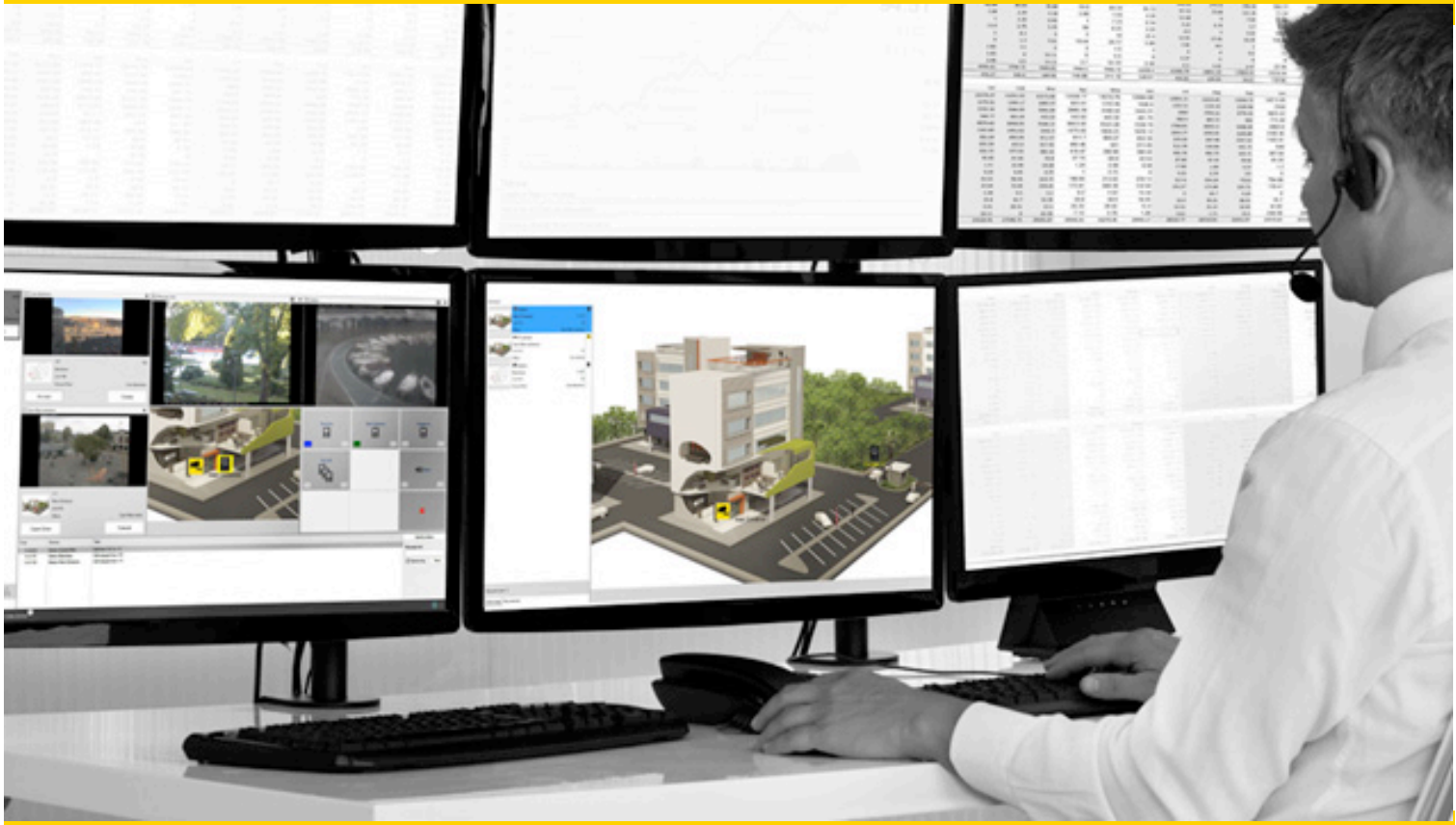
VINGTOR STENTOFON
by ZENITEL GROUP



TECHNICAL MANUAL

Vingtor-Stentofon Operator Configuration & Operation

Contents

1 Introduction

1.1 Document Scope

This document describes the configuration and operation of Vingtor-Stentofon Operator. The information in this document is aimed at System Administrators, System Installers, and End Users.

1.2 Product Information

Vingtor-Stentofon Operator is a PC based system management package for the AlphaCom audio server. It shows the status of intercom devices in maps and makes it easy for operators to handle calls and call requests. It is primarily intended for use in a control room environment.

1.3 Publication Log

Revision	Date	Author	Status
1.2	20.9.2016	HKL	Updated, Proofread and Published
1.3	28.9.2016	HvD	Vingtor-Stentofon Operator server Windows compatibility updated
1.4	17.10.2016	HvD	Additional information in appendix C regarding problems with logging in
1.5	20.10.2016	HvD	License change
1.6	22.2.2017	HvD	Version 1.1.3.2, additional functionality
1.7	8.9.2017	HvD	Version 1.2.3.1; associate up to 4 cameras to an intercom; Optionally show video from call/call request in video tile – desktop space saving
1.8	8.1.2018	HvD	Version 1.3.3.1; Linking of maps; Standard templates; import and export of templates; Copying of device properties
1.9	31.5.2018	HvD	Name change to Vingtor-Stentofon Operator
1.10	20.6.2018	HvD	Updated screenshots from AlphaView to Vingtor-Stentofon Operator; Added description Move from AlphaView to Vingtor-Stentofon Operator
1.11	12.9.2018	HvD	Version 1.4.3.0, see software release notes
1.12	27.8.2019	HvD	Version 1.5.3.2, see software release notes
1.13	30.9.2019	HvD	Version 1.5.3.4, see software release notes
1.14	11.3.2020	HvD	Version 1.6.3.0, snapshot functionality added
1.15	1.9.2020	HvD	Updated description for using the credential fields for IP cameras

1.4 Related Documentation

Doc. no.	Documentation
A100K10805	AlphaCom XE Installation, Configuration & Operation
A100K10602	AlphaCom E/XE User Guide
A100K11931	ICX-AlphaCom Quick start guide
Zenitel Wiki	https://wiki.zenitel.com/wiki/Category:ICX-AlphaCom_Platform

2 Installation

2.1 Move from AlphaView to Vingtor-Stentofon Operator

Please visit AlphaWiki for details: https://wiki.zenitel.com/wiki/Moving_from_AlphaView_to_Vingtor-Stentofon_Operator

To move from AlphaView to Vingtor-Stentofon Operator:

- Install Vingtor-Stentofon Operator, server and client
- In the installation dialog, enable **Use existing configuration file (ConfigData.xml)**
- Manually uninstall AlphaView, server and client

2.2 Server

2.2.1 Minimum Requirements and Prerequisites

The minimum hardware requirements for the Vingtor-Stentofon Operator server are:

- Memory: >= 2GByte
- Processor speed: >= 1.6 GHz

The Vingtor-Stentofon Operator server requires Windows 10 or higher or Windows Server 2012 or higher.

Vingtor-Stentofon Operator requires AMC-IP software version 11.2.3.1 or higher.

The Vingtor-Stentofon Operator server requires that the following software is available on the server:

- .NET 4.6.2
- SQL Express or SQL server – SQL express is the free version of SQL server
  Check the Microsoft website for compatibility between the operating system and SQL server versions – in general, use the latest available SQL server version

2.2.2 Server Installation

- Run **VS-OP_SetupServer_x.x.x.x.exe**

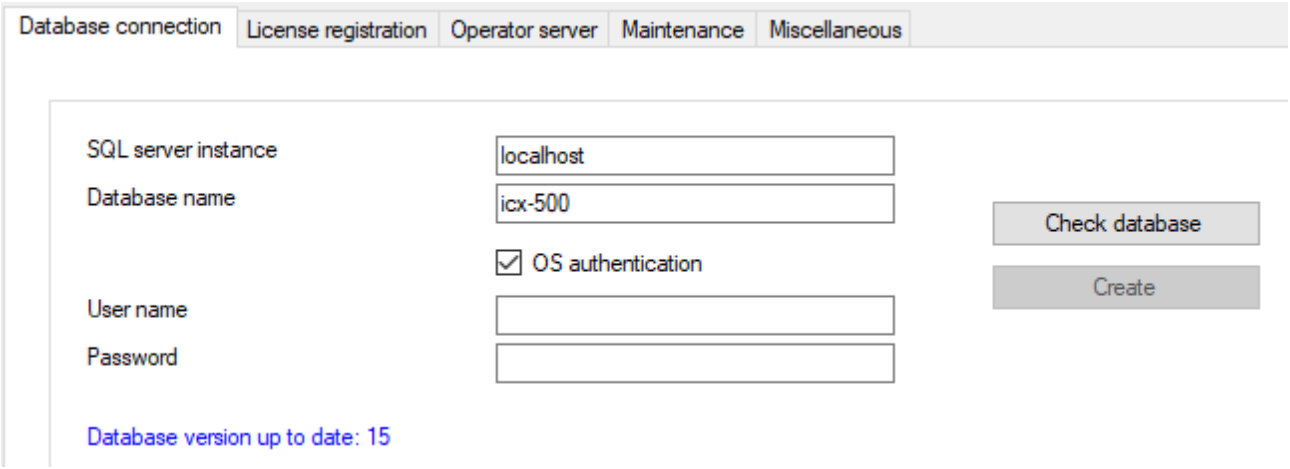
x.x.x.x denotes the version number.

By default, the server will be installed in:

C:\Program Files (x86)\Vingtor Stentofon\Vingtor-Stentofon Operator\Vingtor-Stentofon Operator Server

At the end of the installation process the Vingtor-Stentofon Operator Config Tool will be started. It has five Tabs:

Database connection

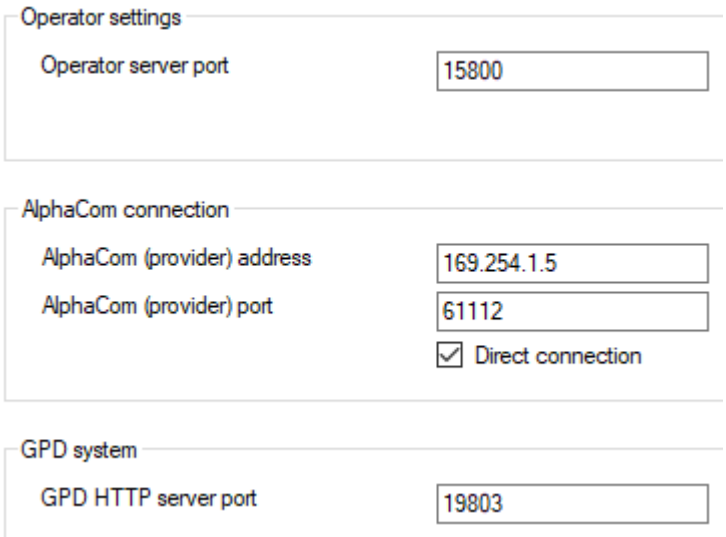


- Enter the name of the **SQL server instance**
- Enter a name for the **Database**
- Click **Create** – via the start menu it is possible to start this dialog later and click **Check database** in case there are unexpected problems
- It is possible to use either Windows Authentication or SQL Server Authentication (see Appendix C, also when experiencing login problems when starting the Vingtor-Stentofon Operator client)

After an upgrade of an existing installation there may be a warning 'Incorrect database version'. As soon as the Vingtor-Stentofon Operator is started, Vingtor-Stentofon Operator will automatically convert the database to the new format.

License registration (see section 2.4.3.2)

Vingtor-Stentofon Operator server

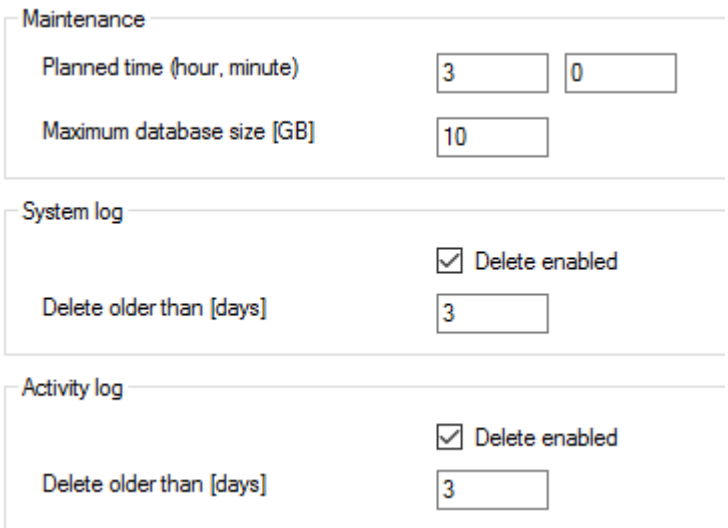


- **Operator server port**: Enter the server port number to which Vingtor-Stentofon Operator clients can connect. Make sure that all clients are configured with the same port number as entered in this field.

- **AlphaCom (provider) address**: Enter the IP address of the AlphaCom that the Vingtor-Stentofon Operator Server should connect to (in an AlphaNet it is only necessary to connect to one AlphaCom)
- **AlphaCom (provider) port**: Enter the API/OPC port number, either **61112** or **61113** - make sure that the relevant port is opened in AlphaWeb in all exchanges in an AlphaNet
- **Direct connection:** By default enabled. When enabled, the server connects directly to AlphaCom; It is possible to not directly connect to AlphaCom, but to connect through the AlphaNetServiceProvider; In that case the **Provider address** is the IP-address of the PC on which AlphaNetServiceProvider is running, and the **Provider port** is the port which is assigned as to be connected to
  For details consult https://wiki.zenitel.com/wiki/AlphaNetServiceProvider
  Using the AlphaNetServiceProvider is only necessary when for some reason all API/OPC ports are in use. The AlphaNetServiceProvider itself connects to either port 61112 or 61113; the already connected integration which will be disconnected must itself also support connection via the AlphaNetServiceProvider, as otherwise that existing integration will no longer be operational
- **GPD HTTP server port**: Enter an appropriate GPD HTTP server port number.
  GPD (General Purpose Devices) is a separate part of the Vingtor-Stentofon Operator Server. External systems can send status information to the server through which the appearance of GPD icons can be manipulated. External systems can also receive information from Vingtor-Stentofon Operator. For more information, see appendix D.

Maintenance

Vingtor-Stentofon Operator can log activity (calls and call requests) and system events to the database. To make certain that the database does not grow too much, this TAB allows criteria to be set for deleting logging information. During the deletion process, Vingtor-Stentofon Operator will also clean the database.



- **Planned time (hour minute)**: The server will do the maintenance every day at the time entered here.

- **Maximum database size [GB]**: The maximum size the database is allowed to grow, this is especially important if it is decided not to delete logs based on number of days. SQL Express has a limit for the database size of 10 GB.
- **Delete enabled** and **Delete older than [days]:** Enable deletion of the **System log** and **Activity log** and enter the number of days for which the information must be kept.

Miscellaneous

Sometimes it happens that there is a mismatch between the date/time format as used by the SQL database server and the format which VS-Operator uses to retrieve history data. This results in the VS-Operator client seemingly hanging for about 45 seconds after a data retrieve attempts, followed by an error message. If this is the case then a choice can be made in this TAB to rectify the issue:

- **Auto**: the default selection, which will work in most cases
- **Custom**: Select one of the pre-defined formats or enter a format as required in case one of the pre-defined formats does not solve the problem; Please refer to documentation from Microsoft which can be found on-line for information about possible formats which is normally related to language specific setup of the Windows Operating System
  After a change, the VS-Operator server must be stopped and started again

- After any changes click **Save**
- Start the server by clicking **Start**

2.2.3 Windows Firewall

In case of connection problems from clients on remote computers, make certain that the relevant ports are opened on the PC on which the Vingtor-Stentofon Operator server is running.

2.3 Client

2.3.1 Minimum Requirements and Prerequisites

The minimum hardware requirements for Vingtor-Stentofon Operator client are:

- Memory: >= 2 GByte
- Processor speed: >= 1.6 GHz

Vingtor-Stentofon Operator client supports video cameras. Although there is no restriction to the number of cameras that can be configured, the possibility to view multiple streams simultaneously depends on the capability of the client PC hardware. The minimum hardware requirement as specified in this paragraph supports the possibility to associate one camera with each intercom station and view streams belonging to a call request and an established call simultaneously.

Vingtor-Stentofon Operator client requires Windows 8.1 or higher.

Vingtor-Stentofon Operator client requires that the following software is available:

- .NET 4.6.2

2.3.2 Client Installation

- Run **VS-OP_SetupClient_x.x.x.x.exe**

x.x.x.x denotes the version number.

By default, the client will be installed in:

C:\Program Files (x86)\Vingtor Stentofon\Vingtor-Stentofon Operator\Vingtor-Stentofon Operator Client

2.3.3 Client Configuration

Any client can be used to configure the Vingtor-Stentofon Operator system. All information will be stored by the Vingtor-Stentofon Operator server in an SQL database - see chapter 3 for details.

There is a minimum amount of configuration to be done for each individual Vingtor-Stentofon Operator Client.

To configure the client:

- Start the Vingtor-Stentofon Operator Client
- User name: **admin**
- At initial startup there is no password (define one as soon as possible)
- At the bottom right of the Vingtor-Stentofon Operator screen, click on the down-arrow  and select **Options**
- Enter the **Host name** or **IP address** – the host is the computer on which Vingtor-Stentofon Operator Server is running
- Enter the **Port number** (default is 15800) – the port number must be the same as entered in section 2.2.2 for the Vingtor-Stentofon Operator Server
- Enter the identity of this Vingtor-Stentofon Operator client - the identity is the client-associated intercom station, AlphaCom **Node number** and **Directory number**
- Select a **Language** - see appendix G for language availability
- Normally a double click on an icon in a map is required to start its default action; Select **Single click as double** a single click is the preferred mode of operation – note that this behavior only applies to icons in maps
- Enabling **Use thin bars** maximizes the available screen area by making the bars at top and bottom (configuration mode) or only bottom (operational mode) thinner. Note that on small touch screens it may then become difficult to select menu options
- **Own station visibility** – Select from **Default**, **Visible** or **Hidden**; Default takes the visibility from the global setting see paragraph 3.12.3; This setting determines whether the with this client associated station is shown on a map or not
- Operators can be assigned the right to take **Snapshots**
    - Define the **Directory** where the snapshots shall be stored; note that the snapshots will be stored in a subdirectoy of the defined directory, the subdirectory name is related to the date the snapshot is taken
    - Snapshots will be deleted automatically after the specified number of days

- Vingtor-Stentofon Operator is able to play H.264 coded video streams; depending on the network conditions it may be necessary to change the buffer size for H.264 streams; Size is set in msec, default is 1100
  The change must be entered directly into the file **ConfigData.xml**, which can be found here: **C:\Program Files (x86)\Vingtor Stentofon\Vingtor-Stentofon Operator\Vingtor-Stentofon Operator Client**
  Find the line **<string>NetworkCachingRtsp=1100</string>** and change the value to an appropriate number

2.4 Licensing

Vingtor-Stentofon Operator is a licensed product. Two types of licenses are required:

- AlphaCom or ICX-AlphaCom OPC/API license
- Vingtor-Stentofon Operator license

2.4.1 AlphaCom OPC/API license

The number of intercom stations that will be shown in Vingtor-Stentofon Operator depends on the presence of the appropriate license in AlphaCom.

