSpatial repositories. Note that the password is stored in plain text. It's not used used if the property UseLocalFile is set to true.or if the property 'SQLTrustedConnection' is set to true. 'SQLTrustedConnection' is set to true.
- **SQLTrustedConnection**: Boolean property that indicates if the connection to the SQL server is done with integrated OS security (true) or by the use of a native SQL user/password (false). To avoid security issues, it's recommended to set this property to true.
- **UseLocalFile**: Boolean property that indicates if the connection is to an SQL server or an GISIZE GEOSpatial Runtime Repository standalone file (.GRR)
- **LocalFile**: Contains the full local path (file included) to the location of a GISIZE GEOSpatial Runtime Repository standalone file (.GRR). The control will use this file as its repository data. The property **UseLocalFile** must be true.

All these properties are Read/Write properties.

Note that overwriting the value of these properties will not take effect at the runtime until the **UpdateData** function is triggered by editing the property **DoUpdateData**.

5.5.2.2. GISIZE Behavior Properties

These properties are located under the **GISIZE Properties** group. They allow controlling the behavior of the application at the runtime. These properties are:

- **MapProvider**. String property that indicates the Map Repository to be visualized at the runtime.
- **Position:** Indicates in Latitude and Longitude format the location where the map provider will center. Default the value is '0,0'.
- **Rotation:** Indicates in degrees the rotation of the content on the GISIZE GEOSpatial App. By default the value is 0.
- **Zoom:** Indicates the current zoom level of the map provider.
- **CenterVisible:** Shows/hides the central cross.
- **GISLanguage**: Indicates the name of the GISIZE GEOSpatial language to be used at the runtime. By default the value is 'Default'.
- **ToolTipDelay:** Indicates in milliseconds the time that takes a tooltip to appear. Default value is 5000.

- **UserName:** Indicates the name of the current GISIZE GEOSpatial user.
- **DesignTime**. Boolean property that indicates if the GISIZE GEOSpatial App will accept Design Time Repositories or not. If the value is False, the GISIZE GEOSpatial control will work only with Runtime Repositories. By default the value is False.
- **NavigationModel**. String property that indicates the Navigation Model to be visualized at the runtime. Optional use.
- **ControlVersion**: Read only property. It indicates the installed version of GISIZE GEOSpatial 2019.
- **LicenseStatus**: Read only property. It indicates the current status of the license installed.

5.5.2.3. GISIZE Events Properties

Read only properties whose values keep updated with the result of some GISIZE GEOSpatial events that may occur at the runtime. These properties are located under the **GISIZE Events** group, and are:

- **Location_Changed_Name**: The name of the current location if any. The location is the shape where the central cross is located.
- **Marker_Clicked_Name**: the name of the last left-clicked marker at the runtime.
- **Marker_Clicked_ArchestrAName**: The ArchestrA Name of the last left-clicked marker at the runtime.
- **Marker_Clicked_ArchestrAGraphicName**: The ArchestrA Graphic Name of the last left-clicked marker at the runtime.
- **Marker_DoubleClicked_Name**: The name of the last double-clicked marker at the runtime.
- **Marker_DoubleClicked_ArchestrAName**: The ArchestrA Name of the last double-clicked marker at the runtime.
- **Marker_DoubleClicked_ArchestrAGraphicName**: The ArchestrA Graphic Name of the last double-clicked marker at the runtime.
- **Marker_Entered_Name**: The name of the last marker that the mouse is moved over at the runtime.
- **Marker_Entered_ArchestrAName**: The ArchestrA Name of the last marker that the mouse is moved over at the runtime.
- **Marker_Entered_ArchestrAGraphicName**: The ArchestrA Graphic Name of the last marker that the mouse is moved over at the runtime.
- **Marker_Left_Name**: The name of the last marker that the mouse left at the runtime.
- **Marker_Left_ArchestrAName**: The ArchestrA Name of the last marker that the mouse left at the runtime.
- **Marker_Left_ArchestrAGraphicName**: The ArchestrA Graphic Name of the last marker that the mouse left at the runtime.
- **Marker_RightClicked_Name**: The name of the last right-clicked marker at the runtime.

- **Marker_RightClicked_ArchestrAName**: The ArchestrA Name of the last right-clicked marker at the runtime.
- **Marker_RightClicked_ArchestrAGraphicName**: The ArchestrA Graphic Name of the last right-clicked marker at the runtime.
- **Marker_RightDoubleClicked_Name**: The name of the last double right-clicked marker at the runtime.
- **Marker_RightDoubleClicked_ArchestrAName**: The ArchestrA Name of the last double right-clicked marker at the runtime.
- **Marker_RightDoubleClicked_ArchestrAGraphicName**: The ArchestrA Graphic Name of the last double right-clicked marker at the runtime.
- **Shape_Clicked_Name**: the name of the last left-clicked shape at the runtime.
- **Shape_Clicked_ArchestrAName**: The ArchestrA Name of the last left-clicked shape at the runtime.
- **Shape _Clicked_ArchestrAGraphicName**: The ArchestrA Graphic of the last left-clicked shape at the runtime.
- **Shape _DoubleClicked_Name**: The name of the last double left-clicked shape at the runtime.
- **Shape _DoubleClicked_ArchestrAName**: The ArchestrA Name of the last double left-clicked shape at the runtime.
- **Shape _DoubleClicked_ArchestrAGraphicName**: The ArchestrA Graphic Name of the last double left-clicked shape at the runtime.
- **Shape _Entered_Name**: The name of the last shape that the mouse is moved over at the runtime.
- **Shape _Entered_ArchestrAName**: The ArchestrA Name of the last shape that the mouse is moved over at the runtime.
- **Shape _Entered_ArchestrAGraphicName**: The ArchestrA Graphic Name of the last shape that the mouse is moved over at the runtime.
- **Shape _Left_Name**: The name of the last shape that the mouse left at the runtime.
- **Shape _Left_ArchestrAName**: The ArchestrA Name of the last shape that the mouse left at the runtime.
- **Shape _Left_ArchestrAGraphicName**: The ArchestrA Graphic Name of the last shape that the mouse left at the runtime.
- **Shape _RightClicked_Name**: The name of the last right-clicked shape at the runtime.
- **Shape _RightClicked_ArchestrAName**: The ArchestrA Name of the last right-clicked shape at the runtime.
- **Shape _RightClicked_ArchestrAGraphicName**: The ArchestrA Graphic Name of the last right-clicked shape at the runtime.
- **Shape _RightDoubleClicked_Name**: The name of the last double right-clicked shape at the runtime.
- **Shape _RightDoubleClicked_ArchestrAName**: The ArchestrA of the last double right-clicked shape at the runtime.

- **Shape _RightDoubleClicked_ArchestrAGraphicName**: The ArchestrA Graphic Name of the last double right-clicked shape at the runtime.

5.5.2.4. GISIZE Functions Properties

These are write only properties which can trigger GISIZE GEOSpatial functions at the runtime. These properties are located under the **GISIZE functions** group, and are:

- **DoUpdateData**: Boolean value. Triggers a complete update of the data in the GISIZE GEOSpatial App. It's necessary for the Connection properties to take effect after the startup. It must be configured as Read/Write to work properly.
- **CenterItem**: String value. On data change, it centers the GISIZE GEOSpatial App over the specified item (marker or shape).
- **SetInvisibleContainer**: String value. On data change, it turns the specified container invisible.
- **SetInvisibleItem**: String value. On data change, it turns the specified item (marker or shape) invisible.
- **SetInvisibleLayer**: String value. On data change, it turns the specified layer invisible but not its sub layers.
- **SetInvisibleLayerAndChildren**: String value. On data change, it turns the specified layer and its sub layers invisible.
- **SetInvisibleMarker**: String value. On data change, it turns the specified marker invisible.
- **SetInvisibleShape**: String value. On data change, it turns the specified shape invisible.
- **SetVisibleContainer**: String value. On data change, it turns the specified container visible.
- **SetVisibleItem**: String value. On data change, it turns the specified item (marker or shape) visible.
- **SetVisibleLayer**: String value. On data change, it turns the specified layer visible but not its sub layers.
- **SetVisibleLayerAndChildren**: String value. On data change, it turns the specified layer and its sub layers visible.
- **SetVisibleMarker**: String value. On data change, it turns the specified marker visible.
- **SetVisibleShape**: String value. On data change, it turns the specified shape visible.
- **AboutControlVisible**: Boolean value. On data change, it shows or hides the About Control of the GISIZE GEOSpatial App.
- **ConfigPanelVisible**: Boolean value. On data change, it shows or hides the Configuration Panel of the GISIZE GEOSpatial App.
- **ContainersPanelVisible**: Boolean value. On data change, it shows or hides the Containers Panel of the GISIZE GEOSpatial App.
- **LayersPanelVisible**: Boolean value. On data change, it shows or hides the Layers Panel of the GISIZE GEOSpatial App.

- **MovePadMobileVisible**: Boolean value. On data change, it shows or hides the mobile Move Pad control of the GISIZE GEOSpatial App.
- **MovePadVisible**: Boolean value. On data change, it shows or hides the Move Pad control of the GISIZE GEOSpatial App.
- **NavigationModelPanelVisible**: Boolean value. On data change, it shows or hides the Navigation Model Panel of the GISIZE GEOSpatial App.
- **SearchFieldVisible**: Boolean value. On data change, it shows or hides the Search Field of the GISIZE GEOSpatial App.
- **TopBarMaximized**: Boolean value. On data change, it expands or collapses the Top Bar of the GISIZE GEOSpatial App.
- **TopBarVisible**: Boolean value. On data change, it shows or hides the Top Bar of the GISIZE GEOSpatial App.
- **UILeftVisible**: Boolean value. On data change, it shows or hides the whole left-sided User Interface of the GISIZE GEOSpatial App.
- **UIRightVisible**: Boolean value. On data change, it shows or hides the whole right-sided User Interface of the GISIZE GEOSpatial App.
- **ZoomBarVisible**: Boolean value. On data change, it shows or hides the vertical Zoom Bar of the GISIZE GEOSpatial App.
- **ZoomButtonsVisible**: Boolean value. On data change, it shows or hides the Zoom Buttons of the GISIZE GEOSpatial App.
- **ZoomButtonsMobileVisible**: Boolean value. On data change, it shows or hides the mobile Zoom Buttons of the GISIZE GEOSpatial App.

