Qognify

SITUATOR

Qognify Situator

Administrator Guide

March 2023

Version 9.1

Rev. 00

Revision History

Revision	Purpose for Change	Date
00	Version 9.1 GA	March2023

Contents

CHAPTER 1 About This Guide

Situator is a platform for mission-critical incident management and actionable business intelligence. Using a client-server architecture, Situator can handle significant loads of data and triggers from a large variety of third-party systems and sensors.

Situator can be deployed in various topologies, including load-balancing clustered environments.

The information and procedures described in this document are for use by Qognify personnel or system administrators qualified to install and use Situator.

Use this document's information and requirements as a basic guideline for a typical Situator deployment and analyze the system requirements per project case by case.

All Situator documents are available on the The Q (the Qognify Partner Portal).

CHAPTER 2 Hierarchical Site Installations

Situator has built-in support for multiple-hierarchy installations. A hierarchical installation has multiple layers. Each layer manages its own local installation and is unable to view higher layers. Situator supports various escalation mechanisms between the hierarchical layers.

2.1 Hierarchy Options

Each entity and object in the Situator database is associated with corresponding site information, whether it is administrator-defined (such as sensor groups or maps) or derived from the context (such as incidents).

The visibility and manageability of each entity and object is derived from its settings according to the following options:

- *Global* – visible anywhere throughout the full hierarchy
- *Standard* – visible according to its position in the hierarchy and up
- *Regional* – visible throughout its own hierarchy branch
- *Local* – visible only in the local site

The following are examples of hierarchy options:

Global

Standard

Regional

Local

2.2 Hierarchy Settings

Hierarchy layers are defined in the Situator database.

Site Hierarchy Schema:

To define a hierarchy in the database:

1. In the *Planning Tool* **Advanced Setup** view, expand the **Sites** section.
2. Change the parameter **HasSites** value to **1**. The organization's sites are then defined in the *dbo.Sites* table.

Table - dbo.Sites | Table - dbo.SystemConfiguration | Summary

SiteID	Name	Description	ParentSite	Sequence
1	HQ	HQ	NULL	0
2	Region 1	Region 1	1	0
3	Region 2	Region 2	1	1
4	Region 3	Region 3	1	2
14	Site 11	Site 11	2	0
16	Site 12	Site 12	2	1
19	Site 21	Site 21	3	0
20	Site 22	Site 22	3	1
21	Site 31	Site 31	4	0
22	Site 32	Site 32	4	1
NULL	NULL	NULL	NULL	NULL

3. Complete the fields for the parent site:

a. Add in the name and description for the parent site.
b. Since there is no parent site for the main site, the **ParentSite** column should remain "NULL".
c. The **Sequence** column for the main site should be **0** (zero).
d. Assign a **SiteID**. For the parent site this should preferably be **1**.

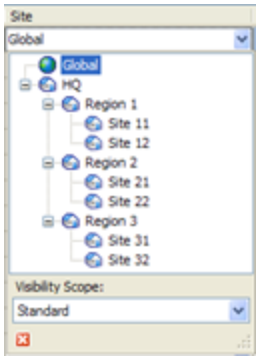
4. Complete the fields for the child site:
   a. Add in the name and description for the child site.
   b. In **ParentSite**, put the "SiteID" of the child's parent site.
   c. In **Sequence**, assign to the children sites starting with **0** (zero). Zero is the first of the children under the parent site, 1 is the second, 2 the third, etc.
   d. Assign a **SiteID**.
5. Click **Save**.
6. Restart the Operational Service.

2.3 Applying Site Settings in Control Room (Refreshing Cached Data)

After making site changes, administrators can refresh cached data in *Control Room* to reflect the changes to the hierarchy of the sites .

To apply site settings:

1. Open Situator *Control Room*.
2. From the **Tools** menu, click **Reload Sites**. The to re-evaluates all sites and re-caches all related entities after which a notification is sent to all currently connected clients.



CHAPTER 3 Bin Folder Configuration Files

3.1 Overview

This chapter describes the common functions of the various Situator configuration files and configuration parameters in the database. A configuration file contains configuration information for a particular program. When the program is executed, it refers to the configuration file to check what parameters are in effect. Configuration files are identified by their *.config* or *.xml* suffix. Situator configuration files are located in various *bin* folders. Configuration parameters in the database are located in the *dbo.SystemConfiguration* table, and can be modified from the *Planning Tool* application under the **Advanced Setup** tab.

IMPORTANT: Do not use Wordpad as a text-editor as it may corrupt the configuration file and prevent *Control Room* from starting.

3.2 Multiple Control Room Applications

Multiple *Control Room* applications can run on the same system. In the configuration file *Stabilis.Situator.ControlRoom.UI.exe.config*, system administrators can, for example, determine how the system should treat multiple running Control Room applications upon startup.

This solution only applies when using the same Windows user account and not if different sessions are used.

Example:

Locate the following line:

```
<add key="HandleMultipleCrProcesses"
   value="AllowMultiple">
</add>
```

- Changing the value to equal "`AutoTerminate`" will automatically terminate all existing processes upon start-up.
- Changing the value to equal "`AskUser`" will prompt the user to decide what action needs to be taken upon start-up.

3.3 Control Room Monitor Layout

The Control Room monitor layout settings are saved in the configuration file *Dockpanel.config*, located in the Control Room *bin* folder. System administrators can, for example, save a copy of the file to use for backup.

3.4 Incident Tasks Layout Configuration

In the Incidents Task Tab, the **View** button menu includes two layout options: **Default** and **Collaboration**.

- The **Default** option restores the default column settings as well as the entire layout for all the gadgets. On new client installation and first startup, the default layout is taken from the *SystemLayout* file in the *dbo.SystemConfiguration* database table. Subsequently, on Control Room shutdown, the layout is stored locally in the Bin Configuration file *IncidentsLastLayout.xml*, which is used on the next startup to apply the user-defined layout changes.
- The **Collaboration** option applies the layout taken from the *dbo.SystemConfiguration* table reference *TaskCollaborationView*. This xml file defines the Collaboration layout, and should be edited only by Qognify Professional Services.

3.5 Setting User Interface Languages

The User Interface (UI) language is supported for a designated user and can be changed by administrative users. Language setting is available in the following Situator applications:

- Control Room
- Planning Tool
- Reporting Tool

The languages presented to a user upon login are defined and saved in the following configuration files:

Application	Configuration file	Location
Control Room	Stabilis.Situator.ControlRoom.UI.exe	Control Room / bin
Planning Tool	Comergency.UI.exe.config	Planning / bin

Application	Configuration file	Location
Reporting Tool	Orsus.Situator.Reports.WizardUI.exe.config	Control Room / bin
	Stabilis.Situator.ReportingService.exe.config (update the key "**ReportsCulture**" value with required language). See the example image below.	Server\Operational\bin

The following is an example of Report language configuration:

Stabilis.Situator.ReportingService.exe.config

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="nhibernate" type="System.Configuration.NameValueSectionHandler,
    </section>
  </configSections>
  <appSettings>
    <!--   User application and configured property settings go here.-->
    <!--   Example: <add key="settingName" value="settingValue"/> -->
    <add key="OpThreadPoolSize" value="5">
    </add>
    <add key="ReportsStoragePath" value="Reports">
    </add>
    <add key="UserName" value="pLK4*enc*SNn6rJ8eZDWlVSy4Gg==">
    </add>
    <add key="Password" value="pLK4*enc*SNn6rJ8eZDWlVSy4Gg==">
    </add>
    <add key="Machine" value="YHc3*enc*Zxg4mKfRQSHbkjhMeQ==">
    </add>
    <add key="ReportsStoragePathType" value="Relative">
    </add>
    <add key="ReportsCulture" value="es-ES">
    </add>
    <add key="channel" value="tcp">
    </add>
    <add key="port" value="5030">
    </add>
```

After applying the language, it is required to restart the reporting services:

1. Stop 'Situator Reporting Service'. In the server RAM, terminate any instance of the Situator Reporting Service.
2. Start 'Situator Reporting Service'.

Only languages that appear in the file are presented to the user in the login screen, as shown below:

```
119     <add key="PlumPointPrecision" value="1">
120     </add>
121     <add key="MainForm.PreviewFunctionality" value="false">
122     </add>
123     <add key="LanguageKey" value="en-US">
124     </add>
125     <add key="SystemLanguages" value="en-US,es-ES,es-HN,he-IL,mk-MK,pl-PL,pt-PT,ru-RU,th-TH">
126     </add>
127     <!-- Toggle Threat Level Panel Visibilty -->
128     <add key="ThreatLevelPanelVisibile" value="true">
129     </add>
130     <!-- If set to false, the 'Screens' list in Video pane will be hidden, and only Main video screen
131          will be accessible. -->
132     <!--<add key="AllowChangeScreens" value="false" />-->
```

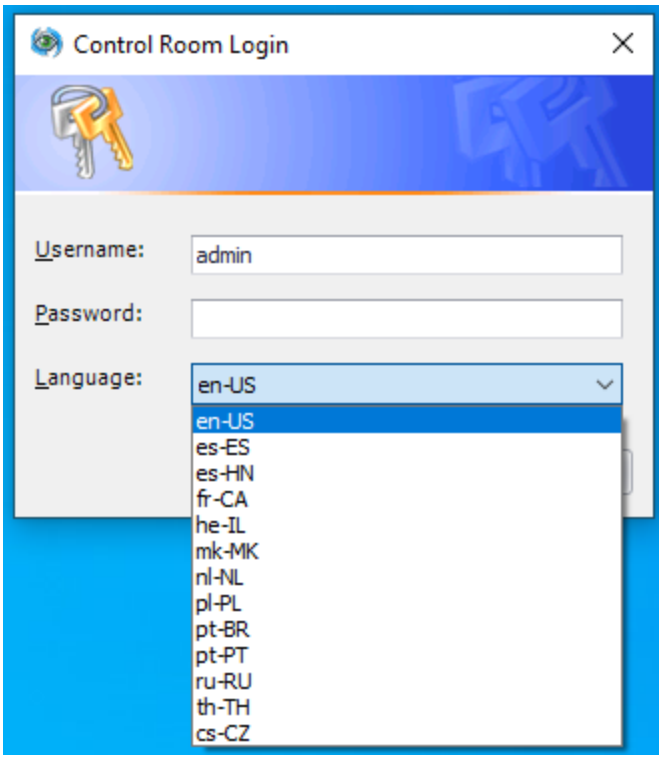
In addition, the default language which will be selected for the first login can be set. Upon subsequent login sessions, the UI language will be the user's last choice.

The default language is set in the "`LanguageKey`" key, and the list of languages presented to the user is set in the "`SystemLanguages`" key.

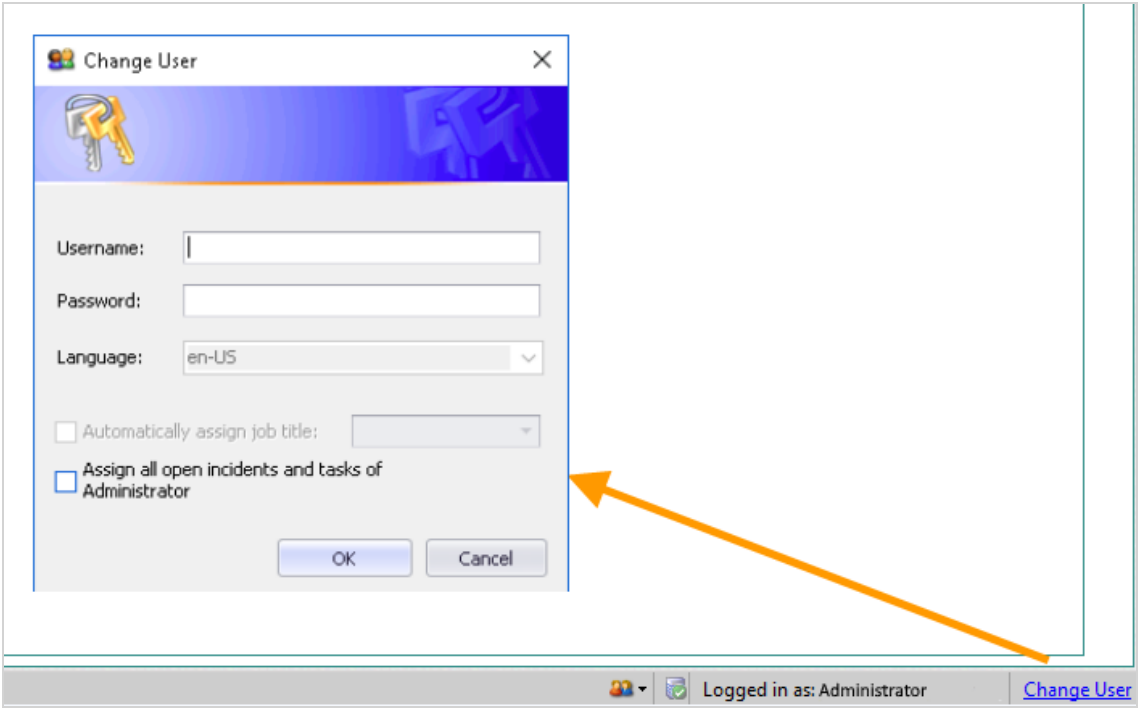
To change the UI language:

Do one of the following:

- During the login process, in the login window, select the language.



- While using the Control Room, click **Change User** and define the language.



3.6 Map Adapter Configurations

Situator's Multiple Visualization Platform (MVP) provides users the flexibility to choose from a range of industry-leading visualization adapters and the ability to toggle between 2D and 3D views.

Supported adapters include:

- ESRI ArcGIS Engine 10.0 SP4
- ESRI ArcGIS 10.2 Runtime WPF
- ESRI ArcGIS 100.5 Runtime Quartz

> NOTE: Skyline is currently not supported.

To enable a Quartz map adapter:

The configuration files are in the following folder:

<Your_Installation_Path>\ControlRoom\bin\GISAdapters\ Orsus.Situator.GisAdapter.Esri.ArcGISQuartz.dll.xml

1. Open the relevant xml file in an XML editor, such as Notepad.
2. Under GISAdapterConfiguration, locate the following:

   `IsEnabled="False" RunAsAddin="True" RunInDifferentProcess="True"`
3. Change the `IsEnabled` value to "True".
4. Recommended: Leave the `RunInDifferentProcess` value set to "True" (the default value).
   - When set to "True", the adapter will run as an AddIn in a different process.
   - When set to "False", the adapter will run inside Control Room's AppDomain.

   For information on monitoring an AddIn process, refer to Monitoring an AddIn Process on page 13.

   > NOTE: In some of the configuration files, there is also a `Title` value. The `Title` value determines the text that will display in the Control Room Maps view if more than one adapter is enabled.
   > In the ESRI ArcGIS Quartz configuration files, the `Title` value is " Quartz 2D/3D".
5. Under **GISAdapterConfiguration.AdapterAttributes**, add the license key to the `ArcGISRuntime_LicenseKey` value, as shown in the following example:

   `x:Key="RuntimeLicenseKey">runtimebasic,***,rud861174764,none,3M1NDT************<`

To enable a WPF map adapter:

1. Open the relevant xml file in an XML editor, such as Notepad.
2. Under **GISAdapterConfiguration**, locate the following:

   `IsEnabled="False" RunAsAddin="True" RunInDifferentProcess="True"`
3. Change the `IsEnabled` value to "True".
4. Recommended: Leave the `RunInDifferentProcess` value set to "True" (the default value).
   - When set to "True", the adapter will run as an AddIn in a different process.
   - When set to "False", the adapter will run inside Control Room's AppDomain.

   For information on monitoring an AddIn process, refer to Monitoring an AddIn Process on page 13.

NOTE: In some of the configuration files, there is also a `Title` value. The `Title` value determines the text that will display in the Control Room Maps view if more than one adapter is enabled.
In the ESRI ArcGIS Engine or **WPF**[1] configuration files, the `Title` value is "2D".

5. If you are enabling ESRI AcrGIS WPF, under **GISAdapterConfiguration.AdapterAttributes**, add the license key to the `WPFRuntime_LicenseKey` value, as shown in the following example:

```
x:Key="WPFRuntime_
LicenseKey">runtimebasic,***,rud861174764,none,3M1NDT**************<
```

(For WPF, if you are using a coordinate system other than WGS84) - Set the URL for the relevant Geometry server to the `WPFRuntime_GeometryServerUrl` value key, as shown in the following example:

```
x:Key="WPFRuntime_
GeometryServerUrl">http://GISSERVER/arcgis/rest/services/Utilities/Geometry/Geom
etryServer<
```

6. Restart the *Control Room* application.

3.6.1 Enabling Windows Authentication for ESRI ArcGIS WPF Map Layers

For ESRI Service layers using the ESRI ArcGIS WPF adapter, Situator supports both ArcGIS authentication (user/password for the ArcGIS server) and Windows authentication. Windows authentication credentials must first be configured in the *Orsus.Situator.GisAdapter.Esri.ArcGISWPF.dll.xml* file.

NOTE: ESRI does not support tokens for WMS and KML layers. Therefore authentication is not supported for these layers in Situator. If possible, convert WMS and KML layers to dynamic map layers.

To enable Windows authentication:

1. Open the *Orsus.Situator.GisAdapter.Esri.ArcGISWPF.dll.xml* file, located in: *...\Qognify\Situator\ControlRoom\bin\GISAdapters* in an XML editor, such as Notepad.
2. Make the necessary configurations, as follows:
   - `WPFRuntime_UseDefaultCredentialsWithProxy`: Set to "`True`" to initialize the layers request to be sent with default web credentials. Default value is "`False`". It is recommended to set this to "`True`".

[1] Windows Presentation Foundation

- `WPFRuntime_ArcGisServerGenerateTokenURL` – The token URL generated by the ArcGIS server (used for layers with user name and password and for layers with Windows authentication). The default value is `{ServerURLOfLayer}/arcgis/tokens/generateToken`.
- `WPFRuntime_ArcGisServerWithWindowsAuthentication` – The ArcGIS server that should be called with Windows authentication for its layers. If empty, no server will be called with Windows authentication. If the value equals "`ALL`", all servers will be called with Windows authentication. It should look like this: *http://{Server}*
- `WPFRuntime_ArcGisServerTokenDurationInMinutes` – The expiration value (in minutes) for the token requested from the ArcGIS server. The default value is a day (1440).

3. Save the changes in the file and only then restart Control Room.

3.6.2 Enabling Map Background Color

When using ESRI ArcGIS v.10 or ESRI ArcGIS WPF, system administrators can enable/disable a map background color in Control Room by making modifications in the relevant configuration file:

- *Orsus.Situator.GisAdapter.Esri.ArcGIS10.dll.xml* for ESRI Engine
- *Orsus.Situator.GisAdapter.Esri.ArcGISWPF.dll.xml* for ESRI ArcGIS WPF

To enable a background map color:

1. In the ControlRoom>bin>GISAdapters folder, open the relevant xml file in an XML editor, such as Notepad.
2. Under **GISAdapterConfiguration.AdapterAttributes**, locate the following:
   - For ESRI ArcGIS v.10, `<x:String x:Key="ESRI_MapBackgroundColor">#ffffff</x:String>`
   - For ESRI ArcGIS WPF, `x:String x:Key="WPFRuntime_MapBackgroundColor">#ffffff</x:String>`
3. Change the `#ffffff` value to the color you want.

> NOTE: Use hexadecimal color codes.

3.6.3 Monitoring an AddIn Process

The Process Monitor provides the ability to monitor any component that hosts an AddInProcess and the state of the hosted process, as well as recover the process when necessary.

The AddInProcess monitor runs in a background thread and validates the state of the process. It sends a *MonitoredProcessStateChanged* event when a process state changes upon which the AddInProcess host will perform a recovery.

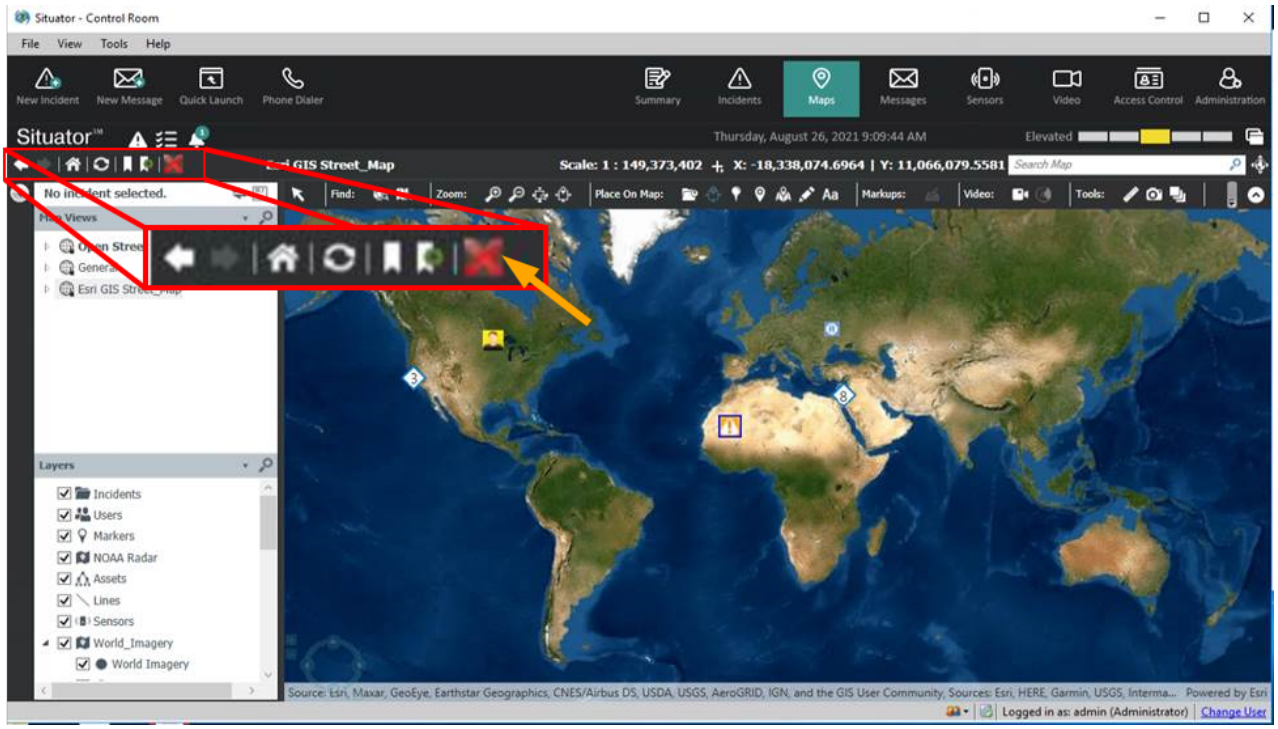
AddInProcessMonitor activity can be viewed in the logger file *ControlRoomAddInProcessMonitor-Log.txt*, located in the *ControlRoom\bin\Logs* folder.

The following monitoring settings can be made in *GIS.Configuration.xml*, located under *...\\Qognify\Situator\Client\ControlRoom\bin\Config*:

```
<GISConfiguration GISShowDebugToolBarItems="False" xmlns="clr-namespace:Nice.Situator.ClientGISCommon.GIS;assembly=Nice.Situator.ClientGISCommon"
                  xmlns:scg="clr-namespace:System.Collections.Generic;assembly=mscorlib"
                  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <GISConfiguration.GISAttributes>
    <x:Boolean x:Key="ShowEntityOutOfExtent">True</x:Boolean>
    <x:Boolean x:Key="IndicateLockedUnlockedDoors">True</x:Boolean>
    <x:Int32 x:Key="GoToMapLocationScale">5000</x:Int32>
    <x:Int32 x:Key="MapSearchTimeDelaySeconds">2</x:Int32>
    <x:Int32 x:Key="MinCharNumForInstantSearch">3</x:Int32>
    <x:Boolean x:Key="ServiceDiscovery_EnablePreviewImage">True</x:Boolean>
  </GISConfiguration.GISAttributes>

  <GISConfiguration.AddinProcessMonitor>
    <AddinProcessMonitorConfiguration IsEnabled="True" MonitorIntervalMilliseconds="5000" RulePredicateTimeoutMilliseconds="10000" MonitoringRules=
    "IsAliveByPollingRule,GDIObjectesThresholdRule" ShowKillProcessButton="True" />
  </GISConfiguration.AddinProcessMonitor>
  <GISConfiguration.Geoprocessing>
    <GeoProcessingConfiguration IsEnabled="True" Provider="Runtime"/>
  </GISConfiguration.Geoprocessing>
  <GISConfiguration.Addins>
    <AddinConfiguration GISAdapterAddIn_MouseMoveEventFiringInterval="200" GISAdapterAddIn_AllowedGDIObjectsCount="9500" AddInsLibraryPath="AddInsLibrary"/>
  </GISConfiguration.Addins>

  <GISConfiguration.GISAdapters>

    <scg:List x:TypeArguments="GISAdapterConfiguration">

    </scg:List>
  </GISConfiguration.GISAdapters>
</GISConfiguration>
```

Configuration key	Set this to...	In order to...
AddInProcessMonitorConfiguration_ IsEnabled	True	Enable AddiInProcessMonitor to recover a failed AddInProcess
MonitorIntervalMilliseconds	A designated time interval in milliseconds	Determine the time interval in which the AddInProcessMonitor validates its monitored processes
RulePredicateTimeoutMilliseconds	A designated time interval in milliseconds	Determine the time interval for process monitoring rule validation before the rule is aborted

Configuration key	Set this to...	In order to...
MonitoringRules	IsAliveBy PollingRule	Defines the supported monitoring rules for the AddInProcessMonitor. NOTE: By concatenating the rules, it can contain more than one rule. Available rules: » `IsAliveByPollingRule`: Monitors availability of AddInProcess by polling the GIS Adapter proxy and expecting a simple string value. An unexpected result (exception or timeout invocation) means that the adapter is no longer available. » `IsAliveByQueryingPIDRule`: Monitors availability of AddInProcess by querying the machine's processes using the PID of the hosted AddInProcess.exe. It will not identify a hanging process like the previous monitoring rule. » `TimeoutExceededRule`: Emulates timeout and should be used for debugging purposes only.
ShowKillProcessButton See the figure below.	True	Adds a toolbar button in the *Maps* pane, which kills all AddInProcess.exe instances. NOTE: This is for debugging purposes only.



3.6.4 ESRI WPF Adapter Geolocator Configurations

Configurations for the geolocator feature are made in the following XML files:

- *gis.configuration.xml*
- *GPActionsRepository.xaml*

To define the ESRI WPF adapter in gis.configuration.xml:

To define the ESRI WPF adapter as an engine with local ArcGIS or runtime with server ArcGIS, edit the *gis.configuration.xml* in the *bin/Config* folder and set the provider attribute of the **GeoProcessingConfiguration** to '**Runtime**' or '**Engine**'.

For example:

```
<GeoProcessingConfiguration IsEnabled="True" Provider="Runtime"/>
```

To define the ESRI WPF adapter in GPActionsRepository.xml:

The *GPActionsRepository.xaml* file is in the *bin/Geoprocessing* folder.

Every command (e.g. **GeocodingAddressCommand** or **ReverseGeocoding**) has its own configuration and needs to be configured separately, as detailed in the following table:

NOTE: Currently, only Engine with local ArcGIS and Runtime with server ArcGIS are supported.

Command	GeocodingAddressCommand	ReverseGeocoding
Engine>local	**ShapeFieldName** – the name of the shape field in the returned results. Currently, the value is "**Shape**" .	
	ScoreFieldName – the name of the score field in the returned results. Currently the value is "**Score**".	
	DisplayFieldName – the name of the display field in the returned results. Currently, the value is "**Match_addr**".	
		ReturnIntersection - Whether to return the nearest intersection as the geocoded address. Currently the value is "**False**".
		LocalLocatorName – The local locator name e.g. "**AtlantaLoc**".
Runtime>server	**LicenseKey** – runtime arcgis install key (standard or basic)	
	InstallPath – the path of the arcgis runtime on the computer	

Command	GeocodingAddressCommand	ReverseGeocoding
Global	**LocatorPath** – the path of the data source. In local mode, the value should be the GDB file path and in the server it should be the URL of the Geocode server.	
	MinimumScore – results under this score will not be shown (values from low 1 to high 100).	
	MaximumResults – the number of maximum results that will be returned from the search provider. Default is "1000" (all).	
		CustomFormat – defines the result format that will be presented on the GUI. For example, %userContent% - (%locatorName%) will show the resulting address and then the relevant locator name in parenthesis. Currently, only the address (%userContent%) and locator name (%locatorName%) are supported.
		SearchDistanceInMeters – defines the radius of the search perimeter in meters.

3.6.5 Map Caching

This feature increases the speed of switching to a map view that was already loaded.

Upon closing a map view, the most recent ESRI Map document is saved in the MapsCacheManager. As a result, when opening a map view, the adapter checks if there's a cached Map document that was already built and opens the requested map view quickly.

The number of map-caches stored in the MapsCacheManager is controlled by the following configuration entry in the configuration file *Stabilis.Situator.ControlRoom.UI.exe.config*, located in the Control Room *bin* folder:

```
<!-- Specifies the number of maps to remember in the adapter's inner cache -->
<add key="ESRI_NumberOfMapsToKeepInCache" value="10"/>
```

3.6.6 Geocoding Configuration

Situator allows searching for a point on a map from a street address to a geocoordinate point on a map and vice versa (reverse geocoding). To do so, administrators need to configure an ESRI mapping file to coded parameters and enable this feature in the Control Room configuration file: *C:\Program Files\Qognify\Situator\Control Room\bin\Geoprocessing\GPActionsRepository.xaml*.

NOTE: This feature is currently supported by ESRI ArcGIS. While parameter mapping is done in the ESRI file, the defined parameters may be applicable to all other GIS applications.

To configure the Provider Type File from a local file:

The provider type is a GDB file located on a local network or a local file system. The locator file is based on a project specific map file.

Configure the following parameters in the Control Room configuration file:

- LocatorName
- AddressFieldName
- ShapeFieldName
- LocatorPath

To configuring Action parameters:

Administrators can define action parameters as part of customizing a user's search criteria. For example, for reverse geocoding, administrators can set the distance value in the "SearchDistance in Meters" parameter. Another example (for reverse geocoding) is to redefine the "relevant intersection" parameter which is set to false (as the default) to true so that the geocoding shows the pinpoint at the closest street intersection.

In the Control Room configuration file, reconfigure the relevant action parameters as shown in the example below:

```
<gpc:GPAction Name="ReverseGeocoding" x:TypeArguments="opServerTypes:GisEntityLayer">
    <gpc:GPAction.ActionCommand>
        <wpf:Binding>
            <wpf:Binding.Source>
                <wpf:ObjectDataProvider ObjectInstance="{x:Static GPCommandsFactory.Instance}" MethodName="GetCommand">
                    <wpf:ObjectDataProvider.MethodParameters>
                        <x:String>ReverseGeocodingCommand</x:String>
                    </wpf:ObjectDataProvider.MethodParameters>
                </wpf:ObjectDataProvider>
            </wpf:Binding.Source>
        </wpf:Binding>
    </gpc:GPAction.ActionCommand>