- 1009649901 - VS API License Supporting 40 Stations
- 1009649902 - VS API License Supporting 80 Stations
- 1009649903 - VS API License Supporting 160 Stations
- 1009649904 - VS API License Supporting 240 Stations
- 1009649905 - VS API License Supporting 320 Stations
- 1009649906 - VS API License Supporting 400 Stations
- 1009649907 - VS API License Supporting 552 Stations

Note that each AlphaCom in an AlphaNet will require its own API license

2.4.2 ICX-AlphaCom OPC/API license

The number of intercom stations that will be shown in Vingtor-Stentofon Operator depends on the presence of the appropriate license in ICX-AlphaCom.

- 1002602306 - API License Supporting 64 Stations
- 1002602309 - API License Supporting 512 Stations

Note that each ICX-AlphaCom in an AlphaNet will require its own API license

2.4.3 Vingtor-Stentofon Operator License

2.4.3.1 General

The Vingtor-Stentofon Operator server is the licensed entity. Licensing is based on:

The maximum number of clients that can be connected simultaneously

- Functionality: Basic and Enhanced; Appendix H lists the by the Basic and Enhanced licenses supported functionality.

Without license the server runs in demo mode; Demo mode is similar to the Basic license mode, but with the following restrictions:

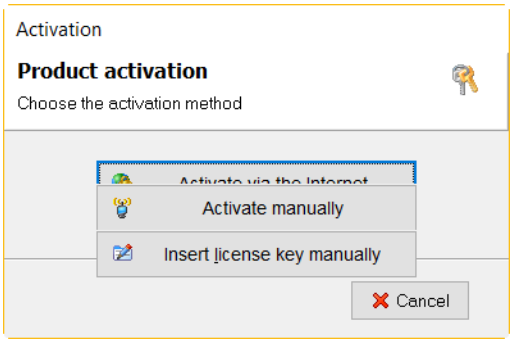
- Maximum 7 devices can be defined
- Maximum 4 devices of the same type can be defined

A server does not support a mix of clients with different functionality licenses.

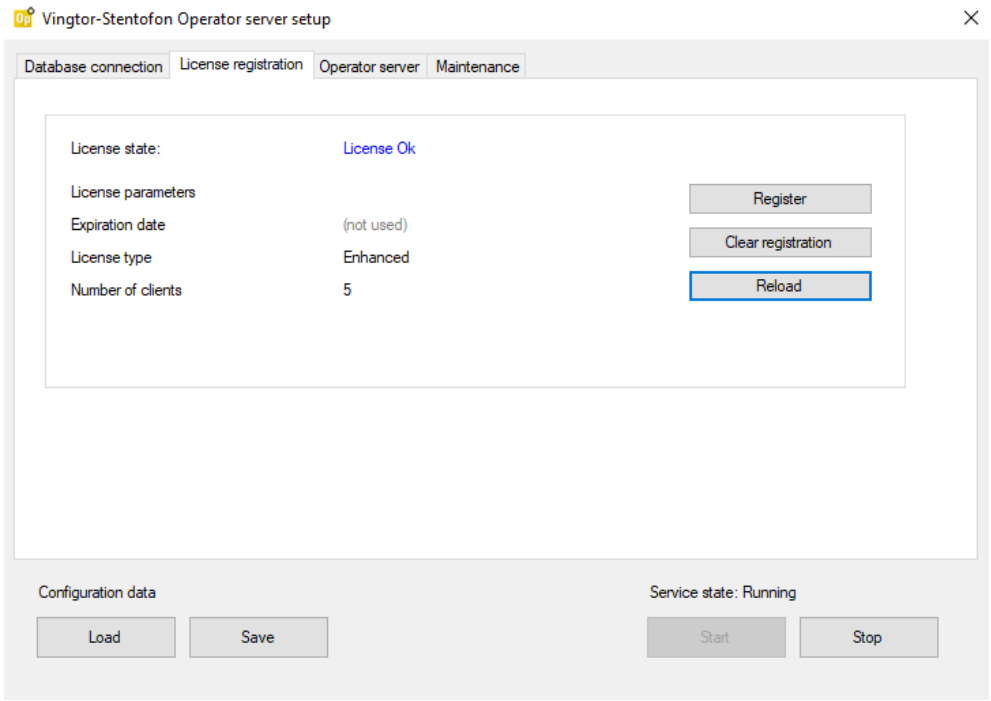
2.4.3.2 Installing a License

When the product has been purchased, a license serial number will be mailed to the purchasing organization. The license can be installed using the Vingtor-Stentofon Operator Config Tool. The easiest way to install a license is via the on-line procedure - please make certain the server PC is connected to the Internet.

- Start **Vingtor-Stentofon Operator Config Tool**
- If the Vingtor-Stentofon Operator server is running, click **Stop**
- Select the tab **License registration** and click **Register**
- Select **Activate via the Internet**
- Enter the license serial number in the input fields and click **OK**
- Normally the registration will only take a few seconds. If for any reason the registration process fails (timed out), or the computer is not connected to the Internet, use the **Activate manually** procedure
    - Enter the license serial number in the input fields and click **OK** - you will get a dialog-box that contains a **Product Activation** code
    - Mail the Product Activation code to CustomerService@zenitel.com and you will be provided with the license registration code – depending on the time zone you are in, this may take some hours.
    - Click **Insert license key manually** and enter the code into the input field



If the registration is successful, the **License state** will show **License Ok**.



Select **Clear registration** to delete the registration file and Vingtor-Stentofon Operator will revert to demo-mode.

3 System Configuration

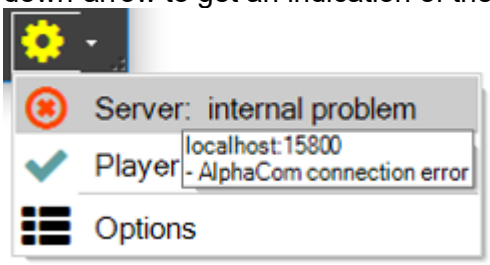
3.1 General

3.1.1 Client startup

Start the Vingtor-Stentofon Operator Client.

The bottom right indicator displays connection statuses

-  - connection client-server and server-AlphaCom are OK
-  - One of the connections client-server or server-AlphaCom are failing, click on the down-arrow to get an indication of the error



If there are no connection errors, log in into the client. The default username is admin, by default there is no password. It is highly recommended to add a password for the admin user.

If there is a connection error between client and server, login will fail with an error message.

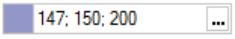
If there is a connection error between server and AlphaCom, login is possible. Note that in this case in configuration mode auto-population of intercom stations is not possible, see paragraph 3.2.5.2. All other configuration actions are possible. Login in operational mode with a failing server-AlphaCom connection will mean that no data can be sent or received to and from the AlphaCom.

3.1.2 Selecting Items to Configure

Different parts of Vingtor-Stentofon Operator can be configured by using the icons on the right-hand side of the screen.

3.1.3 Use of Colors

In certain menus, it is possible to select colors from a palette. This normally results in a color coded by three values in the range 0-255 for Red, Green and Blue. It is possible to add a 4th value as the first parameter, which indicates the transparency: 0 for fully transparent (the color is not visible) and 255 for a solid color (255 will 

not 

be shown).

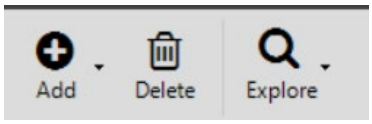
3.2 Objects

3.2.1 Object Types

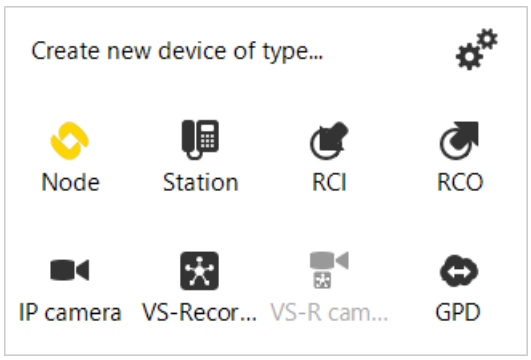
Vingtor-Stentofon Operator recognizes different types of objects such as:

- AlphaCom node
- AlphaCom connected station: (IP) intercoms and (SIP) phones
- RCI
- RCO
- Vingtor-Stentofon Recorder 2.0
- VS-Recorder connected cameras
- IP cameras with a direct connection to Vingtor-Stentofon Operator
- General Purpose Devices (GPD)
- Defined areas in maps to link to other maps in the hierarchy

3.2.2 Entering device parameters manually or automatically



- Click **Add** to select the device types



The following device types can be added manually:

- AlphaCom nodes
- AlphaCom connected intercoms and phones
- Directly to Vingtor-Stentofon Operator streaming IP cameras
- VS-Recorder 2.0
- RCIs
- RCOs
- GPDs (General Purpose Devices)

The following devices can be imported automatically:

- AlphaCom nodes
- AlphaCom connected intercoms and phones
- Cameras which stream to VS-Recorder 2.0 (VS-Recorder 2.0 must first be added manually)

When importing intercoms and cameras it is possible to select which devices will be imported. Note that all in the VS-Recorder 2.0 defined devices will be shown, so make certain to only select cameras.

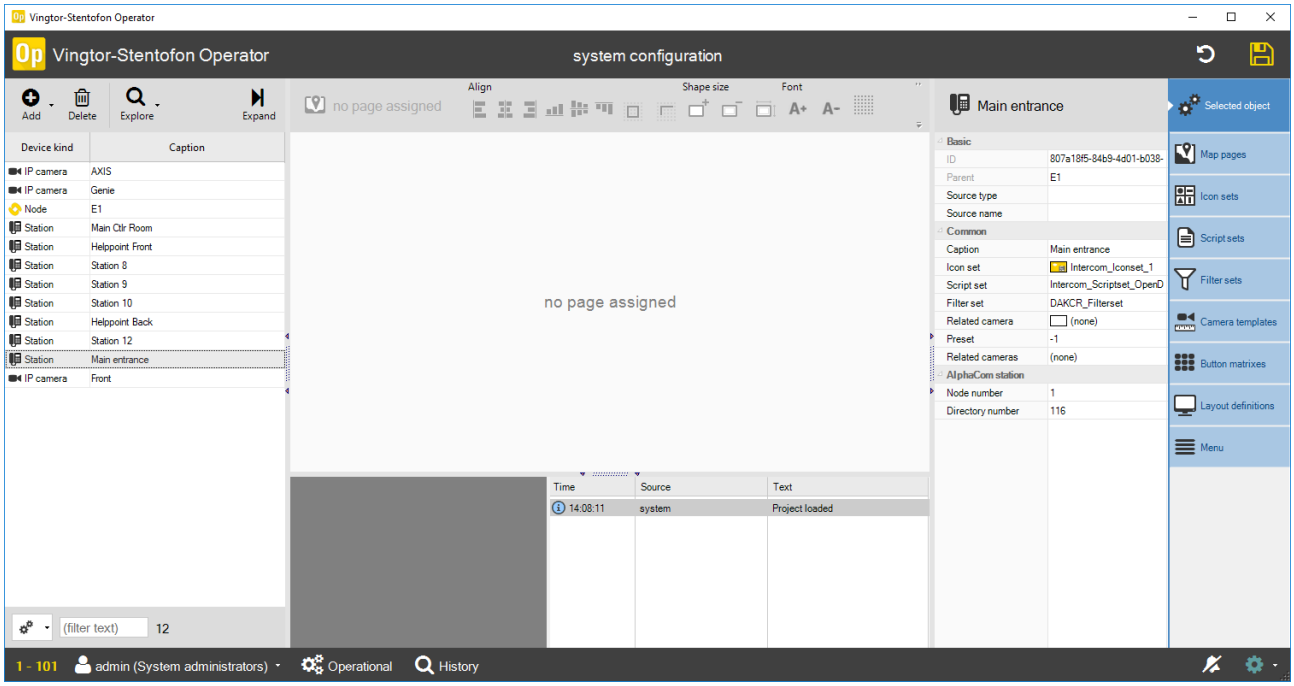
The menu option **Explore** has a submenu: **Auto register mode**. If this is selected, Vingtor-Stentofon Operator will attempt to add devices, which send status information, and were not in the device list yet, to the list. For instance, an RCI which has not been defined manually and which sends status information to Vingtor-Stentofon Operator will be added to the device list.

3.2.3 Deleting Devices

After deleting a device, it is recommended to **Save** the configuration. If this is not done and the same device is added again, Vingtor-Stentofon Operator will attempt to write the new device data to the SQL database, which will respond **Cannot insert duplicate key …** .

If this happens, delete the device again and save before inserting. It may be necessary to use the Reload button  to get the actual data as stored in the database showing in the device list before attempting to delete the device again.

3.2.4 Device Parameters



3.2.4.1 Basic

Under **Basic,** there are four parameters:

- **ID** – this parameter is given a GUID value automatically which cannot be edited

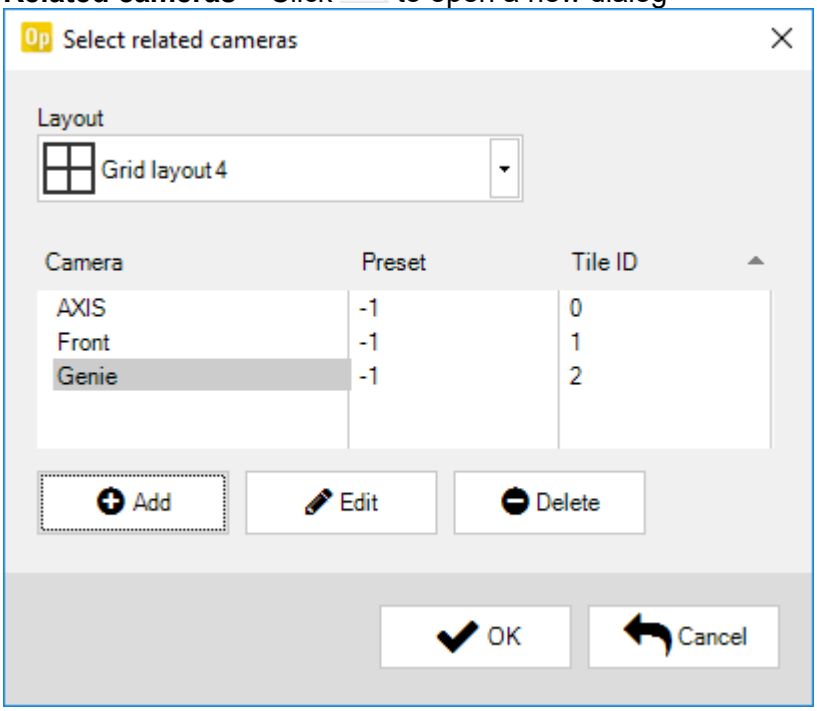
- **Parent** – this parameter is automatically added depending on other parameters that have been entered and this value cannot be edited directly
- **Source type** – added automatically for VS-Recorder connected cameras
- **Source name** – added automatically for VS-Recorder connected cameras

3.2.4.2 Common

Under **Common,** there are six parameters:

- **Caption** – the name of the device that appears in the device list
- **Icon set** – for icon sets, see section 3.5
- **Script set** – for script sets, see section 3.6
- **Filter set** – for filter sets, see section 3.7
- **Related camera** and **Related cameras** – two separate ways to associate one or more cameras to a device
    - The camera(s) will be shown when there is a call request or call with the station to which the cameras have been associated
    - When a device other than a camera is dragged into a camera view, the related camera(s) will be shown
    - Note: Although it is possible to relate cameras to another camera, this has no practical use

  The fields **Relate camera** and **Preset** are still used as in previous versions of Vingtor-Stentofon Operator for backward compatibility reasons
    - **Related camera** – Enter a single related camera; the field **Preset** is only used when the related camera is connected to a VS-Recorder. When the camera is selected, it will also be set to the PTZ preset entered here
    - **Related cameras** – Click ... to open a new dialog



Maximum 4 cameras can be associated to a device in this way

The cameras will be placed in a grid, on a single Tile View, see paragraph 3.9.10; Select the preferred grid view in the dropdown box **Grid layout**

Add, **Edit** or **Delete** the cameras; The columns **PTZ** and **Tile ID** can be edited directly in the dialog. **PTZ** is a Pan, Tilt, Zoom preset, which is only used when the related camera is connected to a VS-Recorder; When the camera is selected, it will also be set to the PTZ preset entered here

3.2.4.3 Symbol

These device parameters are only visible after the related icon has been placed in a map using drag-and-drop. Using these parameters, it is possible to accurately place the icon in the map and set parameters related to label font, size, color and placement. The names of the parameters make this section self-explanatory.