5.6. Lab 18 - Integrating GISIZE GEOSpatial App in InTouch OMI

In this lab, you will integrate GISIZE GEOSpatial in InTouch OMI.

5.6.1. Objectives

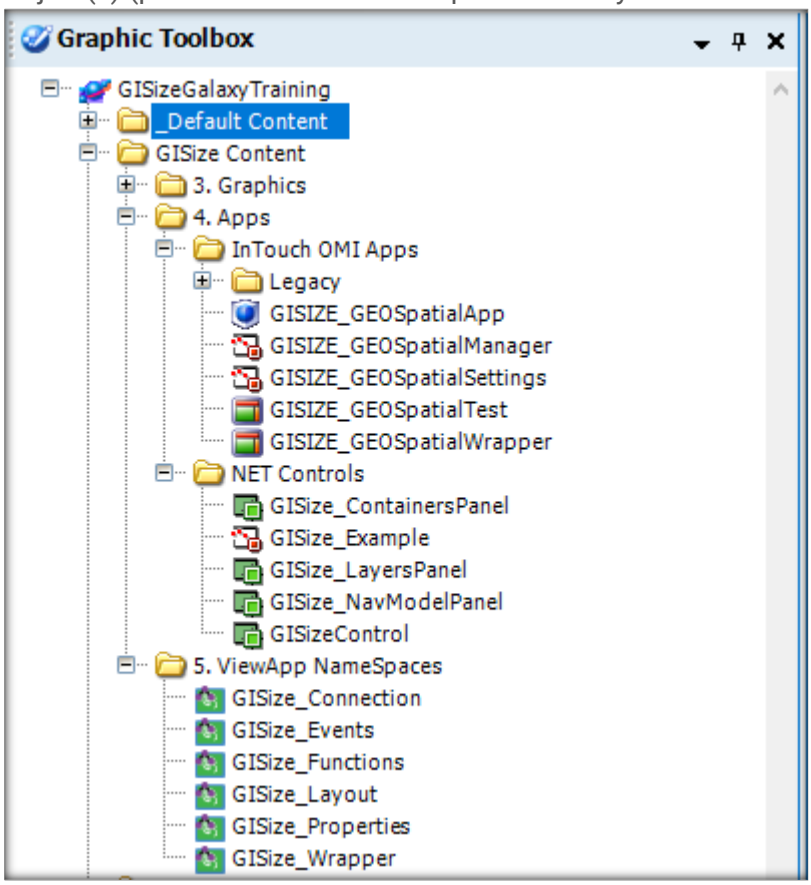
Upon completion of this lab, you will be able to:

- Import the GISIZE GEOSpatial ArchestrA App.
- Use the **GISIZE_GEOSpatial_Manager** ArchestrA Graphic to configure and manage the GISIZE GEOSpatial App.
- Use ArchestrA Graphics in the GISIZE GEOSpatial App.

5.6.2. Import and configure the GISIZE GEOSpatial ArchestrA App

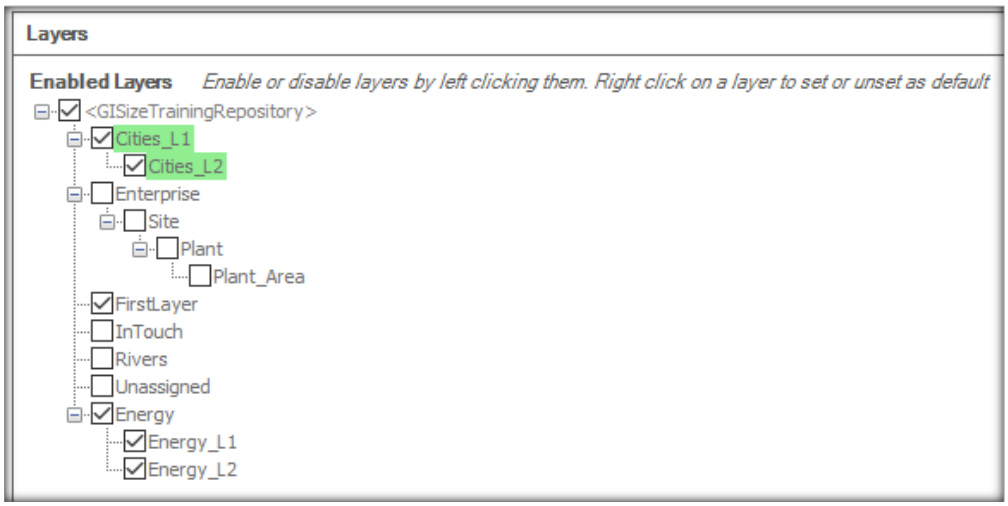
In the following steps, you will import and configure the GISIZE GEOSpatial ArchestrA App.

14. Import the ArchestrA package file **Apps\InTouch OMI Apps\ GISIZE InTouch OMI Apps.aaPKG** that can be found in the **Program Files/GISIZE** path of the computer, and must be imported from ArchestrA IDE with the option Galaxy -> Import -> Object(s) (please note that this step was already done in lab 2 if you did it).



15. Import the ArchestrA packages that can be found in the **Lab 18 - Integrating GISIZE App in InTouch OMI** folder located in your training files.

16. At the OpenStreetMap Map Provider disable the **InTouch** layer and enable by default the layers **Cities_L1** and **Cities_L2**.

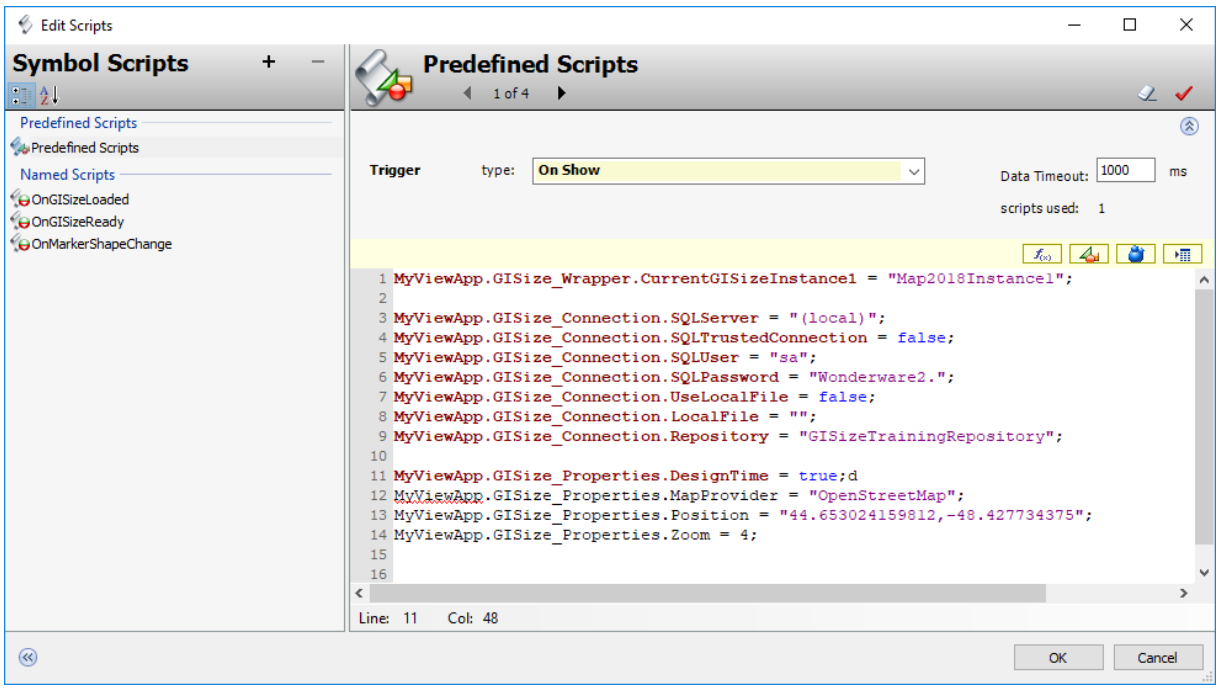


17. Open the **GISIZE_GEOSpatialManager** ArchestrA Graphic and verify at least the value of the following properties:

    a. MyViewApp.GISIZE GEOSpatial_Connection.Repository: **GISIZETrainingRepository**

    b. MyViewApp.GISIZE GEOSpatial_Connection.SQLTrustedConnection: **True**

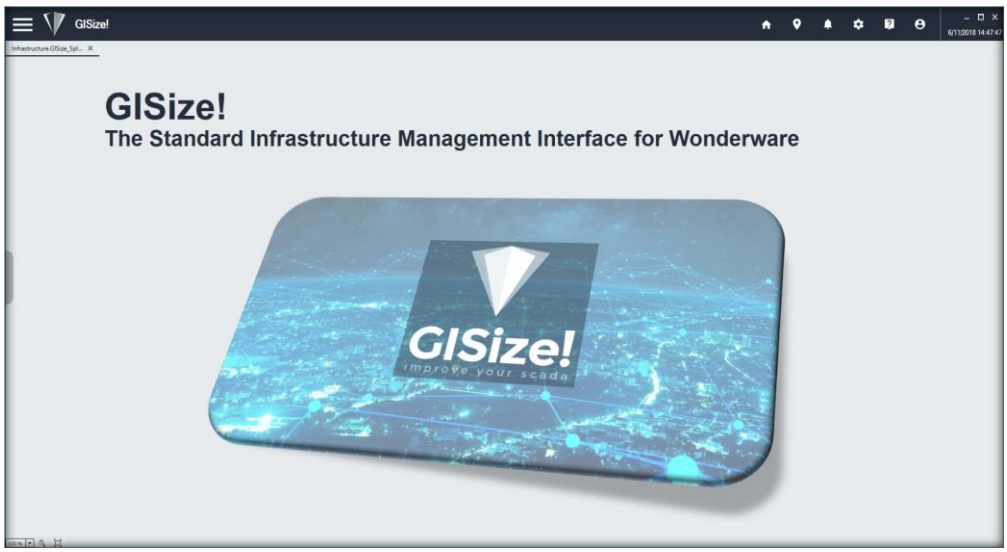
    c. MyViewApp.GISIZE GEOSpatial_Properties.DesignTime: **True**