    <gpc:GPAction.AuthorizationContext>
        <data:Actions>AdmnGIS</data:Actions>
    </gpc:GPAction.AuthorizationContext>

    <gpc:ActionParameter x:TypeArguments="x:String" Name="LocatorName" Value="Locators/TA_Address_NA_10" />
    <gpc:ActionParameter x:TypeArguments="x:String" Name="AddressFieldName" Value="SingleLine" />
    <gpc:ActionParameter x:TypeArguments="x:String" Name="ShapeFieldName" Value="Shape" />
    <gpc:ActionParameter x:TypeArguments="x:String" Name="ScoreFieldName" Value="Score" />
    <gpc:ActionParameter x:TypeArguments="x:String" Name="DisplayFieldName" Value="Street" />
    <gpc:ActionParameter x:TypeArguments="gp:EGeocodingSearchProviderType" Name="ProviderType" Value="File" />
    <gpc:ActionParameter x:TypeArguments="x:String" Name="LocatorPath" Value="Geoprocessing\Data\Atlanta.gdb" />
    <gpc:ActionParameter x:TypeArguments="x:String" Name="LocalLocatorName" Value="AtlantaLoc" />
    <gpc:ActionParameter x:TypeArguments="x:Int32" Name="SearchDistanceInMeters" Value="50" />
    <gpc:ActionParameter x:TypeArguments="x:Boolean" Name="ReturnIntersection" Value="False" />
</gpc:GPAction>
```

3.6.7 Nearest Resource Configuration

Control Room provides an option that shows the closest resources to an incident: Show Nearest Route: Displays a suggested route on the geo-referenced map from nearest resource to the incident.

For using the Nearest Resource feature, the ESRI Engine version 10 SP4 and WPF version 10.2 with a working license of ArcGIS Engine should be properly installed on any client machines that will be using this feature.

To configure the client Control Room options and the [1]file location:

1. Start the *Control Room* application and log in. Wait for Control Room to load and to become functional.
2. Exit the client by selecting **File**>**Exit**.

   WARNING: Do not terminate the process; exit the client using Control Room UI only.

3. Edit the file: *C:\Program Files (x86)\Qognify\Situator\ControlRoom\bin\Configuration\NARouteSetting.xml*:

   Change the following value from the default value of false to true to enable configuring any of the feature's parameters:

[1]Database file created by MapSource

```
<EnableNearestResourceFeature>false</EnableNearestResourceFeature>
```

To:

```
<EnableNearestResourceFeature>true</EnableNearestResourceFeature>
```

4. Set the specific *gdb* folder name (in place of "*gdb path name*") in the following folder path: *<NARoutePath>gdb path name</NARoutePath>*.

> NOTE: The *gdb* folder path can be to a network (i.e. *<NARoutePath>\\netapp\QA\Situator_QA\xxx\GIS\xxx\xxx\ND_Geo.gdb</NARoutePath*) or to a local location.

5. Save the changes in the file and only then restart the *Control Room*.

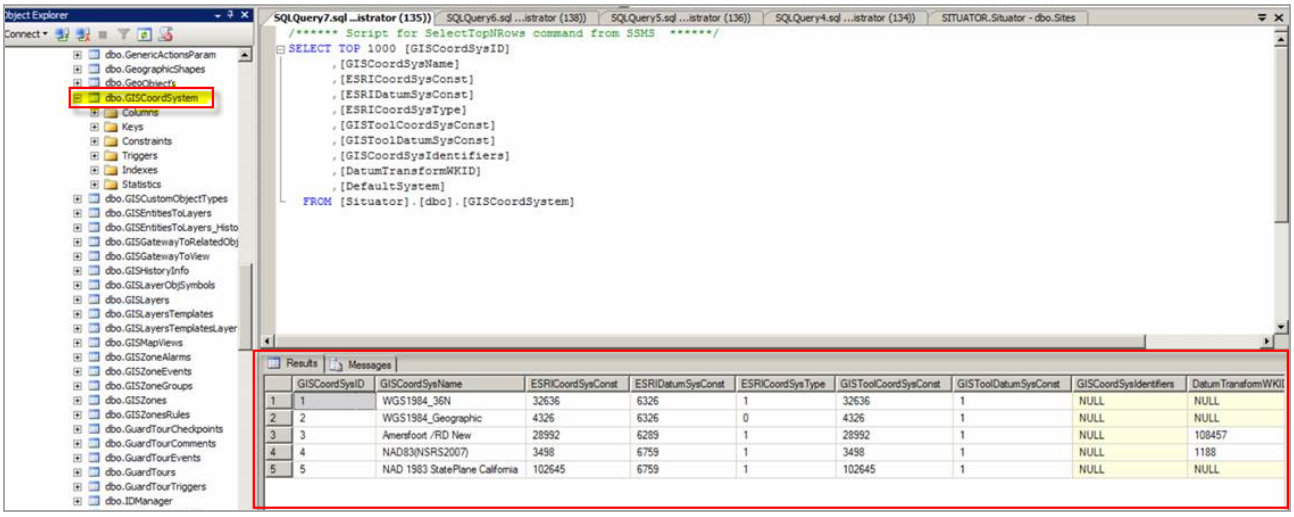
3.6.8 Adding a GIS Coordinate System

Situator supports ESRI ArcGIS coordinate systems. You may add an ESRI ArcGIS coordinate system by manually entering the coordinate system parameters to the Situator database file dbo.GISCoordSystem.

For information on ESRI ArcGIS coordinate systems, refer to:
http://desktop.arcgis.com/en/arcmap/10.3/map/working-with-arcmap/specifying-a-coordinate-system.htm.

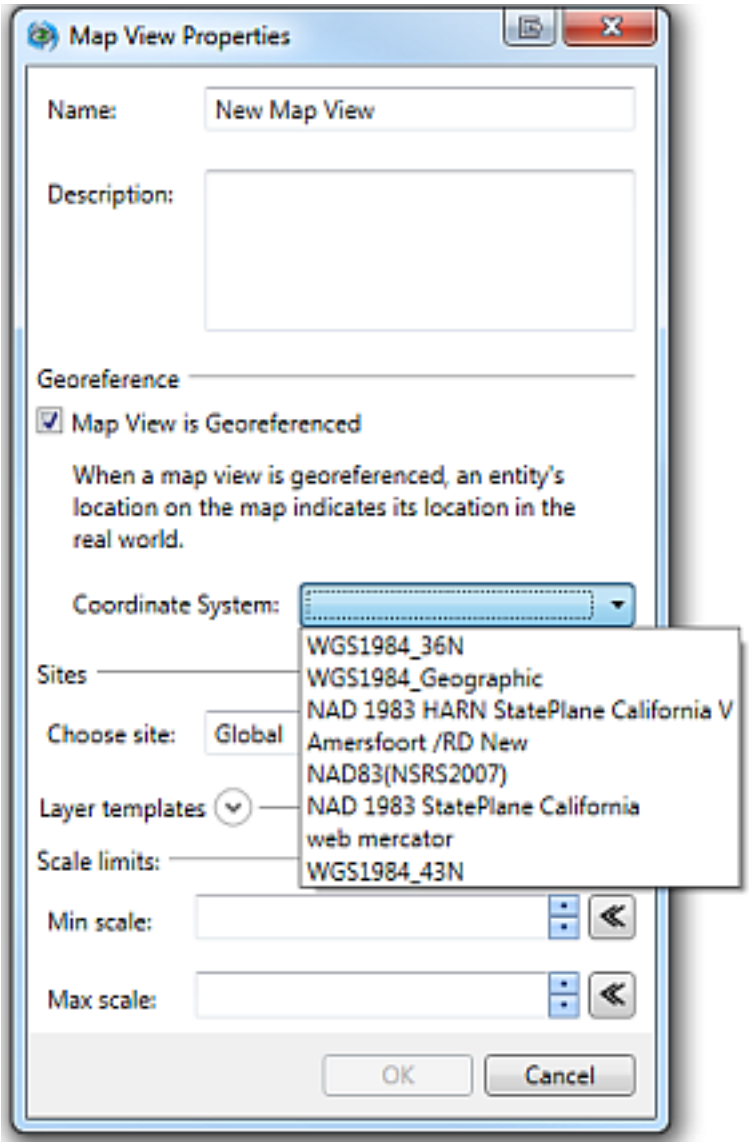
To add a new coordinate system:

1. Using **SQL Server Management Studio Object Explorer**, open *dbo.GISCoordSystem*. Currently installed coordinate systems will be listed in the column GISCoordSysName, as shown in the example below:



2. For the new coordinate system, manually enter the parameters as a new entry to the DB table.
3. Verify the entry by rerunning the query on the table.

4. In the *Control Room* **Maps** view pane toolbar, click the **Edit** button to display the *Map View Properties* dialog box. The new coordinate system is available from the **Coordinate System** menu.



3.7 Entities Coordinate-Projection Methodology Options

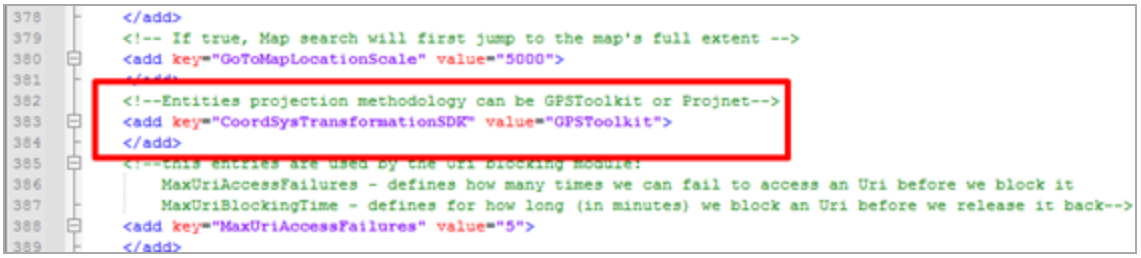
Situator *Control Room* uses a third-party coordinate-transform application to enable real-time display of entity locations on the map. By default, *Control Room* uses Microsoft's *GPSToolkit*, but you may reconfigure it to use *ProjNet*, as described herein. ProjNet may provide more coordinate system options, depending on the map and its original coordination system.

Coordinate systems supported by ProjNet and tested for *Control Room* currently include WGS 84_Ellipse / Pseudo-Mercator, WGS 84_Courtusian3D / Pseudo-Mercator, and WGS 84 / Pseudo-Mercator. For other systems that may be supported, contact Qognify Customer Support.

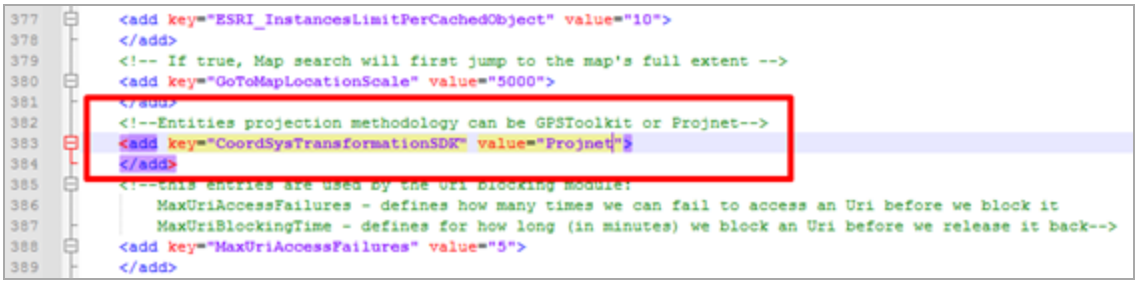
To change the entities coordinate projection methodology:

1. Close the *Control Room* application.
2. Open the Control Room configuration file located at *C:\Program Files (x86)\Situator\Client\ControlRoom\bin\Stabilis.Situator.ControlRoom.UI.exe.config*.

   By default, Entities projection methodology is GPSToolkit:



3. Change the **CoordSysTransformationSDK** value as required, either GPSToolkit (default) or Projnet.



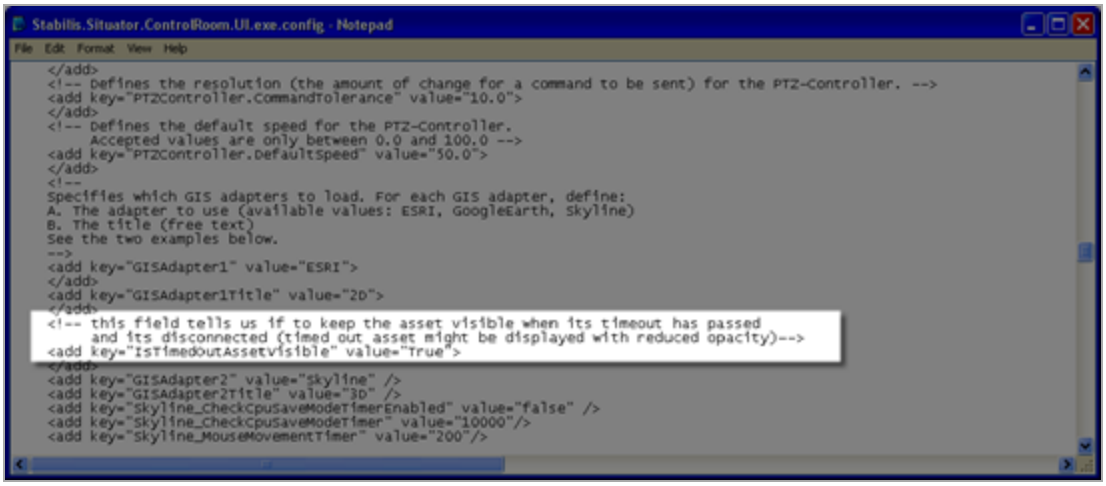
4. Open the *Control Room* application.
5. Confirm that entities placed on the map have the correct X,Y coordinates.

3.8 Handling Timed-out Assets on Maps

Administrators can select how a timed-out asset might display on maps. These assets may display:

- with reduced opacity, or
- completely removed from the map

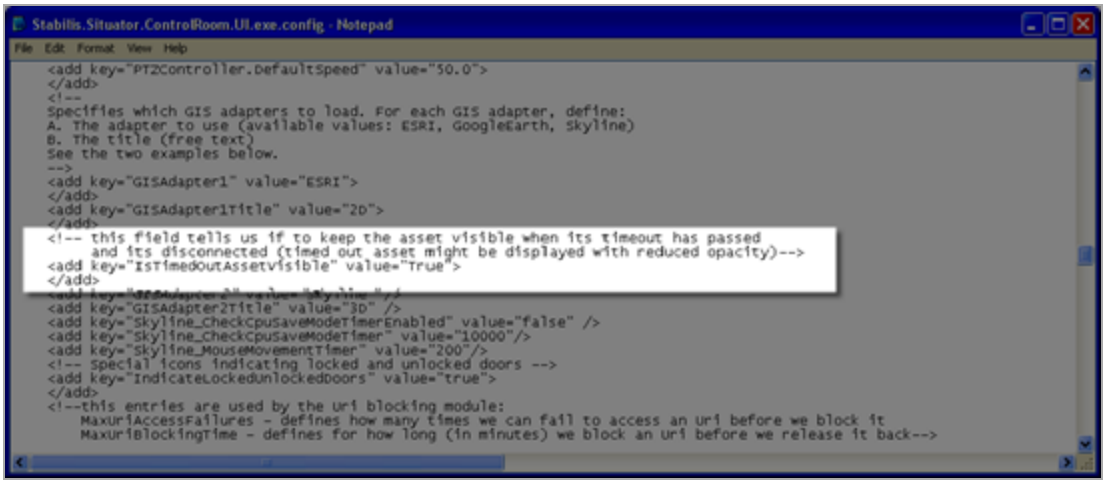
The configuration isavailable in the configuration file **Stabilis.Situator.ControlRoom.UI.exe.config**, located in the Control Room *bin* folder.



The time of the last updated location is added to the asset's tooltip.

3.9 Vehicle Assets Map Icon Configuration

By default, icons representing vehicle assets whose location has become obsolete (timeout) continue to show on maps. System administrators can prevent timeout icons from displaying on the map by changing the "**IfTimeOutAssetVisible**" value to "**False**" in the configuration file **Stabilis. Situator.ControlRoom.UI.exe.config**, located in the Control Room *bin* folder.



In the tooltip properties box of a vehicle asset placed on a map, the *License Plate Number* (*LPN*) of that vehicle displays for a designated period of time (default is 12 hours).

The time period the LPN shows in the tooltip can be defined in the **dbo.SystemConfiguration** table in the database, as described below.

Located in the AssetManagement section, the "**ShowLPNIfHoursPassedLowerThan**" field's default time is set to "**12**". This means that a vehicle's LPN is reported to the *License Plate Recognition* (*LPR*) system for a period of 12 hours. After the defined number of hours, a report is not sent and the asset's LPN no longer displays in the tooltip on the map.

To change the time period the LPN shows in the tooltip:

1. In the *Planning Tool* **Advanced Setup** view, expand the **AssetManagement** section. By default, the **ShowLPNIfHoursPassedLowerThan** default time is set to 12. This means that a vehicle LPN is reported to the License Plate Recognition (LPR) system for a period of 12 hours.
2. Click in the field in the *Value* column, and change the value as required.
3. Click **Save**.
4. Restart OpService for the change to take effect.

3.10 LPR Attachment Names Format

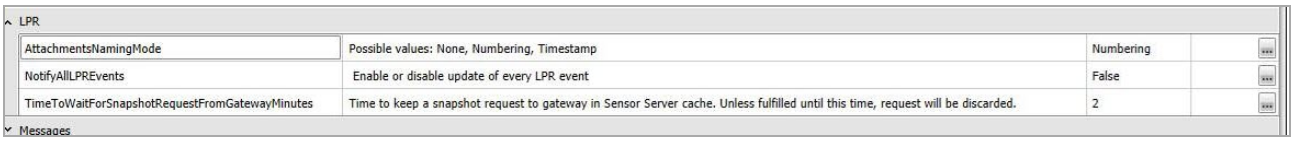
LPR event-triggered snapshots come through a LPR third-party system, which sends the images via the LPR Gateway to Situator, making the snapshots available in the system for use as incident attachments.

Situator provides several format options for the names of the automatically generated LPR attachments. The parameter **AttachmentsNamingMode**, in the Database System Configuration table, may be set for the following formats:

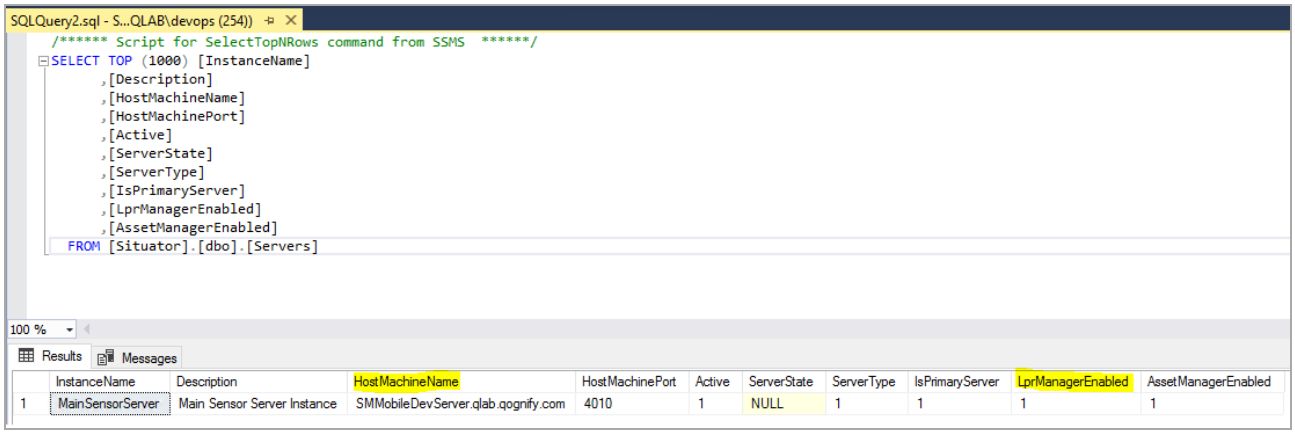
- **None** – all the LPR image attachments will have the same name as before, i.e., "LPR Snapshot"
- **Numbering** – (the default) all the LPR image attachments will have an addition of "_" and a unique identifier.
- **Timestamp** – all the LPR image attachments will have an addition of "_" and the timestamp when they were added.

To configure the LPR attachments numbering format:

1. Do one of the following:
   - In the *Planning Tool* **Advanced Setup** view, expand the **LPR** section.
   - Use a *Database Management Tool* to open the *dbo.SystemConfiguration* table.
2. For the parameter **AttachmentsNamingMode**, under the *Value* column, set the option required: None, Numbering, or Timestamp.



3. Click **Save**.
4. For the LPR to be enabled, in the DB Servers table, verify that:
   - The **LPRManagerEnabled** is **True**.
   - The value of the **HostMachineName** is the Operational server machine name.



3.11 Video Sources in the Video View Matrix

All live video sources in the Situator *Control Room*, which are currently open in the video matrix, are defined in the *configuration file VideoScreenStates.config*, located in the Control Room *bin* folder.

Situator supports multiple screen displays. Each Situator "video screen" is differentiated by a `<ViewerContainerState>` XML element. Within `<ViewerContainerState>`, all open video sources are defined by a VideoSlotState XML element, according to the relevant SlotID attribute, as shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<ViewerContainerStates xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ContainerStates>
            <ViewerContainerState ContainerID="0" Rows="3" Columns="3"
            Locked="false" FullScreen="false">
      <OpenedSlots>
            <VideoSlotState SlotID="0" ViewType="LIVE_VIDEO" ViewID="28"
            Locked="false" />
      </OpenedSlots>
    </ViewerContainerState>
<ViewerContainerState ContainerID="1" Rows="3" Columns="3" Locked="false"
FullScreen="false">
   <OpenedSlots>
            <VideoSlotState SlotID="0" ViewType="LIVE_VIDEO" ViewID="29"
            Locked="false" />
      </OpenedSlots>
    </ViewerContainerState>
  </ContainerStates>
            </ViewerContainerStates>
```

While the *Control Room* is closed, system administrators can manually redefine cameras, virtual tours, and multiple screen displays in *VideoScreenStates.config* file.

Example:

Deleting the following line will close that specific video source:

`<VideoSlotState SlotID="0" ViewType="LIVE_VIDEO" ViewID="28" Locked="false" />.`

> WARNING: Deleting the configuration file *VideoScreenStates.config* entirely when the Control Room is closed will clear all open cameras and Virtual Tours from all screens. When Control Room is reopened, the Video view matrix will be empty. The predefined "Screens" will still exist, but they too will be empty. The *VideoScreenStates.config* file will be automatically recreated by the Control Room upon exit.

3.12 Video Source Tree Lock Options

By default, the Video Source tree is unlocked on slot selections. Administrators can change the lock settings of the video source tree.

To change the default Video Source tree lock setting:

1. Open the Control Room configuration file *Stabilis.Situator.ControlRoom.UI.exe.config* and find the parameter **<add key="LockTreeOnSlotSelection" value="False">**.
2. Set the parameter value to **True**. The cameras tree will be locked on slot selection each time *Control Room* is started.
3. Save the file.

3.13 Configuring the Incident Report Filename Structure

When generating a periodic or incident report, the report filename has the following default structure: Report name, date, and time.

You can modify this default filename to include other parameters, such as incident type, owner ID, report generation time, and more. Configuring the structure of the report's filename may help in managing the report more efficiently.

To customize the reports filename structure:

1. In the file *C:\Program Files (x86)\Qognify\Situator\Server\Operational\bin\Stabilis.Situator.ReportingService.exe.config* change the "**default**" value of parameter **IncidentReportFileNames**. Follow these guidelines:
   - Use one or more of the supported parameter values:
     - For incident report filename - ReportType (incidentReport), Name, IncidentID, IncidentTypeName, OwnerID, StartTimeDateTime
     - For periodic and statistic report filename - ReportType (periodic report, incident statistic report), startTime, endTime
   - Use "`%<value>%`" format
   - Add spaces or underscores (_) between the values as necessary.

   Example of filename configuration:

   ```
   <add key="IncidentReportFileName"
   value="%ReportType%_%Name%_%IncidentID%_%IncidentTypeName%_%OwnerID%_%StartTimeD
   ateTime%"
   ```

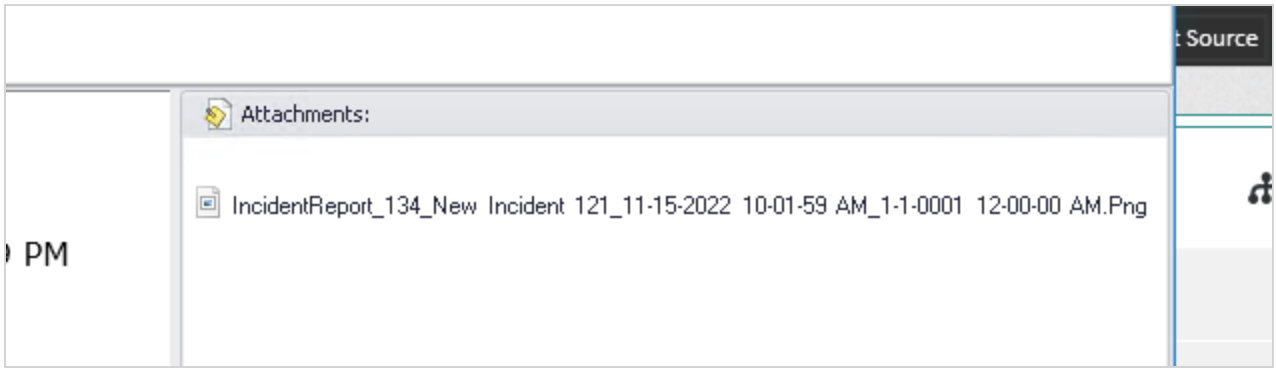
   The resulting report filename will be similar to the following:

   IncidentReport_223_IncName_01-01-2023 2-00-00 PM_01-02-2023 2-45-21 PM 8-15-2022 2-45-21 PM

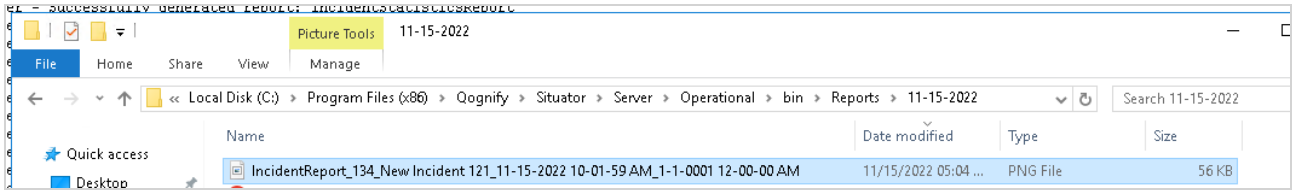
```
</add>
<!--  *
*   Default: "Default" - report name + date and time.
* Supported params: ReportType (constant- incidentReport),IncidentID, Name, IncidentTypeName, StartTimeDateTime, ClosureTimeDateTime, OwnerID
* Format example: %ReportType%_%IncidentID%_%Name%_%StartTimeDateTime%_%ClosureTimeDateTime% .
* Output will be $ReportType_$IncidentID_$Name_$StartTimeDateTime_$ClosureTimeDateTime .
* Output with values: for the values ReportType =IncidentReport, IncidentID=123, Name="new incident", StartTimeDateTime = 01/01/2020 14:00,
   ClosureTimeDateTime = 01/02/2020 14:45:21 : IncidentReport_123_New incident_01-01-2020 2-00-00 PM_01-02-2020 2-45-21 PM . 8-15-2022 2-45-21 PM
-->
<add key="IncidentReportFileName" value="Default">
</add>
<!--  * Default: "Default" - report name + date and time.
* Supported params: ReportType (periodic report, incident statistics report), startTime - Report data start time, endTime - Report data end time
* Format example 1: %startTime%_%endTime% .
* Output will be $startTime_$endTime .
* Output with values: for the values startTime = 01/01/2020 09:45:30,
   endTime = 01/02/2020 14:45:21: 01-01-2020 9-45-30 AM_01-02-2020 2-45-21 PM .
* Format example 2: %ReportType%_%startTime%_%endTime% .
* Output will be be $ReportType_$startTime_$endTime .
* Output with values: for the values startTime = 01/01/2020 09:45:30,
   endTime = 01/02/2020 14:45:21 PM ReportType: Peridoic report : Peridoic report_01-01-2020 9-45-30 AM_01-02-2020 2-45-21 PM .
* Also supports Incident statistics report-->
<add key="PeridocReportFileName" value="Default">
</add>
<add key="ReportsCulture" value="en-US">
</add>
<add key="channel" value="tcp">
```

2. Restart Situator Services and *Control Room*.
3. Generate a report via *ControlRoom* or *Planning Tool* task.

   - The report is saved as an attachment with the customized filename format as configured above:



   - A copy of the report is stored with the customized format in *C:\Program Files (x86)\Qognify\Situator\Server\Operational\bin\Reports\date*:



3.14 Report Snapshots and Maps

By default, Situator disables snapshots and map images from appearing in reports generated by Client and Server. System administrators can enable snapshots and map images in the configuration file *ActionsConfiguration.xml*.

To enable report snapshots – server:

1. Open the configuration file *ActionsConfiguration.xml* in both the *Operational bin* folder and *Sensor Server bin* folder.
2. Locate the line `<TakeSnapshots>false</TakeSnapshots>` in both.
3. Change `false` to `true` in both.
4. Save both configuration files.

To enable report snapshots – client:

1. Open the configuration file *ActionsConfiguration.xml* in the *Control Room\bin* folder.
2. Locate the line `<TakeSnapshots>false</TakeSnapshots>`.
3. Change `false` to `true`.
4. Save the configuration file.

To enable report map images:

1. Copy all site maps found in the *Control Room\bin\Map* folder.
2. Create a new folder named **Map** under *<Program Files>\Situator\Operational\bin*, and paste all map images in it.
3. Restart the Sensor Server and Notification Server.

3.15 Zone Transparency Display on Maps

By default, camera FOV and zone fill-color on maps are hidden, and only the border is displayed. This improves performance when there are many zones and/or FOVs drawn on maps.

System administrators can enable camera FOVor zone fill-color in the configuration file *Stabilis.Situator.ControlRoom.UI.exe.config*, located in the Control Room *bin* folder.