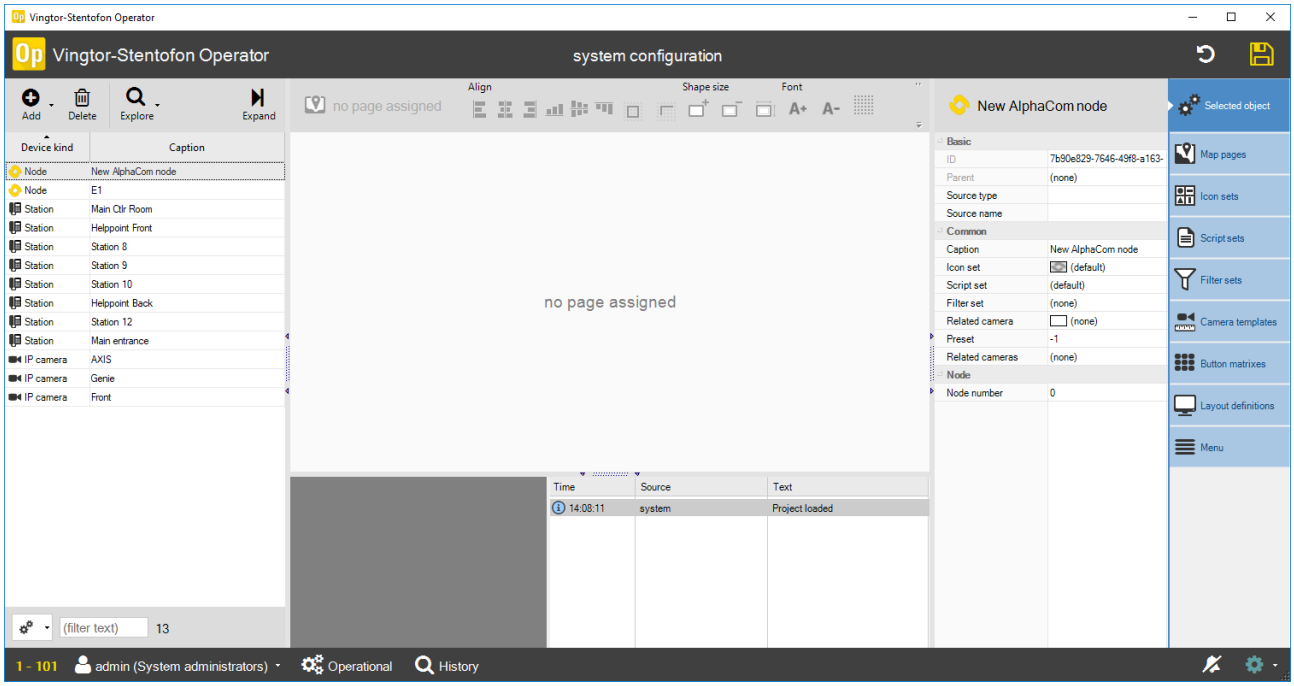
3.2.5 Node – AlphaCom node

An AlphaCom node can be added to manually or imported from the Vingtor-Stentofon Operator connected AlphaCom network.

3.2.5.1 Manual

- Click **Add**
- Select **Node**

A **New AlphaCom node** will be added to the list of devices.



- Edit the node name in the **Caption** field - see section 3.2.4.2.
- Edit the node number in the **Node number** field

3.2.5.2 Automatic

- Click **Explore**
- Select **Explore AlphaCom stations…**
- In the new dialog, click **Load nodes**

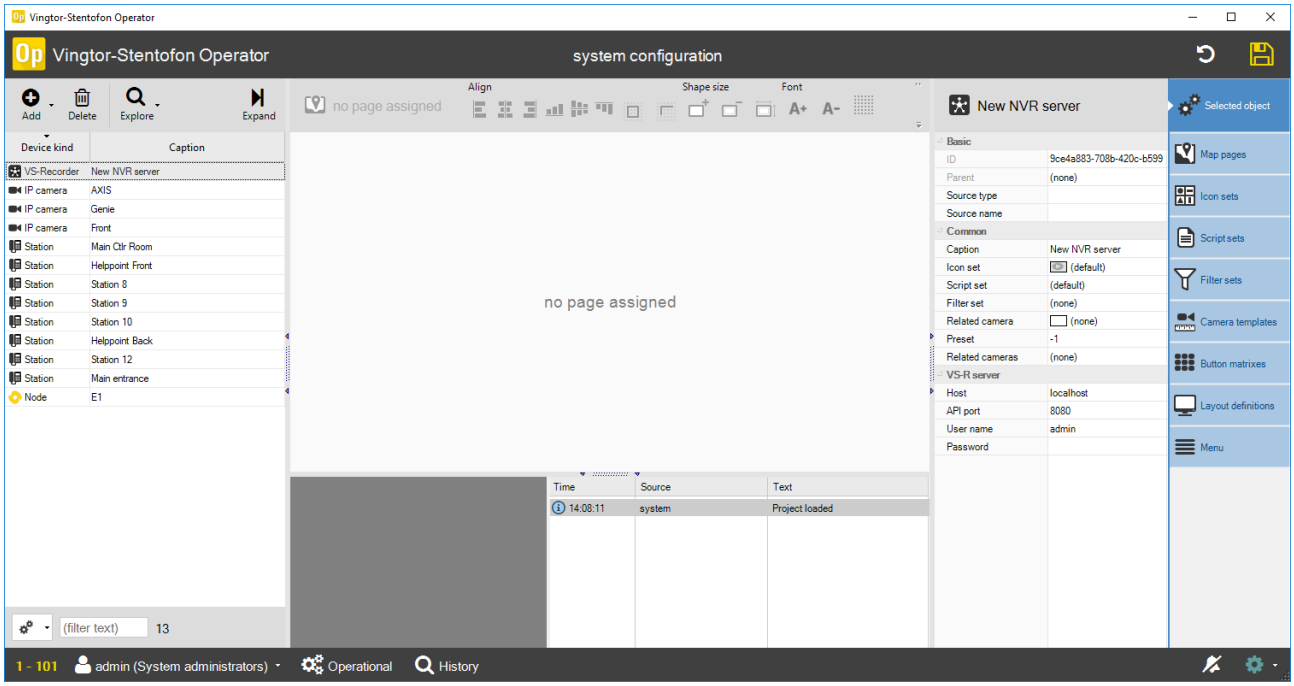
All nodes that are in the AlphaNet will now be imported into Vingtor-Stentofon Operator.

3.2.6 VS-Recorder 2.0

A VS-Recorder server can only be added manually.

- Click **Add**
- Select **VS-Recorder**

A **VS-Recorder** will be added to the list of devices.



- Edit the VS-Recorder name in the **Caption** field - see section 3.2.4.2.
- Edit the VS-Recorder specific details under **VS-R Server**
    - **Host** – Host name or IP address of the computer on which the VS-Recorder server is running

- **API port** – The API port number that Vingtor-Stentofon Operator must use to connect to the server. The port number is assigned when the VS-Recorder is installed using the VS-Recorder utility
- A valid VS-Recorder **User name** and **Password**

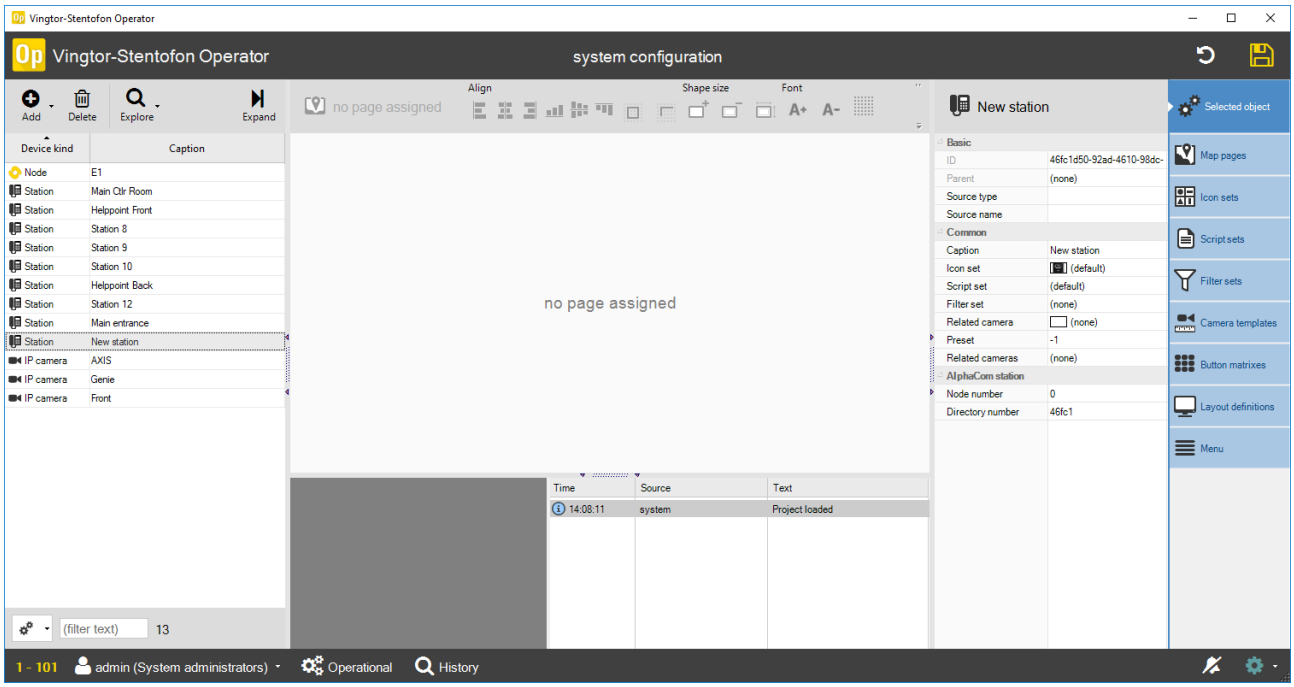
3.2.7 Intercom

An intercom or phone can be added both manually or imported from/to the Vingtor-Stentofon Operator connected AlphaCom network

3.2.7.1 Manual

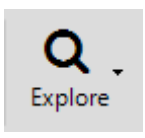
- Click **Add**
- Select **Station**

A **New station** will be added to the list of devices.

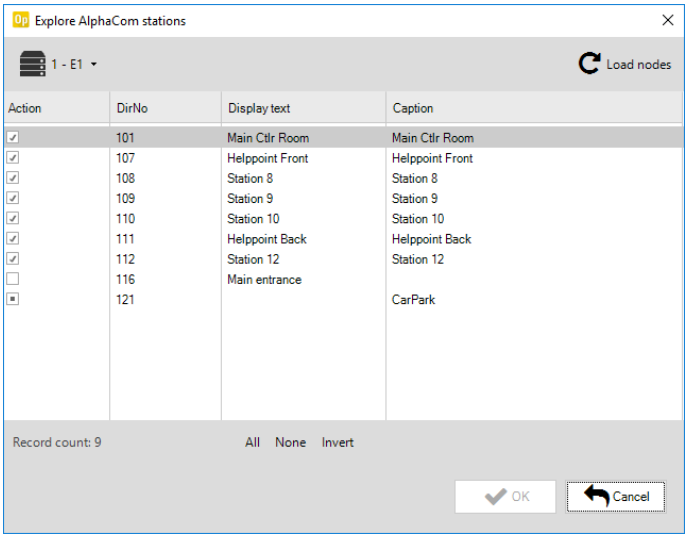


- Edit the station name in the **Caption** field - see section 3.2.4.2.
- Enter the **Node number** to which the station is connected under **AlphaCom station**
- Enter the **Directory number** of the station under **AlphaCom station**

3.2.7.2 Automatic



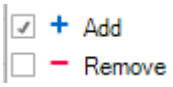
- Click **Explore**
- Select **Explore AlphaCom stations…**
- In the new dialog, click **Select a node…** and select a node from the drop-down list.



All the stations that are defined in the selected node will be listed.

- Stations that are already defined in Vingtor-Stentofon Operator are indicated by a **checkmark**.
- Stations that are defined in Vingtor-Stentofon Operator, but are not present in the selected node are indicated by a ▪ sign.

By clicking individual stations or **All**, **None**, **Invert** it is possible to select stations for either being added or deleted from Vingtor-Stentofon Operator. Whether a station will be added or deleted is clearly indicated.



3.2.8 Camera

Vingtor-Stentofon Operator recognizes two different types of camera connections:

- Cameras connected to a VS-Recorder 2.0
- Cameras that directly stream to the Vingtor-Stentofon Operator server

With some cameras, it may take too long to establish a connection or to retrieve video, in which case the video will not be displayed. It may be possible to resolve this by changing a value directly in the Vingtor-Stentofon Operator client configuration file **ConfigData.xml**; this file is by default located in **C:\Program Files (x86)\Vingtor Stentofon\Vingtor-Stentofon Operator\Vingtor-Stentofon Operator Client**.

Locate the line

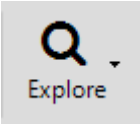
<string>NetworkCachingRtsp=1100</string>

and increase the value 1100 to an appropriate value. The client must be restarted for the change to take effect.

As there are many cameras from many manufacturers on the market, it may occasionally happen that a camera is not compatible with Vingtor-Stentofon Operator. Zenitel will try to solve such issues in Vingtor-Stentofon Operator.

3.2.8.1 VS-Recorder connected camera

Cameras connected to a VS-Recorder 2 cannot be added manually but must be imported. These cameras are listed as **VS-R camera**.



- Click **Explore**
- Select Explore **VS-R cameras…**
- In the new dialog, click **Select VS-R server…** and select a server from the drop-down list. Servers that cannot be contacted are greyed-out - if the server should be available click **Reset VS-R connections** to re-establish the connection with the VS-Recorder

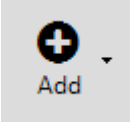
All to the VS-Recorder connected devices will be listed. These may not all be cameras - make certain to only select cameras to import into Vingtor-Stentofon Operator, as Vingtor-Stentofon Operator is not able to handle streams from audio devices.

A V-sign indicates cameras that are already defined in Vingtor-Stentofon Operator.

By clicking individual cameras, **All**, **None** or **Invert** it is possible to select cameras for either being added or deleted from Vingtor-Stentofon Operator. Whether a camera will be added or deleted is clearly indicated.

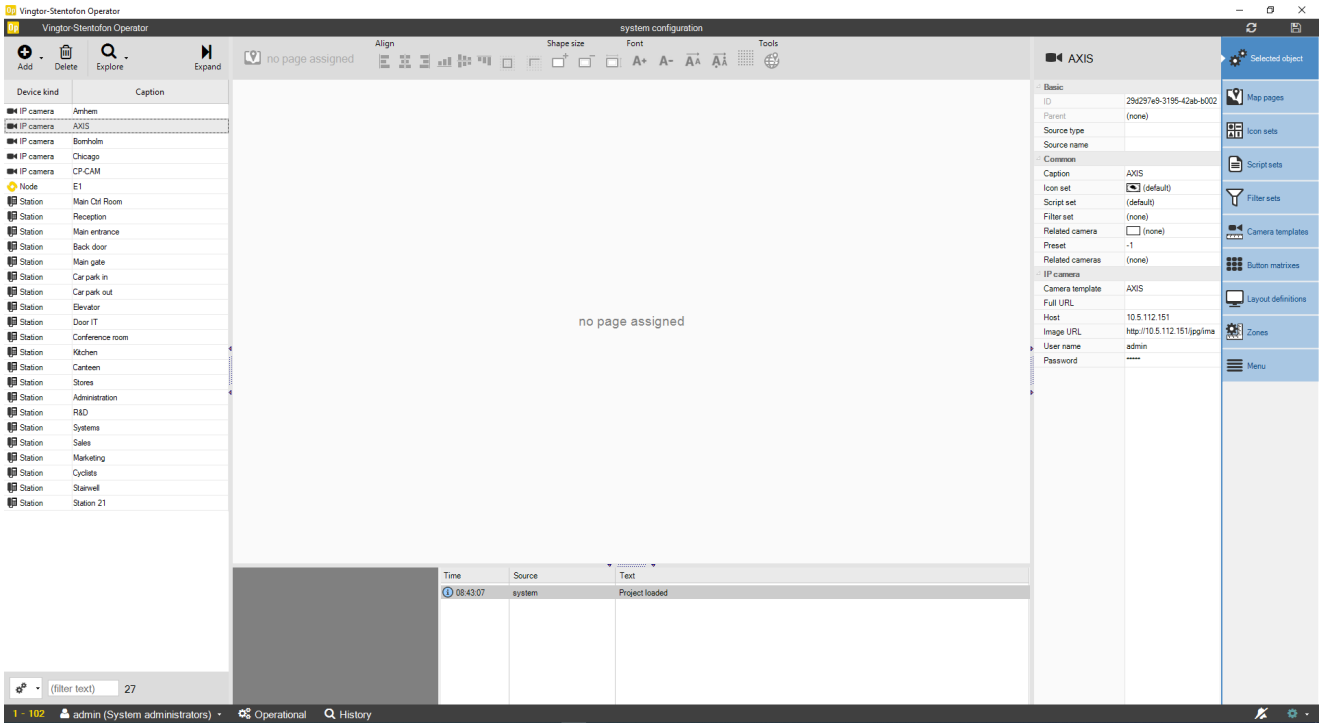
3.2.8.2 Cameras directly streaming to Vingtor-Stentofon Operator

Directly to the Vingtor-Stentofon Operator streaming cameras are listed as ‘IP camera’; these can only be added manually.



- Click **Add**
- Select **IP camera**

A **New IP camera** will be added to the list of devices.



- Edit the camera name in the **Caption** field - see section 3.2.4.2.

To access the IP camera a number of fields need to be filled in under the section **IP camera**.

- The IP camera can be addressed by entering its full URL - for a Vingtor-Stentofon TCIV station: this could be http://IP_ADDRESS:8090/mjpg/video.mjpg
- Alternatively it is possible to define a **Camera template** and only enter the host name or IP address in the field **Host** - see section 3.3
    - Enter User name and Password for the camera; these fields are only used as parameters for the camera template -- see section 3.3

If both the fields **Camera template** and **Full URL** are empty, the default template will be used. In that case, only the host name or IP address is required.

If there is an entry in the field **Full URL,** it will be used even if the fields **Camera template** and **Host** contain valid information.

As from version 1.6.3.0 it is possible to take a snapshot from a video view. By default the snapshot is taken from the video viewer; If the camera is able to take a photo, it is possible ot use that as a snapshot instead, define the full URL in the field **Image URL**.

3.2.9 RCI

RCIs can only be added manually.

- Click **Add**
- Select **RCI**

An RCI is an AlphaCom Remote Control Input. It must be given a Node number and an RCI number.