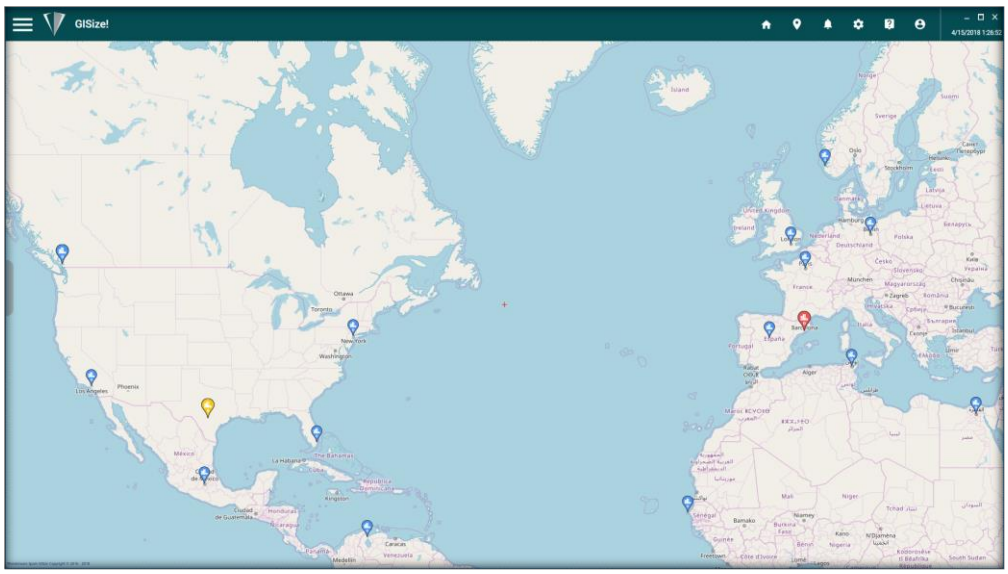
5.6.3. Test View App

In the following steps, you will test GISIZE GEOSpatial embedded in a View App.

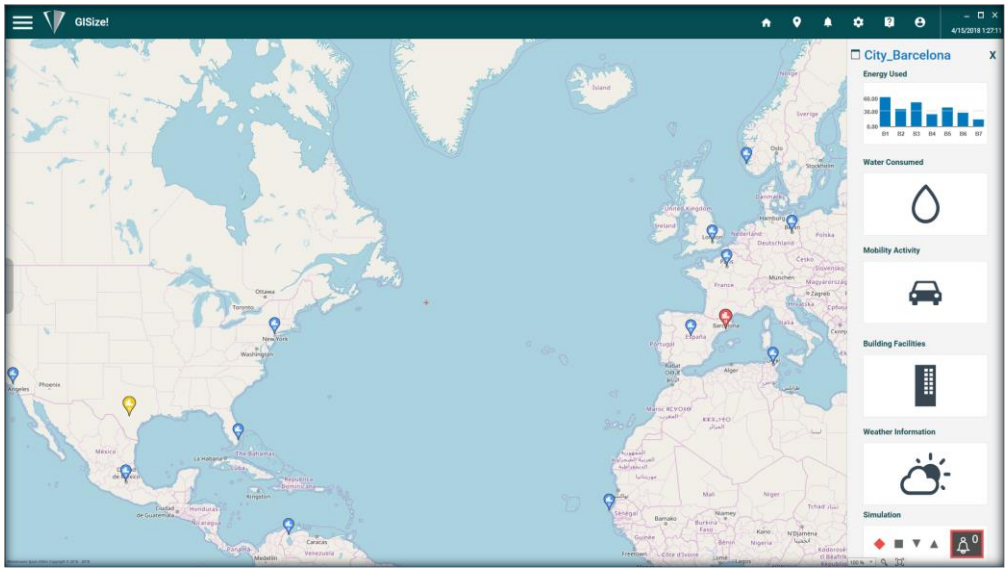
1. Deploy the **GISIZE_ViewApp_001.**
2. Launch the ViewApp **GISIZE_ViewApp_001**.



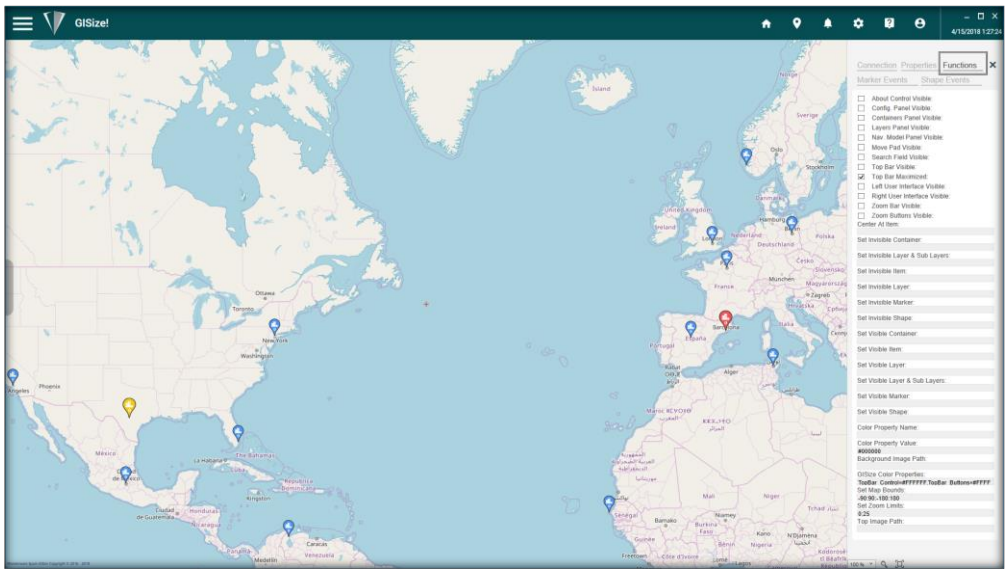
3. Click the **Map** button to open GISIZE GEOSpatial ArchestrA App embedded in **GISIZE_Wrapper** Layout.

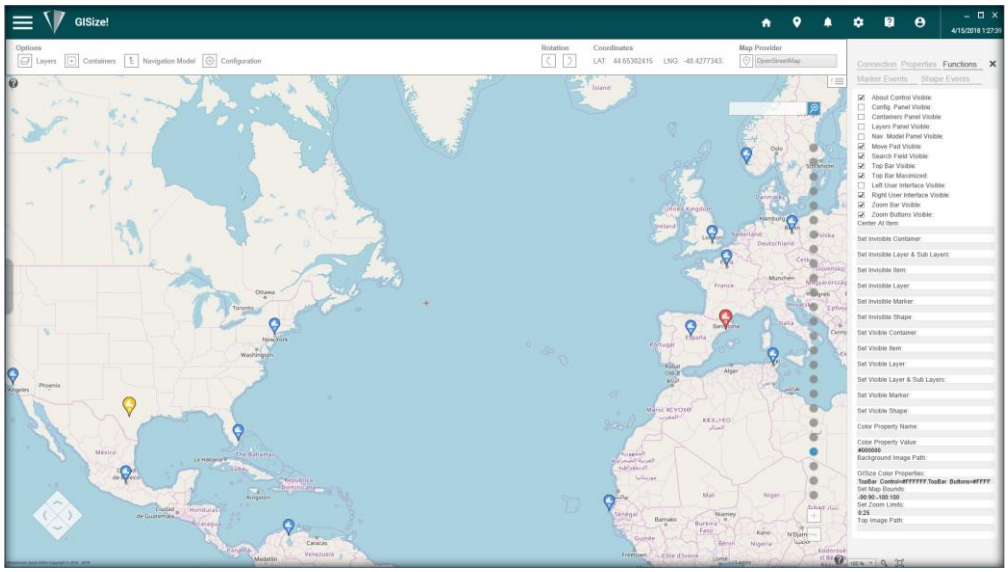


4. Test the behavior of the application clicking the **Markers** and **Shapes**.



5. Click the **Help** button to interact with the ArchestrA App using the View App System Tags (Connection, Properties, Functions and Events).