To configure camera FOV/zone camera fill-color transparency:

1. Open the Stabilis.Situator.ControlRoom.UI.exe.config configuration file.
2. Locate the add key **"ESRI_EnableFOVAndZoneTransparency"** and change the value to "**true**".

```
<!-- Specifies if the FOV Geometries will be half transparent -->
<add key="ESRI_EnableFOVAndZoneTransparency" value="true">
</add>
```

3.16 Communication Settings Configuration

Situator supports the following communication providers:

- Skype -To use Skype communication, you need to purchase SkypeOut credits. (When upgrading your Skype version, it is recommended to uninstall and then reinstall Skype.
- SIP (Session Initiation Protocol) - SIP is a signaling protocol, widely used for controlling multimedia communication sessions such as voice and video calls over Internet Protocol (IP). Situator supports SIP enabling users to make outbound SIP calls from the Phone Dialer or to initiate a SIP call to a user directly from a map. The SIP Gateway must be installed prior to the configurations below. Refer the SIP Gateway Setup help file for installing the SIP Gateway or contact your Situator representative.
- Dial-up modem
- Cisco IP Communicator

Situator's communication configurations are defined in the following configuration files:

- *Orsus.Situator.OperationalService.exe.config*
- *Stabilis.Situator.ControlRoom.UI.exe.config*

The hang-up interval option allows you to control the interval in seconds before the modem hangs up the call. Hang-up intervals can be configured in the Situator database.

To configure communication settings:

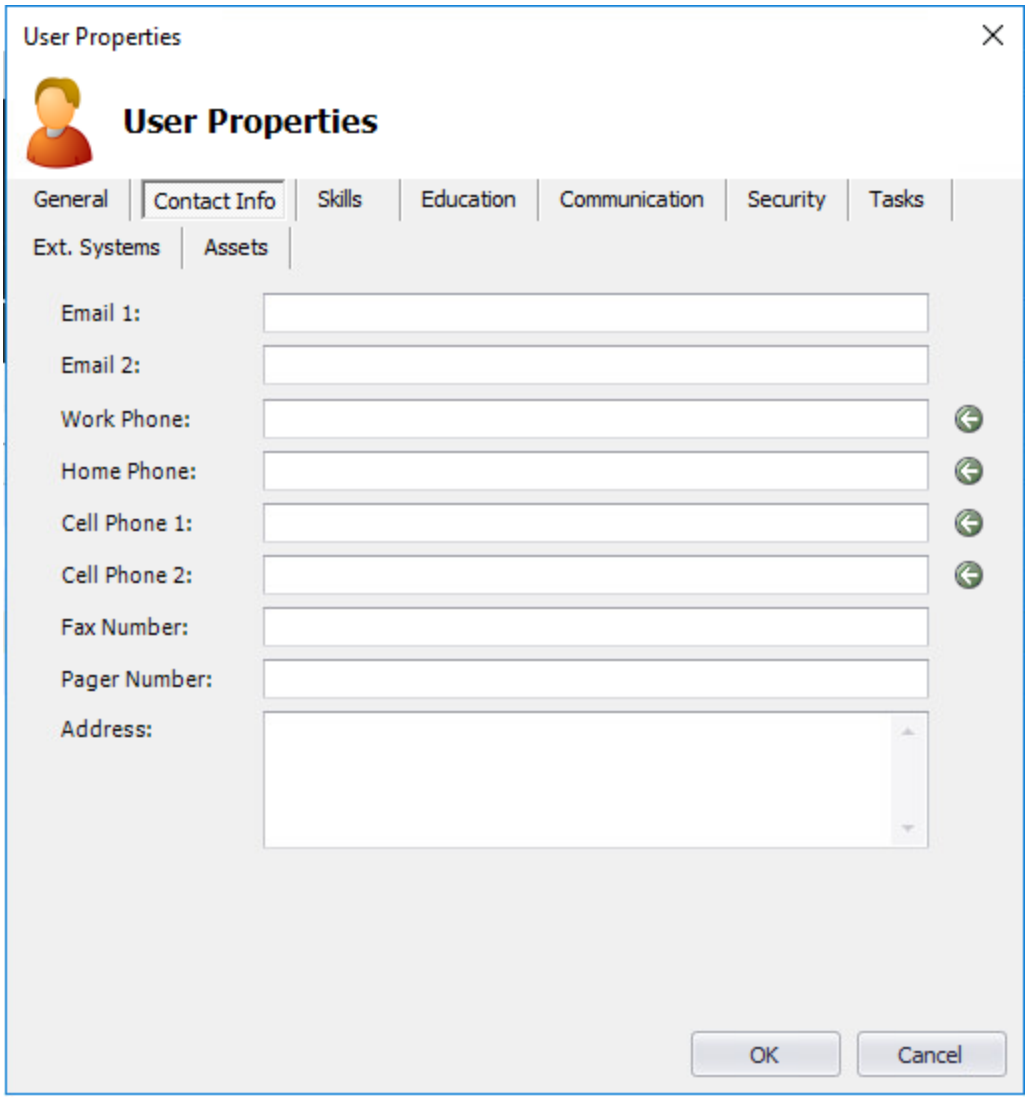
1. Open the Control Room configuration file *Stabilis.Situator.ControlRoom.UI.exe.config*, and set the phone provider as follows:
   - For a dial-up modem: <add key="PhoneProvider" value="DialupModem"/>
   - For a Skype connection: <add key="PhoneProvider" value="Skype"/>
   - For a SIP connection: <add key="PhoneProvider" value="SIP"/>

   NOTE: Make sure the SIP Provider (*Qognify.Situator.SIPProvider.dll*) is in the *ControlRoom\bin\PhoneProviders* folder.

» For a Cisco IP Communicator connection: <add key="PhoneProvider" value="Cisco"/>

> NOTE: Make sure the Cisco Provider (*Qognify.Situator.CiscoPhoneProvider.dll*) is in the *ControlRoom\bin\PhoneProviders* folder.

2. Restart *Control Room*.
3. In *Control Room*, in the *Administration* view *Users* workspace, right-click the user for whom you want to add contact details and then select **Properties**. The *Users Properties* dialog box opens.
4. Select the **Contact Info** tab.
5. Type the user's phone number into the appropriate fields. If a prefix number is required to "dial out" of an organization, the user's phone number should begin with that number.

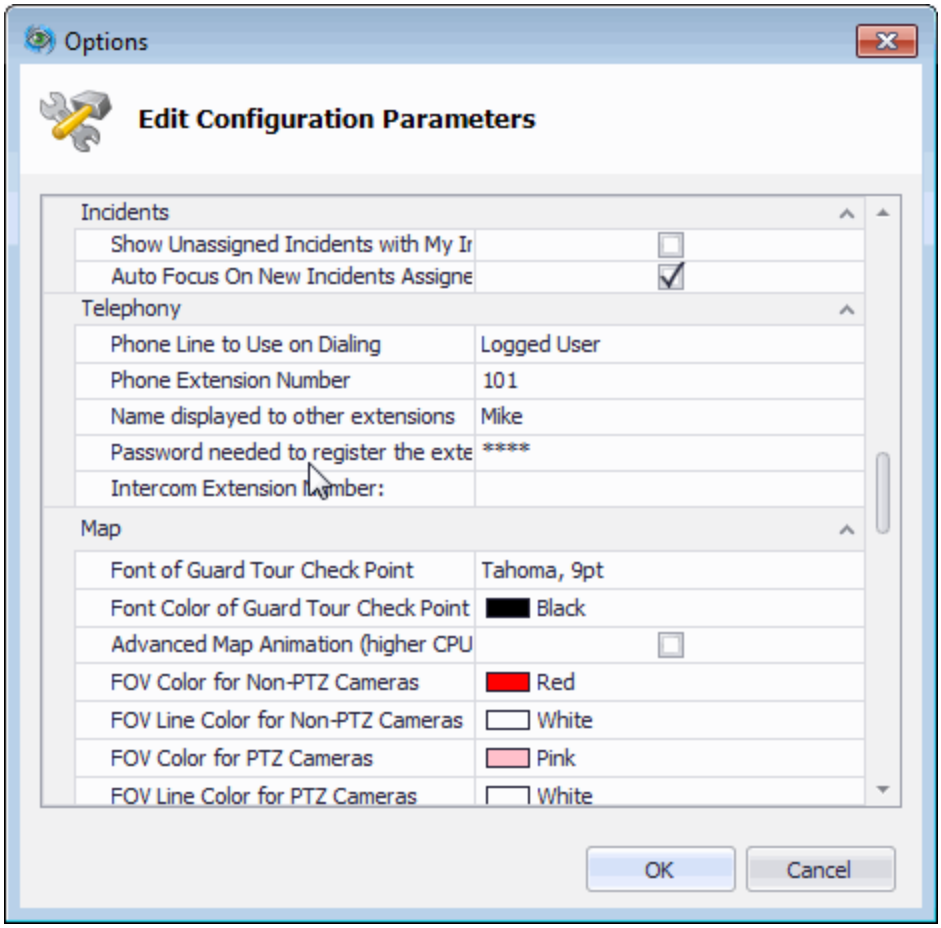


To configure SIP phone extension parameters in Control Room:

Most of the SIP configuration fields are taken from the *dbo.SensorGatewayParams* table from the "SIP Gateway" row. However, phone extension parameters such as username, password, and display name are defined in *Control Room*.

1. In *Control Room*, from the **Tools** menu, select **Options**. The *Options* dialog box opens.
2. In the **Telephony** group, add the **SIP** parameter configurations:
   - In **Phone Line to Use on Dialing**, select **This Terminal**.
   - In **Phone Extension Number**, add the number of the SIP extension.

- In **Name displayed to other extensions**, add the name you would like to be displayed to other SIP extensions when making an outgoing call.
- In **Password needed to register the extension to PBX**, type a password.



To configure the hang-up interval option:

1. Do one of the following:
   - In the *Planning Tool***Advanced Setup** view, expand the **PhoneProviders** section.
   - Use a database management tool to open the *dbo.SystemConfiguration* table.
2. For the parameter **HangUpInterval**, under the *Value* column, set the value (in seconds) as required.
3. Click **Save**.
4. Restart the OpService for the change to be applied.

3.17 Video Analytics Context Menu Options

When integrated with a supported video analytics system, suspicious movement in the camera feed is pinpointed with an alerting visual "ellipse" and users can use on-screen operations to notify the video analytics recorder that a detected object is considered suspicious or to remove the ellipse and move the object to the background.

The context menu options are facilitated in *Control Room* by moving the *VideoAnalysisConfiguration.xml* into the Control Room *bin* folder.

3.18 Intercom Call Management

To disable an "on hold" phone state in Situator, you must modify the *Control Room* configuration file *Stabilis.Situator.ControlRoom.UI.exe*.

To disable placing calls on hold:

1. Open the *Stabilis.Situator.ControlRoom.UI.exe* file in the Control Room bin folder: *<Your local or network drive>:\Program Files\Qognify Systems\Situator\ControlRoom\bin*.
2. Find the **IntercomCallManagement_HoldFeatureEnabled** add key and change the value to **false**.

3.19 Mass Notification Message Configurations

When an external Mass Notification System (MNS) is installed, the following configuration options are available:

- Delivery-status parameter names translation: the delivery statuses returned from the external MNS are translated into Situator *Control Room* display names. The translation is performed according to a configurable .xml file.
- Configuring a message template - create a template (in HTML format) so that it displays as an available message template parameter option in the Send Message BPM action.
- Message Group Recipient Status Calculation: parameter **IsAllGroupMembersMustAcknowledge.**

These configurations are described in the next sections.

3.19.1 XML Translation Files

The XML translation file converts the statuses returned from an external Mass Notification (MN) system to appropriate user-friendly names for display in Situator *Control Room*. The display names for each returned status can be customized in the xml file.