The RCI state is set via the AlphaCom protocol message RCI_CHANGED (see AlphaWiki).

This protocol message is not sent automatically when the state of a physical RCI changes, but the message must be configured in the AlphaCom event handler:

@nnpp $RCIC Wx Uy

- nn is the node number to which the Vingtor-Stentofon Operator server is connected – 01-FE for node 1-254
- pp is a value for the API port number which is used – C7 for 61112, C4 for 61113; As from AMC-IP 12.3.3.1: D1 is the broadcast address for the 2 API-ports (information will be sent out on both)
- x is the RCI number – the protocol message does not allow the AlphaCom node number to which the RCI is connected to be used. RCIs must therefore have a unique RCI number in Vingtor-Stentofon Operator - the node number as configured in Vingtor-Stentofon Operator is operationally not relevant.
- y is the state – 0 for inactive, 1 for active

3.2.10 RCO

RCOs can only be added manually.

- Click **Add**
- Select **RCO**

An RCO is an AlphaCom Remote Control Output. It must be given a Node number and an RCO number.

The RCO state is set via the AlphaCom protocol message RCO_CHANGED (see AlphaWiki). This protocol message is sent automatically when the state of a physical RCO changes. The message can also be configured in the AlphaCom event handler in case the RCO is used for other purposes then an actual RCO in AlphaCom terminology:

@nnpp $RCOC Wx Uy

- nn is the node number to which the Vingtor-Stentofon Operator server is connected – 01-FE for node 1-254
- pp is a value for the API port number which is used – C7 for 61112, C4 for 61113; As from AMC-IP 12.3.3.1: D1 is the broadcast address for the 2 API-ports (information will be sent out on both)
- x is the RCO number – the protocol message does not allow the AlphaCom node number to which the RCO is connected to be used. RCOs must therefore have a unique RCO number in

Vingtor-Stentofon Operator - the node number as configured in Vingtor-Stentofon Operator is operationally not relevant.

- y is the state – 0 for inactive, 1 for active

3.2.11 General Purpose Device - GPD

GPDs can only be added manually.

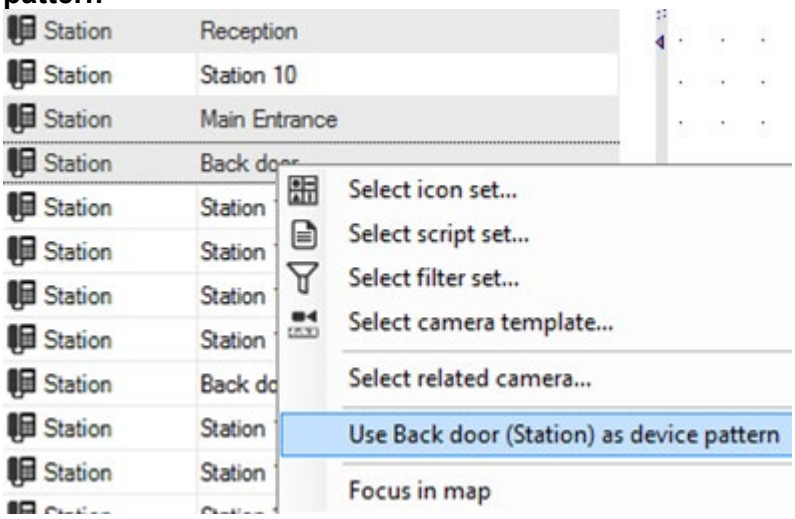
- Click **Add**
- Select **GPD**

More information about GPDs can be found in appendix D.

3.2.12 Assigning Icon, Script and Filter sets

It is possible to copy the settings of Icon, Script and Filter sets from one device to many other devices in one action.

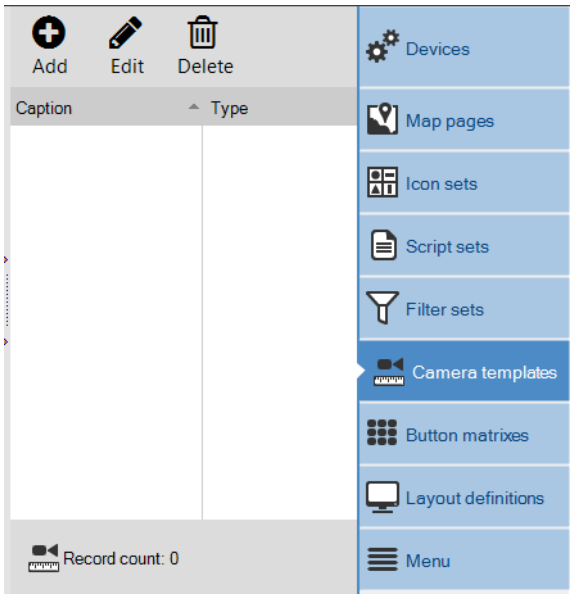
- Define the Icon, Scripts and Filter set for 1 device
- Multiselect the just defined device and all devices that should have the same set-properties (click on the first device, press the CTRL-key on the keyboard and click the other devices)
- Right click on the device which has the required set-properties and click on **Use as device pattern**



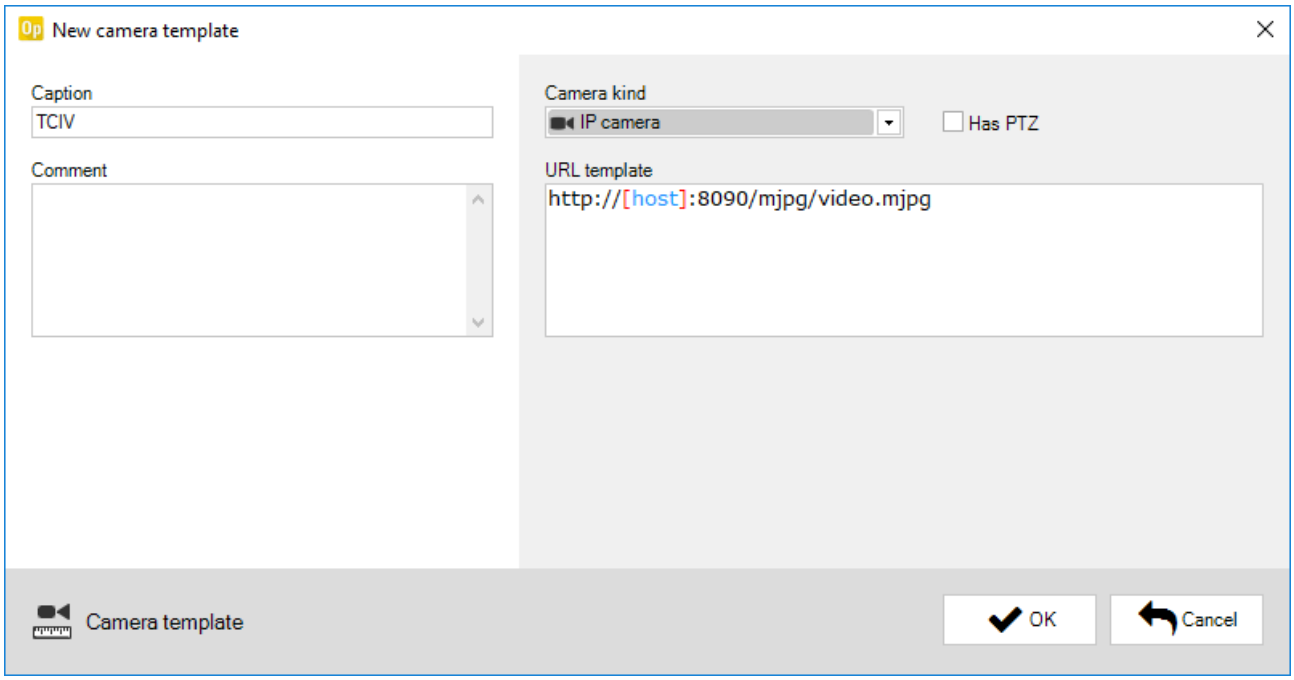
3.3 Camera Templates

If many cameras of a similar type need to be added it is advisable to define a **Camera template**. Camera templates can be added, edited or deleted.

- Click the button **Camera templates**



- Click **Add**



When adding or editing a camera template, enter the following information in the **Camera template** dialog:

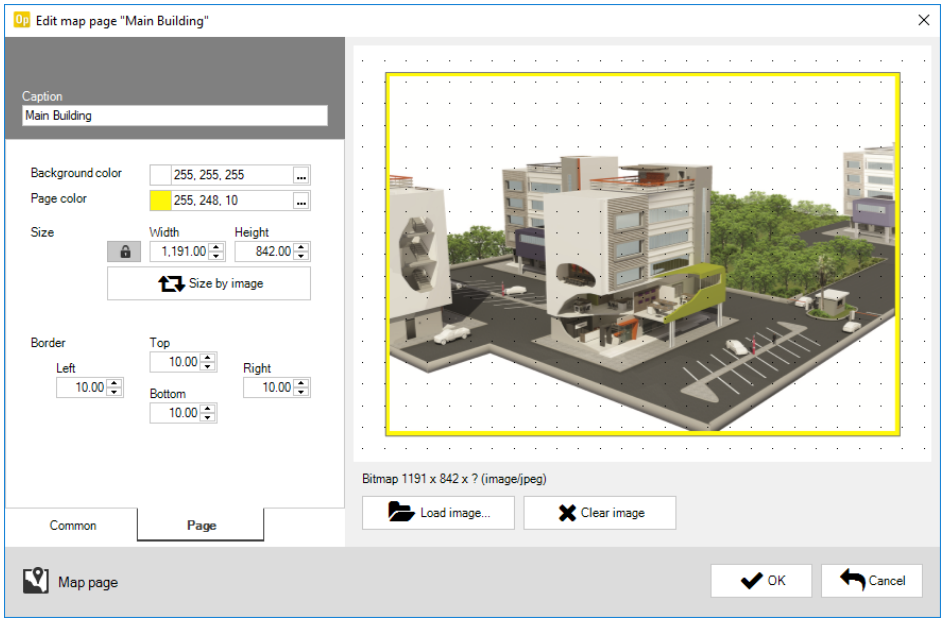
- A meaningful name in the field **Caption** - this name will be available in the dropdown box when selecting a template to be used for an IP camera
- The **URL template** - using the default port number of a TCIV camera, the template for a TCIV station would be
    - If no authentication is used: http://[host]:8090/mjpg/video.mjpg
    - If basic authentication is used for the camera: http://[username]:[password]@[host]:8090/mjpg/video.mjpg

The default template is http://[host]/mjpg/video.mjpg

3.4 Map Pages

3.4.1 Defining Maps

- Click the button **Map pages**
- Click **Add**



It is now possible to add, edit or delete maps. Maps can be imported in three bitmap (png, bmp, jpg) and two vector (emf, wmf) formats.

A **Page color** can be defined, which will be shown inside the map area. Note that this color will only be visible if the map itself is transparent.

It is possible to define a border around the map area - the border will be filled with the defined **Page color**.

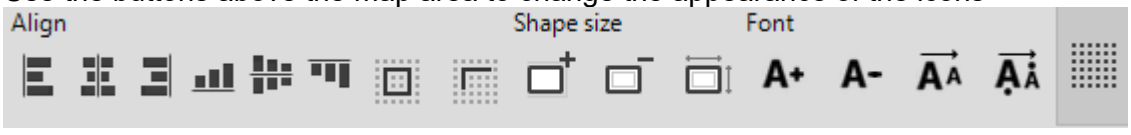
Alternatively, it is also possible to

- Click the button **Map pages**
- Drag and drop a map graphics file directly into the map-list

3.4.2 Adding devices to maps

Once a map has been defined, it can be dragged to the middle of the screen, after which devices can be dragged onto it. When a device is selected in the map, it is possible to change the device properties:

- Click on the button **Devices** and edit the device properties
- Use the handles of the device to change the size of the icon on the map
- Use the buttons above the map area to change the appearance of the icons



  - Multi-select a number of device icons: Align all icons as required to the icon which was selected as the first

- Multi-select a number of device icons: Make all icons the same size as the icon which was selected as the first by using
- Multi-select a number of device icons: Copy all label text properties (font, size, color, …) of the icon which was selected as the first to all other icons
- Multi-select a number of device icons: Copy all label text properties (font, size, color, …) and position of the icon which was selected as the first to all other icons

It is possible to change the map area size by clicking on the area size handles to the right, left and the bottom of the map area.

3.4.3 Adding and deleting linked map areas

It is possible to link maps, by placing map-link objects in a map. By clicking on the map-link object, the next map in the hierarchy is shown. The map-link object can change appearance when there is activity in a map lower in the hierarchy, see paragraph 3.12.

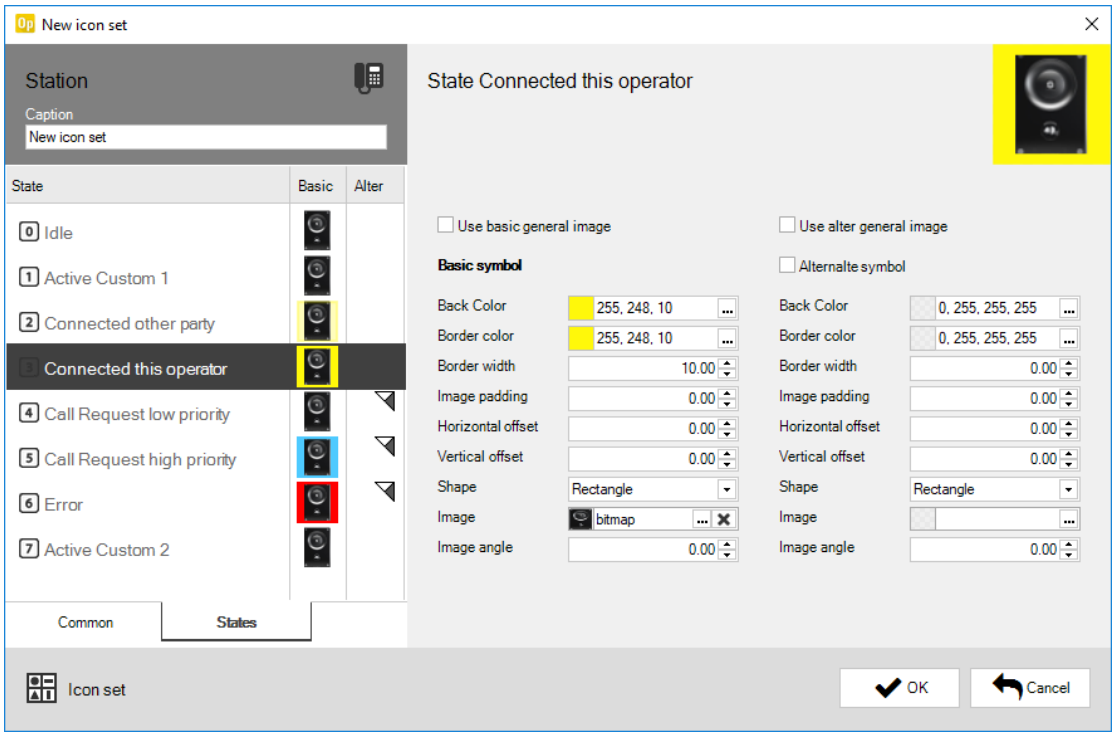
- Drag and drop the -sign in the **Tools** section above the map area onto the map in the map area to define a map-link object

- Once placed, use the handles to size the map-link object and place it accurately
- Select **Selected object** in the menu list and change the text-properties as required
- Click on the field **Referenced map page** to show a small button with 3 dots; click this button to select a map

To delete a map-link object, select it in the map and use the Delete button on the keyboard to delete the object.

3.5 Icon Sets

- Click the button **Icon sets**
- Click **Add**
- Select the device such as **Station**, **IP camera**, etc.



3.5.1 General

An icon set defines the graphics for the different states an icon can be in. An icon set can be assigned to a device. In this way, it is easy to make devices, which have the same functionality in the system appear the same. Icon sets can be added, edited and deleted. It is possible to have multiple icon sets for the same device type. This gives for instance the possibility to define different icon sets for intercoms in lifts and at gates.

3.5.2 Graphics

For each device, there are a number of possible states depending on the device type.

Each state has a **Basic symbol** and optionally an **Alternate symbol**. If the Alternate symbol is used, the icon will appear alternating between the two graphics. It is possible to define the use of the Alternate symbol, but not assign graphics to it. In that case, the icon will appear to flash.

When **Use basic general image** and/or **Use alter general image** is selected, the image as defined for the **Idle** state will be used for all other states where no specific image has been defined. It is always possible to override the image for a specific state by defining an image for it.

3.5.3 Intercom icon set

An Intercom icon can have two special states:

- Active Custom 1
- Active Custom 2

These states can be set via the **On Custom Parameter** command from the AlphaCom event handler.