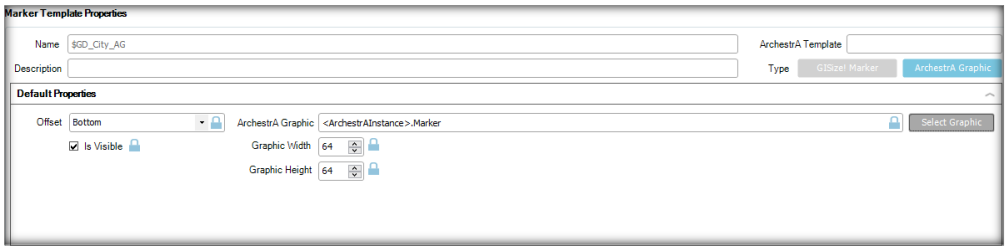




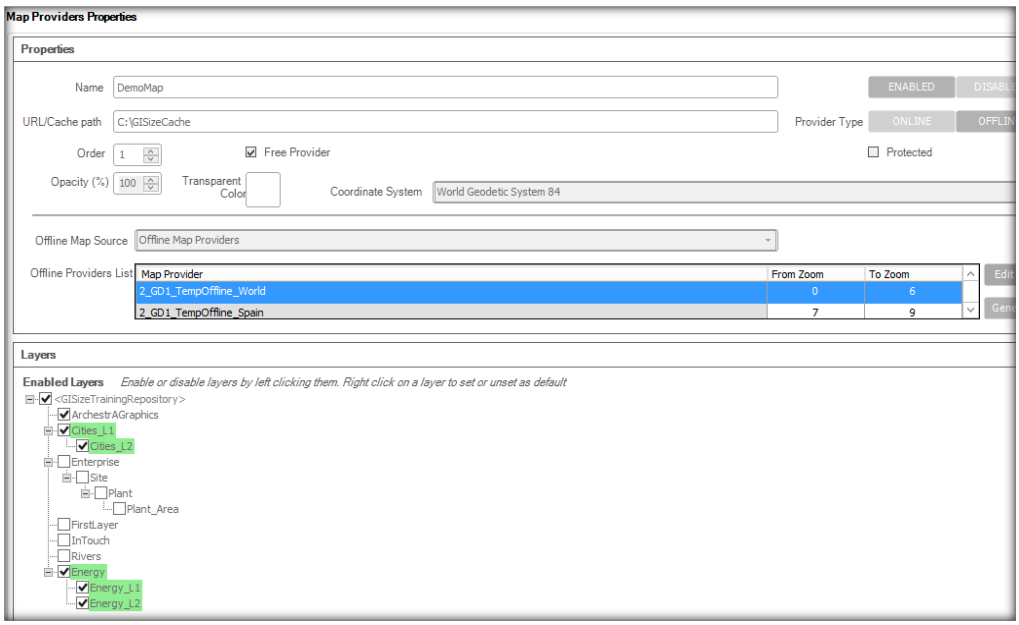
5.6.4. Use ArchestrA Graphics in the GISIZE GEOSpatial Makers

In the following steps, you will configure new City markers to use ArchestrA Graphics.

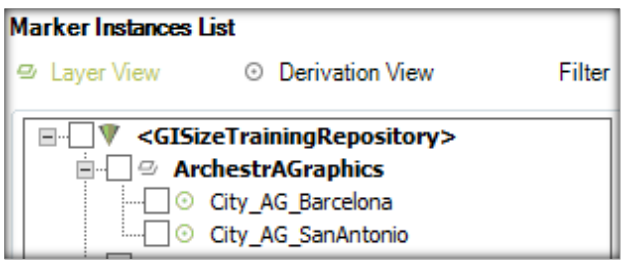
1. Duplicate the **$GD_City** Marker Template and rename it to **$GD_City_AG**.
2. Edit the **$GD_City_AG** Marker Template.
3. Click **ArchestrA Graphics** button to use an ArchestrA Graphics for the marker.
4. Enter **<ArchestrAInstance>.Marker** in the field **ArchestrA Graphic**.
5. Enter **64** in the **Graphic Width** and **Graphic Height** fields.



6. Create a new Layer named **ArchestrAGraphics.**
7. Edit the **DemoMap** Map Provider, and enable the **ArchestrAGraphics** layer.

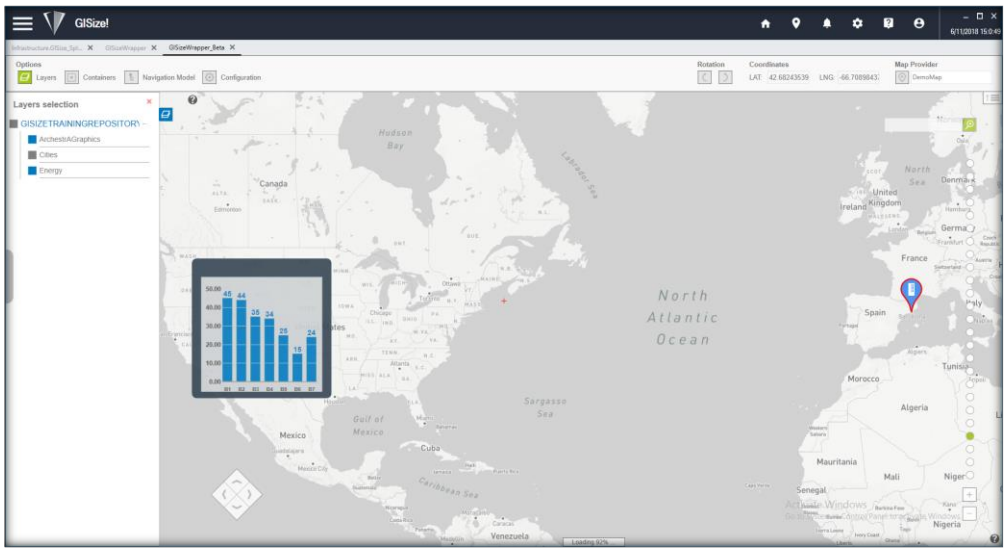


8. Go to **Marker Instances** section and import the assets located in the file MarkerInstances_CitiesAG.xml (folder **Lab 18 - Integrating GISIZE App in InTouch OMI**).



9. Deploy the **GISIZE_ViewApp_001.**

10. Launch the ViewApp **GISIZE_ViewApp_001** and enable the ArchestrA Graphics layer.



6. Multi Language and Security

The objectives of this module are:

- Explain how to translate and use GISIZE GEOSpatial in different languages.
- Explain how to configure the GISIZE GEOSpatial security and create systems defining which layers and instances are available for the different roles and users.

6.1. Multi Language

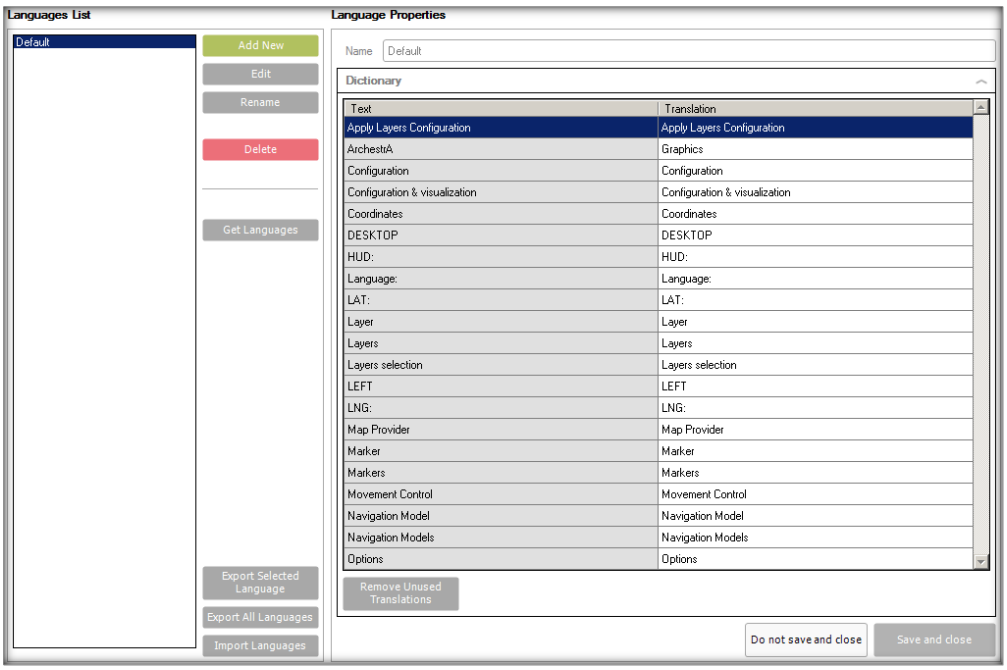
This section explains how to translate and use GISIZE GEOSpatial in different languages.

6.1.1. Introduction

The Multilanguage panel allows the definition of languages and the configuration of the existing ones.

Conceptually a GISIZE GEOSpatial language is a dictionary, a set of texts that will replace the default texts of GISIZE GEOSpatial at the runtime. The replaceable text comprehends all the texts that are potentially viewable at the runtime, including static texts of the instances.

The current visualized language is a writable property of the GISIZE GEOSpatial control.



The panel has two parts. On the left side of the panel, there is a list of currently defined languages and a set of buttons to perform operations. On the right side of the panel there is a collection of controls to modify the properties of a selected language.

6.1.2. Languages List

The Language List shows the currently available languages. Note that a 'Default' language exists and cannot be deleted. At the runtime, if no language is indicated, the 'Default' language is loaded.

Upon the selection of a language of the tree, a set of operations can be performed through the buttons next to it. From top to bottom these buttons are:

- **Add New:** This button allows creating a new language. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The minimum requirement to add a new language is a unique name. By default, a complete set of the dictionary will be filled.

- **Edit:** This button allows editing the properties of a previously existing language. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The name of the language being edited cannot be changed with this functionality.

- **Rename:** This button allows changing the name of a selected language. Upon clicking it a popup window will appear to type the new desired name. Duplicated names are not allowed. The 'Default' language name cannot be renamed.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

- **Delete:** Clicking this button will delete the selected language with all its properties. Before deleting a language, the user will be asked for confirmation. This operation cannot be undone.

- **Get Languages:** Clicking this button will refresh the list of available languages. Usually there is no need to use this button because after performing any operation locally, the list is refreshed automatically.

- **Export Selected Language:** This button allows exporting a selected language with all its properties to a XML file. This is a way to backup languages to be imported later. The file can be edited externally with any text editor.