For example, the MiR3 system returns a status called "CONFIRMATION" that is translated to the Situator status name "Acknowledged". You can edit these parameters in the file *MirMassNotificationConfiguration.xml*, as shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
- <MirMassNotification xsi:noNamespaceSchemaLocation="MirMassNotificationConfiguration.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    - <Settings>
        <Connection ConnectionConfigFilePath="MirMassNotification.config" BindingName="mir3"/>
      - <Message Locale="nl_NL">
          - <AcknowledgeRequests>
              <AcknowledgeRequest Locale="nl_NL" Text="Click to send acknowledge"/>
            </AcknowledgeRequests>
        </Message>
      - <Mapping>
          - <Severity>
              <SeverityType MirName="Lowest" Name="VeryLow"/>
              <SeverityType MirName="Low" Name="Low"/>
              <SeverityType MirName="Medium" Name="Medium"/>
              <SeverityType MirName="High" Name="High"/>
              <SeverityType MirName="Highest" Name="VeryHigh"/>
            </Severity>
          - <DeliveryState>
              <DeliveryStatus SituatorStatus="Error" MirStatus="AUTO_RESPONSE"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="BUSY"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="CALL_FAILED"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="CALL_REJECTED"/>
              <DeliveryStatus SituatorStatus="Acknowledged" MirStatus="CONFIRMATION"/>
              <DeliveryStatus SituatorStatus="Waiting" MirStatus="CONNECTED"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="CONNECTED_COMPLETE"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="DESKTOP_SENT"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="DISCONNECTED"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="DISCONNECTED_COMPLETE"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="EMAIL_SENT"/>
              <DeliveryStatus SituatorStatus="Waiting" MirStatus="FAX"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="FAX_SENT"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="INVALID_PHONE_NUMBER"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="INVALID_RESPONSE"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="LEFT_MESSAGE"/>
              <DeliveryStatus SituatorStatus="Waiting" MirStatus="MACHINE_CONNECTED"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="MACHINE_DISCONNECTED"/>
              <DeliveryStatus SituatorStatus="Waiting" MirStatus="MACHINE_UNDELIVERED"/>
              <DeliveryStatus SituatorStatus="Waiting" MirStatus="NETWORK_BUSY"/>
              <DeliveryStatus SituatorStatus="Waiting" MirStatus="NO_ANSWER"/>
              <DeliveryStatus SituatorStatus="Waiting" MirStatus="NO_RESOURCES"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="NOT_HERE"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="NOT_OPTED_IN"/>
              <DeliveryStatus SituatorStatus="Acknowledged" MirStatus="NOTIFICATION_RETRIEVAL"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="OPERATOR_INTERCEPT"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="OTHER_MACHINE"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="PAGE_REJECTED"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="PAGE_SENT"/>
              <DeliveryStatus SituatorStatus="Acknowledged" MirStatus="RESPONDED"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="RESTRICTED_TELNO"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="RIMP2P_REJECTED"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="RIMP2P_SENT"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="SENDING_FAILED"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="SMS_DEACTIVATED"/>
              <DeliveryStatus SituatorStatus="Acknowledged" MirStatus="SMS_HANDSET_ACK"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="SMS_REJECTED"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="SMS_SENT"/>
              <DeliveryStatus SituatorStatus="Acknowledged" MirStatus="SMS_SERVER_ACK"/>
              <DeliveryStatus SituatorStatus="Acknowledged" MirStatus="SMS_SERVER_C_ACK"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="TDD_CONNECTED_COMPLETE"/>
              <DeliveryStatus SituatorStatus="Waiting" MirStatus="TDD_CONNECTED_PARTIAL"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="TDD_DISCONNECTED_COMPLETE"/>
              <DeliveryStatus SituatorStatus="Arrived" MirStatus="TDD_DISCONNECTED_PARTIAL"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="TDD_UNDELIVERED"/>
              <DeliveryStatus SituatorStatus="Waiting" MirStatus="UNKNOWN_STATUS"/>
              <DeliveryStatus SituatorStatus="Error" MirStatus="WRONG_ADDRESS"/>
            </DeliveryState>
        </Mapping>
      - <Search>
          <UserDisplaying>First-LastName</UserDisplaying>
        </Search>
    </Settings>
</MirMassNotification>
```

3.19.2 Configuring a Message Template

As part of the Messaging feature, you need to configure a template (in HTML format) so that it displays as an available message template parameter option in the Send Message BPM action.

To insert a template:

1. Navigate to *\Qognify\Extra\DBMaintain\MessageTemplate*.
2. Select **InsertMessageTemplate**. The *Windows SQL Database* window opens.

   NOTE: Make sure that the OpServer is down when making any changes to the MessageTemplate table.

3. Fill in the following fields:
   - MyTemplate
   - Put Description in here
   - Put subject in here
   - TemplatePath

   NOTE: For the **Template Path**, it is recommended to use a designated folder on the database server.

4. Copy the (HTML) template to the selected path.
5. Select **Execute SQL**.
6. In the Situator database tables, select **dbo.Messagetemplates**. All the templates that you added appear in the list. When there is more than one template, you can select the one to be used as the default message template, by configuring the **IsDefault** variable to **True**.

3.19.3 Message Group Recipient Status Calculation

The delivery status of Mass Notification (MN) message group is calculated based on the member statuses according to the value of the parameter **IsAllGroupMembersMustAcknowledge**, located in *dbo.NotifyDataConfigurations* database table, as described in the table below.

Group member statuses	IsAllGroupMembersMustAcknowledge value	
	True	False (default value)
All members in error state	Failed	Failed
Some group members in error state (but not all)	Failed	PendingAcknowledged\ Acknowledged Depending on whether there is an acknowledged member
Some group members acknowledged (but not all)	PendingAcknowledged	Acknowledged
All members acknowledged	Acknowledged	Acknowledged
All members Waiting\Arrived	PendingAcknowledged	PendingAcknowledged

3.19.4 Defining Polling Interval Between Processing Messages Sent to Mass Notification System (MNS)

When messages are sent to the MNS from *Control Room*, the message status changes to “Sent” when it reaches the driver and then ”Pending Acknowledged” when it reaches the MNS. In the Situator database, you can configure the polling interval time between processing messages.

To configure the polling interval between processing messages:

1. Using your database management tool, open the Situator database tables, and then open the **dbo.NotifyDataConfigurations** table.
2. Locate the **MonitorMessageIntervalInSeconds** parameter. In the *ParamVal* column, type a time interval (in seconds).

   NOTE: The recommended interval is 5 seconds. This interval should also be set in the **MonitorMessagesInterval** parameter in the **dbo.SystemConfiguration** table. Refer to Database System Configuration Table.

3. Locate the **WillMonitorSentStatus** parameter. In the *ParamVal* column, type **True**.

3.20 GIS Sensor and FOV Tables

GIS services data used for displaying sensors and sensor FOVs is in two tables:

- **dbo.ArcGISSensorsView**: lists all sensors that can be displayed on the map
- **dbo.ArcGISFieldofViewView**: lists each sensor and its FOV

To display sensors, FOVs, views, incident markers, the parameter **duplicateArcGISTables** must be **True** (default) in the database system configuration table.

3.21 Actions Collaboration Behavior

Default collaboration behavior is defined in the table **dbo.Actions**.

Tne variable **IsOwner** applies to Assignees and Stakeholders:

- Value=0: cannot update
- Value=1: can update

3.22 Disabling the New Incident Button in the Navigation Bar

Usually, the authorization to create new incidents is assigned based on individual user types, in the Administration view Authorization workspace. However, it is possible to set the client configuration such that the New Incident button is hidden for a specific client.

To disable and hide the New Incident button in the Control Room navigation toolbar:

1. Locate the file *Stabilis.Situator.ControlRoom.UI.exe.config* in the Control Room *bin* folder, and open in edit mode.
2. Locate the line `<add key="HideNewIncidentIcon" value="False"></add>`.
3. Change its value attribute to "**True**".
4. Save the file.
5. Restart *Control Room*.

3.23 Configuring the Number of Pop-up Notifications

You can configure the maximum number of pop-up notifications that can simultaneously display on a workstation. The default number is 5 (five).

To configure the maximum number of simultaneously displayed notifications:

1. On the workstation to be configured, navigate to *C:\Program Files (x86)\Qognify\Situator\ControlRoom\bin* and open the file *Stabilis.Situator.ControlRoom.UI.exe.config*.
2. In the field, **<add key="MaxNotificationsOnScreen" value="5">**, change the value to the maximum number that you want. The recommended maximum is 7 (seven), to avoid extending into the application toolbar.
3. Save the file.
4. Start *Control Room*.

3.24 Configuring Video Slot Maximize Button

By default, *Control Room* is configured to display a **Maximize** button in the video slot header. This button opens the video slot in full screen.

You can change the configuration so that an operator can also double-click a video slot to view it full screen. The behavior is determined by the parameter **EnableVideoSlotMaximizeOnDoubleClick** in the file *Stabilis.Situator.ControlRoom.UI.exe.config*.

To configure the video slot full screen options:

In the *Stabilis.Situator.ControlRoom.UI.exe.config* file, change the value of the parameter **EnableVideoSlotMaximizeOnDoubleClick** for the desired behavior:

- **<add key="EnableVideoSlotMaximizeOnDoubleClick" value="False">**: The default value, only the **Maximize** button will expand the video slot to full screen.
- **<add key="EnableVideoSlotMaximizeOnDoubleClick" value="True">**: Video slot can be expanded to full screen either by the **Maximize** button or by double-clicking in the video slot.

CHAPTER 4 Database SMTP Server Configuration

Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (e-mail) transmission across IP networks.

To facilitate e-mail transmission in Situator using the SMTP protocol, the following configurations need to be made in the Situator database.

To configure the SMTP server:

1. Open the table *dbo.NotifyDataConfigurations*.
2. In the `"SMTPSenderEmail"` row, type a valid user email address in the *ParamVal* column.
3. In the `"SMTPHost"` row, type a valid organization SMTP Server address in the *ParamVal* column.
4. In the `"SMTPUser"` row, type the valid SMTP Server username in the *ParamVal* column.
5. In the "`SMTPPass`" field, type the valid SMTP user password in the ParamVal column.

CHAPTER 5 LRAD Sensor Configurations

After installing the LRAD gateway:

- Verify that the sensor feature SQL database script was run.
- Add the desired MP3 files you want to play while using the **Hail Recorded Broadcast** option to the *GatewaysHost* folder.

To run the sensor feature script:

1. Add the LRAD camera feature and feature template:

   INSERT INTO [SensorFeaturesRepository] ([Name], [Description], [Category]) VALUES ('Camera\LRAD',NULL,0)

   INSERT INTO [SensorFeaturesTemplates] ([Name], [TemplateType]) VALUES ('LRAD Camera', 1)

   INSERT INTO [SensorFeaturesTemplateItems] ([TemplateID],[FeatureID]) VALUES (IDENT_CURRENT ('SensorFeaturesTemplates'),IDENT_CURRENT('SensorFeaturesRepository'))

   INSERT INTO [SensorFeaturesTemplateItems] ([TemplateID],[FeatureID]) VALUES (IDENT_CURRENT ('SensorFeaturesTemplates'),1)

   INSERT INTO [SensorFeaturesTemplateItems] ([TemplateID],[FeatureID]) VALUES (IDENT_CURRENT ('SensorFeaturesTemplates'),2)

2. Add new action to operate LRAD camera:

   ```
   DECLARE @user_acl_id int
   ```

3. Define new action:

   INSERT INTO [Actions] ([ActionKey],[ActionName],[ActionCategory],[Description],[AdminOnly], [IsTypeAllowed],[IsTyped]) VALUES ('OperateLRAD','Operate LRAD Camera',4,NULL,0,0,0)

4. Authorize the users to use the action:

   SELECT @user_acl_id = AclID FROM ACLs WHERE AclName = 'Administrator'

   INSERT INTO [AclActionPermissions] ([AclID],[ActionID],[ActionTypeID])VALUES(@user_acl_id,IDENT_ CURRENT('Actions'),NULL)

   SELECT @user_acl_id = AclID FROM ACLs WHERE AclName = 'User'

   INSERT INTO [AclActionPermissions] ([AclID],[ActionID],[ActionTypeID])VALUES(@user_acl_id,IDENT_ CURRENT('Actions'),NULL)

GO

> NOTE: After running the script, restart the client and server.

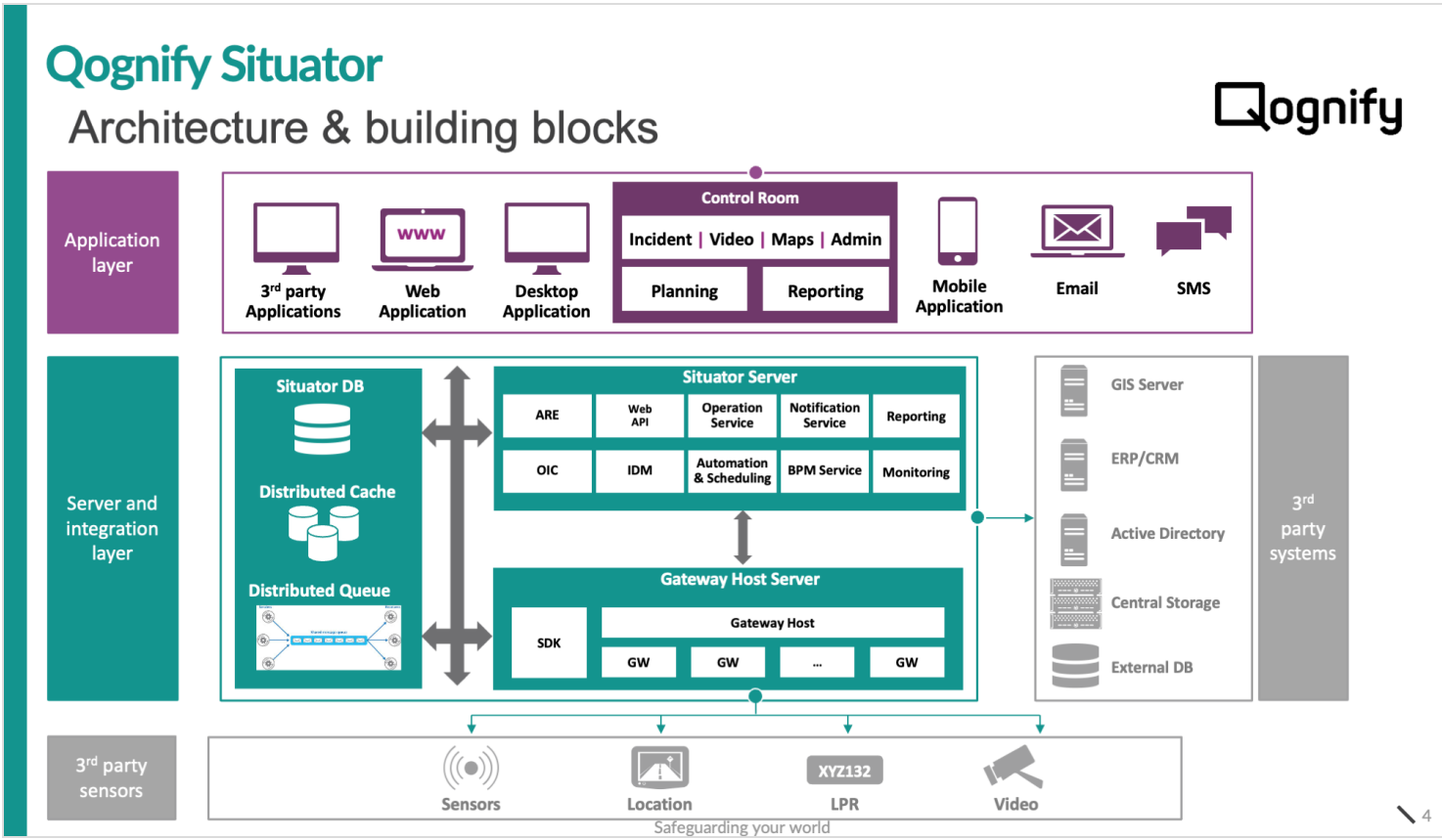
To add the desired MP3 files you wish to play while using the Hail Recorded Broadcast option in Situator:

1. Navigate to \\..*Program Files**Qognify**Situator**GatewaysHost*\.
2. Move the desired MP3 files into the *GatewayHosts* directory.

CHAPTER 6 Server and Client Port Communication Definitions

6.1 Situator Network Connectivity

The following diagram shows a typical Situator Network Connectivity implementation.



6.2 Network Ports

For a complete list of the default incoming ports and connection types each component uses, refer to the *Defining Network Ports* in the *Situator System Requirements Guide*.

6.3 Switching Clients between Servers

Using the Situator Clients Environment Configurator tool, installers can capture configuration details of servers within a Situator environment (OP, WebAPI, DB, Messaging, etc.) and then easily connect Situator client applications to a different environment.

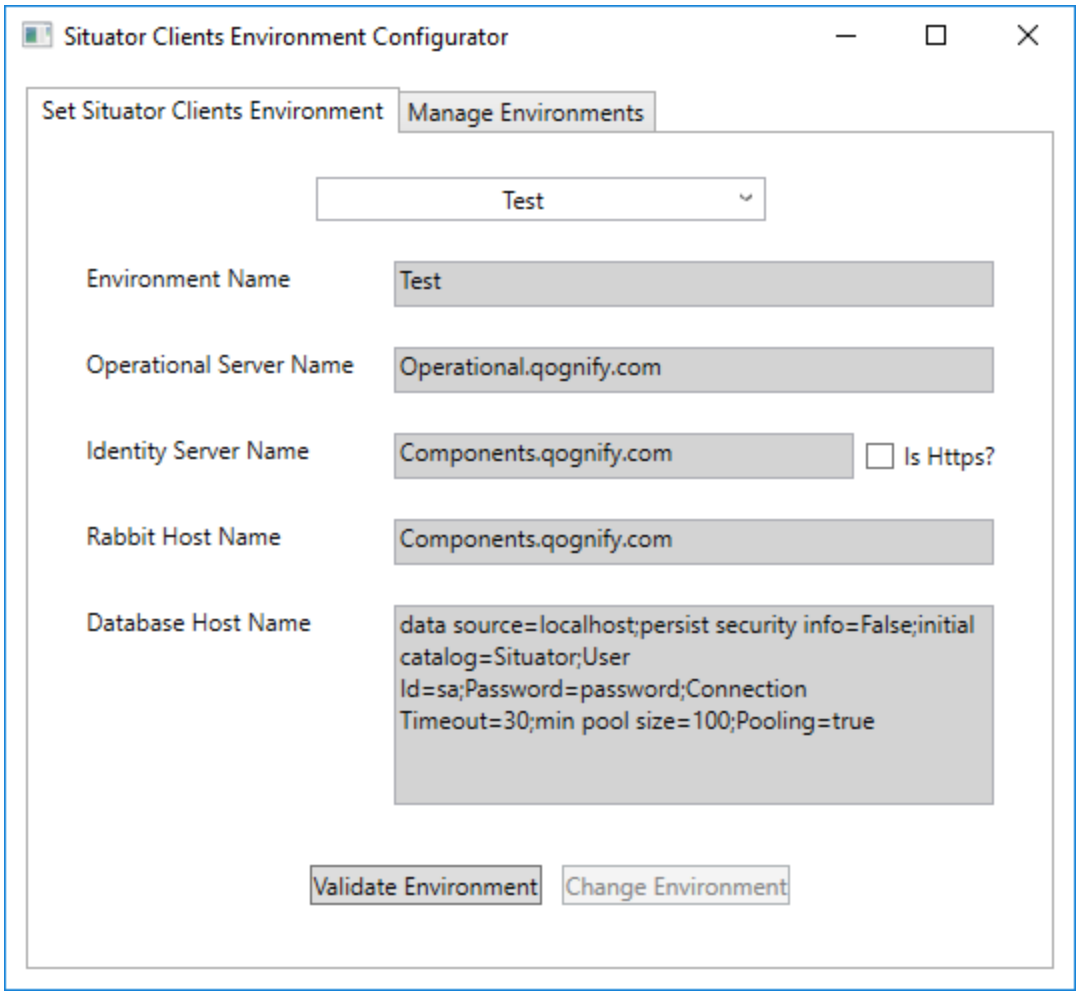
The tool is located in the *Extra/SituatorClientsEnvironmentConfigurator* folder in your Installation directory.

NOTE: *Control Room* and *Planning Tool* must be closed before changing environments. If you try to open the tool while one or both of the client applications is open, you will be prompted by a message to first close the applications before running the tool.

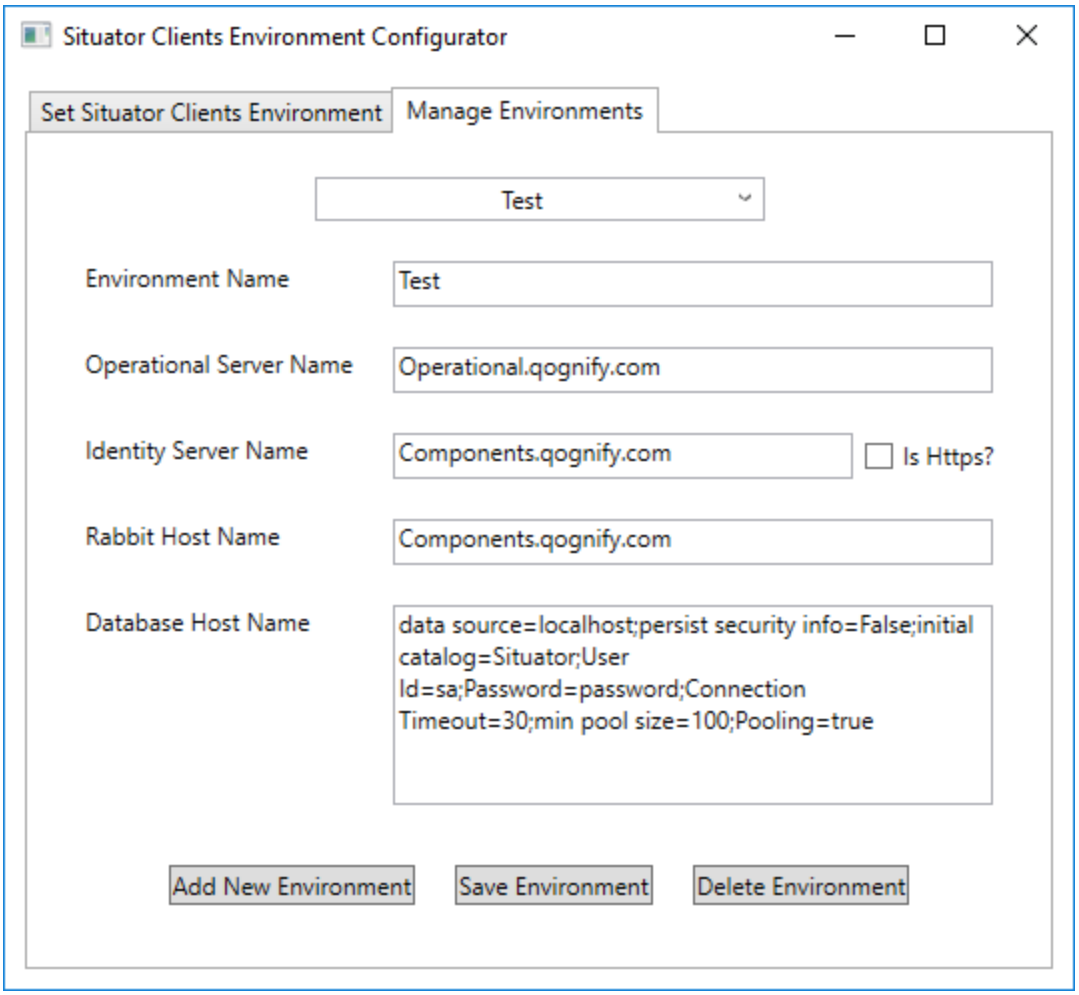
You must first add environment details to the tool.

To add environment details:

1. Open the *SituatorClientsEnvironmentConfigurator* folder and double-click **SituatorClientsEnvironmentConfigurator.exe**.
2. If the User Account Control (UAC) setting is enabled, a message apppears: “Do you want to allow the following program to make changes to this computer?” Click **Yes** to continue. If the UAC is disabled, the user will not be prompted. The tool opens with the **Set Situator Clients Environment** tab in focus.

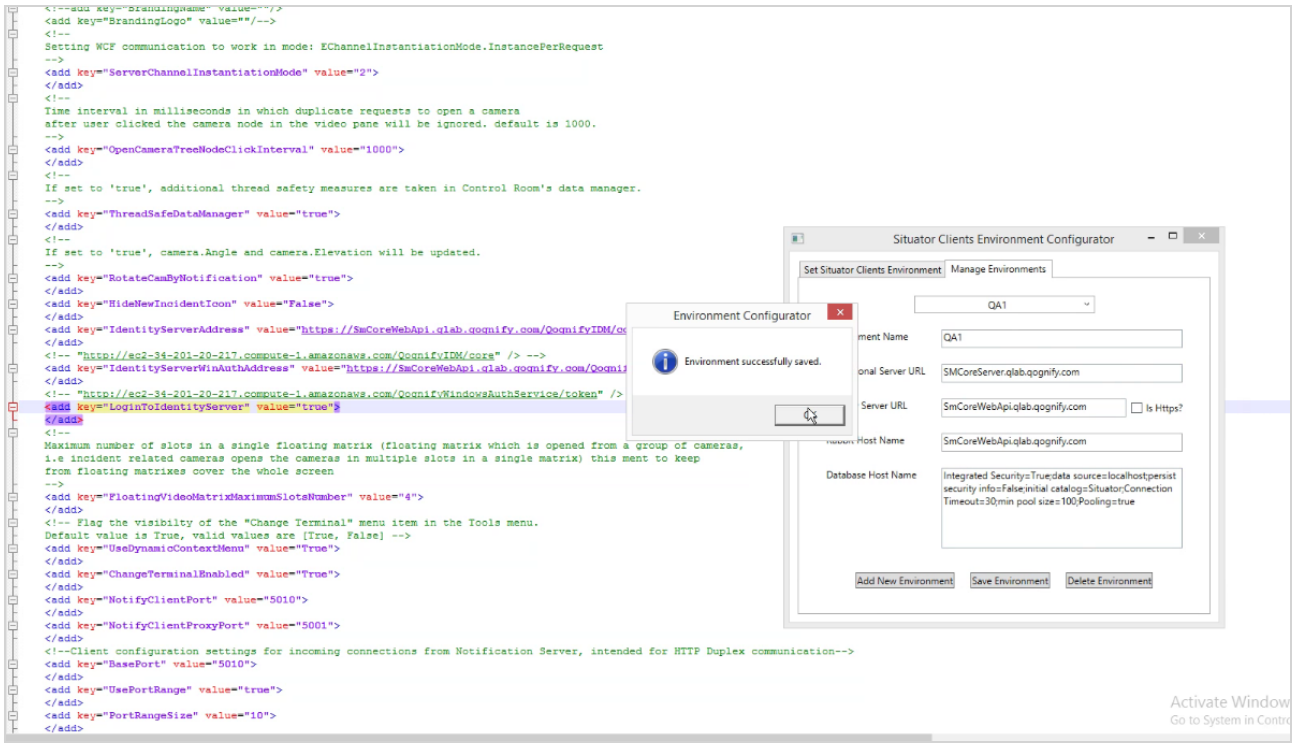


3. Select the **Manage Environments** tab.



4. Click **Add New Environment**.
5. In the **Environment Name** field, change the default name to a logical name for the environment. For example, Maintenance, Production, etc.
6. In the **Operational Server Name**, **Identity Server Name** and **Rabbi Host Name** fields, type the IP addresses or full names of the servers.

   If the clients connect to the Identity Server by Https, select the **Is Https?** check box. Once you save your environment details, the URL to the Identity Server will automatically change to https in the *Stabilis.Situator.ControlRoom.UI.exe.config* configuration file, as shown in the following figure.



7. In the **Database Host Name** field, in the connection string, replace the following:
   - `data source 'local host'` with the correct IP address or full name of the server
   - `correct initial catalog name, default is 'Situator'` with the correct catalog name
   - UserID `'sa'` with the correct SQL user
   - `'password'` with the correct SQL user password
8. After editing all your entries, click **Save Environment**. A successful environment confirmation message opens.
9. Click **OK**.
10. Repeat steps 4-9 to add additional environments.
11. To delete environments, in the drop-down field, select the environment you want to delete and click **Delete Environment**.

To switch client applications to a different environment:

1. Select the **Set Situator Clients Environment** tab.
2. In the **Environments** drop-down list, select the environment you want to switch to. The fields and connection string are populated with environment details.

3. Click **Validate Environment**.
   - If the connection details are incorrect, a connection error message prompts you. Click **OK** and fix the environment details.
   - If the connection details are correct, click **Change Environment**.

CHAPTER 7 Monitoring Server Services

7.1 Overview

Situator has a built-in watchdog tool that monitors server software components according to administrator definitions. Upon identifying a faulty process, the monitoring tool can be configured to restart software and send notifications.

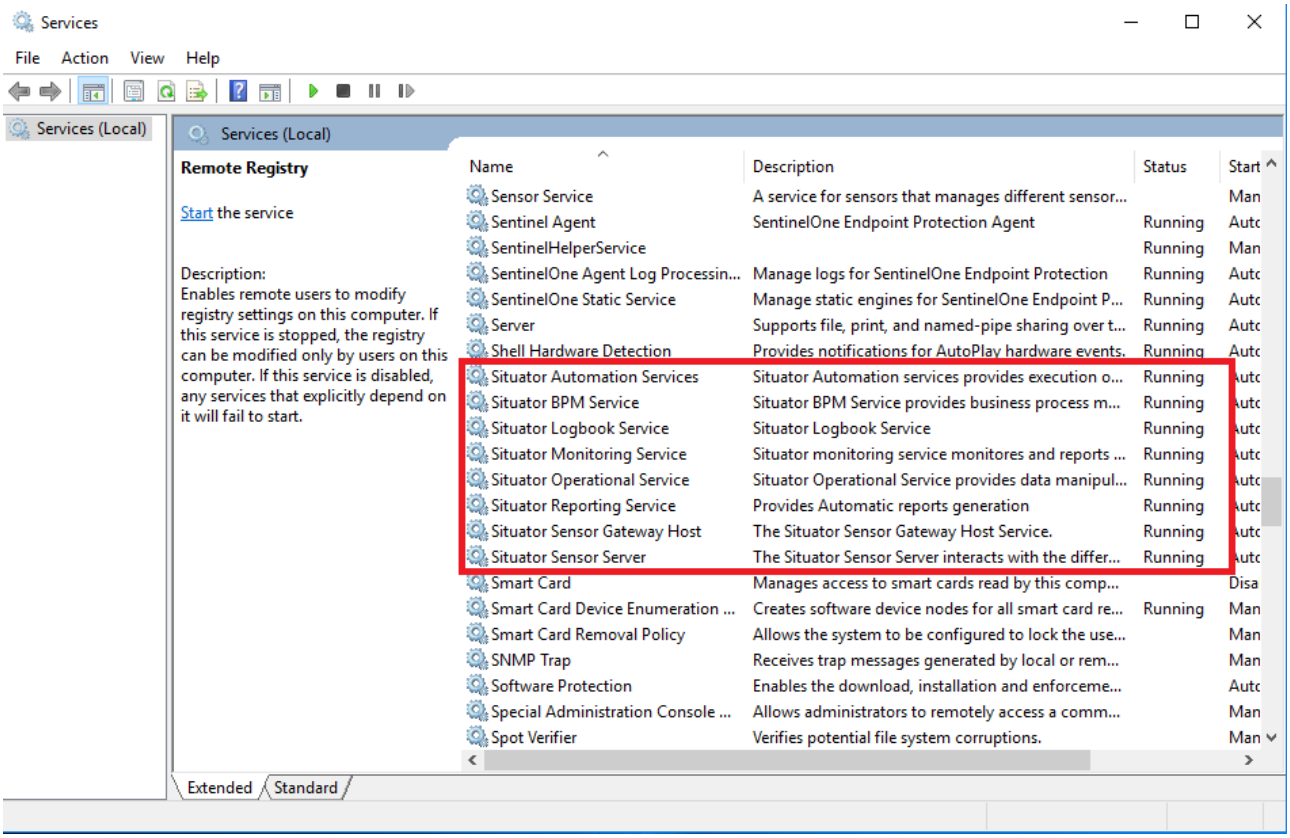
In addition, the monitoring service also sends data to the *Qognify Umbrella* health monitoring dashboard. The main purpose of the monitoring service in Umbrella is to provide faster information from sites, especially related to the cameras, services and systems. Refer to Monitoring Situator Services with Umbrella on page 58.

The monitoring definitions are kept on the Operational server in the XML configuration file *MonitoringServiceConfiguration.xml*, located in the *Operational bin* folder. The tool has a simple configuration interface.

The following servers provide various services to Situator clients:

- Situator Server
- Situator Gateway Host Server
- Situator Web API Server

A description of the Situator Server Services is provided in the figure and table below, and in the following sections.



Service Name	Process Name	Description
Situator Operational Service	Orsus.Situator.OperationalService.exe	This service handles most of the internal business logic and all database operations needed by the clients (with the exception of planning tool and certain gateways).
Situator Automation Service	Stabilis.Situator.AutomationServices.exe	This service handles all automatic scheduled tasks functionality (timing an action to specific time of day or in fixed intervals), It is also responsible for escalation system of incidents, tasks and triggers.

Service Name	Process Name	Description
Situator Monitoring Service	Stabilis.Situator.MonitoringService.exe	This is a watchdog service responsible for monitoring and recovery of all other services, as well as sending notifications about failed services.
Situator BPM Service	Orsus.Situator.BPMService.exe	This service is a Business Process Management engine, running BPM rules and workflows.
Situator Reporting Service	Stabilis.Situator.ReportingService.exe	This service generates all reports throughout the Reporting Tool and automatic actions (both Crystal and DevExpress).
StreamBase Service	StreamBase 7.6 64-bit Service - sbd64.7.6 *C:\Program Files (x86)\Qognify\Situator\ARE\AREServer\bin64\sbd.exe*	This service is also referred to as Advanced Rules Engine (ARE) - a stream-based event processing engine that processes and analyzes huge amounts of data in real time.

7.1.1 Situator Gateway Host Server

This service comes in two options: 32 bit and 64 bit, as some gateway instances require 32 bit (for example NiceVision VMS GW) and other require 64 bit (for example the PicturePerfect ACS GW).

Service Name	Process Name	Description
Gateway Host Service	Orsus.Situator.Gateways.GatewaysService.exe	The purpose of this service is to host specific gateways instances (one or more instance for each sub system).

7.1.2 Situator Web API Server Services

Situator Web API is an integration layer with 3rd party systems. The integration layer allows 3rd party services to communicate with Situator via a web-based API. The Web-API module acts as the front gate of integrations into Situator. It serves as a client of the Situator server by logging in as a new special administrative user, using a Global terminal. It also connects to the notification service to publish notifications to the subscribers.

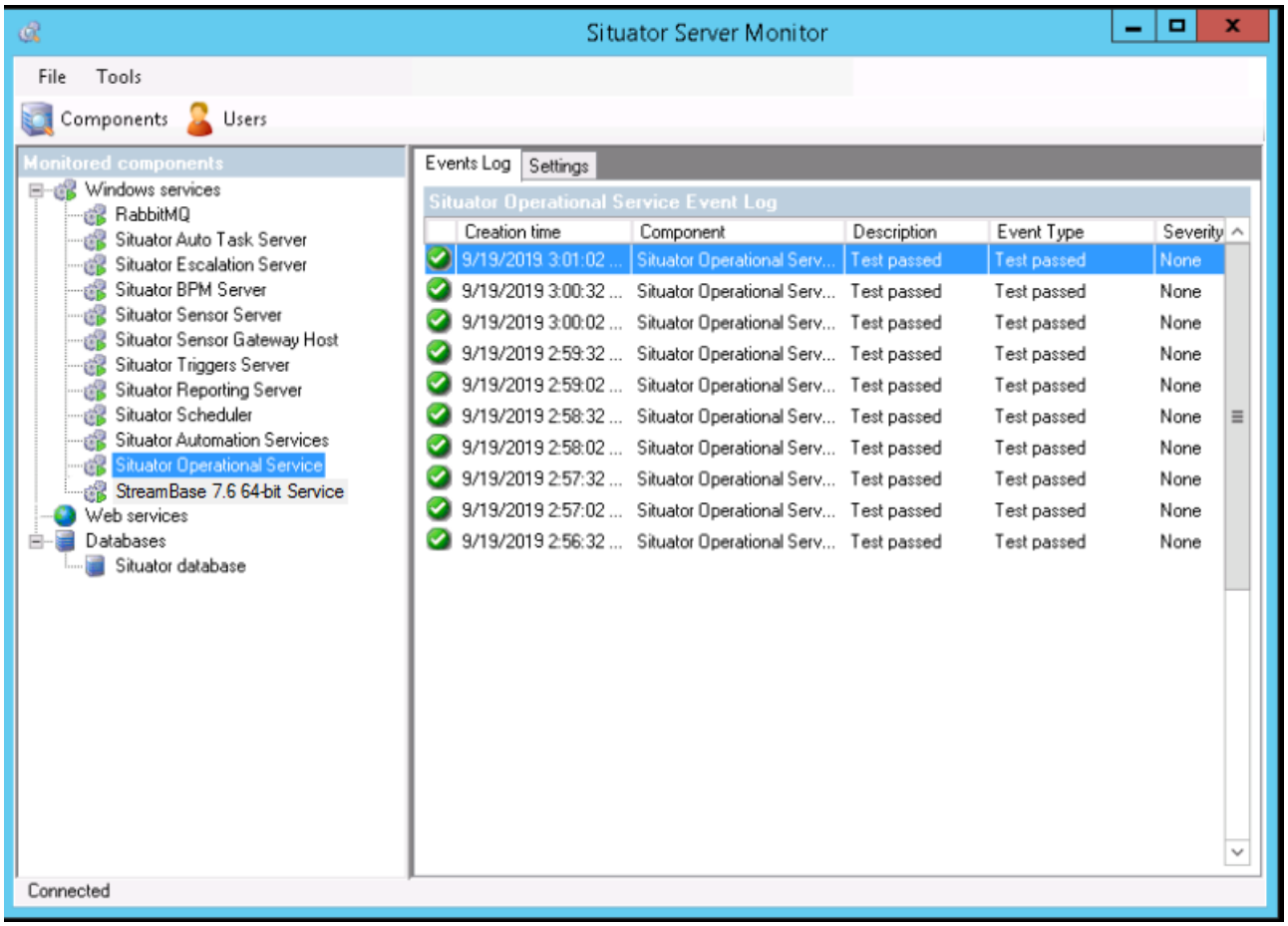
This web-based API is a two-way communication mechanism allowing services to create, retrieve, update, delete Situator entities, and to retrieve a notification on changes that occur in the system including custom update notification data files.

The API is based on Open Data Protocol (OData) that enables the creation of REST-based data services. The REST-based data services, allow resources, identified using Uniform Resource Locators (URLs), to be published and edited by Web clients using simple HTTP messages. This Help document defines a set of recommended rules for constructing URLs to identify the data and metadata exposed by an OData server as well as a set of reserved URL query string operators, when accepted by an OData server.

7.2 Setting Up the Situator Server Monitor

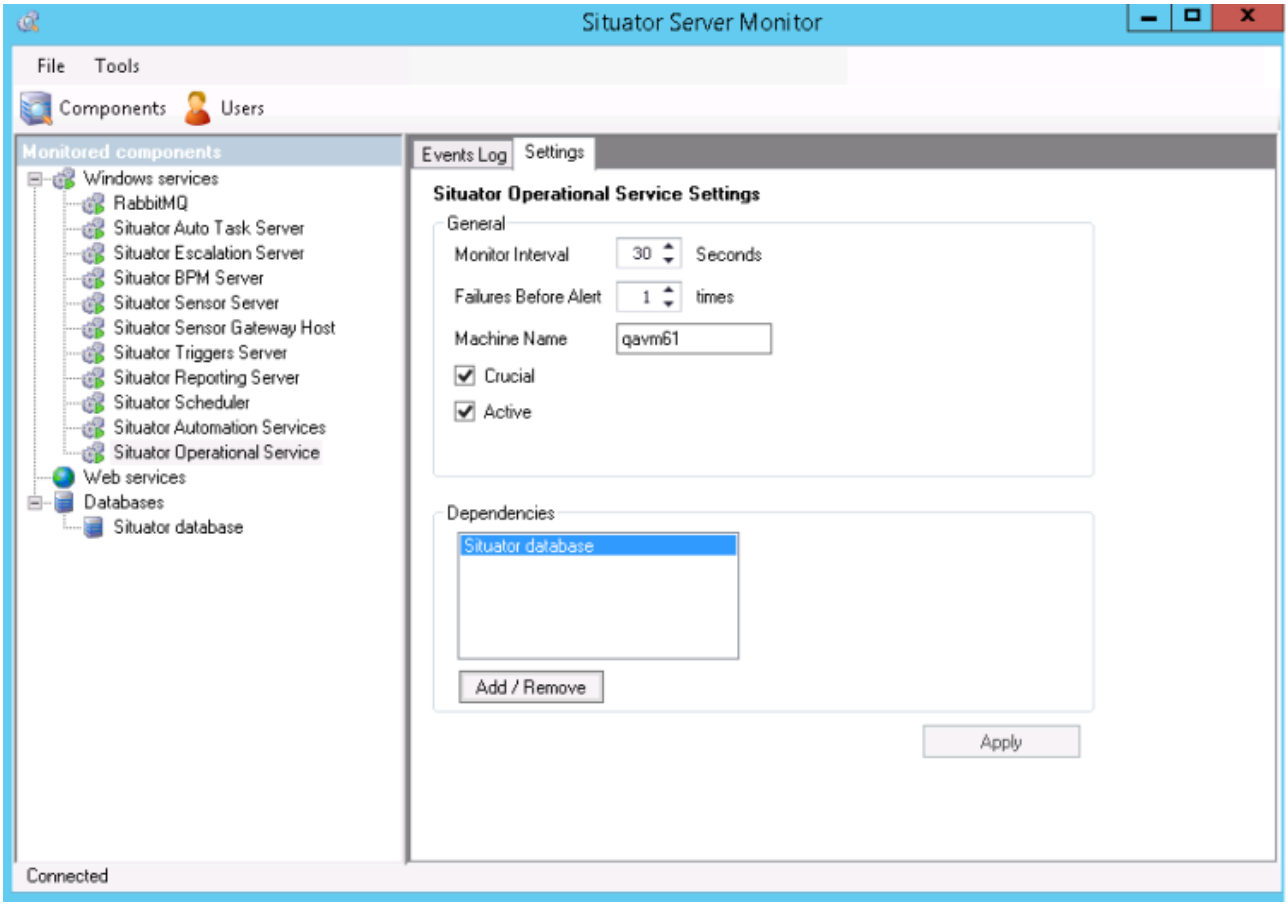
To access the Server Monitor:

Select **All Programs > Situator Server Tools > Situator Server Monitoring**. The *Situator Server Monitor* window opens, displaying the **Event Log** tab, which provides a list of all monitoring events and information on whether they succeeded or failed.


Situator Server Monitor
File
Tools
Components
Users
Monitored components
Windows services
RabbitMQ
Situator Auto Task Server
Situator Escalation Server
Situator BPM Server
Situator Sensor Server
Situator Sensor Gateway Host
Situator Triggers Server
Situator Reporting Server
Situator Scheduler
Situator Automation Services
Situator Operational Service
StreamBase 7.6 64-bit Service
Web services
Databases
Situator database
Events Log
Settings
Situator Operational Service Event Log
Creation time
Component
Description
Event Type
Severity
9/19/2019 3:01:02 ... Situator Operational Serv... Test passed Test passed None
9/19/2019 3:00:32 ... Situator Operational Serv... Test passed Test passed None
9/19/2019 3:00:02 ... Situator Operational Serv... Test passed Test passed None
9/19/2019 2:59:32 ... Situator Operational Serv... Test passed Test passed None
9/19/2019 2:59:02 ... Situator Operational Serv... Test passed Test passed None
9/19/2019 2:58:32 ... Situator Operational Serv... Test passed Test passed None
9/19/2019 2:58:02 ... Situator Operational Serv... Test passed Test passed None
9/19/2019 2:57:32 ... Situator Operational Serv... Test passed Test passed None
9/19/2019 2:57:02 ... Situator Operational Serv... Test passed Test passed None
9/19/2019 2:56:32 ... Situator Operational Serv... Test passed Test passed None
Connected

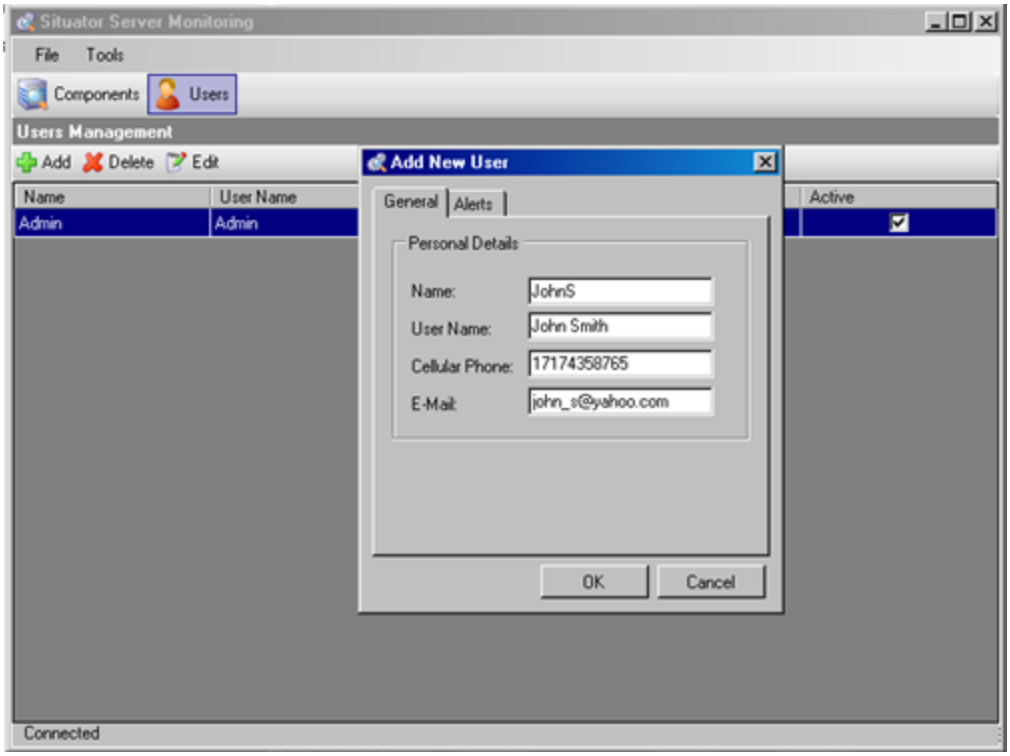
To set up monitoring:

1. In the *Situator Server Monitor* window, click the **Settings** tab.



2. If required, in the **General** section, set up monitoring intervals and number of failures before alert. It is recommended to keep the default values.
3. (Optional) Select the **Crucial** check box for the monitored software component. An indication of whether a downed component is crucial will be sent to operators in the *Control Room* per this definition.
4. (Optional) Select the **Active** check box to enable monitoring this component. If not active, this monitoring will not show in monitoring tools, so you cannot activate it, but if it was activated, you can deactivate it.
5. (Optional The **Dependencies** section provides an indication of software components that are dependent on other components to function correctly. Click **Add/Remove**. to set up dependencies.
6. Clock **Apply**.
7. To define users that will be alerted in the event of failure, do the following:

a. Click **Users** at the top of the window. The *Add New Users* window opens.
b. Enter the name(s) and contact details.
c. Click the **Alerts** tab and select the alert for the specific user(s). Refer tio .
d. Click **OK**.



7.3 Sending Alerts upon Repeated Abnormal Status

From the *Server Monitoring* tool **Alerts** tab you can configure the system to send alert e-mails or SMS to System Administrators when the Server Monitor repeatedly finds abnormal status for a system service.

To enable sending alert e-mails:

1. Configure the **SMTP**[1] settings in the configuration file *GateWaysInitializationData.xml*, located in the Operational *bin* folder:

   Under `<Provider name="SMTP">`, edit the four parameters as follows:

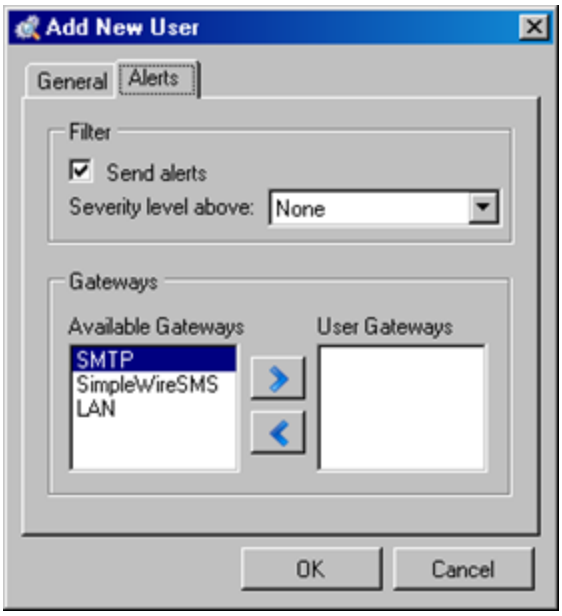
[1]Simple Mail Transfer Protocol is an Internet standard for electronic mail (e-mail) transmission across IP networks

Host	The SMTP Server name in the domain or address
User	Type the login of a user with administrative rights in the SMTP Server
Password	Type the password of the user
From	Type the E-Mail address which the alert email will be sent from

2. After editing these parameters, restart the Monitoring Service.

To enable sending alerts to users from the*Situator Server Monitor*:

1. Access the Server Monitor: **All programs > Situator Server Tools > Situator Server Monitoring**.
2. Click **Users**.
3. Do one of the following:
   - For a new user, click **Add New User** and select the **Alerts** tab.
   - For an existing user, click **Edit** and select the **Alerts** tab.



4. Select the **Send Alerts** check box, and select the severity level from which an alert will be sent to the user.
5. In the *Gateways* section, select the transmission protocol by which the message is to be sent: SMTP (email), SimpleWire (SMS), LAN, etc..

7.4 Monitoring Situator Services with Umbrella

7.4.1 Overview

Umbrella is Qognify's Central **VMS**[1] Management Platform, a Web-based platform to centrally configure, manage and monitor distributed independent VMS installations.