- Parameter 4: 1 for ActiveCustom1; 7 for ActiveCustom2; other values are ignored
- Parameter 5: 1 for ON; 0 for OFF; other values are ignored

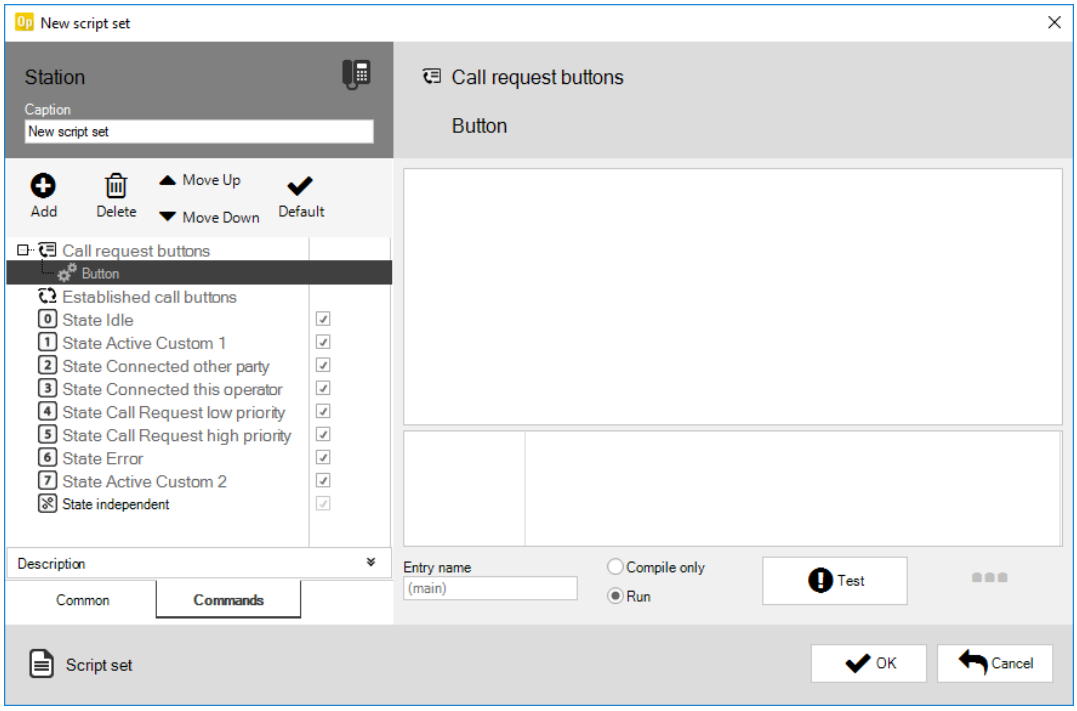
Example action defined in AlphaCom event handler:

- @01C7 M7F01 L(1)101 U0 W0 W0 W0 W7 W1

Note that C7 refers to API port 61112, C4 to 61113; As from AMC-IP 12.3.3.1 it is possible to use D1 (API port broadcast address) to send the information out on both ports simultaneously:

3.6 Script Sets

- Click the button **Script sets**
- Click **Add**
- Select the device such as **Station**, **IP camera**, etc.



3.6.1 General

A script set is a collection of scripts related to the states of the related devices. Script sets make it easy to assign scripts to devices that must initiate the same action when double-clicked on a map or in the device list. Once a script set has been defined, it can be assigned to the appropriate devices.

Script sets can be added, edited and deleted. To add a script set, select the type of device for which the script set will be used.

It is possible to attach multiple scripts to the same device state. The available scripts will be shown in a drop-down list when the relevant device icon is right-clicked.

3.6.2 Device states

It is possible to assign scripts (actions) to all device states. A script will only be accessible if the device is in the associated state. It is also possible to assign **State independent** scripts, as these scripts are always available.

For Intercom station devices it is also possible to assign scripts to buttons that are available when the station is in **Call Request** or **Established Call** mode. These buttons are shown in the call request list, the active call request window and the established call window.

For each state, it is possible to define one script, which will be executed when the icon is double-clicked. Select the script and click **Default**. The default script will be indicated by .

It is possible to preview what the drop-down menu will look like. Select the states that need to be previewed and right-click any state.

3.6.3 Entering scripts

Scripts can be entered in the edit area of the dialog. When finished it is possible to test the script. **Compile only** will always work. It may also be possible to **Run** the script, but that depends whether parameters that are used in the script can be resolved at the time of the test. It is only possible to save a tested script. For more information on scripts, see appendix E.

3.6.4 Default scripts

There is one default script set for intercom devices. A newly defined or discovered intercom device will use this default script set until another one is assigned to it.

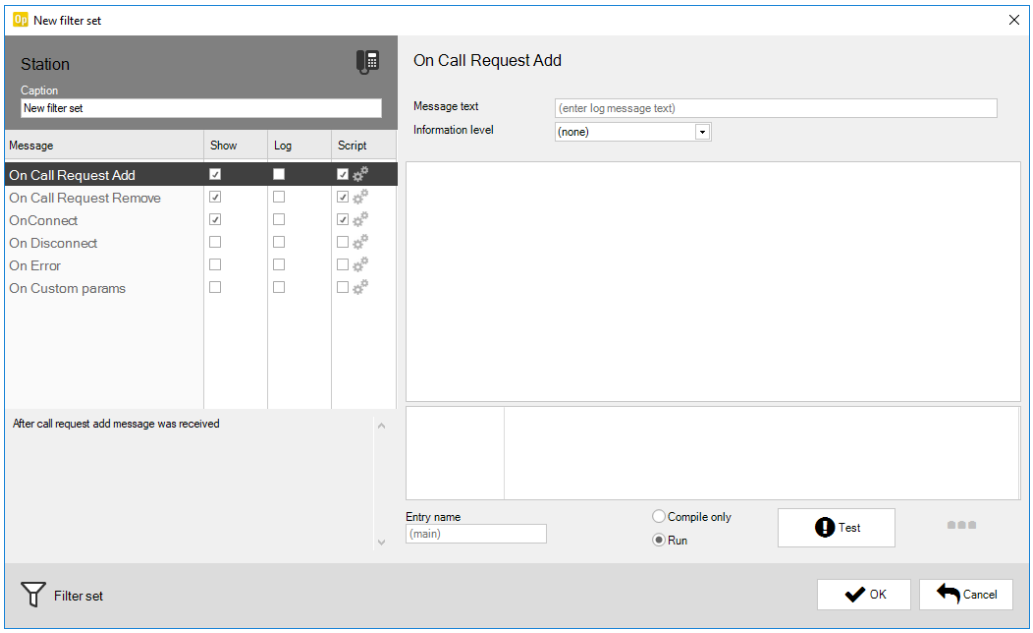
The default script set contains the following scripts:

- App.STE (“@U%NNOP $CALL L%DIROP L%DIRA”);
  - Idle – status 0; default for double click
  - Call request low priority – status 4; default for double click
  - Call request high priority – status 5; default for double click
  - Call request button 1
- App.STE (“@U%NNOP $C L%DIROP”);
  - Connected with this operator (%DIROP) – status 3; default for double click
  - Established call button 2

These scripts are hardcoded and cannot be altered.

3.7 Filter Sets

- Click the button **Filter sets**
- Click **Add**
- Select the device such as **Station**, **IP camera**, etc.



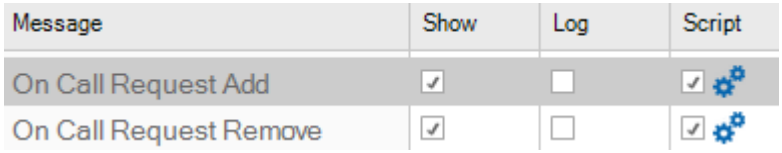
3.7.1 General

Filter sets are used to filter incoming information and perform actions based on that information, depending on what has been enabled for the message:

- Add an entry into the on screen log
- Add an entry to the log in the SQL server
- Trigger a script

Filter sets can be added, edited and deleted. Once a Filter set has been defined, it can be assigned to a device of the appropriate type.

Different device types can filter on different types of incoming information, see appendix B.



Messages related to intercom devices only trigger filters related to the directory number of the A-subscriber.

3.7.2 Logging

Information can be logged into the onscreen log (**Show**) and to disk (**Log**). The information to be logged is defined in the field **Message text**. It is possible to add parameters into this text field. For available parameters, see appendix B.

In the case of **OnConnect** a Message text could be:

- Call %DIROP %DIRB
    - %DIROP will be replaced with the directory number of this Vingtor-Stentofon Operator operator

- %DIRB will be replaced with the directory number of the remote station

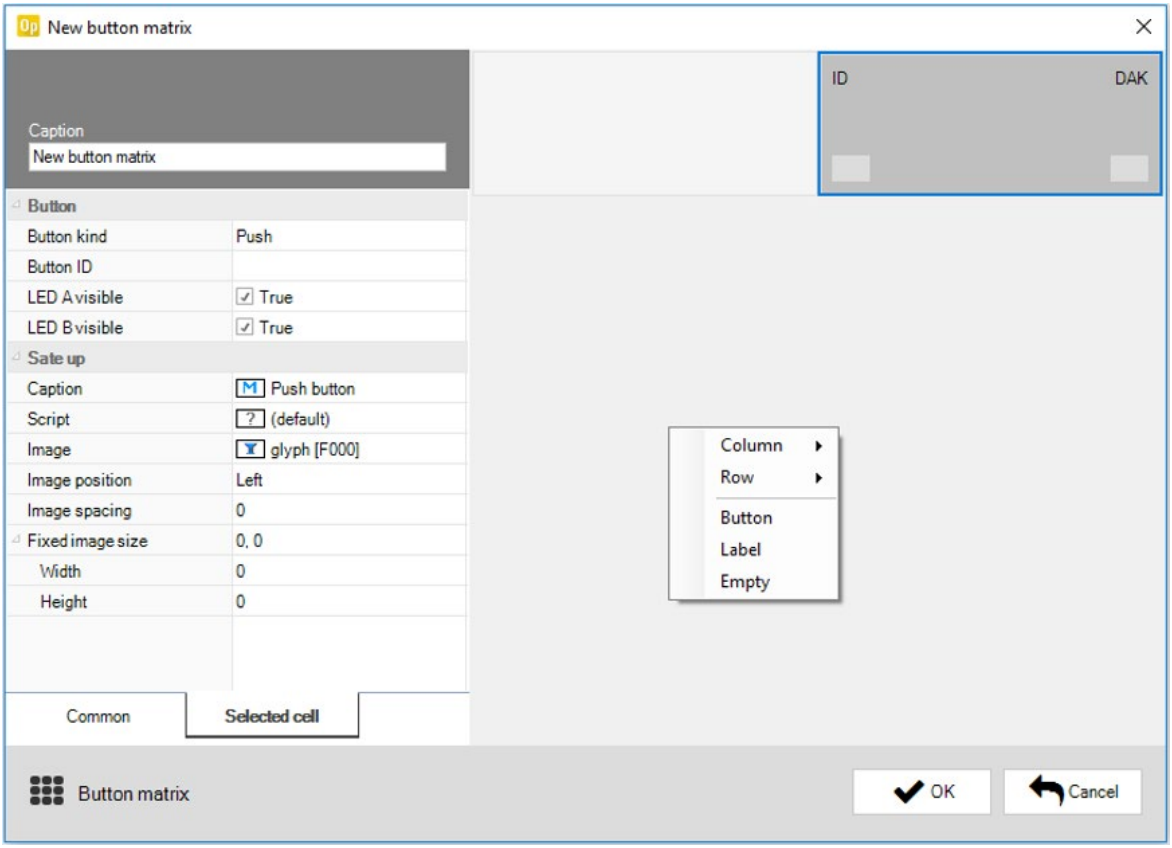
The **Information level** defines the assigned severity to the log entry - possible choices are: **None**, **Info**, **Warning**, **Alarm** and **Error**.

3.7.3 Scripts

Scripts can be entered in the edit area of the dialog. When done it is possible to test the script. **Compile only** will always work. It may also be possible to **Run** the script, but that depends whether parameters that are used in the script can be resolved at the time of the test. It is only possible to save a tested script. For more information on scripts, see appendix E.

3.8 Button Matrixes

- Click the button **Button matrices**
- Click **Add**



3.8.1 General

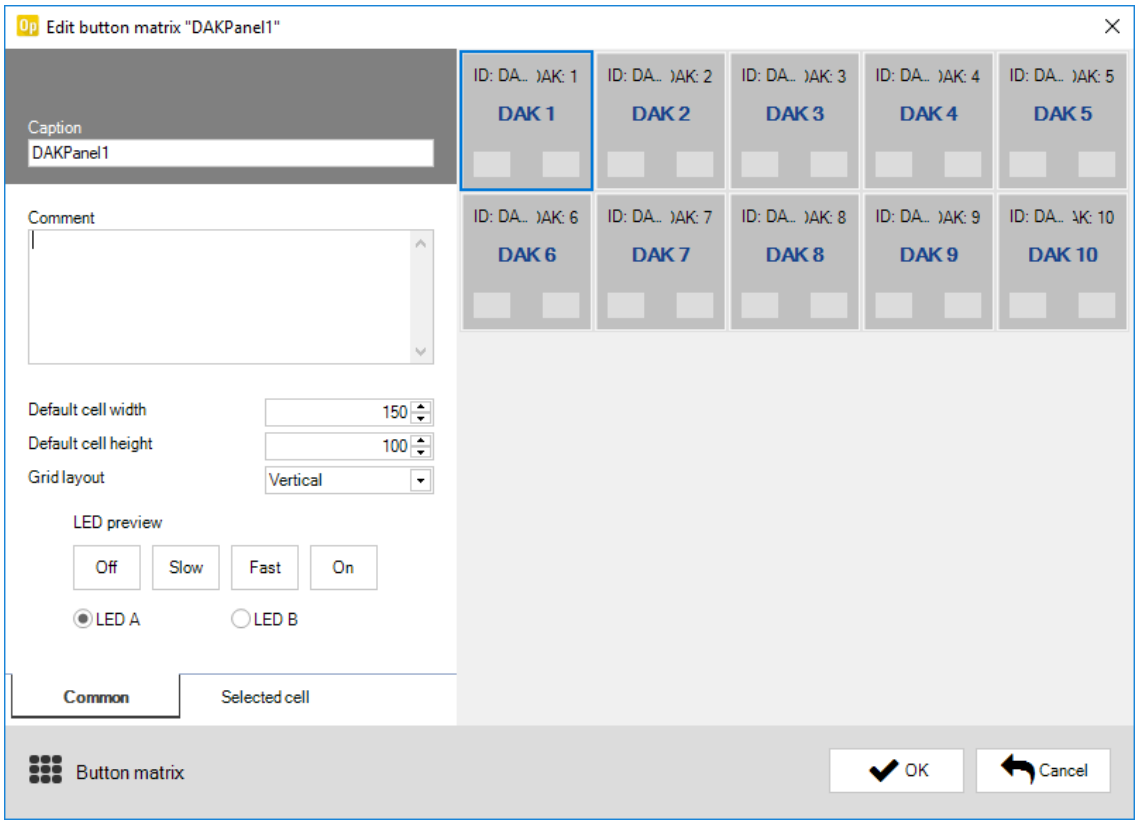
It is possible to add pages of button. Buttons can be defined as:

- Push button
- Toggle button
- DAK button

Right-click anywhere in the button area to add, insert (add above or to the left of the selected row or column) or delete a row or column. The area will be divided in a matrix with the defined number of cells. Click in a cell to activate it, followed by a right-click to define the cell type:

- Button
- Label
- Empty

3.8.2 Common Tab



Enter values for the cell width and height of a cell in pixels.

Grid layout allows the selection of the following behavior:

- **None** – the cells are drawn to the exact width and height as defined
- **Horizontal** – the button matrix will fill the horizontal space as provided when the screen layout is defined. Cells will be squeezed or stretched as appropriate.
- **Vertical** – the button matrix will fill the vertical space as provided when the screen layout is defined. Cells will be squeezed or stretched as appropriate.
- **Stretch** – The combination of Horizontal and Vertical.

LED preview shows the default behavior of the LEDs on the buttons. The default color for LED A is green and the default color for LED B is red.

3.8.3 Cell text

Text can be placed in label cells and buttons. The text can be formatted with simplified markup tags. The following tags are supported:

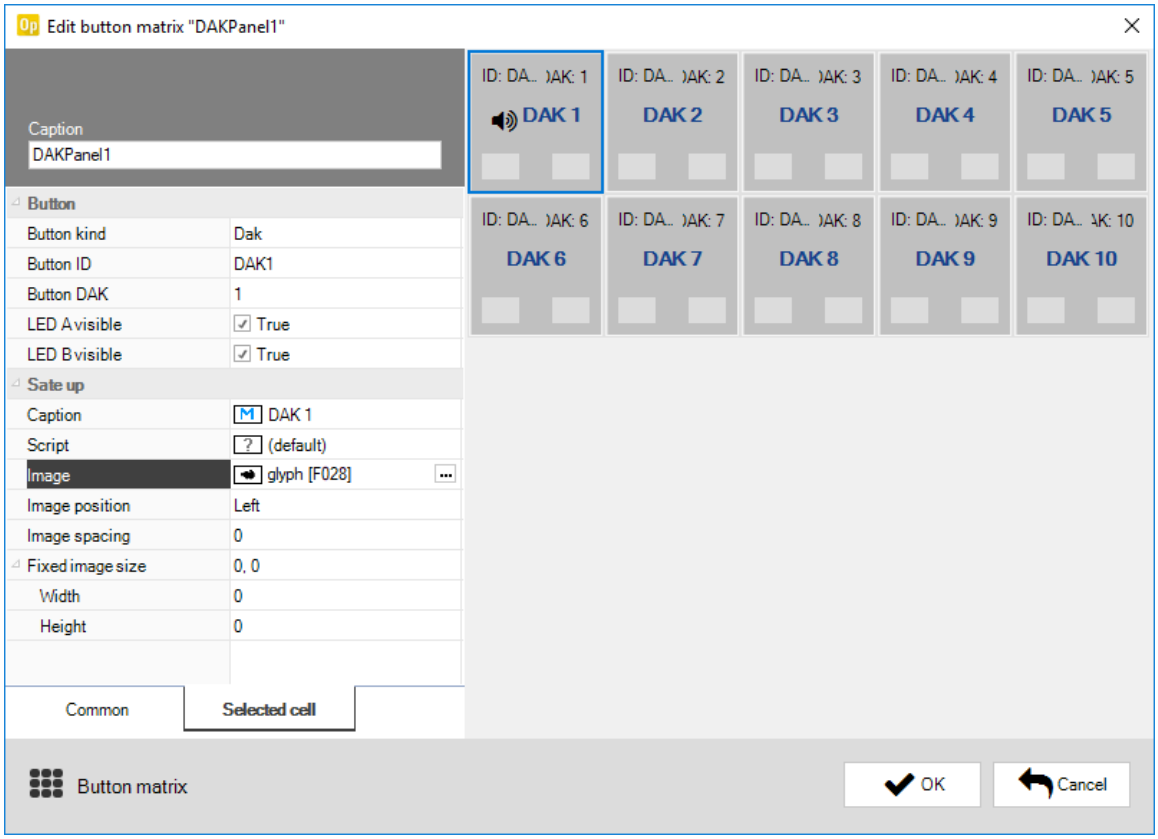
- <b> - bold
- <i> - italic
- <u> - underline

- <p> - <p align="left|right|center">
- <font> - face, size (1-7, default 3), color
-
 - break (line break)

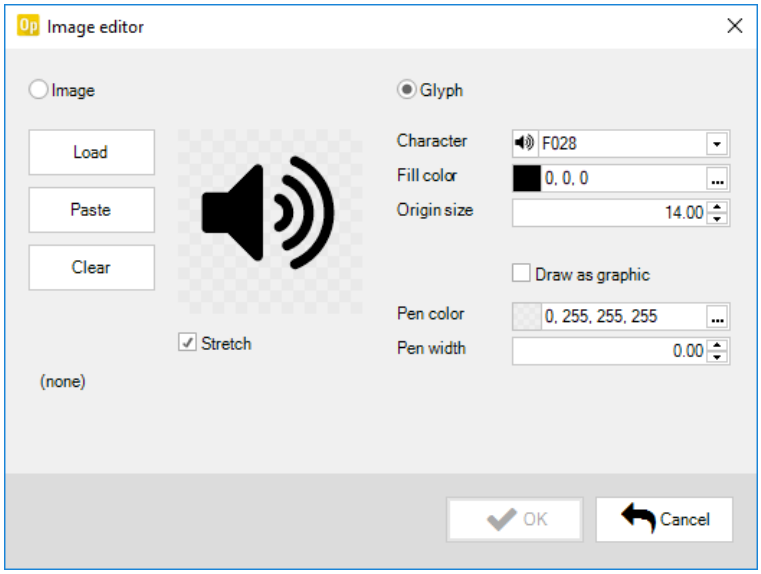
When markup is not used, the text is placed in the middle of the cell.