- **Export All Languages:** This button allows exporting all the existing languages in a repository with all their properties into a single XML file.

  This is a way to back up all languages of a repository to be imported later. The file can be edited externally with any text editor.

- **Import Languages:** This button allows importing an XML file with previously exported languages. Upon clicking it, the user will be asked to select an XML file within an explorer window. After selecting an XML file, the user will have to choose the way to import the file if a language with the same name is found.

  The import preferences are:

  - **Skip – Do not import**: If an existing language has the same name as an imported one, the imported one will be ignored and skipped from the importing operation. The existing language will remain unaffected.
  - **Rename importing item (numeric append)**: If an existing language has the same name as an imported one, the one being imported will be renamed appending an incremental number at the end of its name. The existing language will remain unaffected.
  - **Overwrite existing item**: If an existing language has the same name as an imported one, the existing one will be overwritten with the one being imported. Configuration of the previously existing language will be lost.

6.1.3. Language Properties

The right side of the Language panel contains the property controls of the language. Upon the selection of a shape instance, the property controls will be updated. These controls will remain disabled unless a shape instance is being edited or created.

To accept any modification made during the editing or the creation of a language, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a language, user must click the **Do not save and close** button. Clicking this button while a new language is being created will abort the process. In both cases the ongoing process will come to an end.

Language properties are just the name and the dictionary list:

- **Name**: The unique identifier name of the language. There cannot be two languages with the same name. This control is only enabled during the process of creating a new language. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Name cannot be left empty. The 'Default' name is reserved.
- **Dictionary**: This is a table with two columns that contains the translation texts to be used at the runtime. The left column is read only, and contains the list of the runtime texts that can be replaced. The right column is editable and contains the texts that will replace the left ones at the runtime. By default, both columns contain the same texts. The number of replaceable texts will grow automatically when new texts are added to the repository with the instances.

Additionally, there is the **Remove Unused Translations** button that allows the cleaning the dictionary of unused texts at the runtime.

6.2. Lab 19 - Creating a new language

6.2.1. Introduction

In this lab, you will create a new language and translate some texts.

6.2.2. Objectives

Upon completion of this lab, you will be able to:

- Create a new language.
- Translate texts.
- Change language at the runtime.

6.2.3. Create a new language and translate some texts

In the following steps, you will create a new language and translate some texts.

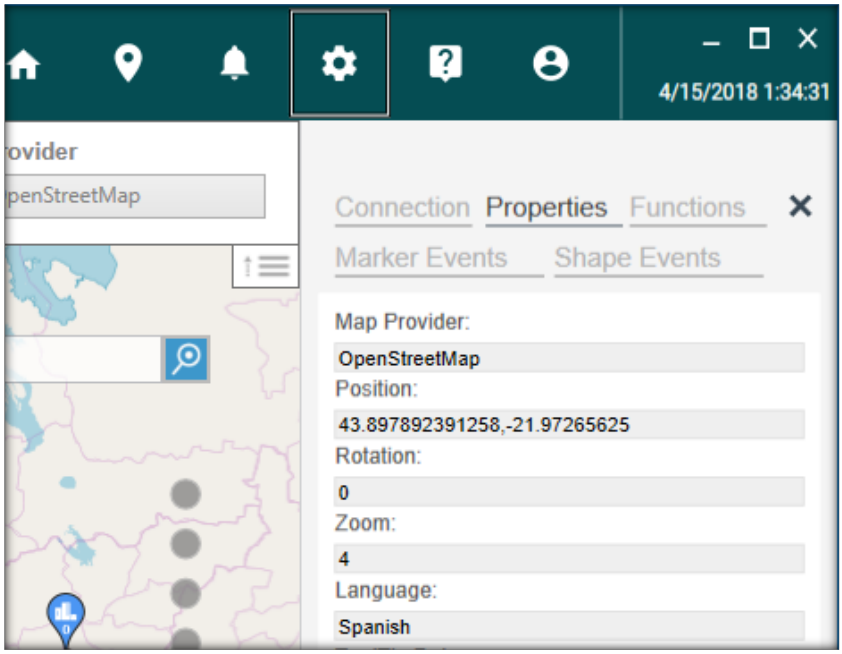
1. Double click the **Multilanguage** button.
2. Click the **Add New** button to create a new Language.
3. Enter **Spanish** in the **New Language Name** field.
4. Click the **Edit** button to translate some texts.
5. Translate the following texts:
    - **Layer**: Capa
    - **Layers**: Capas
    - **Layer Selection**: Selección de capas
    - **Cities**: Ciudades

6. Click the **Save and Close** button.

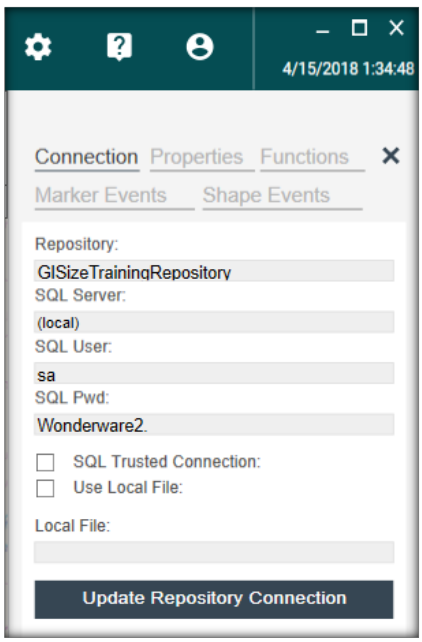
6.2.4. Test View App

In the following steps, you will test GISIZE GEOSpatial embedded in a View App.

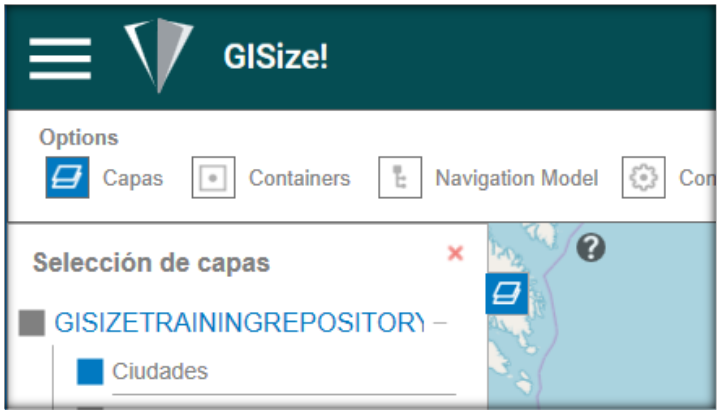
1. Open **View App**
2. Click the **Map** button to open the GISIZE GEOSpatial App.
3. Click the **Settings** button.
4. Select the **Properties** panel and change **Language** field to **Spanish.**



5. Select the **Connection** panel and click the **Update Repository Connection** button.



6. Check how the texts are translated in the Layers panel.



6.3. Security

This section explains how to configure the GISIZE GEOSpatial security and create systems defining which layers and instances are available for the different roles and users.

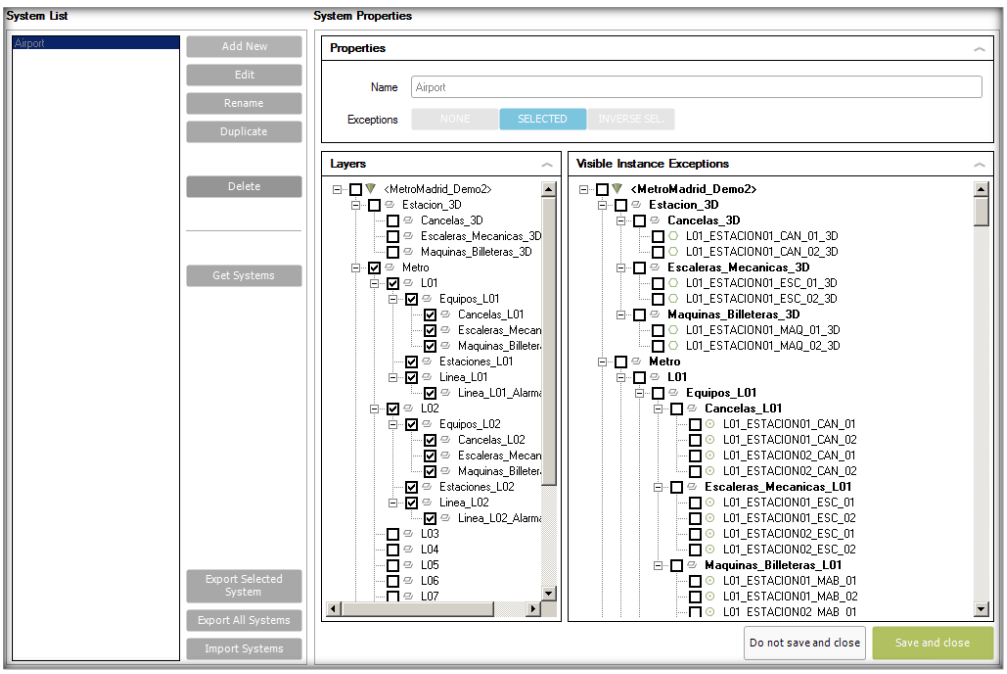
6.3.1. Systems

The Security System panel allows the definition of new security systems and the configuration of the existing ones.

A security system is a group of layers and instance exceptions that will be available at the runtime. Security systems used along with security roles and security users allows defining witch layers and instances are available to which users.

GISIZE GEOSpatial security is effective at the repository level. That means each repository has its own security configuration. Security is enabled or disabled at the Administration panel of each repository.

By default, a GISIZE GEOSpatial repository comes without security and no security system defined.



The panel is divided into left and right sides. On the left side of the panel there is a list of currently available security systems ordered alphabetically. On the right side of the panel there is a collection of controls to modify the properties of a selected security system.