Umbrella enables the configuration, management and monitoring of all connected Qognify VMS systems across multiple sites. It can be hosted in the cloud as well as on premise. Umbrella provides a consolidated view of all entities (servers, cameras) across all connected installations. Changes in configuration can be made centrally and be rolled out to several or all connected installations.

For more information refer to https://www.qognify.com/products/umbrella/.

Umbrella can also monitor Qognify Situator sites. This can be done in two ways:

- Installing a gateway and activating it in Situator - see Umbrella Integration with Situator via a Gateway below
- Using Umbrella UI to apply an exclusive Situator license module and deploy a Situator site - see Enabling Situator Sites Monitoring in the Umbrella UI on page 60.

7.4.2 Umbrella Integration with Situator via a Gateway

Umbrella integration with Situator via a gateway includes the following processes:

1. Installing Situator
2. Installing the Umbrella platform:
   a. Install the *Umbrella Core* in the Situator Components server.
   b. Install the *Umbrella Gateway* in the Situator Operational server.
   c. Install the *Umbrella Monitoring agent* in servers other than operational (Components and others if relevant).
3. Enabling monitoring of Situator sites by Umbrella.

Steps 1 is described in the *Situator Installation and Upgrade Guide*

Step 2 is described in the Qognify Umbrella documentation (see Umbrella Referenced Documents on the next page).

Step 3 is described in Configuring Situator for Umbrella Monitoring on the next page.

[1]Video Management System

Umbrella Referenced Documents

The following documents are required for completing step 2 above. To access these documents, go to Umbrella documentation on the partner portal.

- **Umbrella <version[1]> Installation on Windows Server 2016 MS SQL** - To get the Umbrella Core running [excluding MS SQL Server installation (already part of Situator)].
- **Umbrella <version[1]> Monitoring Plugin Installation and Configuration** - add Umbrella Monitoring plugins to additional servers. Select **Situator** as a plugin type.
- **Umbrella <version[1]> Gateway Installation and Configuration**. You can install the gateway offline (manually) or by downloading and running the installer . Depending on the requirements, the default configuration may suffice. Otherwise, a configuration of the Umbrella gateway might be necessary.
- **Umbrella Carbon Update Guide** - important migration steps after a new release[2]

Configuring Situator for Umbrella Monitoring

After the Umbrella platform, the plugin and the gateway are installed, enable the Umbrella monitoring on the Situator system as described below.

To enable Umbrella monitoring in Situator:

1. Stop the *Situator Monitoring* Service.
2. Navigate to the to *Umbrella gateway* folder and copy the *key.pub* file.
3. Paste the file into the *OpServer bin* folder.
4. In the same folder open *MonitoringServiceConfiguration.xml*.
5. Search for the *ExternalMonitoringList* section.
6. In *Umbrella ExternalMonitoringInfo*, set **IsActive** to **True** (false by default).
7. Start the *Situator Monitoring* Service.

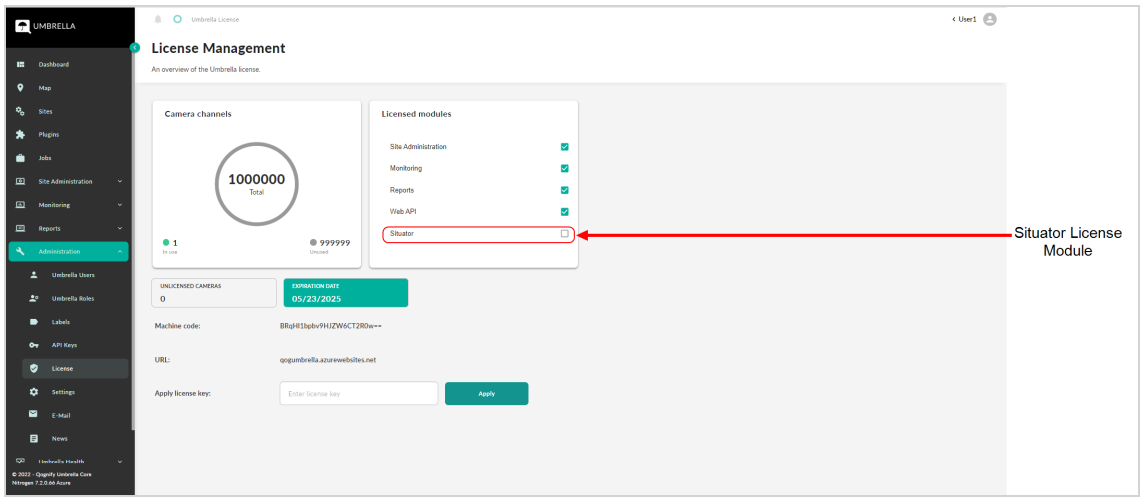
[1]Carbon 6.0 (2022)

[2]Umbrella Nitrogen (2022)

7.4.3 Enabling Situator Sites Monitoring in the Umbrella UI

Refer to the *Umbrella User Guide* for more information.

- In Umbrella *Administration - License Management*, use the specific license module for Situator, which enables working exclusively with Situator.



- Using this mode, when deploying a Situator site in Umbrella, the only available plugin type is **Qognify Situator Plugin**.

> NOTE: After running the Site Installer, adjust the *config.json* file to point to the Situator Plugin / Situator monitoring agent.

- In the *Umbrella Role Details* window, you can configure Situator Entity Types for which to send Events Notifications.
- In jobs, you can filter by Situator job types.

CHAPTER 8 Defining External Login Authentication (SSO)

8.1 Overview

User credentials can be stored in an authentication provider outside of the Situator Server (such as Microsoft **Active Directory** - AD). External login authentication, or **Single Sign-On** (SSO), enables privileged users to access all Situator applications without being prompted for their credentials, as these are taken from the Windows login.

The authenticity of users accessing Situator can be achieved by verifying the end user's name and password against the Situator database or against the organizational Active Directory server. The AD acts as a centralized repository for user credentials and approves or denies access requests forwarded to it by Situator via a secure protocol (e.g. NTLM or Kerberos).

AD helps system administrators to manage user credentials in a centralized and secured location and thus reduces administrative overhead by eliminating the need to store multiple copies of usernames and passwords in several databases.

External login authentication is supported by *Control Room* and *Planning Tool*. Once an external login provider is configured in Situator, administrators can either add local user accounts or external login provider accounts. However, in the Situator login screen, while a user's password can be determined by the external login directory service, the username must be a Situator user.

Enabling external login authentication in Situator includes:

- Defining AD or AD SSO in the database
- If using Azure AD, registering Situator in the Azure portal as defined in Enabling Active Directory/ Azure AD Login Authentication on page 70.
- Enabling SSO capability in *Control Room* and *Planning Tool* configuration files (for AD SSO only)

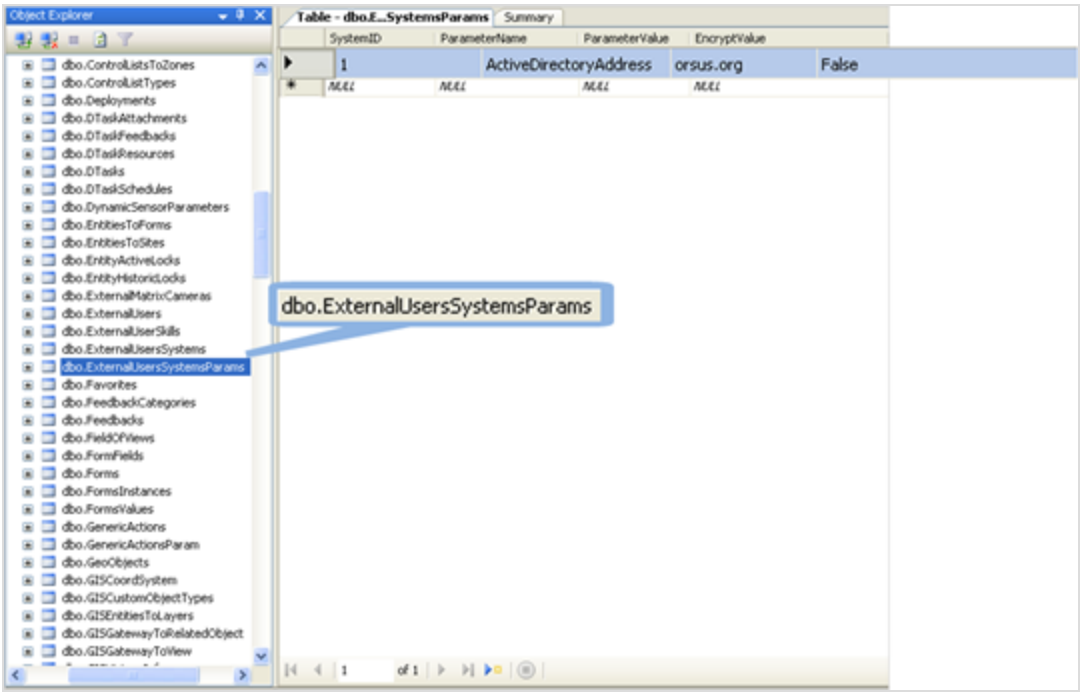
- Defining an authentication policy with two login parameters in *Control Room*
- Defining a role that inherits this authentication policy in *Control Room*
- Defining a Situator User that is associated with this role

The next sections outline the steps for activating Azure AD and AD SSO in Situator.

8.2 Defining Active Directory in the Situator Database

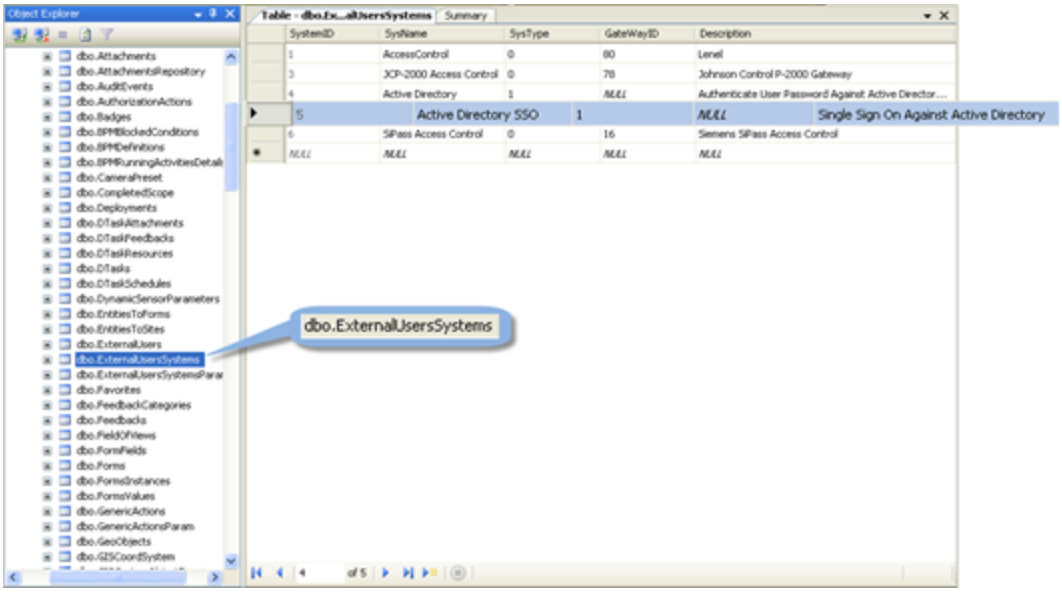
To enable Active Directory:

1. Open the Situator database table *dbo.ExternalUsersSystemsParams*.
2. Enter an Active Directory system provider in the syntax of `<domain_Name>` in the *ParameterValue* column.



> NOTE: Active Directory can only work with one domain.

3. Verify that Active Directory is defined properly in the *dbo.ExternalUsersSystems* table.



8.3 Defining Login Authentication Policy

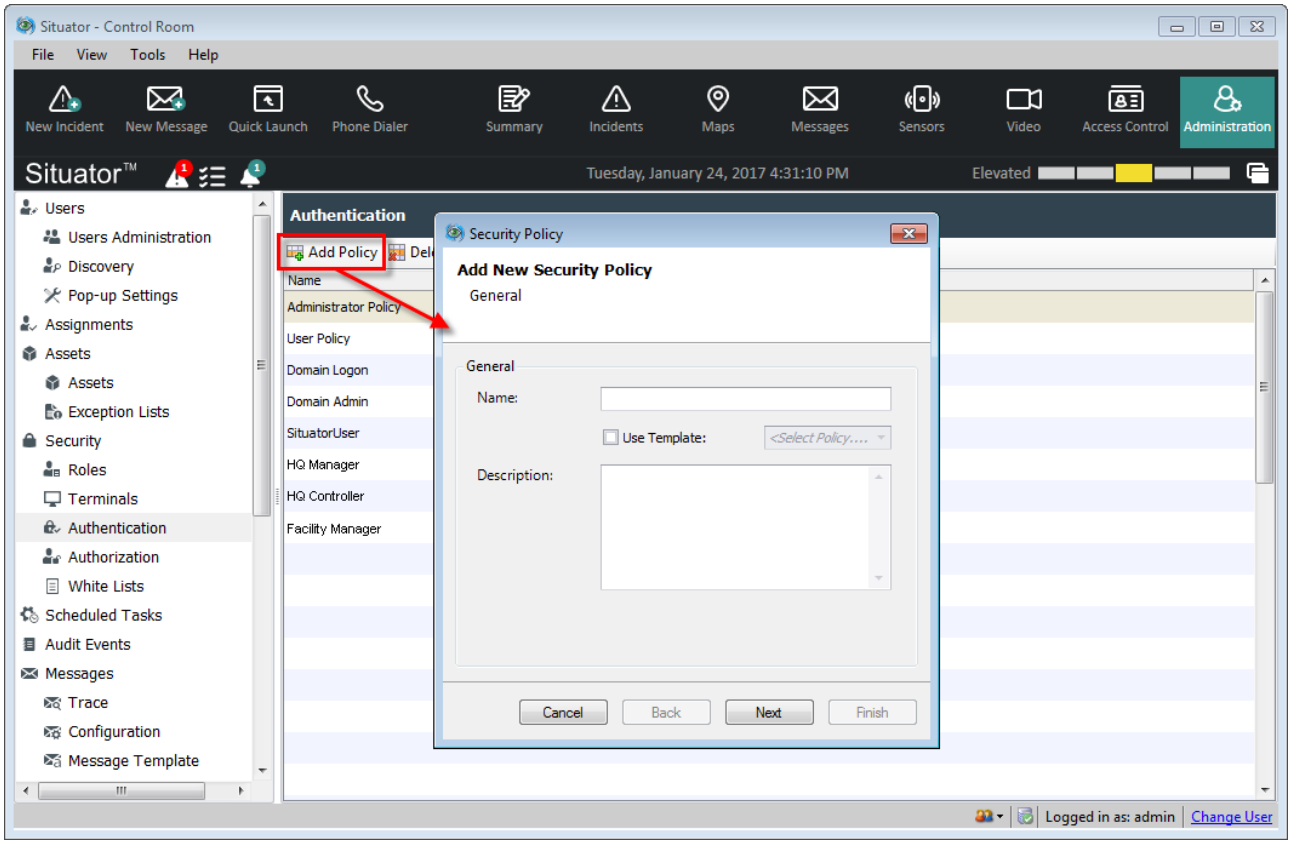
Situator validates users before granting *Control Room* access using passwords. Users cannot perform operations without a valid session. Successful login credentials are required.

Administrator security capabilities:

- Managing logins and user sessions
- Locking / unlocking users including blocking next login or halting all active sessions for specific users
- Allowing users to log in from specific terminals and/or from specific IP addresses
- Managing passwords
- Resetting user's passwords
- Users can have the system restricted to specific locations or regions according to terminals/computers/network access addresses
- Selecting the authentication policies - Situator provides two predefined authentication policies:
  - Administrator policy - No limitations
  - User policy - Password length is limited to 6 characters and password must contain at least 1 upper case letter and 1 lower case letter

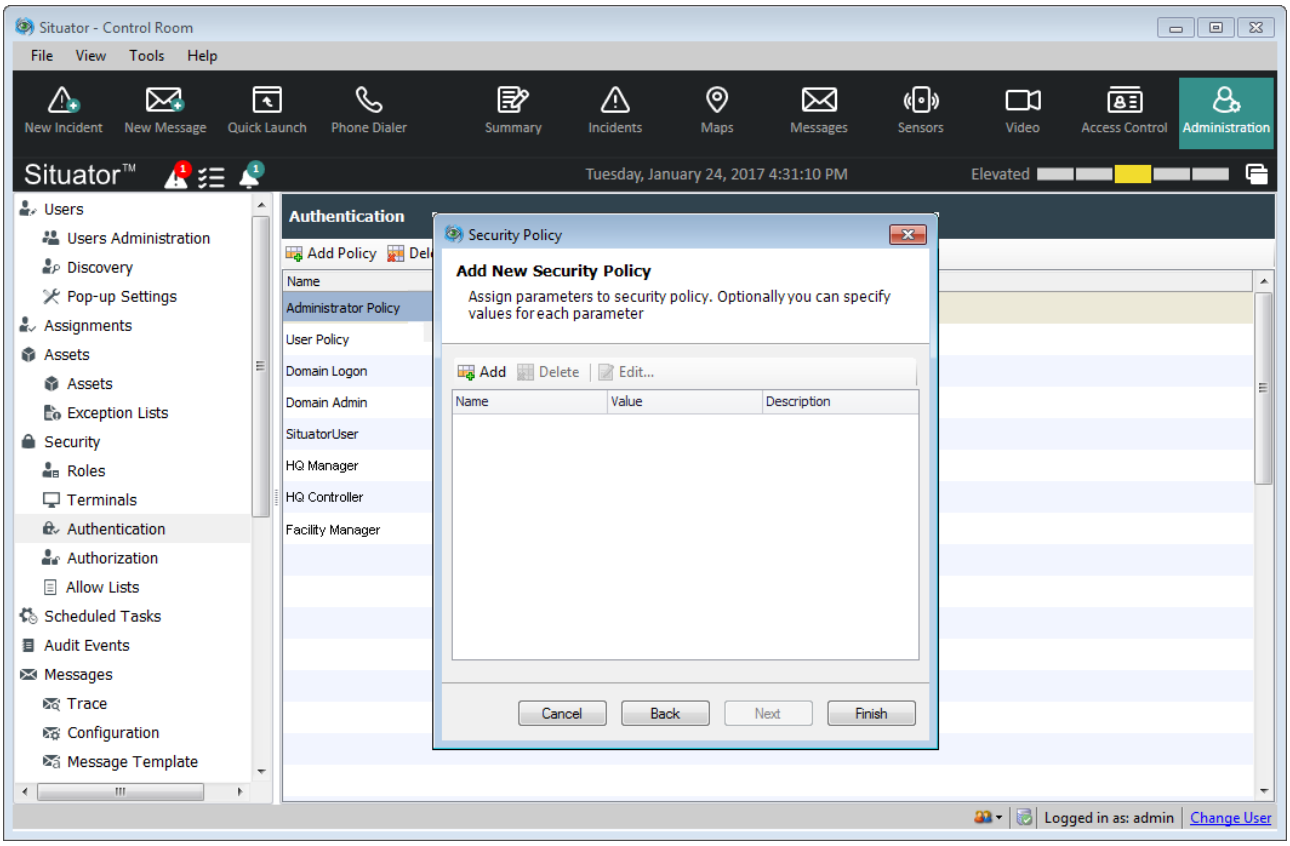
To add a new policy:

1. In the *Control Room* navigation bar, click **Administration**.The *Administration* view opens.
2. In the **Administration tool** pane, click **Authentication**. The *Authentication* workspace opens.
3. In the Authentication workspace, click **Add Policy**. The *Add New Security Policy - General* dialog box opens.

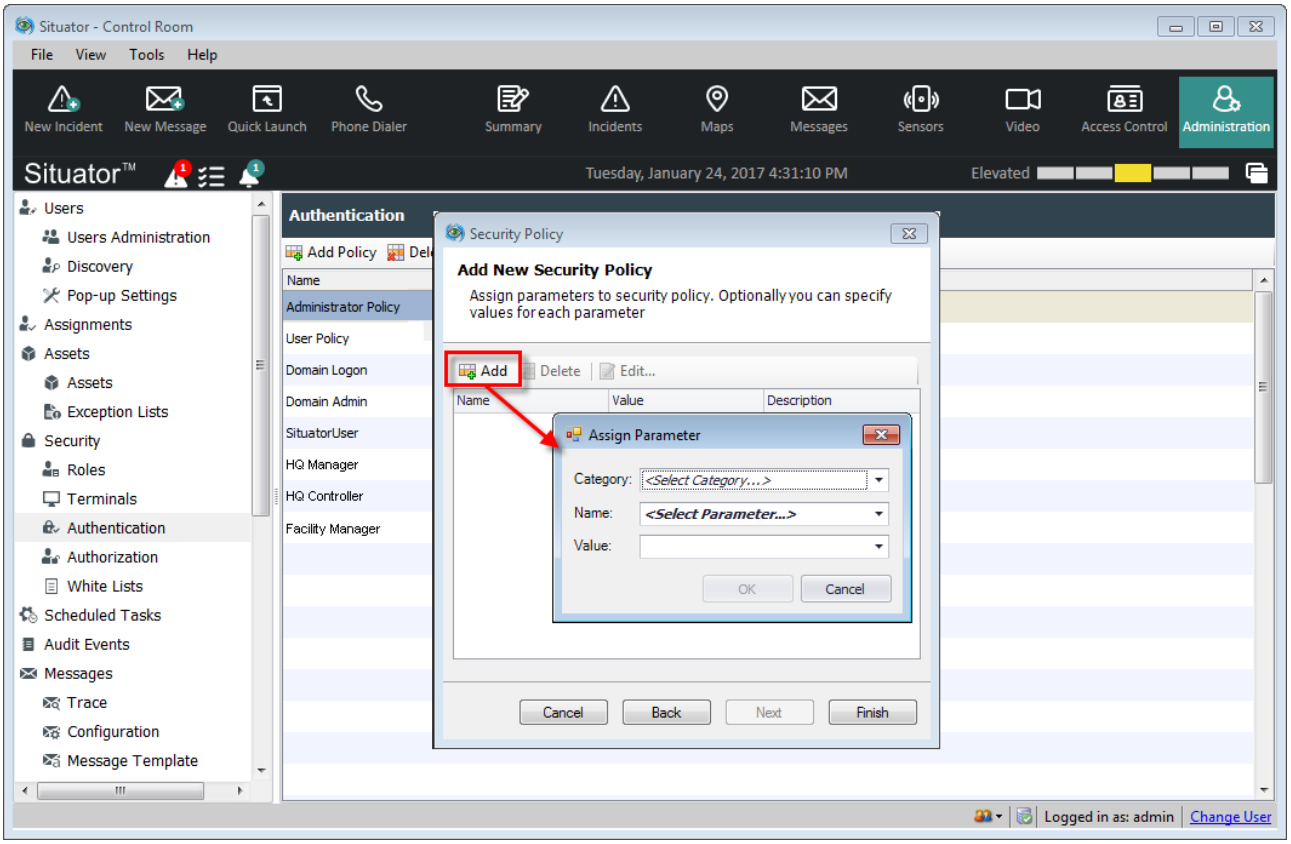


4. In the **Name** and **Description** fields, type the name and description of your policy.
5. Do one of the following:
   - To create a new policy: Click **Next**.
   - To work off of an existing policy:
     a. Select the **Use Template** check box.
     b. From the drop down list, select an existing policy.
     c. (Optional) To change the policy name: In the **Description** field, type a description, and click **OK**.
     d. Click **Next**.

The *Add New Security - Parameters* dialog box opens.



6. (Optional) To add predefined parameters to an existing policy:
   a. In the *Add New Security - Parameters* dialog box, click **Add**. The *Assign Parameter* dialog box opens.



   b. In the **Category** and **Name** fields, select parameters from the drop down lists, and click **OK**.
   c. Repeat to add more parameters.

> NOTE: If you select the predefined User Policy, the two predefined parameters automatically appear in the parameters dialog box.

7. (Optional) Use the following additional option as required:
   - To delete a parameter: Click to highlight a parameter and click **Delete**.
   - To modify policy parameters: Click to highlight a parameter and click **Edit**. Edit as needed.
8. Click **Finish**.

8.4 Authentication Security Parameters

The following table describes the security parameters and their values.

Parameter	Description	Value
Login		
Password Expiration	The number of days before the password expires. When the password expires, the user receives a "Password expired" notification and is then prompted to change passwords.	Number
Verify Roles Machines	When set to true, this parameter determines that roles will be checked for permission to access a terminal upon user login attempt. For example, an organization might decide that only users with the role "Security patrol" can be authorized to access specific terminals	True/False
Single Session Per User	Enforces login of the user account to one client only. For example, if a user is logged in to CR#1 and then tries to log in to CR#2, the login from CR#1 will be disconnected.	
Machine IP Correlation	When set to true, users will have the Terminal's IP property verified before other authentication checks.	True/False
Max Login Attempts	Number of permitted log in attempts. The user is blocked from logging in to Situator after exceeding the permitted number of log in attempts and needs to be unlocked by an administrator in the **Security** tab of the *User Properties* dialog box. Refer to Defining Situator Users on page 82.	Number
Allowed Authentication Method	Enables external login for a role. To avoid user errors, an Authentication Policy should not include both Allowed Authentication Method and Password Expiration parameters. To enable external login capability for a user, refer to Enabling Active Directory/ Azure AD Login Authentication on page 70.	» User Name and Password » Single Sign-On » Both

Parameter	Description	Value
Username and Password Authentication	Determines the external login provider. To enable external login capability for a user, refer to Enabling Active Directory/ Azure AD Login Authentication on page 70.	» Active Directory » Active Directory SSO (Not supported in Web Client)
Password Policy		
Minimum Password Length	This parameter enforces a defined minimum password length.	Number
Enforce Mixed Case Characters	When enabled, this parameter enforces the minimum of one upper case and one lower case character in a password.	True/False
Enforce Digits	This parameter enforces a defined number of digits in a password.	Number

IMPORTANT: Do not use the "#" character in the user name or password. The user will not be able to login to Control Room or Planning Tool if this character is used.

8.5 Configuring Azure AD for Situator

This section describes the one-time setup procedure that is required to enable Microsoft Azure AD to recognize and communicate with Situator clients.

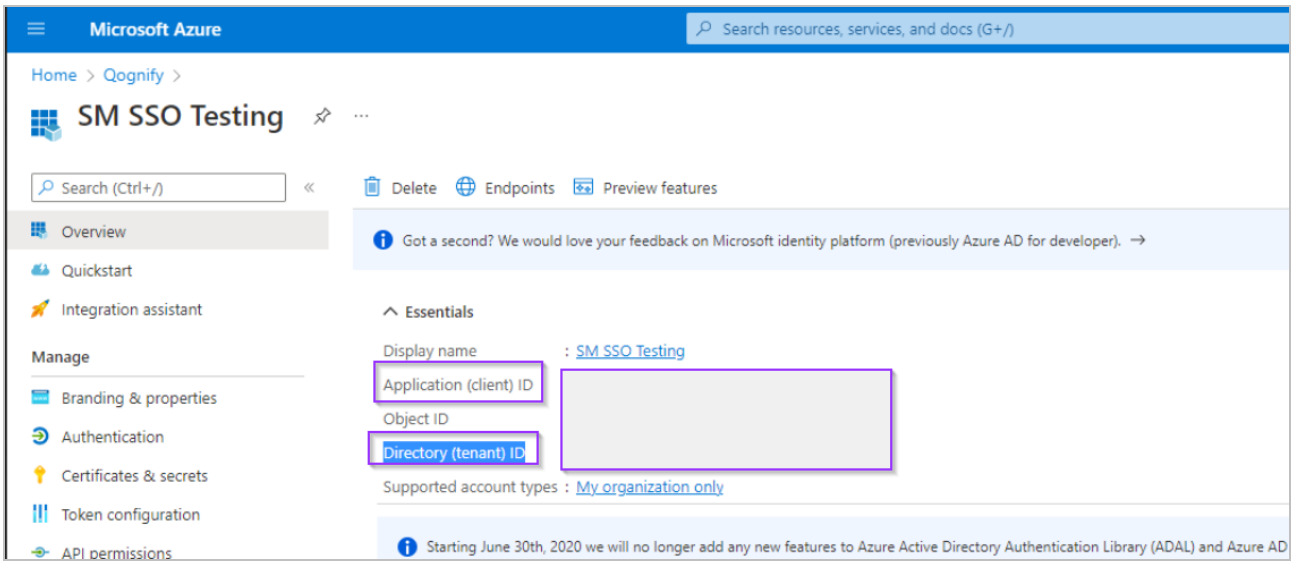
The setup includes the following procedures:

- Register Situator as a desktop application in the Azure AD portal.
- Enter the two IDs provided by Azure into the Situator *Stabilis configuration* file.

After completing this setup procedure, you will need to make some *Control Room* configurations for new policies and roles, as described in the *Situator Control Room Customization Guide*, *Defining Situator User Security*.

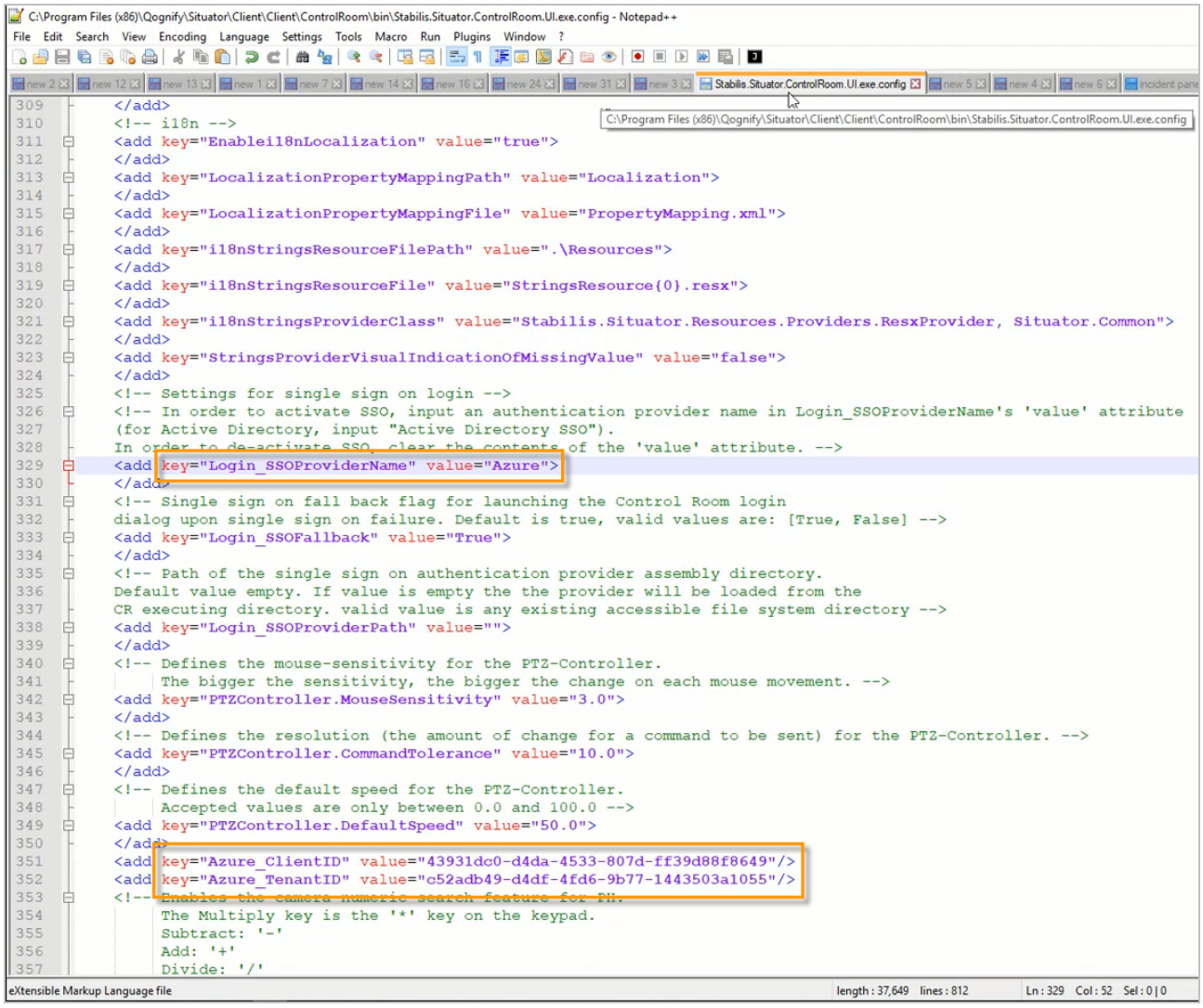
To configure Azure AD for Situator:

1. Open the *Azure AD portal* , and select **App registration**.
2. Under **Owned applications**, select the **Situator registration** from the list. App registration details are displayed. Observe the *Application (client) ID* and *Directory (tenant) ID*; these are the ID numbers required below.



3. Open the file *Stabilis.Situator.ControlRoom.UI.exe.config*, and enter values for the following parameters:
   - **Azure_ClientID**: set value =client ID from the Azure window
   - **Azure_TenantID**: set value =tenant ID from the Azure window
   - **Login_SSOProviderName**: set value as either **Azure** or, if using SSO, **AzureSSO**. (When using active director other than Azure, this parameter is set as either ActiveDirectory or ActiveDirectorySSO.)

   See example below.



4. Save the configuration changes.
5. If *Control Room* is running, restart it.

8.6 Enabling Active Directory/ Azure AD Login Authentication

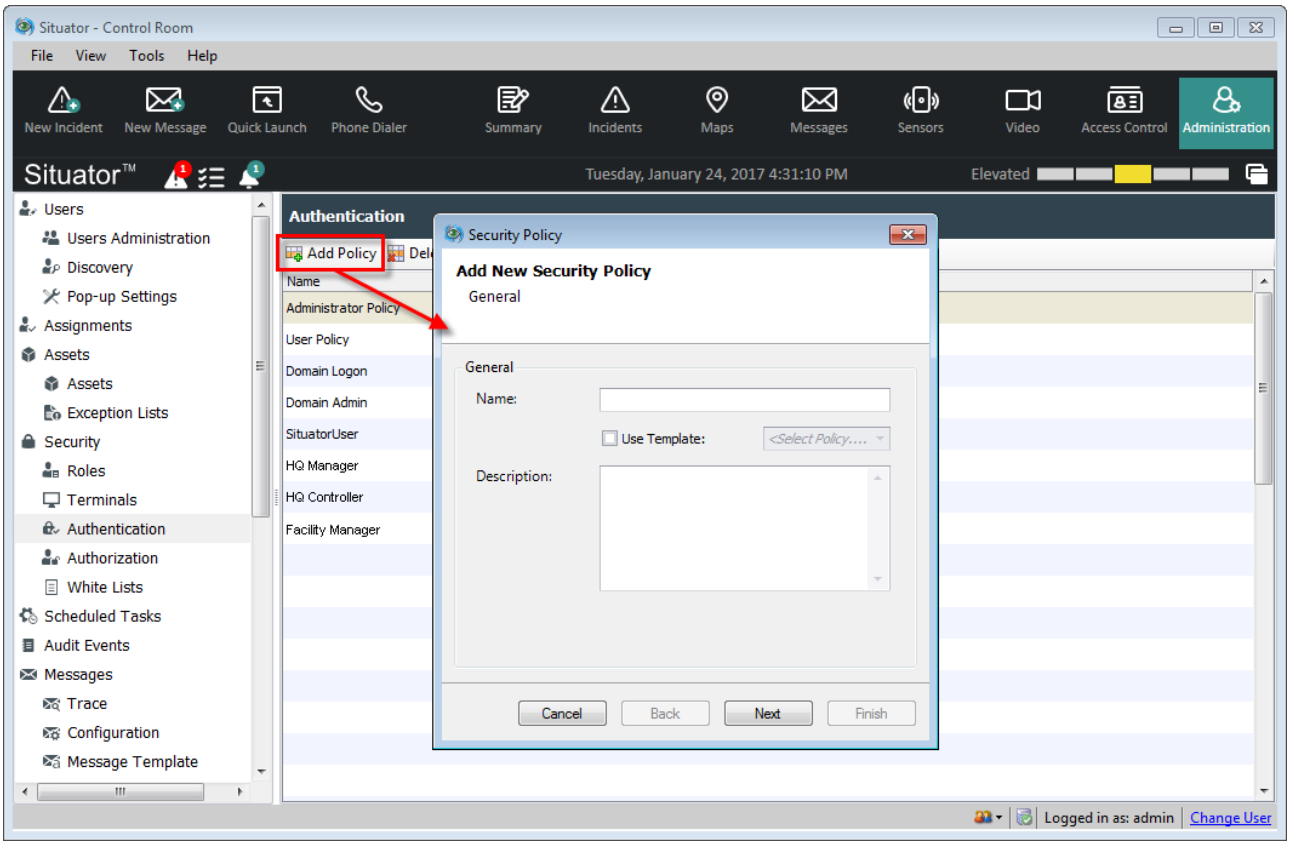
Active Directory/ Azure AD enables Situator users to log in using a single, secure authentication without a prompt for user name and password credentials at each login. External login authentication is supported by the *Control Room*, *Planning Tool*, and *Qognify Web Client*. Once Active Directory/ Azure AD has been added to a Situator authentication policy, Administrators can group users by functional Roles and grant permissions to a particular Role. For example, Active Directory/ Azure AD login permission can be granted to all users assigned to the Role "HQ Manager" but not to users assigned to the Role "User".

You can configure Active Directory/ Azure AD in:

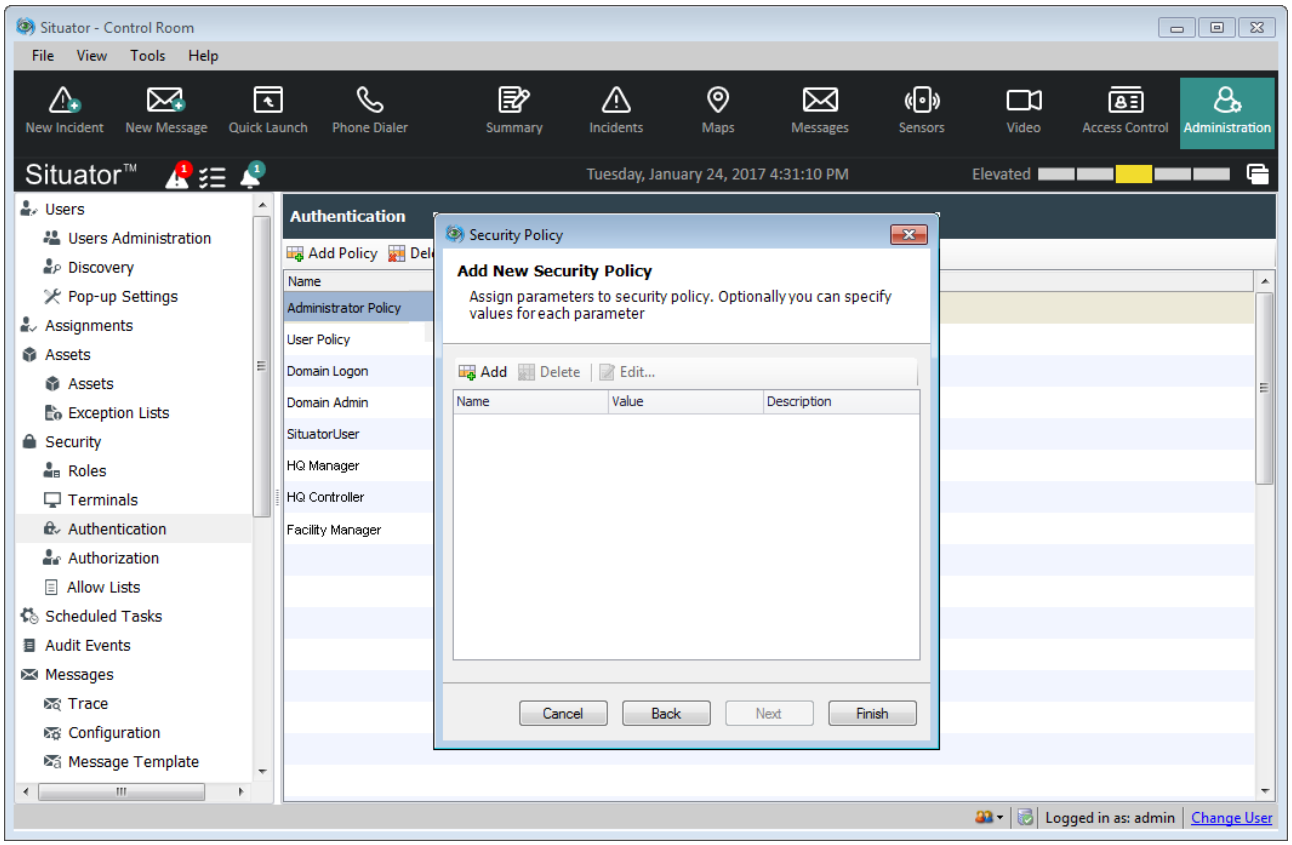
- Situator configuration files, as described in *External Login Authentication* in the *Situator Administrator's Guide*
- *Control Room* Administration view - Authentication, as described below
- *Control Room* Administration view - Users
- *Planning Tool* Advanced Settings view, External Login Authentication

To create a new authentication policy with external login enabled:

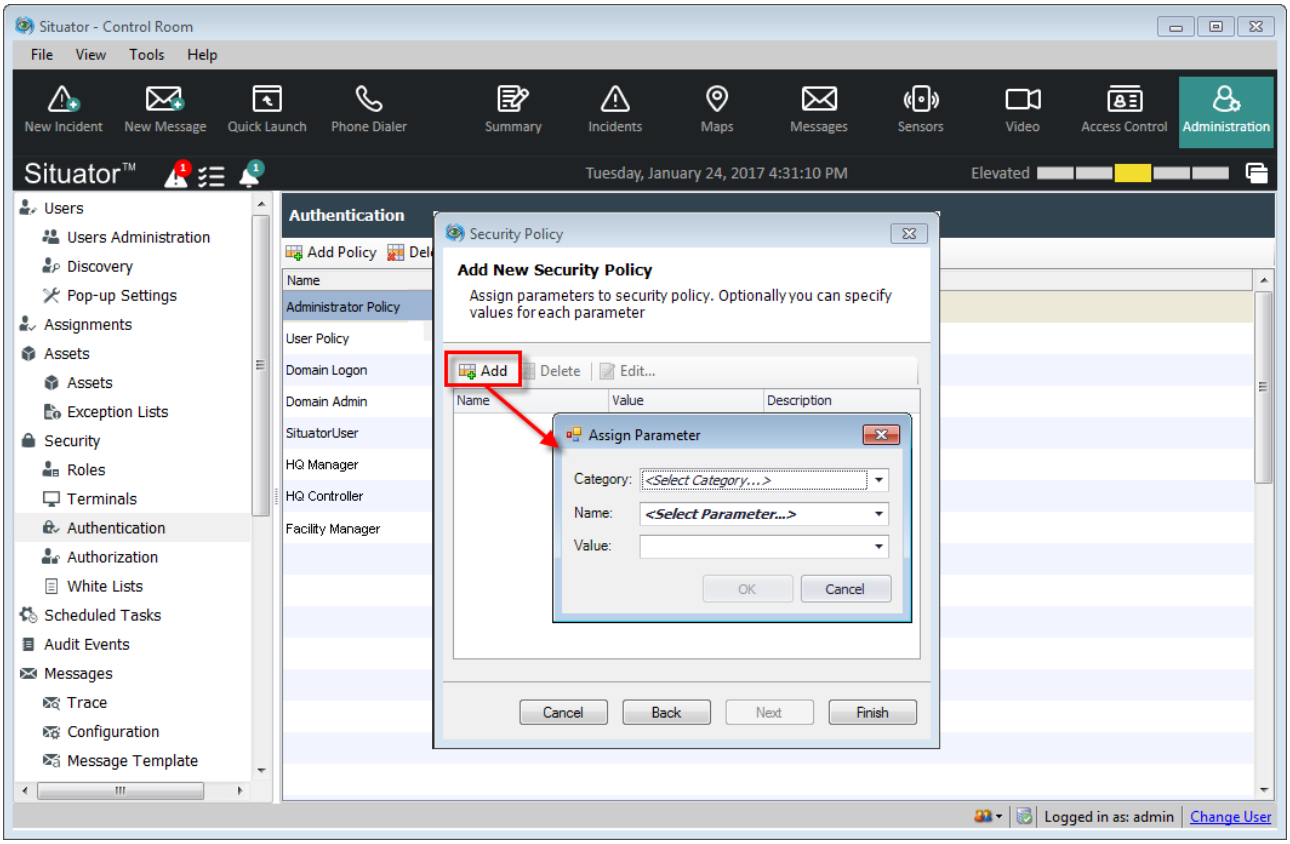
1. In the *Control Room* navigation bar, click **Administration**.The *Administration* view opens.
2. In the **Administration Tool** pane, click **Authentication**. The *Authentication* workspace opens.
3. In the workspace, click **Add Policy**. The *Add New Security Policy - General* dialog box opens.



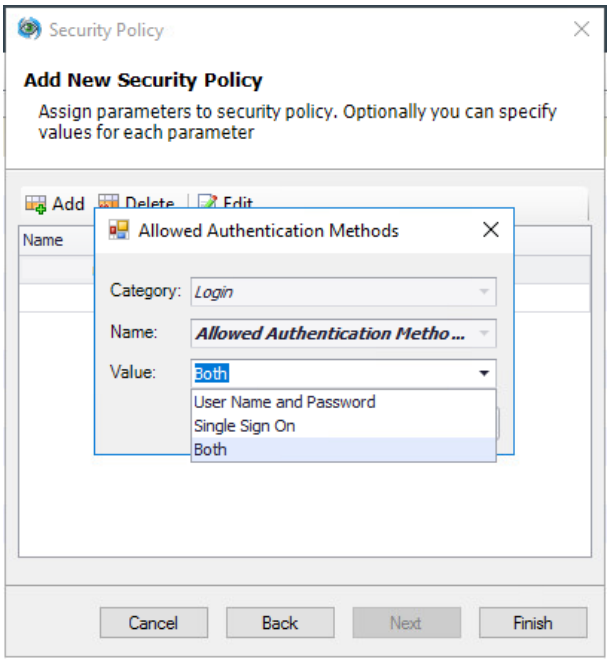
4. In the **Name** and **Description** fields, type the name and description of your policy, and click **Next**. The *Add New Security Policy - Parameters* dialog box opens.



5. Click **Add**. The *Assign Parameter* dialog box opens.



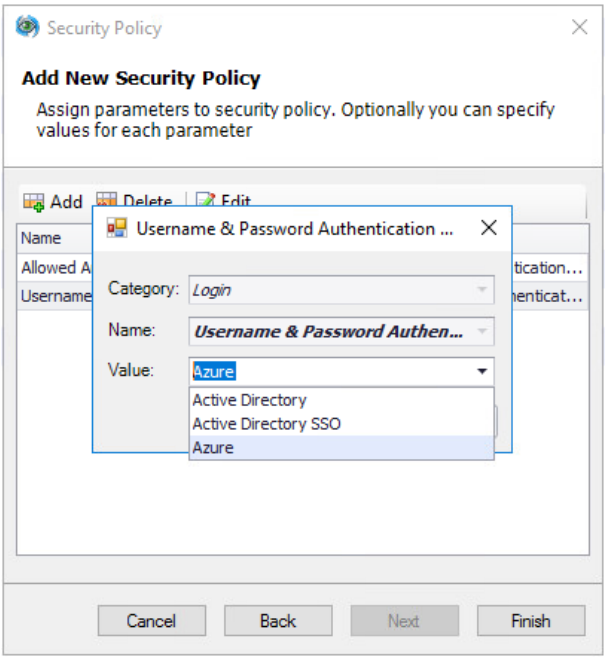
6. From the **Category** drop down list, select **Login**.
7. From the **Name** drop down list, select from the following parameters:
   - **Enable external login for a role:**
     a. Select **Allowed Authentication Methods**.
     b. From the **Value** drop down list, select one of the following:
        - **User Name and Password** to authenticate a user against Active Directory
        - **Single Sign On** to authenticate a user against Active Directory SSO
        - **Both** to authenticate a user against both Active Directory and SSO



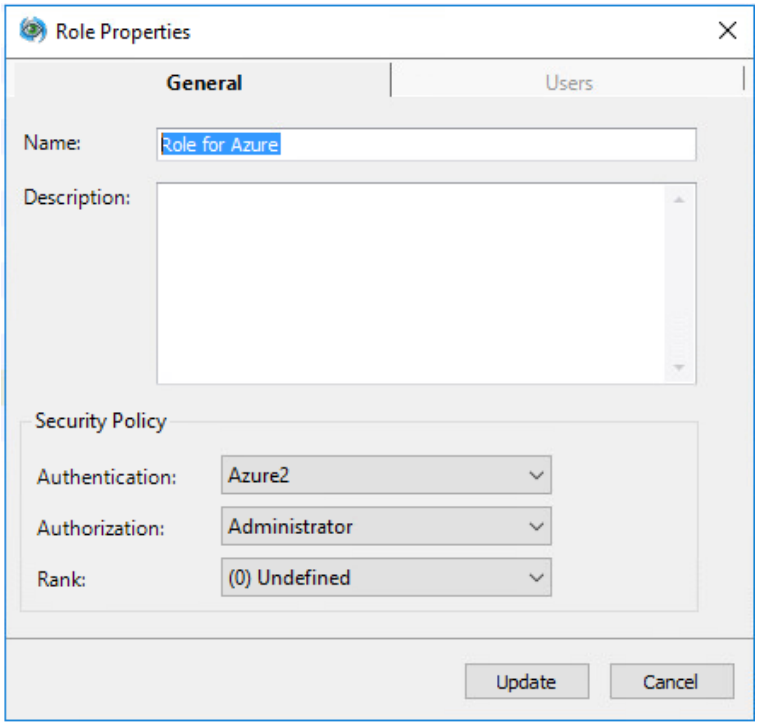
» **Select an external login provider:**

a. Select **Username and Password Authentication Provider**

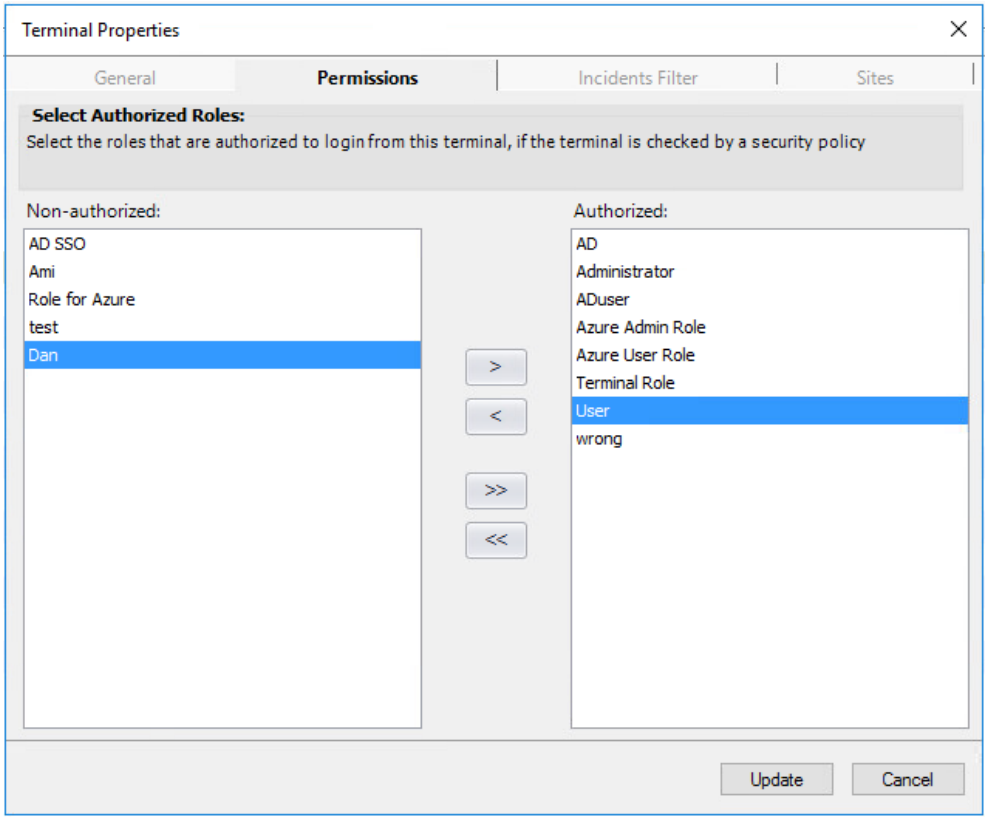
b. From the **Value** drop down list, select **Azure** or **Active Directory**.



8. Click **Finish**.
9. Create a dedicated role for Azure. Refer to *Defining User Security Roles* in the *Situator Control Room Customization Guide*.



10. Assign the Role to a Terminal:
    a. In the **Administration Tools** pane, click **Security**>**Terminals**. The *Terminal Properties* window opens.
    b. Double-click the required terminal to open its properties.
    c. Select the **Permission** tab.
    d. Move the Role defined in previous step to the **Authorized** column.

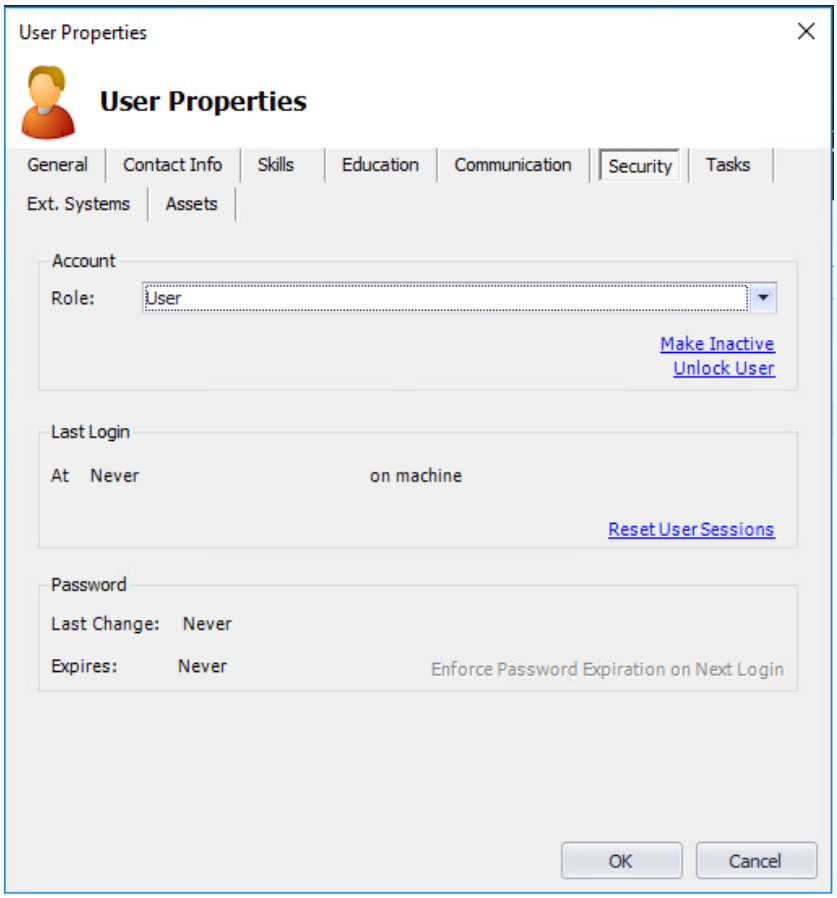


11. Assign the external login capability to a user as described To assign external login capability to a user: below.

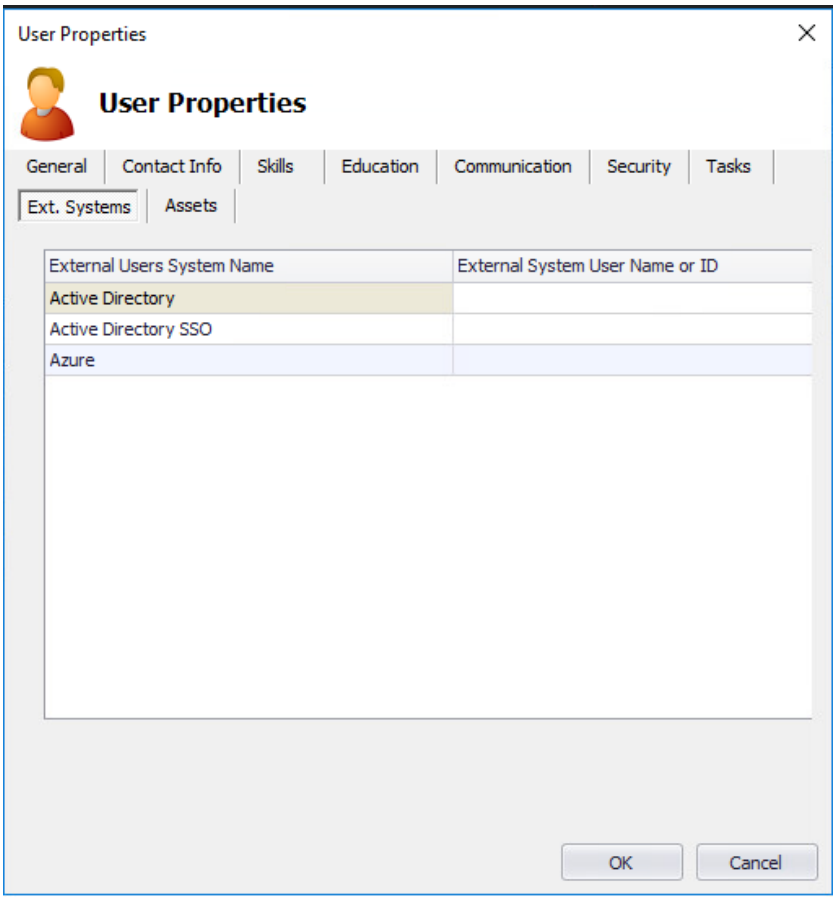
To assign external login capability to a user:

1. In the Control Room navigation bar, click **Administration**. The Administration view opens.
2. In the **Administration Tools** pane, click **Users**. The Users workspace opens.
3. Select one of the following options:
   - To add a new user, refer to Defining Situator Users on page 82 and click **Add & Edit**.
   - To edit the properties of a Situator user, double click a user entry (or right-click and select **Properties**).

   The *User Properties* dialog box opens.
4. Select the **Security** tab, and from the **Role** drop down list, select a Role to which you want assign external login authentication.


User Properties
User Properties
General
Contact Info
Skills
Education
Communication
Security
Tasks
Ext. Systems
Assets
Account
Role:
User
Make Inactive
Unlock User
Last Login
At Never
on machine
Reset User Sessions
Password
Last Change: Never
Expires: Never
Enforce Password Expiration on Next Login
OK
Cancel

5. Select the **Ext. Systems** tab.



6. Do one of the following:
   - To define Azure - In the **Azure** field, enter a **UPN**[1] logon name (for example, *user@Domain.com*). This is a case-sensitive field. Make sure AD and AD SSO are left blank.
   - To define AD or AD SSO - In the **Active Directory**/**Active Directory SSO** fields, enter a relevant external-provided user name or ID.

[1]User Principal Name

The syntax is DOMAIN\USER NAME. For example, org\BobSimmons.

> NOTE: Active Directory and/or Active Directory SSO credentials should be provided by your IT or network administrator.

7. Click **OK**.

To configure Control Room IDM and Qognify Web Client to use Azure AD:

1. Open the folder *C:\Program Files (x86)\Qognify\Situator\Client\ControlRoom\bin*.
2. Open the file **Stabilis.Situator.ControlRoom.UI.exe.config** for editing.
3. Change the parameter:

   ```
   <add key="Login_SSOProviderName" value="Azure">
   ```

   To:

   ```
   <add key="Login_SSOProviderName" value="">
   ```

4. Log in to CR with the default built-in user (“admin”)
5. Create and configure the Azure users.

   For more information, refer to *Configuring Azure AD for Situator* .
6. Close the CR application.
7. Re-open the file **Stabilis.Situator.ControlRoom.UI.exe.config** for editing.
8. Check if `Azure_ClientID` and `Azure_TenantID` fields already contain your IDs. If not, change the key parameters to the following values:

Key	Value
Login_SSOProviderName	Azure
Azure_ClientID	Your ClientID*
Azure_TenantID	Your TenantID*
Azure_Attempts	3

9. Open the folder *C:\Program Files (x86)\Qognify\Situator\IDM Server\IDM*.
10. Open the IDM configuration file **web.config** for editing.

11. Check if `Azure_ClientID` and `Azure_TenantID` fields already contain your IDs. If not, change the key parameters to the following values:

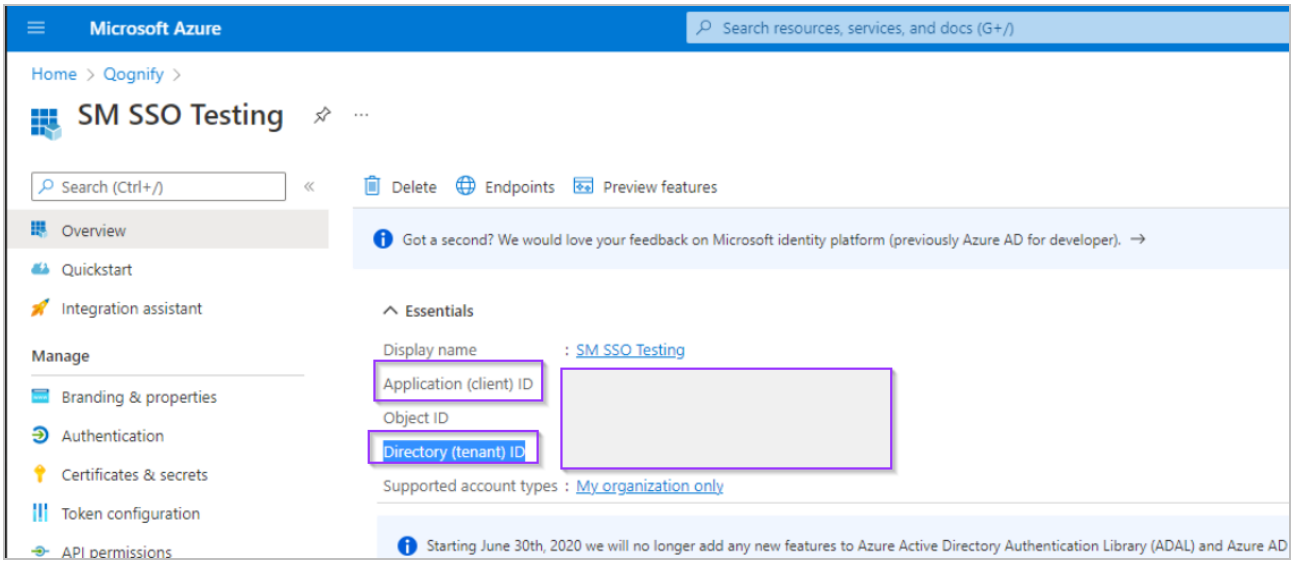
Key	Value
Azure_ClientID	Your ClientID*
Azure_TenantID	Your TenantID*

12. Open the folder *C:\Program Files (x86)\Qognify\Situator\QognifyWebClient\Config*.
13. Open the IDM configuration file **Login.js** for editing.
14. Change the following key parameters to the following values:

Key	Value
LoginProvider	Azure
clientID	Your ClientID*
tenantID	Your TenantID*

15. To check the login using Azure AD, log in to Control Room and Qognify Web Client.

* The Client ID and Tenant ID are available from the Azure Portal (registered Applications):

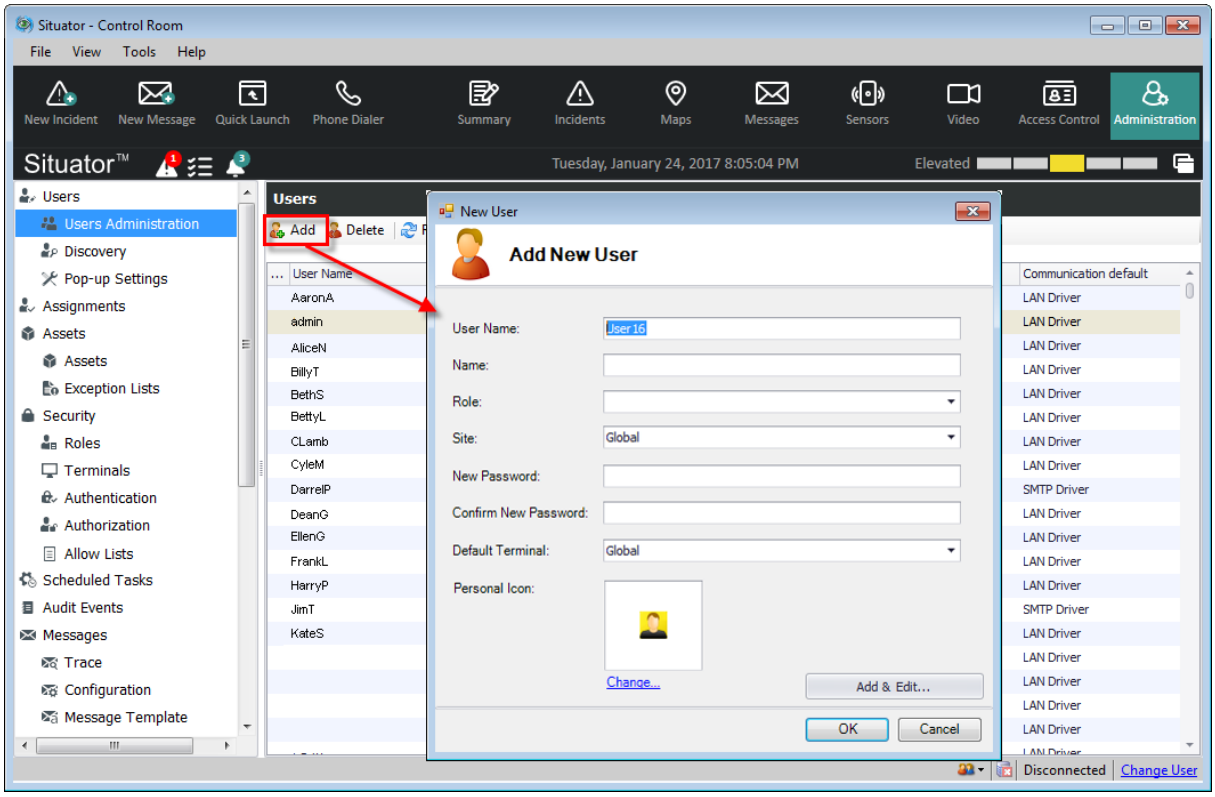


CHAPTER 9 Defining Situator Users

Users are defined by the system administrator. A user definition may include a name, role, password and some other mandatory or optional properties.

To define users:

1. In the *Control Room* navigation bar, click **Administration**. The *Administration* workspace opens.
2. In the **Administration Tools** pane, click **Users**. The *Users* workspace opens.
3. In the workspace, click **Add**. The *Add New User* dialog box opens.



4. In the **User Name** field, type the user name, and in the **Name** field, type the name of the user. Do not use the "#" character in the user name.

5. From the **Role** drop down list, select the role to be assigned to the user.

   NOTE: If the selected role has an authentication policy that enables Active Directory or Azure AD, a valid user name or ID must be entered in the *User Properties* dialog box **Ext. Systems** tab. See table below for more information.

6. In a multi-site installation, from the **Site** drop down list, click the site to be assigned to the user.

   NOTE: Users will be able to see and access sensors/cameras in their sensor groups, in their site, in child sites, as well as those in the global (generic) group, subject to the permissions defined for their role. The logic also applies to Map sensors, cameras in the Video Source tree, and Access Control view.

7. In the **New Password** field, type the password. Do not use the "#" character in the user name.

   Control Room may be configured so that when the selected Role is Azure or Active Directory, a password is not required in the *Add New User* dialog box (the field is grayed out) and the *Login* dialog is not displayed in case of login failure. Refer to *Enabling Active Directory/ Azure AD Login Authentication on page 70*.

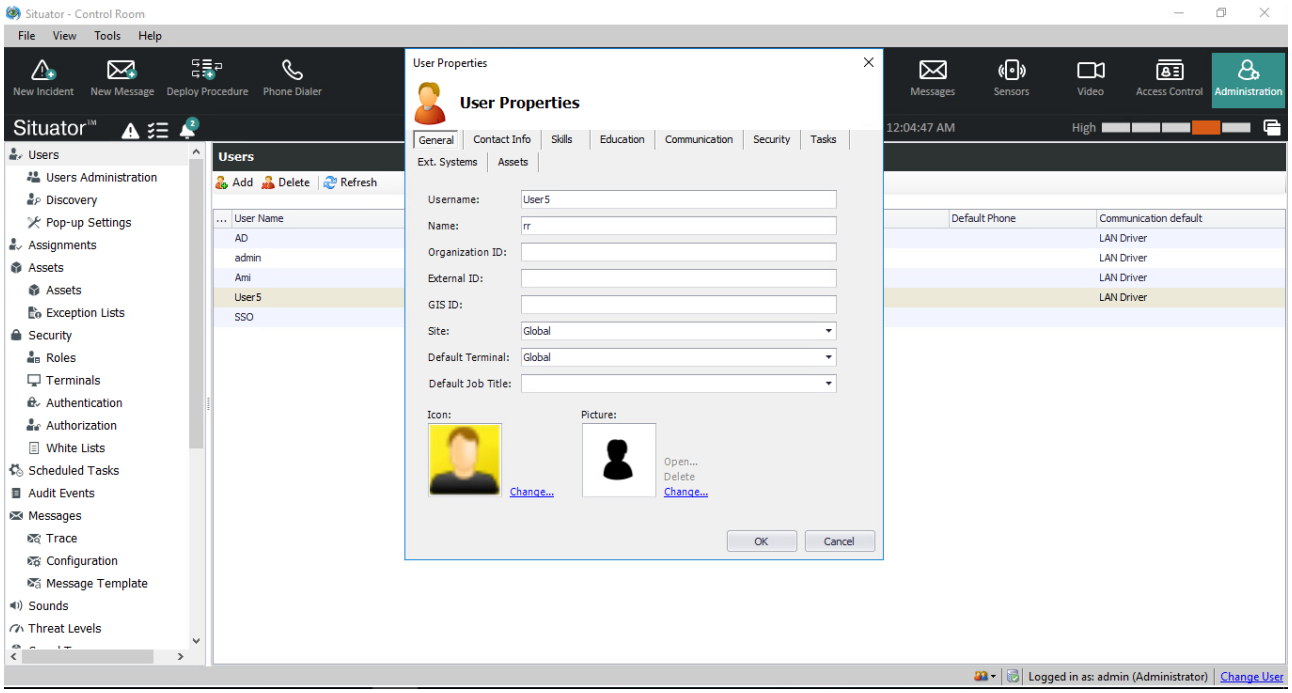
8. In the **Confirm New Password** field, retype the password.

9. (Optional) Make additional changes as described below.

To	Do This	Comments
Specify the user's default terminal	Select the terminal from the list.	If you leave this empty, the terminal definition is assigned by the parameter **Machine Name** in the client configuration file. If that parameter is also empty, then a global definition is applied (full privileges).
Change the icon representing the user	Click **Change** and click to select an icon. The icons represent the various job titles such as security officer, driver, medic, or police officer. The icon displays beside the user's name in the user's main screen.	Administrators in the Control Room Administrator can associate an entity (such as users or sensor groups) with an icon. This icon overrides the Icon Management default base icon. However, the Administrator set icon can be reconfigured in Edit Conditions. When the users have been added the first time, you can change user names, names, sites, icons, and pictures in the *Properties* dialog box by selecting the item you want to change, right-clicking and selecting **Properties**.
Add contact info	a. Click **Add & Edit**. b. In the **Contact Info** tab, add the user's email address, work phone number, home phone number, cell phone numbers, and fax number	See the *User Properties* image below
Add skills details	a. Click **Add & Edit**. b. In the **Skills** tab, assign skills to a user from a list of skills defined in the *Planning Tool* and a time period for which the skills are valid.	Refer to the *Situator Planning Tool Customization Guide* for more information.

To	Do This	Comments
Add education details	a. Click **Add & Edit**. b. In the **Education** tab, add a user's education and training credentials.	
Add communication details	a. Click **Add & Edit**. b. In the **Communication** tab, select a communication method for the user.	The following communication mechanisms are available: » Sensor server drivers used for communicating to mobile devices » SMS providers, such as SMTP and NIMS drivers The SMS drivers are only loaded upon request in Situator if the **IsActive** column is set to TRUE in the NotificationDrivers table in the database.
Add security details	a. Click **Add & Edit** b. In the **Security** tab, assign a user Security credentials such as Situator role, reset user sessions, and reset passwords.	
Add tasks	a. Click **Add & Edit**. b. In the **Tasks** tab, view the tasks assigned to a user.	

To	Do This	Comments
Add external systems details	a. Click **Add & Edit**. b. In the **Ext. Systems** tab, add relevant external system ACS user names or badge IDs and external authentication providers login names or IDs.	
Associate the user to an existing human asset	a. Click **Add & Edit**. b. In the **Assets** tab, select the relevant human asset.	External users, such as mobile users, will be displayed on a map-layer as a human asset.



10. Click **OK**.

CHAPTER 10 Creating Control Room Crash/Hang Dump Files

In the event of an unhandled *Control Room* crash or process termination, it is helpful to create a memory dump to a file, so the information can be later reviewed and analyzed. Situator supports such a memory dump using the third-party application *ProcDump*. Once enabled, a memory dump may be executed either automatically or manually by a .bat script.

10.1 Enabling Crash/Hang Dump Creation

To enable memory dumps, you must install the *ProcDump* application. The installation will enable both automatic and manual memory dumps.

To install the ProcDump application:

1. Download the ProcDump tool from the Microsoft official site:

   https://docs.microsoft.com/en-us/sysinternals/downloads/procdump

2. Install the program. Set the ProcDump path by setting the value of DumperExecutable to one of the following:
   - The default value - *C:\Program Files (x86)\Procdump\procdump.exe*
   - In the CR configuration file *Stabilis.Situator.ControlRoom.UI.exe.config*

10.2 Enabling /Disabling Automatic CR Dump Files Creation

By default, automatic creation of memory dumps is initiated if *Control Room* crashes by an unhandled exception. The name for dump file will be *ControlRoomDump[YYMMDD_HHMMSS].dmp*, in the Control Room Logs folder.

To enable/disable automatic dump file creation:

In the CR configuration file *Stabilis.Situator.ControlRoom.UI.exe.config* set the value of **AutomaticCrashDumpCreation** to **True** or **False**.