Toggle buttons can have different text for the Up and Down state.

3.8.4 Button picture



It is possible to place a picture in a button. Clicking ... in the **Image** parameter brings up an image editor.



In the image editor, it is possible to either select **Image** or **Glyph**. The drawing area will show what the image looks like. If the picture is very small, check the **Stretch** box to be able to see it. Note that **Stretch** has no influence on the appearance of the picture in operational use.

The image position can be at the top, left, right or bottom of the button.

Image spacing sets the space between the image and the text.

With fixed image size, it is possible to define the size of the image on the button (this setting has no influence on the size of a Glyph):

- 0;0 (Width; Height) uses the size of the image itself
- Setting values for Width and Height uses those sizes

Delete the image completely from the button by selecting the image defining text and press the Delete key on the keyboard.

3.8.4.1 Image editor – Image

- **Load** a picture from a file
- **Paste** a picture from the clipboard
- **Clear** the picture completely

By default, the size of the picture will be used as the image size on the button.

3.8.4.2 Image editor – Glyph

A Glyph is a standardized picture. Select a picture and define the appearance using the fields **Character**, **Fill color**, and **Origin size**.

3.8.5 Button ID

Buttons can be identified by an ID. The ID can be assigned as required for the project. A useful way of assigning an ID to a DAK button, which calls the intercom with directory number 111, could be:

- Button ID: DAK111

The Button ID can be used in scripts to identify the button and then, for instance, to set the state of one of the button LEDs. For an example, see Appendix E.4.5.

The button ID is only visible on the button in configuration mode as it is hidden in operational mode.

3.8.6 Button script

See appendix E for further information about scripts.

3.8.7 Push and toggle buttons

A push button always performs the same action when clicked.

A toggle button has an Up and Down state. These states alternate:

- With each click of the button – Direct Press = false
- On mouse down/up – Direct Press = true (can be used as PressToTalk button)

The appearance of the toggle button and the scripts that are called depend on the state of the button.

3.8.8 DAK buttons

Define the button as a DAK-button with the appropriate DAK-code (1-100) in the field ‘Button DAK’. A DAK key sends a DAK-code to the AlphaCom - note that the DAK-key must be programmed in the AlphaCom for the button to have an effect.

3.8.9 Copy/paste

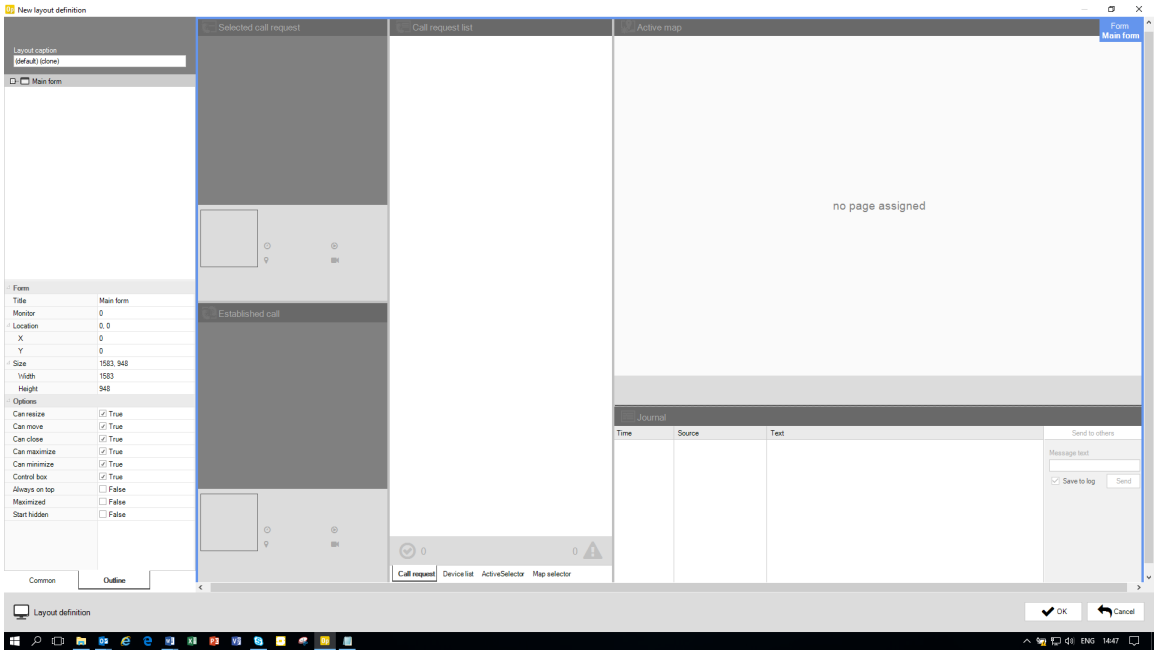
It is possible to copy a button or a label and paste it into another cell.

- To copy: select the cell and use the standard Windows Copy short cut – **CTRL-C**
- To paste: select the cell and use the standard Windows Paste short cut – **CTRL-V**

All properties are copied, including button ID. It is therefore necessary to make relevant changes after copying.
It is also possible to copy/paste between cells in different Button Matrixes.

3.9 Layout Definitions

- Click the button **Layout definitions**
- Click **Add**
- Select either **Empty** or **Default**



The example above is a **Default** form layout.

3.9.1 General

It is possible to define multiple layouts and select one during operation. Layouts can be defined to cover multiple screens.

Layouts can be added, edited and deleted.

When adding a new layout, it is possible to start with an empty layout, or to select an existing or the default layout as a basis. The rest of this chapter assumes that a new empty layout has been selected.

In the **Common**-TAB it is possible to define a script and enable/disable that script from running when a user selects that layout.

3.9.2 Layout element properties

All elements have some specific properties, which can be set. Especially those properties, which are related to size and resizing, must be considered with care.

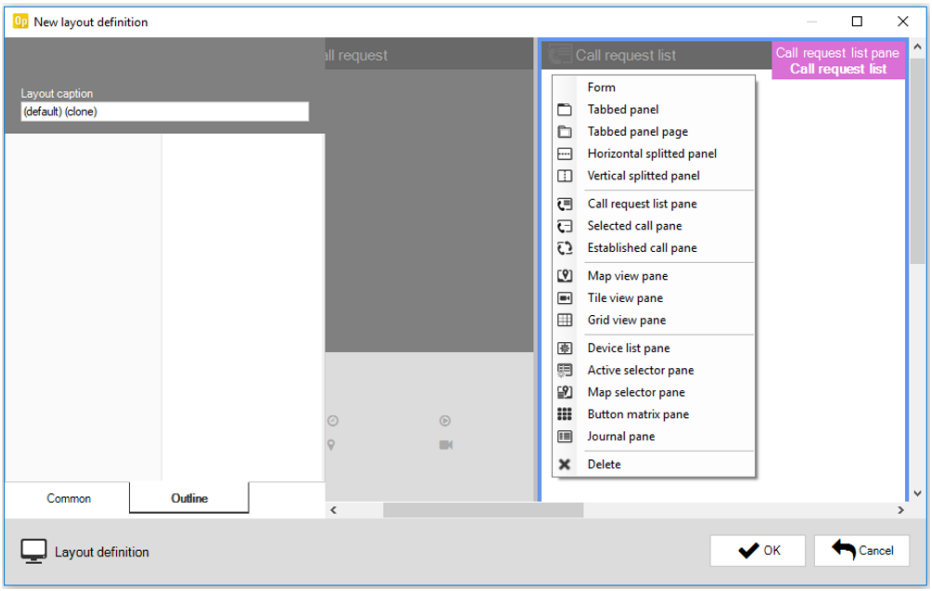
During operational mode, the form will display all elements, which have been placed on the form, no matter how big or small the form and the screen it is on are. If an operator is allowed to resize the form or elements within the form, the overall result may not be easy to work with anymore. It is advisable to fix the size of forms and elements within that form related to the screen resolution on which they will be displayed and rather define some different layouts from which an operator can choose according to his or her personal preference.

Depending on the license, it is possible to define multiple forms on a single screen. Note that this may result in forms, which overlay each other. If this feature is used it should be considered to only place selectors (device list, map selector) in a second form and to take care that this form does not cover the complete screen, such that all critical information is always visible on the primary form. In operational mode, the primary form has a status bar at the bottom indicating that there are additional forms available; the additional forms do not have such a status bar.

3.9.3 Forms

A **Form** is the basic container into which other elements can be placed. It is possible to define multiple forms in a layout. A Form can be fixed to a specific monitor by setting the Monitor property of the Form. If the monitor is not present in the system, that Form will be shown on another monitor. If the **Monitor** property is set to **0**, the main screen as defined in Windows will be used.

The first item to add to an empty layout is a **Form**. Right-click in the empty layout area and start with a Form. Size the form to cover as much of the screen as required.



In the **properties** section for the form it is possible to set:

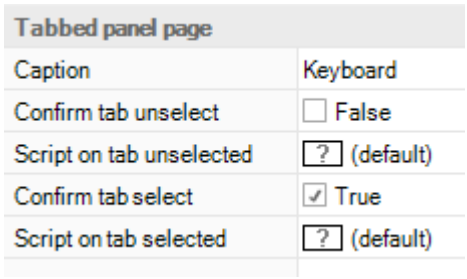
- The monitor on which the form should be displayed – this requires an Enhanced license
- The location and size of the form on the screen
- Different settings to control the ability to Maximize, Minimize, Move, Resize and/or Close the form
- How the form should appear: **Maximized**, **Start hidden**, **Always on top**, **Full screen**
  In Full screen mode, all frames are removed, and the VS-Operator screen extends to the bottom of the monitor, the Windows status bar will not be visible
- **Use pane active flags**: When this flag is enabled it is possible to automatically open and close the form, depending on the active status of some of its functional controls (**Map view**, **Tile view** (only camera status), **Call request list**, **Established call**, **Selected call request**, and an additional flag for the form which can be controlled from the script function **ActivateForm**);
  Note (1): In case of **Map view**, the Map view is only active if it displays a map with active icons; it is therefore advisable, in case of multiple maps, to define a default map for the map view, where this default map has active linking areas to other maps; if there is only a single map, also make it the default map
  Note (2): In case a layout contains multiple forms, the form which is made active (shown) as the last form has the focus, meaning that the controls in it can be clicked on for immediate action; To be able to use controls in other forms, such a form must first be given the focus, i.e. be clicked in

3.9.4 Panes

Right-click in the form area to add elements in which later functions controls can be placed.

It is possible to define the screen layout as required by adding horizontal and vertical split panels. Split panels and pages can themselves be split again. Using tabbed panels in which tab pages are placed are a good way of sharing the same screen space between elements that are used less frequently. Tab pages can be moved withing the tabbed panel after they have been defined by drag/drop. The size of the TAB-handle can be changed in the field **Tab height**.

For tabbed pages, it is possible to define two scripts, one that will run when the tabbed page is selected and one that will run when the tabbed page is deselected. If enabled, **Confirm tab select** and **Confirm tab unselect** will cause a warning to be displayed before the tabbed page is selected or unselected, this before a possibly attached script is run.



3.9.5 Functional Controls

Functional controls are:

- Call request list

- Selected call request
- Established call
- Map view
- Tile view
- Grid view
- Device list
- Active device selector
- Map selector
- Button matrix
- Journal
- Display pane

3.9.6 Call request list

The **Call Request List** contains a list of all the Vingtor-Stentofon Operator-related call requests. Each entry in the call request lists can contain the following elements:

- If the related intercom station has an icon in a map, the map and the location of the device in that map is show as an icon. Clicking the icon will select the relevant map in an **Active Map View**, if such a view is part of the layout.
- When **Show handle buttons** has been enabled:
    - If defined for the device related script set: The button at the top of the list in the section **Call Requests Buttons** for the intercom stations associated script set.
    - If defined for the device related script set: The 2nd button in the list in the section **Call Requests Buttons** for the intercom stations associated script set
- The properties **Show video on active request** and **Video tile ID** will be ignored if a **Selected call request** pane (**Active Call Request**) is present in the layout:
    - Set the property **Video tile ID** to an already defined Tile View with ID 1 or higher
    - If **Show video on active request** is enabled: if one or more (up to 4) cameras have been defined for the relevant intercom, the video will be shown in the defined tile automatically when the call request becomes active
    - Clicking on the camera name in the call request list will show the selected camera in the defined tile
- General information about the remote intercom station
- Time when the call request was made and the number of seconds the request has been in the queue

Clicking the area inside the list entry, but outside the iconized map will show details of the call request in the **Selected call request** pane.

3.9.7 Selected call request

Placing a Selected call request pane in the layout will make the properties **Show video on active request** and **Video tile ID** in the call request list inoperative.

The **Selected call request** view contains the following elements:

- If the related intercom station has an icon in a map, the map and the location of the device in that map is show as an icon. Clicking the icon will select the relevant map in an **Active Map View**, if such a view is part of the layout.
- When **Show handle buttons** has been enabled:
    - If defined for the device related script set: The button at the top of the list in the section **Call Requests Buttons** for the intercom stations associated script set.
    - If defined for the device related script set: The 2nd button in the list in the section **Call Requests Buttons** for the intercom stations associated script set
- General information about the remote intercom station
- Time when the call request was made
- A video pane which shows video from the remote station associated camera
- When the flag **Enable snapshots** is set to **True**, the operator will see a small camera in the righthand bottom corner of the video view; a snapshot will be taken when this icon is clicked

3.9.8 Established call

The **Established Call** view contains the following elements:

- If the related intercom station has an icon in a map, the map and the location of the device in that map is show as an icon. Clicking the icon will select the relevant map in an Active Map View, if such a view is part of the layout.
- If defined for the device related script set: The button at the top of the list in the section **Established Call Buttons** for the intercom stations associated script set.
- If defined for the device related script set: The second button in the list in the section **Established Call Buttons** for the intercom stations associated script set.
- General information about the remote intercom station
- Time when the call was established
- Video from 1 or more (up to 4) with the intercom associated cameras will be shown automatically when the property **Show video on connection is enabled**
    - If the property **Video panel visible** is enabled, the video will be shown inside the Established Call view
    - Otherwise, the video will be shown in the Tile View with the ID set in **Video tile ID**; the ID must be 1 or higher
- When the flag **Enable snapshots** is set to **True**, the operator will see a small camera in the righthand bottom corner of the video view; a snapshot will be taken when this icon is clicked

3.9.9 Map view

A map view is a container to show maps and the map-defined icons. Map views can have different behavior depending on the view defined in **View kind**:

- **On request** – Drag-and-drop a map from view **Map Selector** into the map view.
- **Active** – Drag-and-drop a map from view Active Selector into the map view, or click on an iconized map in the call request list, the Active Call Request view or the Established Call view.
- **Follow** – The map which contains the icon of the remote station with which this Vingtor-Stentofon Operator operator has established a conversation will be shown
- **Active-Follow** – a combination of the behavior described under Active and Follow

Note that it is possible to select a map on request in any of the above defined map views, but that such maps will be replaced automatically in views tagged as Active and/or Follow.

A **Default map** will be shown as soon as the operator logs into operational mode. The default map will also be shown when the activity, which triggered the Active or Follow map to be shown, is finished and no other activity is in progress.
Note that for **Default map** to work an **Active device selector** must be defined in the layout.

Icons that are in a map which is visible are active and it is possible to double-click or right-click them to initiate an action, if the icon is programmed this way.

A map view also has a property **View ID**. If there are multiple map views, the View ID can be given unique values. Scripts can use this View ID to place a map in the map view.

Title visible - defaut **False**; set to **True** to show the title for the map view

Status bar visible – default **True**: Set to **False** to increase the available map area; note that the status bar provides information about the icon on which the mouse is placed – when the sttaus bar is not visible this information will not be available

3.9.10 Tile view

A **Tile view** is a container primarily intended for viewing video from cameras, but they can also be used to view maps or images. For this purpose, a tile view has three type selection properties:

- **Camera enabled**
- **Map enabled**
- **Image enabled**

Maps, which are shown in a Tile view, are not active, and double-clicking or right-clicking an icon does not start any action.

When the flag **Enable snapshots** is set to **True**, the operator will see a small camera in the righthand bottom corner of the video view; a snapshot will be taken when this icon is clicked.

If there are multiple tile views, the **Tile ID** can be given unique values. Scripts can use this Tile ID to place the video from a camera or a picture file into the Tile View view.

Title bar visible: The title bar contains information about the camera that is currently shown. For all contents, it also holds a **Close button**. If several cameras are shown in a grid, the title bar also holds the **Maximize button**.

A picture can be imported in the field **Image**. Select the image by browsing to its location.

It is also possible to refer to a picture by location. Enter the fully qualified path in the field **Image location**; this can be a locally stored image, an image on a shared drive or an image on the Internet. If an image location is entered in this field, any contents of the field **Image** are overridden.

If **Show on start** is enabled, the image is shown immediately when the **Tile view** is created in operational mode.