6.3.1.1. System List

The security system list shows the currently available systems sorted alphabetically. Upon the selection of a system of the list, a set of operations can be performed through the buttons next to it. From top to bottom these buttons are:

- **Add New:** This button allows creating a security system. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The minimum requirement to add a new system is a unique name.

- **Edit:** This button allows editing the properties of a previously existing security system. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The name of the system being edited cannot be changed with this functionality.

- **Rename:** This button allows changing the name of a selected security system. Upon clicking it a popup window will appear to type the new desired name. Duplicated names are not allowed.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

- **Duplicate:** This button will create a copy of the selected security system. The resulting system will be named as the original system name adding the suffix '_Copy'. If a security system already exists with that name, an extra incremental numeric suffix will be added.

- **Delete:** Clicking this button will delete the selected security system with all its properties. Before deleting a security system, the user will be asked for confirmation. This operation cannot be undone.

- **Get Systems:** Clicking this button will refresh the list of available security systems. Usually there is no need to use this button because after performing any operation locally, the list is refreshed automatically.

- **Export Selected System:** This button allows exporting the currently selected security system with all its properties to a XML file. This is a way to back up security systems to be imported later. The file can be edited externally with any text editor.

- **Export All Systems:** This button allows exporting all the existing security systems in a repository with all their properties into a single XML file.

  This is a way to back up all the security systems of a repository to be imported later. The file can be edited externally with any text editor.

- **Import Systems:** This button allows importing an XML file with previously exported security systems. Upon clicking it, the user will be asked to select an XML file within an explorer window. After selecting an XML file, the user will have to choose the way to import the file if a system with the same name is found.

  The import preferences are:

  - **Skip – Do not import**: If an existing system has the same name as an imported one, the imported one will be ignored and skipped from the import operation. The existing system will remain unaffected.
  - **Rename importing item (numeric append)**: If an existing system has the same name as an imported one, the one being imported will be renamed appending an incremental number at the end of its name. The existing system will remain unaffected.
  - **Overwrite existing item**: If an existing system has the same name as an imported one, the existing one will be overwritten with the one being imported. Configuration of the previously existing system will be lost.

6.3.1.2. System Properties

The right side of the Security System panel contains the property controls of the systems. Upon the selection of a system, the property controls will be updated. These controls will remain disabled unless a system is being edited or created.

To accept any modification made during the editing or the creation of a system, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a system, user must click the **Do not save and close** button. Clicking this button while a new system is being created will abort the process. In both cases the ongoing process will come to an end.

The available security system's properties can be classified in three groups: Main properties, Layers and Visible Instance Exceptions.

6.3.1.3. System Main Properties

These are the common properties of security system, and its controls are located at the top-right side of the panel.

From top to bottom and left to right, the properties are:

- **Name**: The unique identifier name of the security system. There cannot be two security systems with the same name. This control is only enabled during the process of creating a new system. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Name cannot be left empty.
- **Exception**: A set of 3 exclusive buttons to indicate if the system doesn't have instance exceptions (NONE), if the instance exceptions are the ones indicated below (SELECTED), or if the instance exceptions are the ones NOT indicated below (INVERSER SEL.) By default, the selected button is NONE.

6.3.1.4. System Layers

This is a tree view control that allows the selection of the layers that will be available for the current security system. All the layers of the repository are selectable independent of the map provider.

6.3.1.5. System Visible Instance Exceptions

This is a tree view control that allows the selection of the marker/shape instances that will be available for the current security system, even if the layer which it belongs to is unavailable.

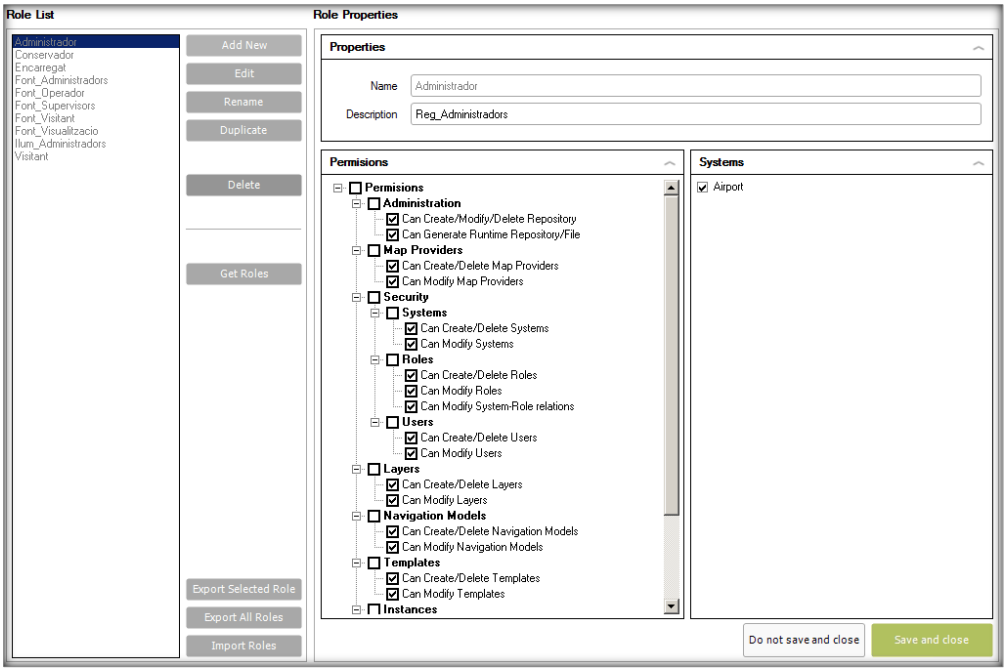
6.3.2. Roles

The Security Roles panel allows the definition of new security roles and the configuration of the existing ones.

A security role is a combination of application permissions and security systems that defines the user's capacity to interact with the repository at the design time and the runtime.

GISIZE GEOSpatial security is effective at repository level. That means that each repository has its own security configuration. Security is enabled or disabled at the Administration panel of each repository.

By default, a GISIZE GEOSpatial repository comes without security but with a read-only 'Administrator' security role defined.



The panel is divided into left and right sides. On the left side of the panel there is a list of currently available security roles ordered alphabetically. On the right side of the panel there is a collection of controls to modify the properties of a selected security role.

6.3.2.1. Role List

The security role list shows the currently available systems sorted alphabetically. Upon the selection of a role of the list, a set of operations can be performed through the buttons next to it. From top to bottom these buttons are:

- **Add New:** This button allows creating a new security role. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The minimum requirement to add a new role is a unique name.

- **Edit:** This button allows editing the properties of a previously existing security role. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The name of the role being edited cannot be changed with this functionality.

- **Rename:** This button allows changing the name of a selected security role. Upon clicking it a popup window will appear to type the new desired name. Duplicated names are not allowed.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

- **Duplicate:** This button will create a copy of the selected security role. The resulting role will be named as the original role name adding the suffix '_Copy'. If a security role already exists with that name, an extra incremental numeric suffix will be added.

- **Delete:** Clicking this button will delete the selected security role with all its properties. Before deleting a security role, the user will be asked for confirmation. This operation cannot be undone.
- **Get Roles:** Clicking this button will refresh the list of available security roles. Usually there is no need to use this button because after performing any operation locally, the list is refreshed automatically.
- **Export Selected Role:** This button allows exporting the currently selected security roles with all its properties to a XML file. This is a way to back up security roles to be imported later. The file can be edited externally with any text editor.
- **Export All Roles:** This button allows exporting all the existing security roles in a repository with all their properties into a single XML file.

  This is a way to back up all the security roles of a repository to be imported later. The file can be edited externally with any text editor.
- **Import Roles:** This button allows importing an XML file with previously exported security roles. Upon clicking it, the user will be asked to select an XML file within an explorer window. After selecting an XML file the user will have to choose the way to import the file if a role with the same name is found.

  The import preferences are:
  - **Skip – Do not import**: If an existing role has the same name as an imported one, the imported one will be ignored and skipped from the importing operation. The existing role will remain unaffected.
  - **Rename importing item (numeric append)**: If an existing role has the same name as an imported one, the one being imported will be renamed appending an incremental number at the end of its name. The existing role will remain unaffected.
  - **Overwrite existing item**: If an existing role has the same name as an imported one, the existing one will be overwritten with the one being imported. Configuration of the previously existing role will be lost.

6.3.2.2. Role Properties

The right side of the Security Roles panel contains the property controls of the roles. Upon the selection of a role, the property controls will be updated. These controls will remain disabled unless a role is being edited or created.

To accept any modification made during the editing or the creation of a role, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a role, user must click the **Do not save and close** button. Clicking this button while a new role is being created will abort the process. In both cases the ongoing process will come to an end.

The available security role's properties can be classified in three groups: Main properties, Permissions, and Systems.

6.3.2.3. Role Main Properties

These are the common properties of security role, and their controls are located at the top-right side of the panel.

From top to bottom and left to right, the properties are:

- **Name**: The unique identifier name of the security role. There cannot be two security roles with the same name. This control is only enabled during the process of creating a new role. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Name cannot be left empty.
- **Description**: A textbox field to type a description of the role. This property plays no role at the runtime, but as any other property, it is accessible by scripting for integrators. It can be left empty.

6.3.2.4. Role Permissions

This is a tree view control that allows the selection of the available design time permissions at the current security role. Existing permissions are just selectable, not editable. Permissions are self-explanatory and are:

- Administration – Can Create/Modify/Delete Repository
- Administration – Can Generate Runtime Repository/File
- Map Providers – Can Create/Delete Map Providers
- Map Providers – Can Modify Map Providers
- Security Systems – Can Create/Delete Systems
- Security Systems – Can Modify Systems
- Roles – Can Create/Delete Roles
- Security Roles – Can Modify Roles
- Security Roles – Can Modify 'System-Role' relations
- Security Users – Can Create/Delete Users
- Security Users – Can Modify Users
- Layers – Can Create/Delete Layers
- Layers – Can Modify Layers
- Navigation Models – Can Create/Delete Navigation Models
- Navigation Models – Can Modify Navigation Models
- Templates – Can Create/Delete Templates
- Templates – Can Modify Templates
- Instances – Can Create/Delete Instances
- Instances – Can Modify Instances
- MultiLanguage – Can Create/Delete Languages
- MultiLanguage – Can Modify Languages
- Can Start Runtime Preview.