```
204     </add>
205     <!-- Path to procdump.exe -->
206     <add key="DumperExecutable" value="C:\Program Files (x86)\Procdump\procdump.exe">
207     </add>
208     <!-- True - automatic dump creation in Log directory
209          False - disabled -->
210     <add key="AutomaticCrashDumpCreation" value="True">
211     </add>
212     <!-- Relative (depends from where you're running it) path to save dumps, default is Logs -->
213     <add key="DumpSavePath" value="Logs">
214     </add>
```

10.3 Creating Manual CR Dump Files

After the *ProcDump* application is installed and *Control Room* is run, the file *CreateCRHangDump.bat* is created in the *bin* directory.

To manually create dump files:

Execute *CreateCRHangDump.bat* file from the *bin* directory. A dump file will be created with name *ControlRoomHangDump[YYMMDD_HHMMSS].dmp*, in the Control Room *Logs* folder.

Control Room must have been executed at least once to create the .bat script.

10.4 Defining the Path to Saved Dump Files

The path to dump files is held in the parameter **DumpSavePath**. By default it has the value **Logs**.

To change the dump files path:

In the CR configuration file *Stabilis.Situator.ControlRoom.UI.exe.config*, set the value of **DumpSavePath** to the path you want the dump files to be saved.

CHAPTER 11 Situator Log Files

A log file lists actions that have occurred in the system. Log files serve as a central method for analyzing errors in the system. This chapter provides a brief introduction to the logging settings (Log4net) used by Situator, and maps out the various Situator log files.

11.1 Log4net Overview

log4net is a tool to help the programmer output log statements to a variety of output targets. log4net is a port of the log4j framework to the .NET runtime. Situator has kept the framework similar to the original log4j while taking advantage of new features in the .NET runtime.

log4net is part of the Apache Logging Services project. The Logging Services project is intended to provide cross-language logging services for purposes of application debugging and auditing.

11.2 Loggers and Appenders

Log4net has three main components: loggers, appenders, and layouts. These three types of components work together to enable developers to:

- Log messages per message type and level
- Control (at runtime) how these messages are formatted and where they are reported

These components are helped by filters that control the actions of the appender and object renderers that turn objects into strings.

11.3 Logging Level

Loggers may be assigned levels: `<level value="INFO" />`

Levels are instances of the log4net.Core.Level class. The following levels are defined in order of increasing priority:

- ALL
- DEBUG
- INFO
- WARN

- ERROR
- FATAL
- OFF

> NOTE: Changing the level value to DEBUG will result in a more extensive log report.

For more information on Log4net, refer to: http://logging.apache.org/log4net/index.html

For Log4net configuration examples, refer to: http://logging.apache.org/log4net/release/config-examples.html

11.4 Log Rotation and Rollback

Each log has an Appender in its corresponding log4net file.

The Appender lines have the following structure:

```
<appender name="EscalationServiceFileAppender"
  type="log4net.Appender.RollingFileAppender" >
<File value="./AutomationServicesLogs/EscalationService-Log.txt" />
<AppendToFile value="true" />
<rollingStyle value="Size" />
<maxSizeRollBackups value="100" />
<maximumFileSize value="10000KB" />
<staticLogFileName value="true" />
<layout type="log4net.Layout.PatternLayout">
  <param name="ConversionPattern" value="%d [%t] %-5p %c - %m%n" />
</layout>
</appender>
```

The `<maxSizeRollBackups>` and `<maximumFileSize>` elements determine whether there will be only one accumulated log or a split one.

The `<maxSizeRollBackups>` value determines the number of log files.

The `<maximumFileSize>` value determines the size of each log file.

For example:

```
<maxSizeRollBackups value="100" />
<maximumFileSize value="10000KB" />
```

indicates that each corresponding log file will be accumulated up to 10MB and rollback will occur once 100 log files of that size are filled up.

This indicates that the total size of the log will be restricted to 1GB.

11.5 Log File Locations

The location of Situator Log4net files and their respective log files are described in the table below.

Situator Component	Log4net Location	Log Location
Automation Service	Operational\bin\AutomationServices.log4net	Operational\bin\Logs\Automation ServicesLogs\AutomationServices-log.txt
Sensor Server	SensorServer\bin\Situator.log4net	SensorServer\bin\Logs\SensorServer.log
Specific Gateway (Not - Hosted)	SensorServer\bin\Logs\Config\Stabilis.Situator.Sensors. Drivers.(GatewayName).log4net	SensorServer\bin\Logs\Gateway-Stabilis.Situator.Sensors.Drivers. (GatewayName).log
Specific Gateway (Hosted)	GatewaysHost\Logs\Config\(Gateway Name).log4net	GatewaysHost\Logs\(Gateway Name).log
Sensor Gateway Host	GatewaysHost\GatewayService.log4net	GatewaysHost\Logs\SensorGatewayHos t.log
Operational Service	Operational\bin\OpService.log4net	Operational\bin\Logs\OpServiceLogs\ OpService-Log.txt
Reporting Service	Operational\bin\ReportingService.log4net Operational\bin\Reports.log4net	Operational\bin\Logs\Reporting ServiceLogs\ReportingService-Log.txt
Control Room	ControlRoom\bin\ControlRoom.log4net	Logs\ControlRoomUI-Log.txt
Reporting Tool	ControlRoom\bin\Reports.log4net	Logs\Reports-Log.txt

CHAPTER 12 Obtaining and Deploying AD Certificates

12.1 Introduction

Active Directory Certificates Services (AD CS) can be used to issue certificates for all purposes. They are based on the Certificate Authority and Enrollment Services.

A standalone CA (Certificate Authority) can be used for demo purposes.

It is recommended to review the following Microsoft resources:

- Active Directory Certificate Services Overview
- Implement and manage AD CS
- Install the Certification Authority

Situator Services require several certificates to accomplish the following functionality:

- RabbitMQ notifications over TLS
- Authentication Method (via Authorization header)

The following certificates are required for the above functionality:

- Root CA Certificate (must be set as Trusted on all Situator machines)
- Public Key Certificate (as *PEM* file)
- Private Key Certificate (as *PEM* file)
- PKCS#12 Certificate (as *P12* file)
  - The P12 format is required by IDM (IdentityServer3)
  - This format is a combined version of Private and Public Key, and protected with a certificate password.

Customers should obtain certificate files from either:

- A well-known certificate authority (CA) - e.g., GoDaddy, DigiCert
- Using AD Enterprise CA

12.2 Extract Required Certificates from AD Certificate Services

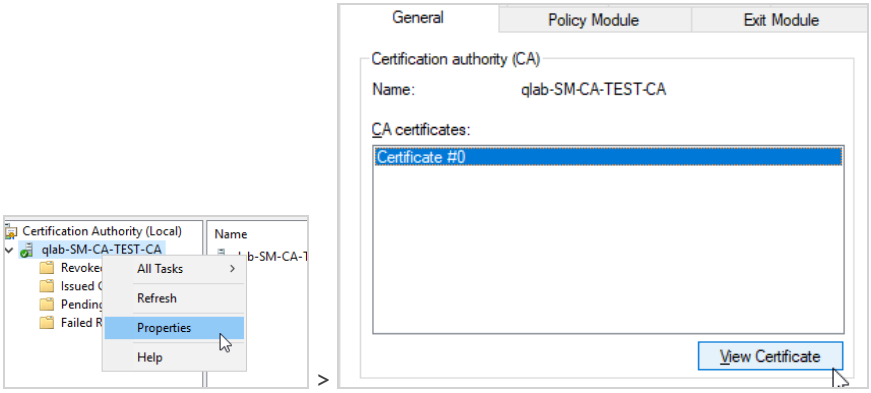
To deploy a standalone Root CA:

Follow the guidelines in this link:

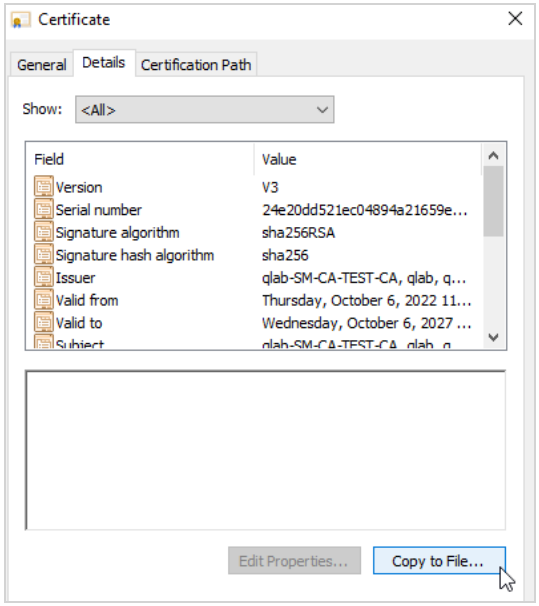
https://thesecmaster.com/step-by-step-procedure-to-set-up-a-standalone-root-ca-on-windows-server/

To obtain a Root CA Certificate:

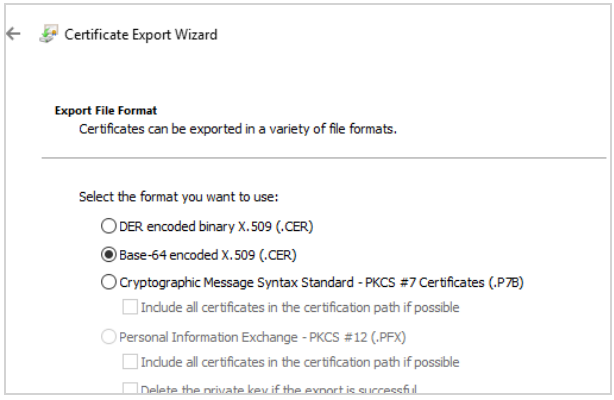
1. Open the **Certificate Authority management** console.
2. Right-click the certificate and select **Properties**. The *Properties* window opens with the **General** tab.



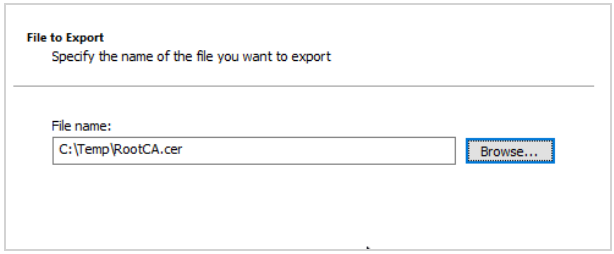
3. Click **View Cerificate**. The *Certificate* window opens.
4. Select the **Details** tab.



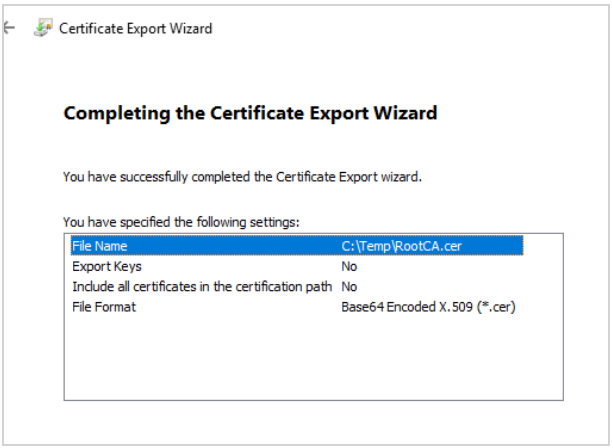
5. Click **Copy to File**. The *Certificate Export Wizard* window opens.



6. Select the format to use for exporting the certificate.
7. Select the file to export.

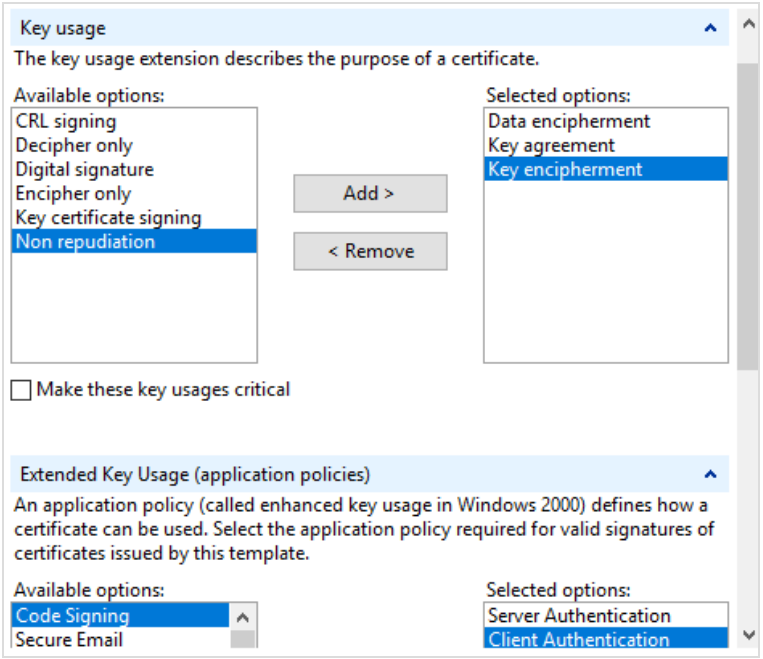


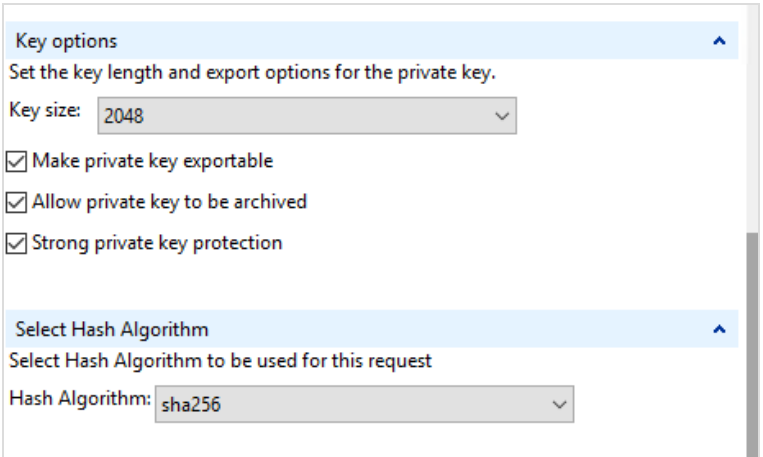
8. Set the destination file to which to export the certificate.
9. Click **Export**. Wait for the following success message window, summarizing the export settings.



To obtain Situator Services Certificate:

1. Submit a *Certificate Request* with the following requirements form the CA:

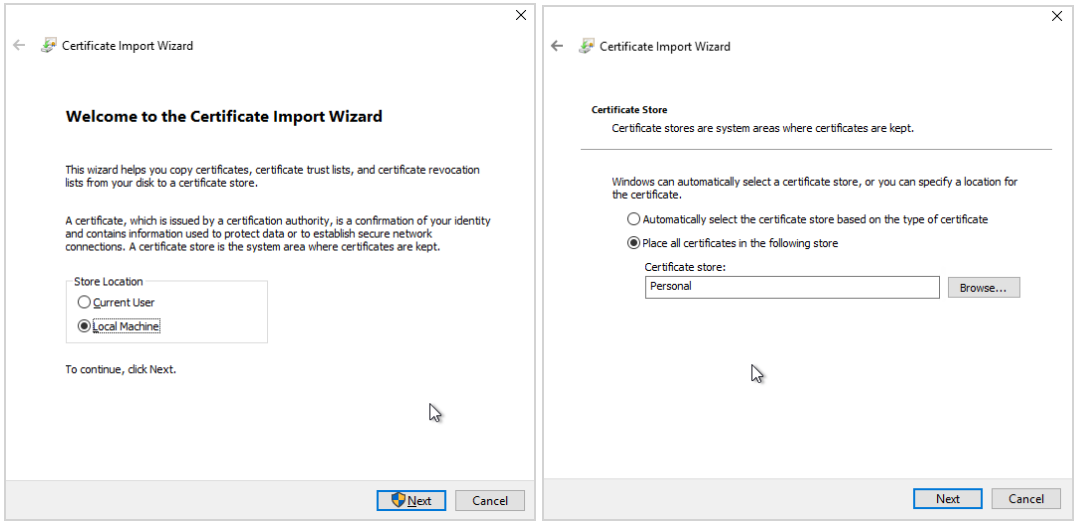




Make sure to select Allow private key to be archived so you can generate the private key PEM file.

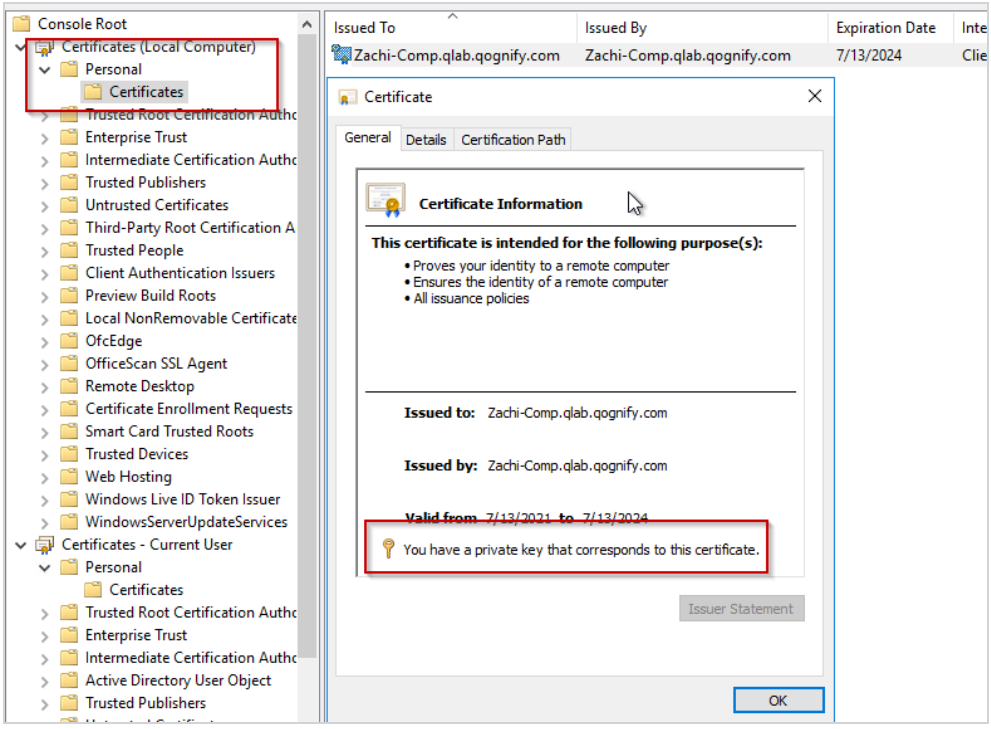
2. Install the certificate on the CA side (assuming the request was made on the CA so the key exists):

   Deploy it on **Local Machine > Personal** store.



3. Check that the key exists by opening the certificate on CA:

a. Open the **Certificates management** console for the "Local Machine":

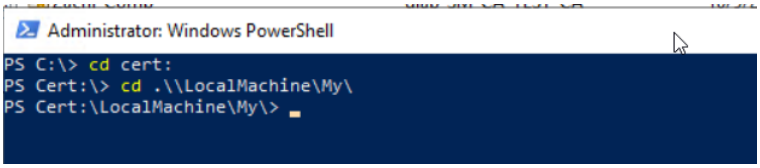


b. Look for the "You have a private key" message. If it does not appear, the key does not exist on CA. In this case, try to create it again.

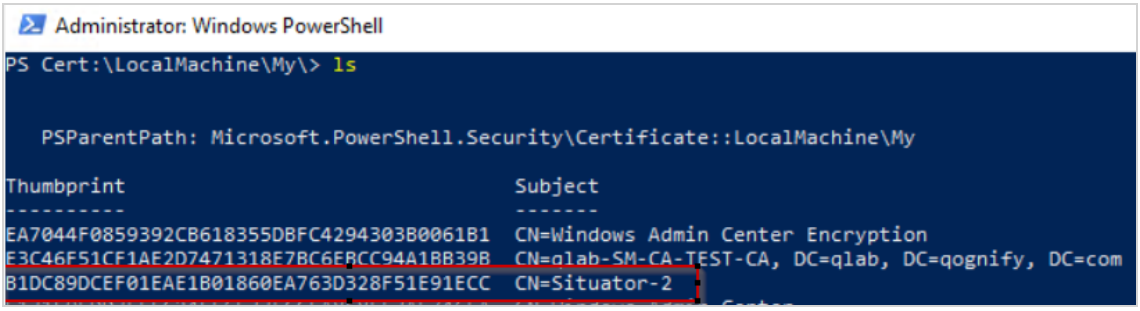
4. Extract the PFX certificate file using following PowerShell commands:

a. Open a **PowerShell** console (run as admin).

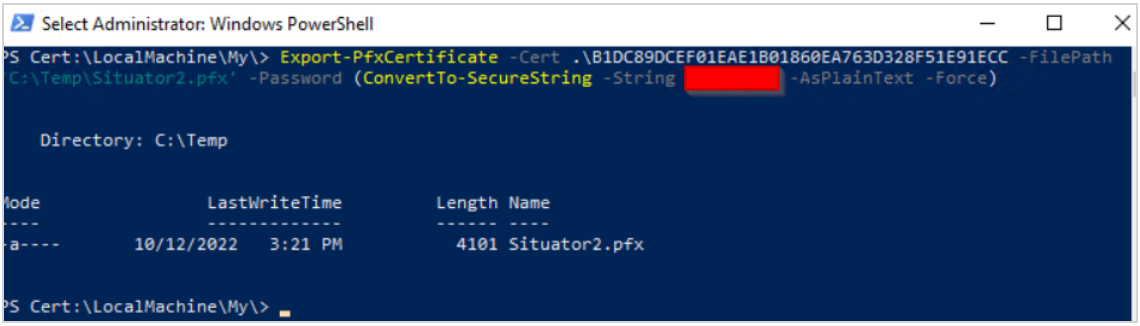
b. Switch to Computer Personal Certificate Store:



c. Check that the certificate is listed:



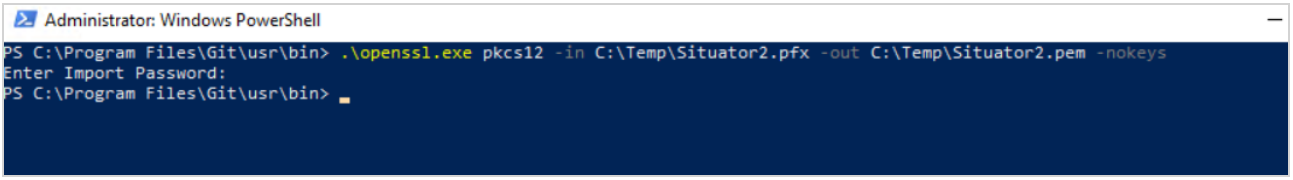
d. Run following command to extract it as PFX:



- Cert <Certificate Thumbprint> (use thumbprint as listed in previous step)
- FilePath <Target PFX filename>
- Password (ConvertTo-SecureString -String <key-password> -AsPainText -Force)

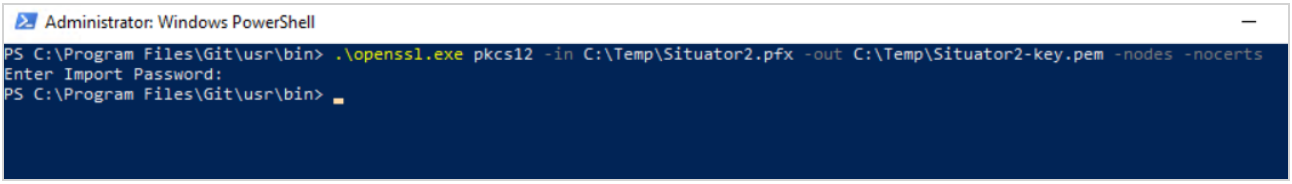
If no error occurs, the file can be found in target folder.

5. Install **OpenSSL** for Windows (can be achieved by installing GIT for Windows).
6. Extract the Public Key Certificate by running the following OpenSSL command:



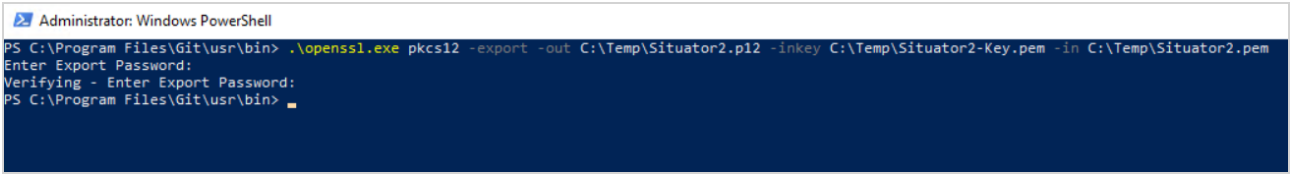
NOTE: A password is required to extract information from the PFX file.

7. Extract the Private Key Certificate by running following OpenSSL command:



NOTE: A password is required to extract information from the PFX file.

8. Add the Private and Public Key into a new P12 certificate file by running the following OpenSSL command:



NOTE: A password is required to protect the P12 file.

You now have the following certificate files:

- Root CA Certificate (e.g., RootCA)
- Public Key Certificate (e.g., Situator2.pem)
- Private Key Certificate (e.g., Situator2-Key.pem)
- PKCS#12 Certificate (e.g., Situator2.p12)

12.3 Deploy Customer Certificates on Situator Services

To deploy and update certificates on Situator hosts:

1. Verify that CA produces the certificates as *Trusted as root CA* on all Situator Hosts (preferably on the "Local Machine" level).
2. When using your own certificates, update the following configuration files (listed per Situator component) with the new certificate name/location. Use the following table for Situator Services locations to update:

Rabbit Server

%RABBITMQ_BASE%\advanced.config

Update the Application Certificates – CA Bundle, Public Key, and Private Key (as *pem* files).