A **Background color** can be specified; This background color will be shown around the picture if the picture does not fill the allocated space completely. It will also be used if the picture is transparent (for instance a vector file without defined background color).

Pictures can be loaded dynamically. See appendix E.2 for details.

3.9.11 Grid view

A Grid view is a combination of a number of Tile Views in one container. It is possible to select one out of 17 layouts, layout 0 to layout 16. Layout 0 is only a placeholder and has no active view area.

 1	 2	 3	 4
 5	 6	 7	 8
 9	 10	 11	 12
 13	 14	 15	 16

A grid view has some additional selection possibilities:

- **Can change layout** – if this is enabled, the operator is allowed to select a layout, and otherwise it is fixed.
- **Can maximize tile** – if this is enabled, the operator can select a tile in the grid view and switch back and forth between the current layout and layout 1.

When the flag **Enable snapshots** is set to **True**, the operator will see a small camera in the righthand bottom corner of the video view; a snapshot will be taken when this icon is clicked.

3.9.12 Device list

The **Device list** will contain all defined devices during operational mode.

3.9.13 Active device selector

The **Active device selector** contains a list of all active (non-idle) devices. Clicking on an item in the list during operational mode will show the map with the associated icon in a map view that has the **Enable device types** property set. If the icon is not placed on a map, clicking the device in the **Active device selector** will have no effect.

The **Active device selector** must be defined in the layout for the property **Default map** in a **Map view** to work.

3.9.14 Map selector

The **Map selector** contains a list of all maps. Double-clicking on an item in the list during operational mode will show the map in a map view that has the ‘View type – on request’ property set. If no map view with this property is available, the map will be placed in a map view with the ‘active’ or ‘follow’ property.

3.9.15 Button matrix

A pre-defined Button matrix can be placed in this container. Select the appropriate matrix from the dropdown list Button matrix. It may be that the defined area does not exactly match the matrix - in that case, it is possible to select that the matrix must be stretched/squeezed horizontally and/or vertically to fit the area exactly.

3.9.16 Journal

The **Journal** is an onscreen history list. Some system messages are shown by default, but in general, all messages that are to be displayed must be configured, normally from scripts in script sets, filter sets, or scripts defined for buttons in button matrices. The Journal also supports an Instant Messaging function where Vingtor-Stentofon Operator operators can send messages to each other. All messages will be visible by all operators with a configured **Journal** pane.

3.9.17 Display pane

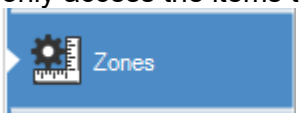
The **Display pane** shows a display with the number of lines as defined in **Number of Lines**. Each line can be written to individualy with the script function **DispWrite**, see paragraph E.2.24.

The default color for background and text can be set through **Background color** and **Label color**.

It is possible to define multiple display panes, each pane must have a unique **Display ID**. In a display pane with display ID=1 it is possible to define one of the lines as **Queue line**; Set **Queue line** to a line number within the range as defined in **Number of Lines**. A Queue line display queue messages which are received by the defined operator station. It is p[possible to select to show all queue information, or only call requests: **Queue entry**: **All** or **Call requests only**.

3.10 Zones

Especially when there are a lot of devices and maps, it might be easier for operators if they can only access the items that are relevant for them. Click **Zones** to define zones.



A zone is a combination of maps and devices. Zones can be added, edited and deleted. If two zones are very similar it is possible to define the first zone, clone it and edit the clone. To clone a zone, click **Commands** followed by **Zone**.

The use of zones requires an enhanced license. Although with a basic license it is possible to define zones, it will not be possible to assign zones to user profiles.

3.10.1 Add a zone

Click **Add** to define a new zone.

In the new dialog, give the zone an appropriate name.

When the flag **All devices** is enabled, all devices are added to the zone, independent of what is shown in the list.

When the flag **All maps** is enabled, all maps are added to the zone, independent of what is shown in the list.

To select which devices or maps should be part of a zone, click . A dialog will be shown in which a selection can be made of items to be part of the zone. Items can be added or removed from the zone in this dialog. Once this dialog is closed, all devices, which are part of the zone, are listed. Delete an individual item from the zone by selecting it and clicking .

Note, all with clients associated intercom stations must be defined as part of the zone that can be handled by an operator from that client. Such an associated station does not need to be placed on a map, and if it is on a map for other reasons, it can be hidden if so desired.

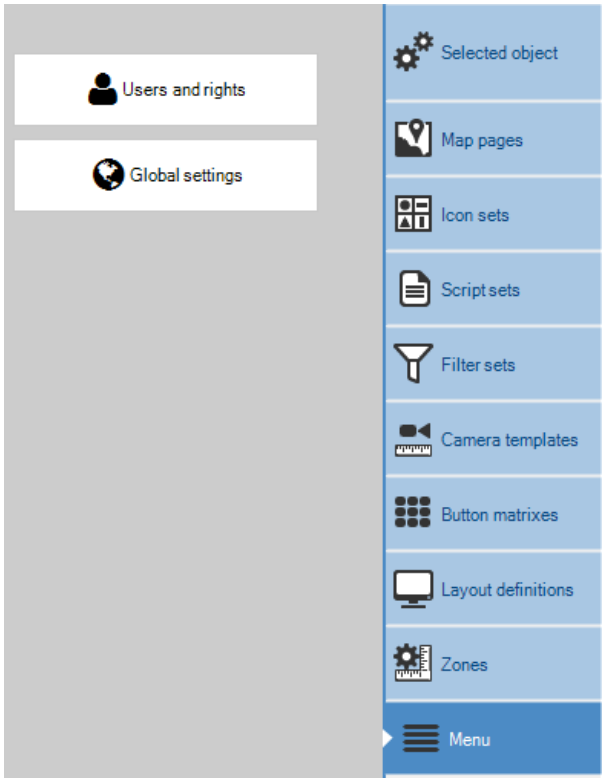
3.10.2 Editing and deleting a zone.

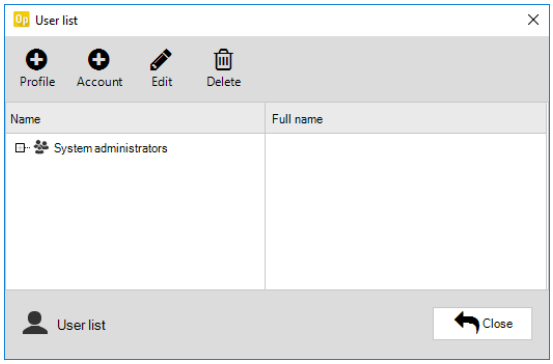
Select a zone and click **Edit** to edit the zone.

Select a zone and click **Delete** to delete the zone.

3.11 Users and Rights

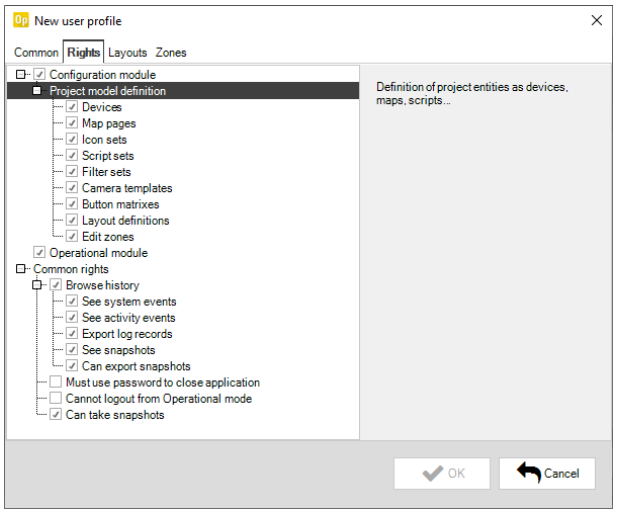
Click **Menu** followed by **Users and rights** to define new user profiles and users.





A **Profile** is intended for a group of users. First, define the profile, and then add users to the profile.

- Click Profile



Different **Rights** can be assigned to **Profile**:

- **Configuration module** - subdivided by different sections which may or may not be changed by members of the profile
- **Operational module** – this module is intended for operators - when an operator logs in he will only see the operational module
- **Common rights**
    - **Browse history** – select this if the operator should have access to the log stored on hard disk
      This is further sub-divided into the possibility to **View system events**, **View activity events** (see paragraph 3.12.1) or **Export log records** (exporting is only available as part of the Enhanced license, see paragraph 2.4.3.1)
    - **Must use password to close application** – This is only valid for operational mode, the user must use a password to close down the application
    - **Cannot logout from Operational mode** – When selected, the **Logout** option is removed from the menu bar at the bottom
    - **Can take snapshots** – when selected the operator with this profile is allowed to take snapshots
- **Layouts** – it is possible to give an operator access to only certain layouts
- **Zones** – it is possible to give an operator access to only certain zones – multiple zones can be assigned to a profile;
  Note that in the Zones-TAB, the default settings **All devices** and **All maps** override the selection made in the list **Assigned zones**
  Note that zoning requires an enhanced license; although zones can be defined even with a basic license, the TAB **Zones** will not be available in the Profiles dialog with a basic license

Once the profile has been defined, it is possible to select the profile and add users. The administrator can also assign an initial password to the user - the user is able to change this password. However, the administrator also has the possibility to reset the password in case this is required.

3.12 Global settings

Click **Menu** followed by **Global settings** to define additional logging options and a startup script.

3.12.1 Additional logging options

Other than the possibility to use the **LOG** script command and the **Show** and **Log** settings in **Filter sets**, Vingtor-Stentofon Operator offers a dedicated logging facility to **Log call requests** and **Log connections** (Calls). When enabled, call requests and calls are stored in a separate database from other logged information.

3.12.2 Startup script

A startup scrip will run every time an operator logs into operational mode.

3.12.3 Maps

When an iconized device button is clicked, the map that this device is defined on will be shown. It is possible to set whether the map should focus on the device and zoom in and if so at what level. Map zoom factor 20 means that the map will not zoom in, a lower value than 20 means that the map will zoom in, to lower the value the less of the complete map will be shown (higher zoom in level).

Hide own station is a global flag with which it can be selected that the with a client associated station is not shown on a map. Note that this setting can be overridden for each client, see paragraph 2.3.3.

Station default label by Directory number – default disabled; By default, the label of a station icon placed on a map is its Display Text; by enabling this checkbox, the default label is the directory number.

3.12.4 Call settings

Vingtor-Stentofon Operator differentiates between low and high priority call requests. This makes it possible to handle call requests from the same intercom station differently, for instance Emergency and Information calls. Set **Call request high priority** to the required value to differentiate between high and low priority call requests. This value corresponds to the call request priority as it can be set in AlphaPro.

Call request low priority can also be used to indicate stations in **private ringing mode** or in **camp-on busy** mode.

- **Call request on private call** means that the intercom device will be handled as if it made a low priority call request when it is in private ringing mode
    - Disable **Show call request buttons / commands when in Private** to suppress the scripts which are defined related to call requests when the station is in private ringing node
- **Call request on busy call** means that the intercom device will be handled as if it made a low priority call request when it is in camp-on busy mode
    - Disable **Show call request buttons / commands when in Busy** to suppress the scripts which are defined related to call requests when the station is in camp-on busy mode

It is advisable to use call requests instead of private ringing and camp-on busy. It is also advisable not to mix private ringing, camp-on busy and ringing groups with call requests in the same system. If a call that is in private ringing mode or made to a ringing group is not picked up by the operator in sequence, the AlphaCom will cancel the first in queue automatically while setting up the call to the

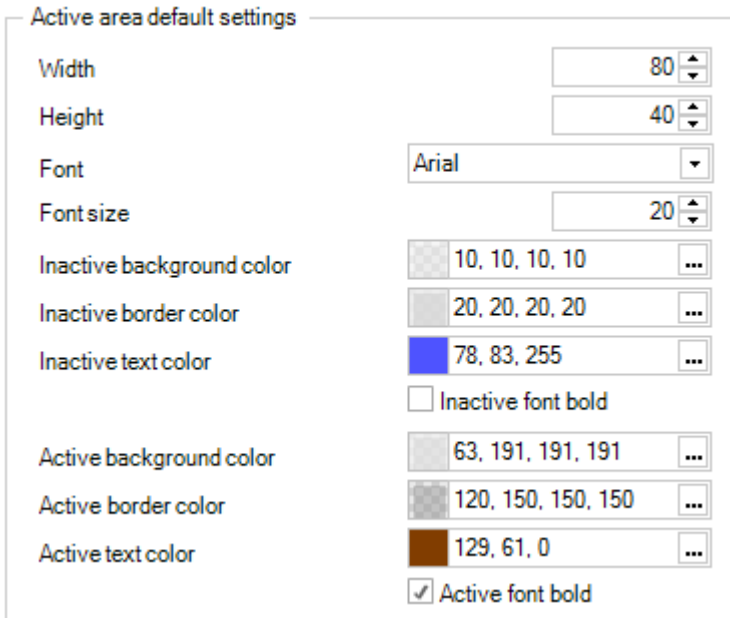
selected destination. This is not a bug in either the AlphaCom or Vingtor-Stentofon Operator software.

3.12.5 Graphics

In the TAB **Graphics**, it is possible to define settings related to icons and map-link objects.

When devices are placed on a map, they can be given some initial graphics properties as defined under **Device default settings**: **Width**, **Height**, **Font**, **Font size** and **Label color**.

Map-link objects can indicate when a device on an underlying map is active. Indication is through properties as depicted below. Note that when a color is fully transparent (the first number of the four number color code = 0), the area is not responsive to a mouse click.



Whether a device is classed as active is determined by the settings in the **Active Device Selector**, see paragraph 3.9.13. If an **Active Device Selector** is not defined in the currently active layout, then Intercom stations, RCO's and RCI's are taken as active when their state is not idle. The states of other device types are in that case not taken into account.

4 Operation

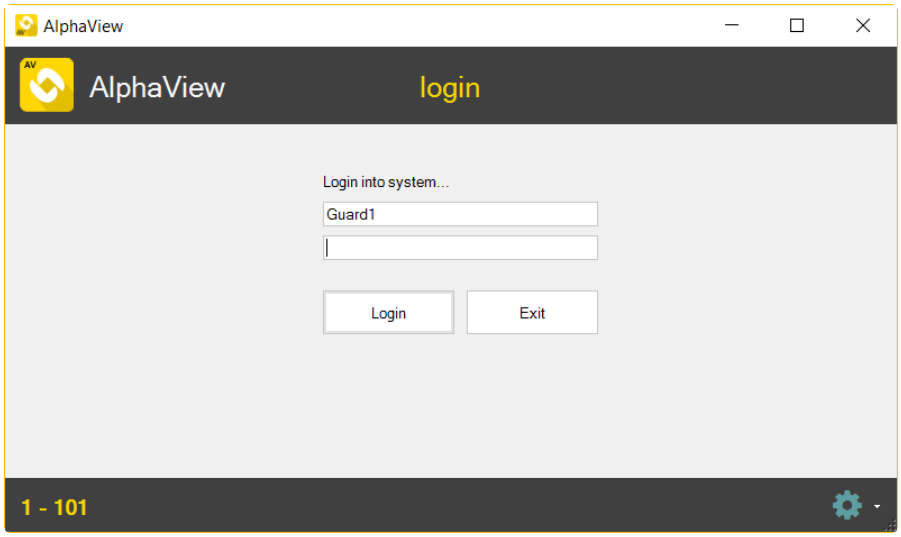
4.1 General

The operation of Vingtor-Stentofon Operator will depend very much on the way the system has been set up. This chapter can therefore only provide some general information about the operation of the system. It is assumed that the operator does not have administrator rights, and that after login, Vingtor-Stentofon Operator will immediately start up in operational mode, as configuration mode is only available to users with administrator rights.

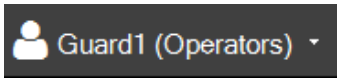
4.2 Login/Logout

Each operator normally has his/her own login credentials.

On Vingtor-Stentofon Operator startup, the login screen will appear automatically. Enter the credentials in the fields provided.



After successful login the username and the user profile is shown at the bottom of the screen.



Clicking on this area will show a dropdown menu with the following options:

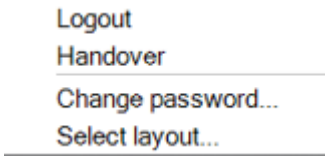
- **Logout** – this may not be visible in operational mode, depending on the assigned user rights
- **Handover** – this is only visible in operational mode
- **Change password…**
- **Select layout…**

4.3 Layout Selection

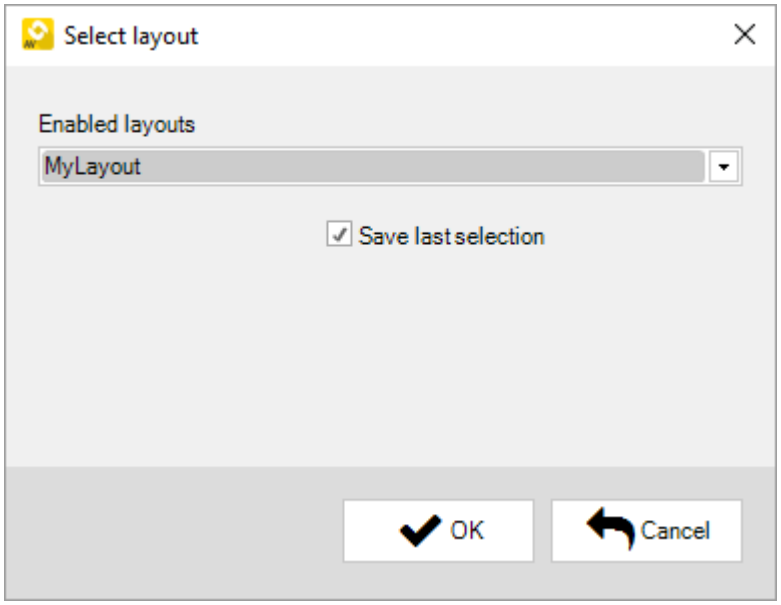
An administrator may have assigned a number of layouts to a profile. The operator can then select the layout that is most suitable.

Normally these different layouts will provide the same information on the screen, but different operators may have a preference where on the screen the information is presented.

- Click the username and select menu option **Select layout…**



A dialog opens from which another layout can be selected.