A user with this role will have the availability to perform the actions that require the selected permissions at the design time.

6.3.2.5. Role Systems

This list allows the selection of the available security systems for the current security role. A user with this role will have access to the layers and instance exceptions defined at the selected systems at the runtime.

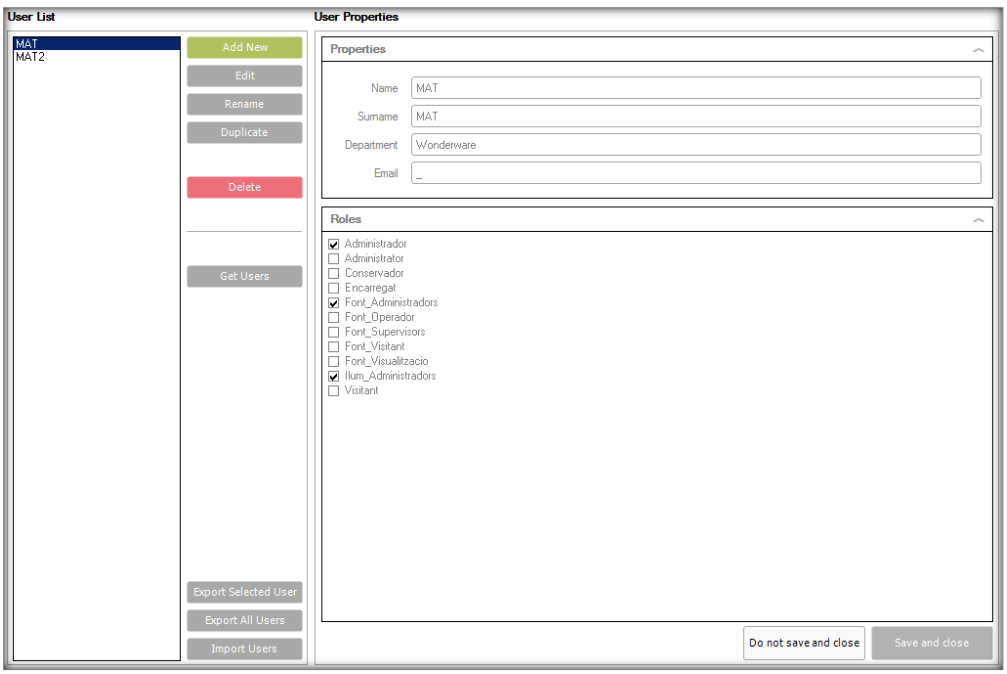
6.3.3. Users

The Security Users panel allows the definition of new security users and the configuration of the existing ones.

A security user is a GISIZE GEOSpatial user with one or more security roles associated. The user will have sum of all the permissions and security systems of the assigned roles that defines the user's capacity to interact with the repository at the design time and the runtime.

GISIZE GEOSpatial security is effective at the repository level. That means that each repository has its own security configuration. Security is enabled or disabled at the Administration panel of each repository.

By default, a GISIZE GEOSpatial repository comes without security and no security user defined.



The panel is divided into left and right sides. On the left side of the panel there is a list of currently available security users ordered alphabetically. On the right side of the panel there is a collection of controls to modify the properties of a selected security user.

6.3.3.1. User List

The security user list shows the current available users sorted alphabetically. Upon the selection of a user of the list, a set of operations can be performed through the buttons next to it. From top to bottom these buttons are:

- **Add New:** This button allows creating a new security user. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The minimum requirement to add a new user is a unique name.

- **Edit:** This button allows editing the properties of a previously existing security user. Clicking it disables the left side of the panel and enables the right side where the property controls for editing are located.

  The name of the user being edited cannot be changed with this functionality.

- **Rename:** This button allows changing the name of a selected security user. Upon clicking it a popup window will appear to type the new desired name. Duplicated names are not allowed.

  Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name.

- **Duplicate:** This button will create a copy of the selected security user. The resulting user will be named as the original user name adding the suffix '_Copy'. If a security user already exists with that name, an extra incremental numeric suffix will be added.

- **Delete:** Clicking this button will delete the selected security user with all its properties. Before deleting a security user, the user will be asked for confirmation. This operation cannot be undone.

- **Get Users:** Clicking this button will refresh the list of available security users. Usually there is no need to use this button because after performing any operation locally, the list is refreshed automatically.

- **Export Selected User:** This button allows exporting the currently selected security user with all its properties to a XML file. This is a way to back up security users to be imported later. The file can be edited externally with any text editor.

- **Export All Users:** This button allows exporting all the existing security users in a repository with all their properties into a single XML file.

  This is a way to back up all the security users of a repository to be imported later. The file can be edited externally with any text editor.

- **Import Users:** This button allows importing an XML file with previously exported security users. Upon clicking on it, the user will be asked to select an XML file within an explorer window. After selecting an XML file the user will have to choose the way to import the file if a user with the same name is found.

  The import preferences are:

  - **Skip – Do not import**: If an existing user has the same name as an imported one, the imported one will be ignored and skipped from the importing operation. The existing user will remain unaffected.
  - **Rename importing item (numeric append)**: If an existing user has the same name as an imported one, the one being imported will be renamed appending an incremental number at the end of its name. The existing user will remain unaffected.

- **Overwrite existing item**: If an existing user has the same name as an imported one, the existing one will be overwritten with the one being imported. Configuration of the previously existing user will be lost.

6.3.3.2. User Properties

The right side of the Security User panel contains the property controls of the users. Upon the selection of a role, the property controls will be updated. These controls will remain disabled unless a user is being edited or created.

To accept any modification made during the editing or the creation of a security user, user must click the green **Save and close** button. It will also finish the editing or creation process.

To cancel any modification made during the editing of a security user, user must click the **Do not save and close** button. Clicking this button while a new user is being created will abort the process. In both cases the ongoing process will come to an end.

The available security user's properties can be classified in two groups: Main properties and Roles.

User Main Properties

These are the common properties of security user, and their controls are located at the top-right side of the panel.

From top to bottom and left to right, the properties are:

- **Name**: The unique identifier name of the security user. There cannot be two security users with the same name. This control is only enabled during the process of creating a new user. Note that only numbers, letters, and the underscore character '_' are valid characters for a name. Invalid characters will be removed from the resultant name. Name cannot be left empty.
- **Surname**: A textbox field to type a surname of the security user. This property plays no role at the runtime, but as any other property, it is accessible by scripting for integrators. It can be left empty.
- **Department**: A textbox field to type a department of the security user. This property plays no role at the runtime, but as any other property, it is accessible by scripting for integrators. It can be left empty.
- **Email**: A textbox field to type an email address of the security user. This property plays no role at the runtime, but as any other property, it is accessible by scripting for integrators. It can be left empty.

6.3.3.3. User Roles

This list allows the selection of the available security roles for the current security user. The user will have the sum of the permissions and security systems available in the selected roles.

6.4. Lab 20 - Configuring security

6.4.1. Introduction

In this lab, you will activate and configure the GISIZE GEOSpatial security.

6.4.2. Objectives

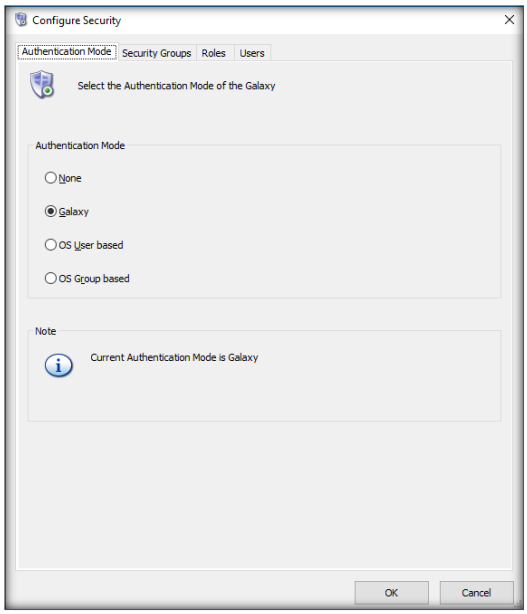
Upon completion of this lab, you will be able to:

- Activate GISIZE GEOSpatial security.
- Create and configure Systems.
- Create and configure Roles.
- Create and configure Users.
- Change User Name at the runtime.

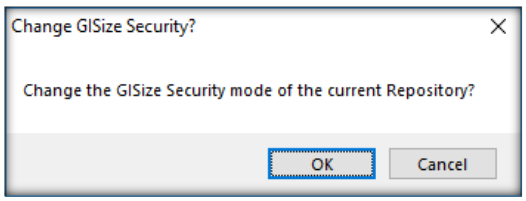
6.4.3. Activate security

In the following steps, you will activate security in Wonderware and GISIZE GEOSpatial.

1. In ArchestrA IDE open the **Configure Security** dialog box (Galaxy -> Configure –> Security).
2. Select **Galaxy** as Authentication Mode.