```
{ssl_options, [{cacertfile,           "C:/Temp/SM-CA-Test/SM-CA-Test-RootCA.pem"},
               {certfile,             "C:/Temp/SM-CA-Test/Situator2.pem"},
               {keyfile,              "C:/Temp/SM-CA-Test/Situator2-Key.pem"},
               {verify,               verify_peer},
               {fail_if_no_peer_cert, true}]}
```

IDM WebService

<InstallDir>\IDM\web.config

Update the Encoded Certificate (Private+Public Keys + Password Protected) (as *P12* file)

```
<appSettings>
    ...
    <add key="CertificateFilePath" value="C:\temp\SM-CA-Test\Situator2.p12" />
```

(P12 is another form of PFX – but it is required due to IdentityServer3 limitation)

<InstallDir>\bin\RabbitSettings.xml

Update the Application Certifications list (the path must be defined in the format shown below)

```
<Certificate>
  <AuthorityCertificatePath>~/bin/Certificates/temp/SM-CA-Test-RootCA.pem</AuthorityCertificatePath>
  <CertificatePath>~/bin/Certificates/temp/Situator2.pem</CertificatePath>
  <KeyPath>~/bin/Certificates/temp/Situator2-key.pem</KeyPath>
  <KeyCertificatePath>~/bin/Certificates/temp/Situator2.p12</KeyCertificatePath>
  <KeyCertificatePathPassphrase>Qognify</KeyCertificatePathPassphrase>
  <ServerName></ServerName>
</Certificate>
```

IDM – WindowsAuth WebService

<InstallDir>\WindowsAuthService\web.config

- Update the Encoded Certificate (Private+Public Keys + Password Protected) (as P12 file)
- Update the Certificate password

```
<appSettings>
  <add key="CertificateFilePath" value="C:\temp\SM-CA-Test\Situator2.p12" />
  <!--EncryptionTypes that will be used for sensitive data: Symmetric, DPAPI
      Default Value: DPAPI (machine specific)-->
  <add key="LocalFilesEncryptionType" value="DPAPI" />
</appSettings>
<CertificateSection>
  <add key="CertificatePassword" value="AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAnip5ll/
</CertificateSection>
```

Operational Services (OP, BPM, Automation, Monitoring, Reporting)

<InstallDir>\bin\RabbitSettings.xml (Server Component)

Update the Application Certifications list (the path must be defined in the format shown below)

```
<Certificate>
  <AuthorityCertificatePath>C:\temp\SM-CA-Test\SM-CA-Test-RootCA.pem</AuthorityCertificatePath>
  <CertificatePath>C:\temp\SM-CA-Test\Situator2.pem</CertificatePath>
  <KeyPath>C:\temp\SM-CA-Test\Situator2-Key.pem</KeyPath>
  <KeyCertificatePath>C:\temp\SM-CA-Test\Situator2.pfx</KeyCertificatePath>
  <KeyCertificatePathPassphrase>Qognify</KeyCertificatePathPassphrase>
  <ServerName></ServerName>
</Certificate>
```

Sensor Server

<InstallDir>\bin\RabbitSettings.xml (Server Component)

Update the Application Certifications list (the path must be defined in the format shown below)

```
<Certificate>
  <AuthorityCertificatePath>C:\temp\SM-CA-Test\SM-CA-Test-RootCA.pem</AuthorityCertificatePath>
  <CertificatePath>C:\temp\SM-CA-Test\Situator2.pem</CertificatePath>
  <KeyPath>C:\temp\SM-CA-Test\Situator2-Key.pem</KeyPath>
  <KeyCertificatePath>C:\temp\SM-CA-Test\Situator2.pfx</KeyCertificatePath>
  <KeyCertificatePathPassphrase>Qognify</KeyCertificatePathPassphrase>
  <ServerName></ServerName>
</Certificate>
```

LogBook Server

<InstallDir>\bin\RabbitSettings.xml (LogBook Component)

Update the Application Certifications list (the path must be defined in the format shown below)

```
<Certificate>
  <AuthorityCertificatePath>C:\temp\SM-CA-Test\SM-CA-Test-RootCA.pem</AuthorityCertificatePath>
  <CertificatePath>C:\temp\SM-CA-Test\Situator2.pem</CertificatePath>
  <KeyPath>C:\temp\SM-CA-Test\Situator2-Key.pem</KeyPath>
  <KeyCertificatePath>C:\temp\SM-CA-Test\Situator2.pfx</KeyCertificatePath>
  <KeyCertificatePathPassphrase>Qognify</KeyCertificatePathPassphrase>
  <ServerName></ServerName>
</Certificate>
```

SM WebAPI

<InstallDir>\bin\RabbitSettings.xml (WebAPI Component)

Update the Application Certifications list (the path must be defined in the format shown below)

```
<Certificate>
  <AuthorityCertificatePath>C:\temp\SM-CA-Test\SM-CA-Test-RootCA.pem</AuthorityCertificatePath>
  <CertificatePath>C:\temp\SM-CA-Test\Situator2.pem</CertificatePath>
  <KeyPath>C:\temp\SM-CA-Test\Situator2-Key.pem</KeyPath>
  <KeyCertificatePath>C:\temp\SM-CA-Test\Situator2.pfx</KeyCertificatePath>
  <KeyCertificatePathPassphrase>Qognify</KeyCertificatePathPassphrase>
  <ServerName></ServerName>
</Certificate>
```

ARE (source > build then deploy)

<InstallDir>\ARE\ARE\src\main\resources\SituatorStreamBaseConfiguration.xml

Update the Rabbit Certificate path and password.

```
<RabbitUserName>mgCGbV3dac2NU5yiA/IFNJShhkLgxYwANnV+EsEj0EM=</RabbitUserName>
<RabbitPassword>Zzk2Yx2hYMnP2Uyyr8JQOJShhkLgxYwANnV+EsEj0EM=</RabbitPassword>
<RabbitCertificatePath>C:\Temp\SM-CA-Test\Situator.p12</RabbitCertificatePath>
<RabbitCertificatePassword>paoeALTfi2PrxRtZXyJtYR2zQmDHzGdGYvQObA+XYvc=</RabbitCertificatePassword>
```

BI Listener (OIC)

<InstallDir>\bin\Dashboard.BIListener.exe.config

Update the Certificate path and passphrase for the Rabbit TLS connection.

```
<RabbitMqSsl>
  <add key="CertificatePath" value="C:\temp\SM-CA-Test\Situator2.p12">
  </add>
  <add key="CertificatePassphrase" value="AQAAANCMnd8BFdERjH*enc*oAwE/C1+sBAAAAE
  </add>
```

Gateway Host

<InstallDir>\RabbitSettings.xml (Gateway Component)

Update the Application Certifications list (the path must be defined in the format shown below)

```
<Certificate>
  <AuthorityCertificatePath>C:\temp\SM-CA-Test\SM-CA-Test-RootCA.pem</AuthorityCertificatePath>
  <CertificatePath>C:\temp\SM-CA-Test\Situator2.pem</CertificatePath>
  <KeyPath>C:\temp\SM-CA-Test\Situator2-Key.pem</KeyPath>
  <KeyCertificatePath>C:\temp\SM-CA-Test\Situator2.pfx</KeyCertificatePath>
  <KeyCertificatePathPassphrase>Qognify</KeyCertificatePathPassphrase>
  <ServerName></ServerName>
</Certificate>
```

CHAPTER 13 Configuring AllowUrlList to Validate a Token from IDM

This procedure describes how to configure a list of valid DNS or IP addresses that WebAPI can trust, allowing login to the Qognify Web Client.

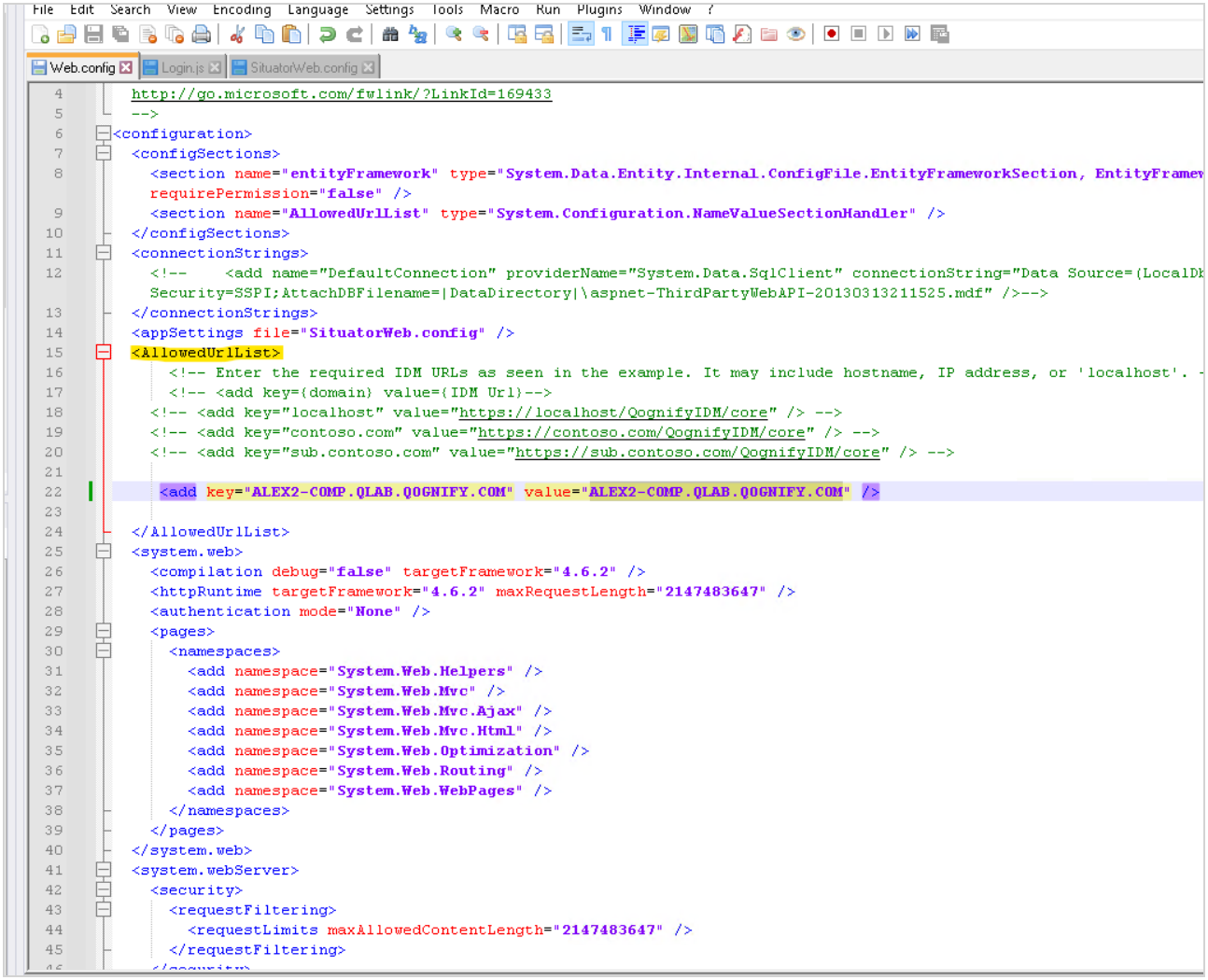
To configure a list of valid DNS or IP addresses:

1. Open the *Web.config* file (*C:\Program Files (x86)\Qognify\Situator\WebAPI\Web.config*).
2. In the **AllowUrlList** parameter, enter the required IDM URLs as required. You can set a Domain/FQDN[1]/IP/localhost or leave it empty.

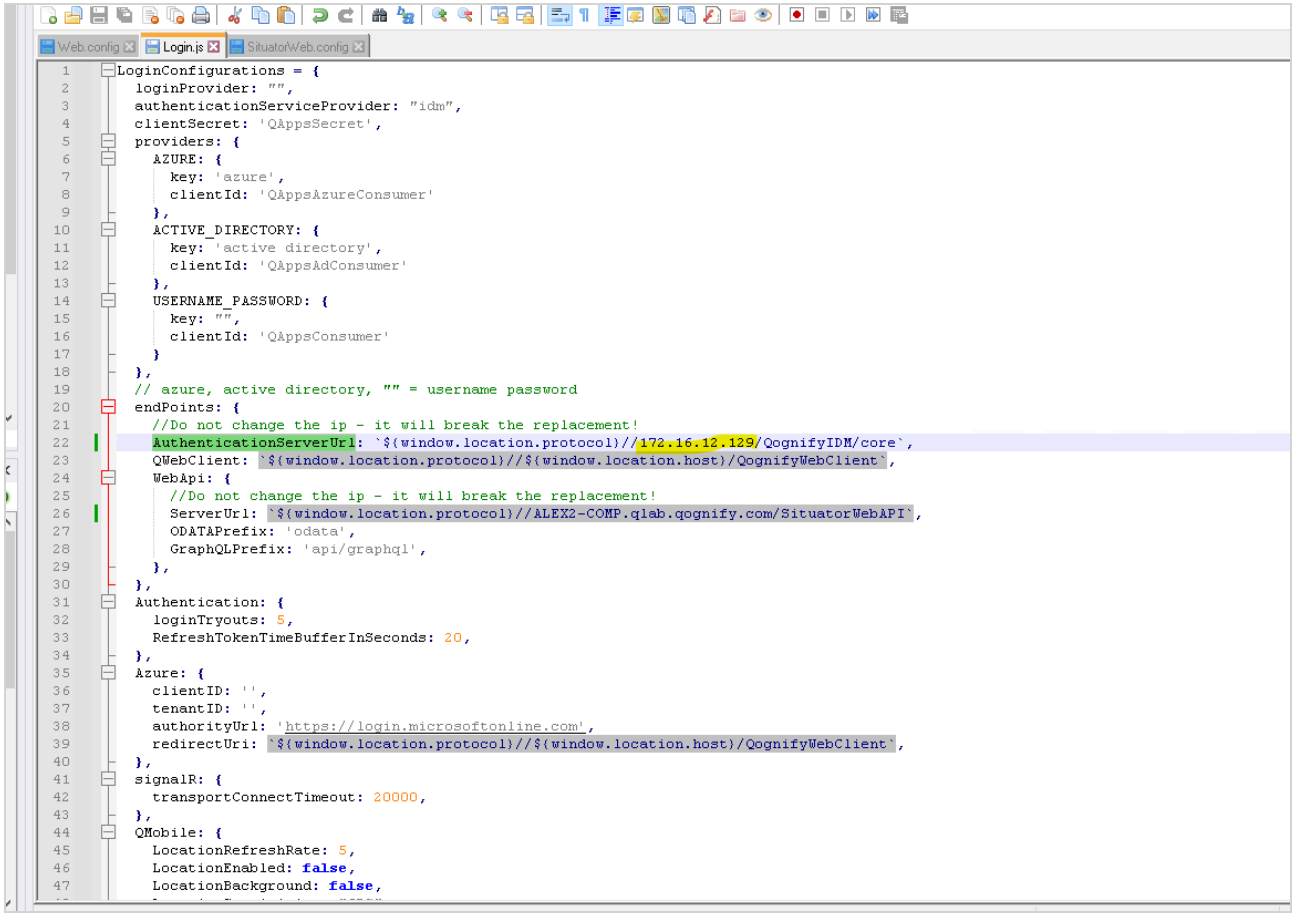
   For example:

   Add key

[1]Fully Qualified Domain Name



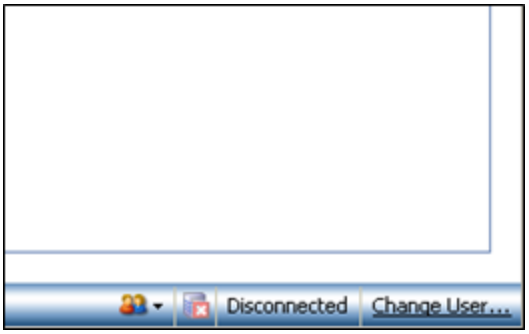
3. Open the *Login.js* file (*C:\Program Files (x86)\Qognify\Situator\QognifyWebClient\Config\Login*).
4. Check that the value of **AuthenticationServerUrl** is the same as set in Step 2.



- If it is different from the value set in Step 2 (for example, IP instead of a hostname), the authorization and login to Web Client will fail. To fix this - add the IP to the `AllowUrlList` parameter in the *Web.config* file.
- If the URL is left empty, the `AllowUrlList` parameter value is taken from the `IdentityServerAddress` parameter in the *Situator.web.config C:\Program Files (x86)\Qognify\Situator\WebAPI\Situator.web.config* file.

CHAPTER 14 Client IP Configuration with Multiple Network Adapters

In *Control Room*, using a client station to log into the Situator Server sometimes results in a successful login but a "Disconnected" status at the bottom-right corner of the main *Control Room* window, as shown below.



In addition, in the Control Room log file, the Notification Client Logger logs that the registration request has failed:

```
NotificationClientLogger – Stabilis.Situator.ControlRoom.UI: The NS Response To Regis-
tration Request Is: Failed
```

This issue arises in a situation where a client station contains more than one network adapter for the Notification Client portion of the application. The login attempt may be made using the wrong network adapter to register to the Notification Server.

Manually setting the client IP in the Control Room configuration file resolves this issue.

To set the client IP in the control room configuration files:

1. Open the file *Stabilis. Situator.ControlRoom.UI.exe.config* in edit mode.

   NOTE: Do not use Wordpad as a text-editor as it may corrupt the configuration file and prevent Control Room from starting. Instead, use either Notepad or Notepad++.

2. Locate the `<add>` XML element containing `"UsePortRange"` as the key attribute value (under the XML element `<appSettings>`). Change its value attribute to `"false"`:

   ```
   <add key="UsePortRange" value="false">
   ```

```
</add>
```

3. Locate the `<application>` XML element under `<system.runtime.remote>`. Insert the following lines immediately after the `<channels>` XML node:

```
<channel ref="tcp" port="5010" machineName="Station'sIP">
<serverProviders>
<formatter ref="binary" typeFilterLevel="Full" />
</serverProviders>
</channel>
```

> NOTE: As the `machineName` attribute value, insert the station's IP address of the network adapter which communicates with the Situator Server.

4. Restart *Control Room*.

APPENDIX A Terms and Abbreviations

The acronyms and abbreviations used in the Situator documents are listed below.

Term	Description
ACS	Access Control System
AD	Active Directory
AMS	Application Management Server
API	Application Programming Interface
ARE	Advanced Rule Engine. An advanced rule engine in Situator's Operational Intelligence Center that detects and reacts to specific business scenarios that require handling multiple data sources in complex ways such as correlating data or aggregating and computing data over a time window
AVI	Video format for exporting files
AVMD	Advanced Video Motion Detection
BPM	Business Process Manager
CEP	Complex Event Processing engine
CCTV	Closed-circuit Television
CSV	Comma-separated values (file format)
CR	Control Room
DB	Database
DMZ	Demilitarized Zone
DNS	Domain Name System
DRP	Disaster Recovery Protocol
DVR	Digital Video Recorder
DPA	Data Processing Agreement
DWH	Data Warehouse

Term	Description
EULA	End User License Agreement
FOV	Field of View
FQDN	Fully Qualified Domain name
GIS	Geographic Information System
GPS	Global Positioning Service
GWH	Gateway Host
GMT	Greenwich Mean Time (also Coordinated Universal Time)
gMSA	Group Managed Service Accounts
HA	High Availability
IDM	Identity Management system
IEP	Incident Extended Properties
IP	Internet Protocol
IR	Infrared (light)
LAN	Local Area Network
LOS	Level of Service
LPR	License Plate Recognition - A sensor type
Mbps	Megabits per second
MIB	The Management Information Base is a text file that defines the interface between the agents and the NMS
MKV	Video container format for exporting files
MN	Mass notification - a MN message delivery option available when third-party MNS installed
MNS	Mass notification system - an external, third party system
MSDTC	Microsoft Distributed Transaction Coordinator service
NIC	Network Interface Card

Term	Description
NMS	Network Monitoring System (NMS) is the monitoring service which monitors VisionHub components via SNMP. The SNMP trap events are logged in its database and enables ticketing and notification policies.
NTP	Network Time Protocol - a protocol for distributing time information between computers on a network
NVF	Video container format (Qognify proprietary)
NVR	Network Video Recorder
OIC	Operational Intelligence Center
OS	Operating System
OSD	OnScreen Display
Packaged OpS	Situator Packaged Operational Solutions
PSIM	Physical Security Information Management
PTZ	Pan Tilt Zoom
RAID	Redundant Array of Independent Disks
RFID	Radio Frequency Identification
RSVR	Redundant SVR (Smart Video Recorder)
SIP	Session Initiation Protocol
SLA	Service Level Agreement. Service Levels are defined in Situator and used in Situator's OIC to display trends or trigger actions.
SNMP	Simple Network Management Protocol. A network protocol used for managing and reporting device status and events
SNTP	Simple Network Time Protocol - a simplified version of NTP
SMTP	Simple Mail Transfer Protocol is an Internet standard for electronic mail (e-mail) transmission across IP networks
SVR	Smart Video Recorder

Term	Description
SOP	Standard Operating Procedure
SSL	Secure Sockets Layer
SSO	Single Sign-On
TCP	Transfer Control Protocol
TEP	Task Extended Property
TLS	Transport Layer Security
UC	Uniqueness Constraint
UTC	Coordinated Universal Time (also Greenwich Mean Time)
VMD	Video Motion Detection
VoIP	Voice over IP - generic term to describe the transport of voice over an IP network
VMX	Video Wall
WAN	Wide Area Network
WPF	Windows Presentation Foundation