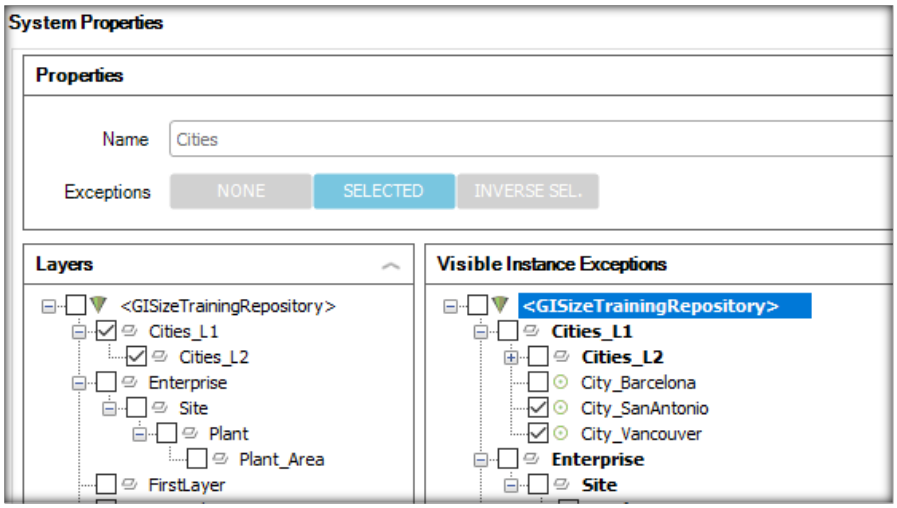
3. Click **OK** and restart the IDE.
4. Double click the **Administration** button.
5. Click the **GISIZE GEOSpatial** button in **User Security** section
6. Click **OK** button in the **Change GISIZE GEOSpatial Security?** dialog box.



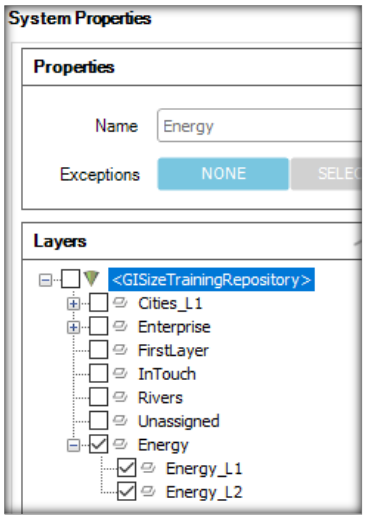
6.4.4. Configure Systems

In the following steps, you will create and configure Systems.

1. Expand **Security** in the tree view and double click the **Systems** button.
2. Click the **Add New** button to create a new System.
3. Enter **Cities** in the **Name** field.
4. Check **Cities_L1** and **Cities_L2** in the **Layers** tree view.
5. Check **City_SanAntonio** and **City_Vancouver** as Instance Exceptions.
6. Click the **Selected** button to exclude this instances from this system.



7. Click the **Save and Close** button.
8. Click the **Add New** button to create a new System.
9. Enter **Energy** in the **Name** field.
10. Check **Energy**, **Energy_L1** and **Energy_L2** in the **Layers** tree view.

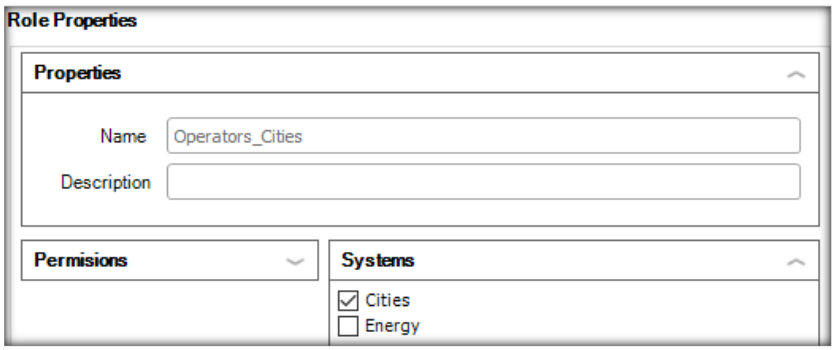


11. Click the **Save and Close** button.

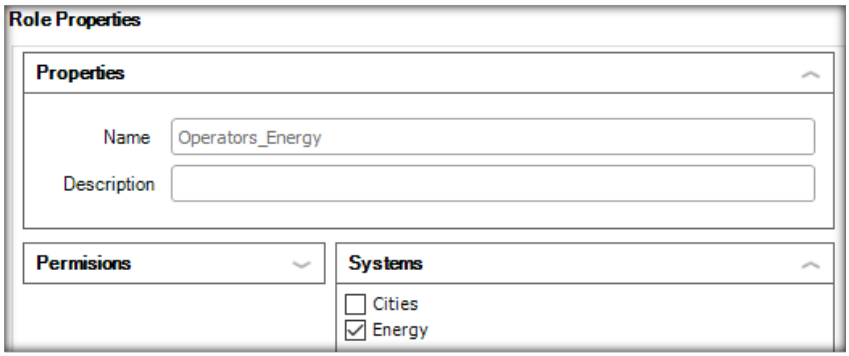
6.4.5. Configure Roles and Users

In the following steps, you will create and configure Roles and Users.

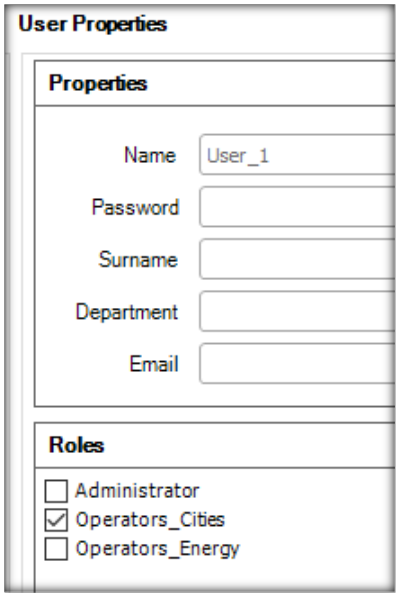
1. Double click the **Roles** button.
2. Click the **Add New** button to create a new role.
3. Enter **Operators_Cities** in the **Name** field.
4. Check **Cities** in the existing systems.



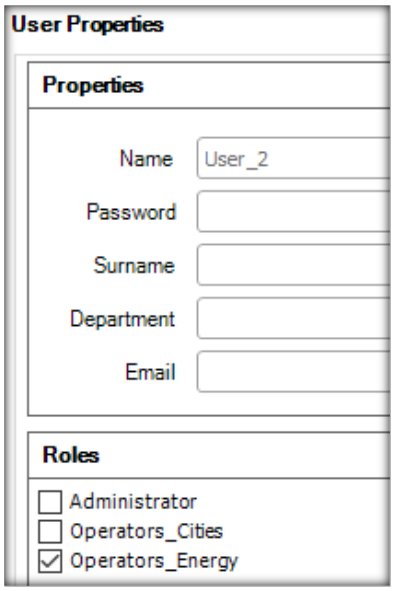
5. Click the **Save and Close** button.
6. Click the **Add New** button to create a new role.
7. Enter **Operators_Energy** in the **Name** field.
8. Check **Energy** in the existing systems.



9. Click the **Save and Close** button.
10. Double click the **Users** button.
11. Click the **Add New** button to create a new user.
12. Enter **User_1** in the **Name** field.
13. Check **Operators_Cities** in the existing roles.



14. Click the **Save and Close** button.
15. Click the **Add New** button to create a new user.
16. Enter **User_2** in the **Name** field.
17. Check **Operators_Energy** in the existing roles.

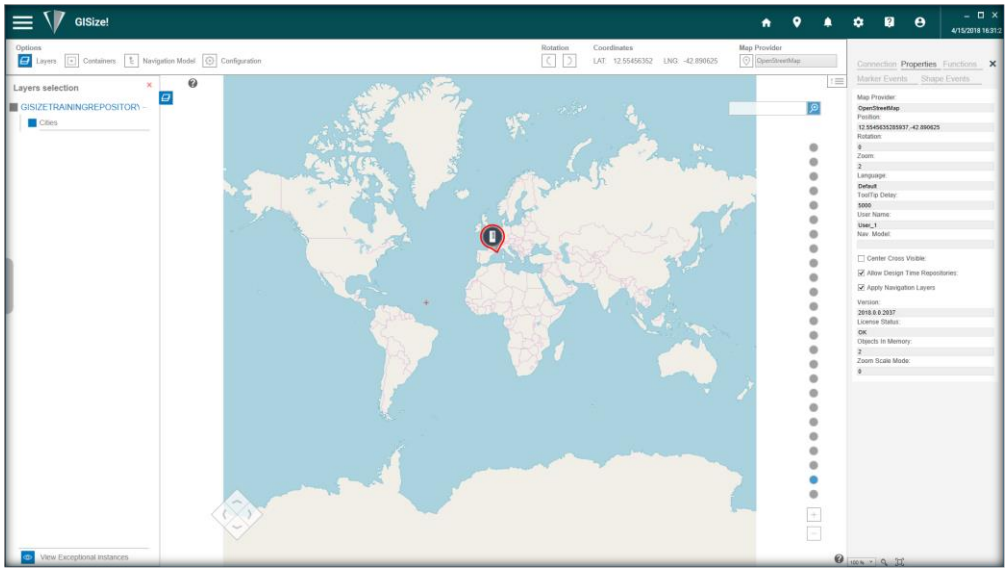


18. Click the **Save and Close** button.

6.4.6. Test View App

In the following steps, you will test GISIZE GEOSpatial embedded in a View App.

1. Open **View App**
2. Click the **Map** button to open the GISIZE GEOSpatial App.
3. Click the **Settings** button.
4. Select the **Properties** panel and change **User Name** field to **User_1.**
5. Select the **Connection** panel and click the **Update Repository Connection** button.
6. Check which layers and instances are available for User_1, User_2 and Administrator users.



Logitek, S.A.

Ctra de Sant Cugat 63, Esc. B, Planta 1ª

08191 - Rubí - Barcelona - Spain

http://www.logitek.es - http://www.wonderware.es - http://www.GISIZE GEOSpatial.com