- Enable **Save last selection** to make certain that when the same operator logs in again the last selected layout is opened.

4.4 Handover

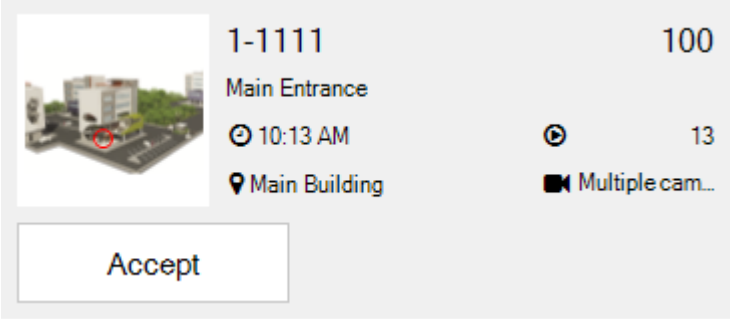
Users can handover to:

- An administrator
- Another user who is a member of the same Profile

Only the user who takes over needs to enter a password. On handover, the selected layout remains, even if the new user used another layout before logout. Any indication on the screen (call requests, icon states, …) remains as is on handover.

4.5 Call Request List

The call request list shows all the pending call requests for this operator. The list is sorted according to the priority of the call request and the time when the call request was made.



Each entry displays some relevant information:

- Directory number of the intercom
- Name of the intercom
- Priority of the call request
- Time when the call request was made
- Number of seconds that the request has been in the list
- Name of a camera associated with the intercom (if a single camera is defined), or the indication **Multiple cameras** if there are more than 1 camera associated with the station
- Name of the map in which the intercom icon is placed (if a map has been defined)
- An iconized picture of that map with a red circle indicating where the station is located on that map
- 0, 1 or 2 buttons to handle the call request

Operation:

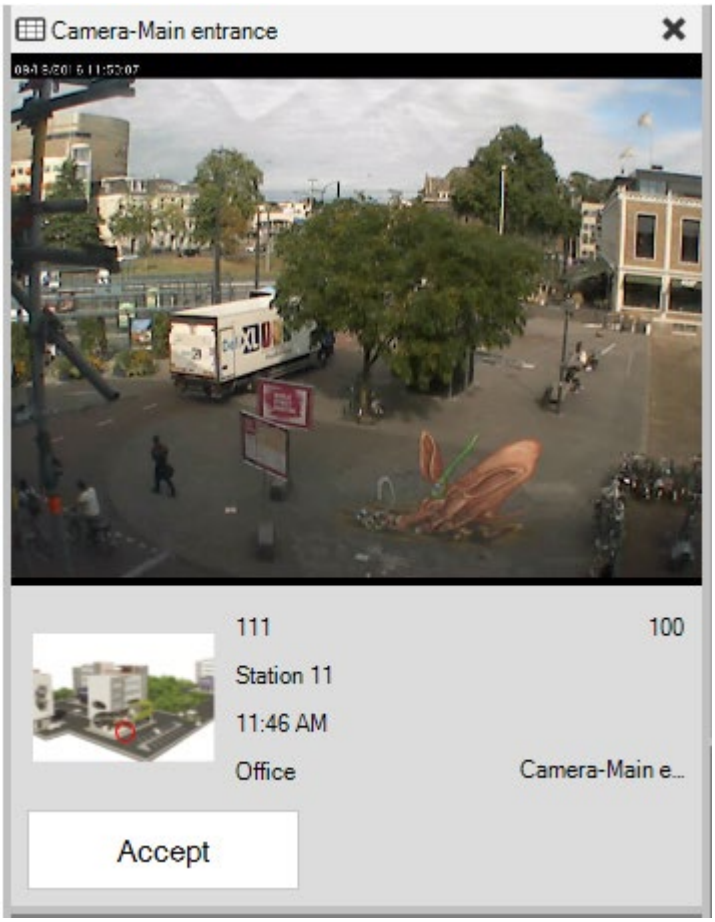
- Click the button to handle the call request
- Click the map icon to show a larger scale map in an Active Map View area on the screen
- Click anywhere else in the entry and the call request will be shown in the **Selected Call Request** pane, only if this pane is available in the selected layout
- Click on the camera name (or **Multiple cameras**) to show the video in a **Tile View**, if a **Tile View** has been defined for this purpose

4.6 Selected Call Request Pane

In addition to the information in the call request list, the **Selected Call Request** pane also shows a video stream from the intercom-associated camera.

The first call request to become active will always be shown in this pane. When a call request is answered, the top-most entry in the call request list will be shown in the selected call request pane.

If a camera icon  is shown in the bottom righthand corner, it is possible to take a snapshot of the contents of the video view.



4.7 Established Call Pane

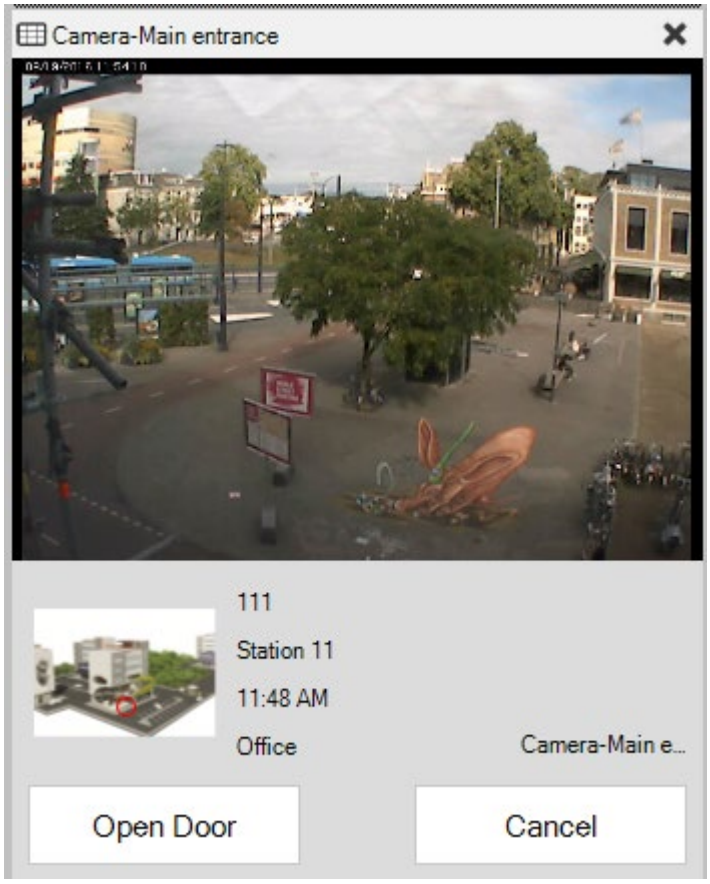
Details of an active call will be shown in the **Established call** pane. The following information can be found in the pane:

- Directory number of the intercom
- Name of the intercom
- Time when the call was made
- Call duration
- Name of a camera associated with the intercom (if a camera is defined)
- Name of the map in which the intercom icon is placed (if a map has been defined)
- An iconized picture of that map
- A video stream from the with the intercom associated camera(s), if the video pane is enabled; The system may also be configured to show the vide in a separate **Tile View**
- 0, 1 or 2 buttons to handle the call

Operation:

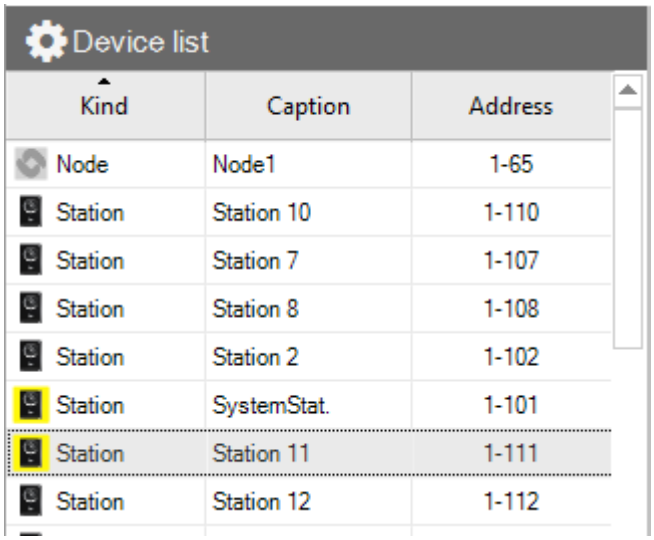
- Click the button to handle the call
- Click the map icon to show a larger scale map in an Active Map View area on the screen

If a camera icon  is shown in the bottom righthand corner, it is possible to take a snapshot of the contents of the video view.

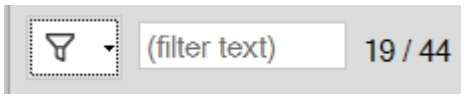


4.8 Device List

The device list shows all devices defined in the system, including **Kind** and **Caption**. The icon appearance depends on the status of the device.

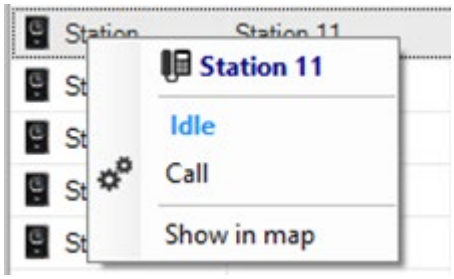


It is possible to easily find a device by filtering the device type, or typing in part of the device name.



- Right-click on the device entry and a dropdown list with commands is shown. Select a command by clicking on it.
- Double-click on the device entry and the action defined for double-click will be executed

The command list which is shown and the action executed on double-click are device state dependent.



On an intercom device the following actions are often defined on double-click:

- Call: on device state **Idle** or **Call Request**
- Cancel: on device state **In Conversation**

4.9 Tile View

One or more **Tile View**s may be set up to display video of with intercom-associated cameras when there is a call request or established call.

Depending on how the system administrator has set up the tile features, it may also be possible to drag-and-drop items onto the tile view:

- Enabled for camera viewing
    - Drag-and-drop a camera device from the device list onto the tile to show the video stream from the camera
    - Drag-and-drop another device type onto the tile – if that device has a camera associated with it the video stream of that camera will be shown

If a camera icon  is shown in the bottom righthand corner, it is possible to take a snapshot of the contents of the video view.

- Enabled to display maps
  - Drag-and-drop a map from the map selection list onto the tile to show the map
  - If a map is already shown in the tile: drag-and-drop a device onto the tile, in which case the map where this device is located will be shown

Maps that are shown in the tile do show device icons in the correct state. However, these icons do not react to right-click or double-click on those icons.

A tile view may also contain an image.

4.10 Grid View

A grid view is a collection of 0-16 Tile Views in different grid layouts.

 1	 2	 3	 4
 5	 6	 7	 8
 9	 10	 11	 12
 13	 14	 15	 16

Operation is very similar to what is described for Tile Views.

Depending on system setup, the following operations may be possible:

- Click in an inactive grid tile or the menu bar of an active grid tile and select **Grid layout** to select 1 of the available layouts
- Click on a grid tile to utilize the complete grid view space for a single grid tile - to go back to grid view, click on the **Restore** control

If a camera icon  is shown in the bottom righthand corner, it is possible to take a snapshot of the contents of the video view.

4.11 Map View

A map view shows maps and the object icons defined in those maps. Object icons can be either map-link objects or device icons.

Map-link objects are areas in maps that when clicked will bring up the next map in the hierarchy. Map-link objects can change appearance depending on whether a device defined in an underlying map is active (state<>0) or idle (state=0). Map-link object which are fully transparent are unresponsive to a mouse click.

- Double click on the map-link object to show the next map in the hierarchy

Device icons are shown depending on the state in which they are.

- Right-click on the device icon and a dropdown list with commands is shown - select a command by clicking on it
- Double-click on the device entry and the action defined for double-click will be executed (default action)

The command list which is shown and the action executed on double-click are device state dependent.

If **Single click as double** has been enabled during system setup, then only a single click is required on a map for the default action or to change a map by clicking a map-link object.

On an intercom device, often the following actions are defined on double-click:

- Call: on device state **Idle** or **Call Request**
- Cancel: on device state **In Conversation**

There may be several map views defined.

Map views may have different behaviors:

- Active map view
  - Click on an iconized map in the call request list, selected call request pane or established call pane to show the map in an active map view
  - Select a map from the Active Maps list to show the map in an active map view
- Follow map view
  - When a call is established the relevant map is shown automatically
- Active-Follow map view
  - This is a combination of Active map view and Follow map view
- On Request map view
  - Drag-and-drop a map from a map selection list onto this view

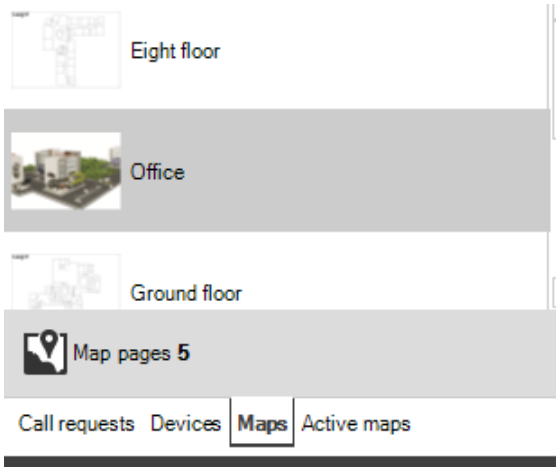
It is always possible to drag-and-drop a map from a map selection list onto any type of map view (on request), but when the view is defined as Active or Follow the map may be replaced by another map by an event as described.

It is always possible to right-click in a map view and select a map from the dropdown list for viewing. When the view is defined as Active or Follow, the map maybe replaced by another map of an event as described.

- To zoom in and out of a map place the mouse pointer inside the map and use the mouse wheel for zooming.
- To move the map in the map view, place to mouse pointer inside the map, click and hold the mouse wheel and drag the mouse.

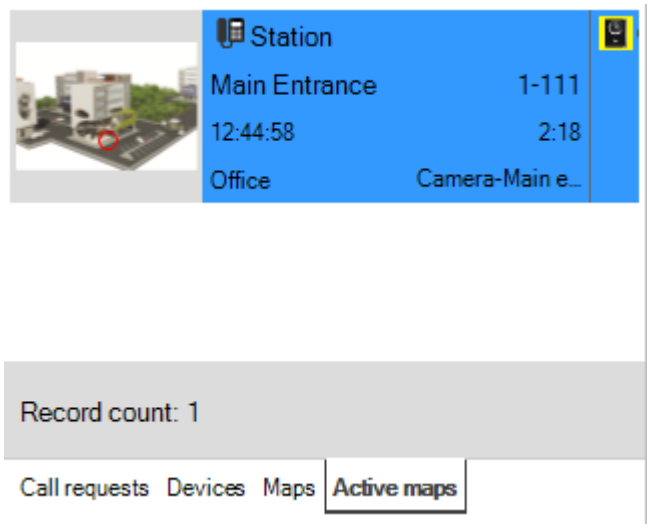
4.12 Map Selector

All in the defined maps in the system are listed in the map selector. Select a map and drag-and-drop it onto a map view or Tile View.



4.13 Active Map Selector

Devices that are in an active state (state different from idle) are listed in the active map selector. Double-click on an entry and the relevant map will be shown in an active map view.



4.14 Button Matrix

A button matrix is a collection of function buttons. Clicking a button will perform the programmed action for that button. Some buttons may be defined as toggle buttons, which means that there are two defined actions. A toggle button will normally have different labels depending on the action that will be invoked on operation.

A button may have zero, one or two (simulated LEDs) that can be in the following modes:

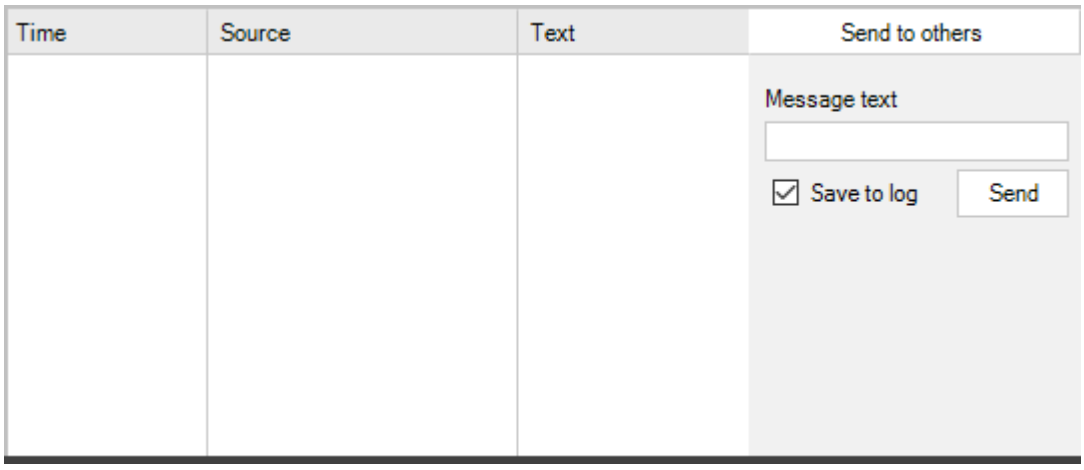
- Off
- Slow flash
- Fast flash
- On

The meaning of these states and the colors that are used for these LEDs will depend on the specific setup.

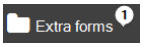
4.15 Journal

The journal shows log entries on screen. The journal also has the possibility of sending messages to other Vingtor-Stentofon Operator operators. Such messages will be sent to

- Click on **Send to others** to show the text input field.
- Click on **Send** and the message will be sent to all operators.
- Enable **Save to log** to store a copy of the message to hard disk.



4.16 Additional Forms

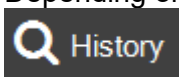
The status bar at the bottom may indicate that there are additional forms available and the quantity of additional forms. 

Clicking on this area will show a list of available forms from which the required form can be selected. This form will now be brought to the foreground.

Note that overlapping forms may cover important operational information on forms below it.

4.17 History

Depending on the rights of the operator, a History button may be available on the status bar.



Click this button to bring up a dialog to view the **System** log, the **Activity** log or available **Snapshots**.

Use the filters to search the entries in the log for the required information. In the Activity log, it is possible to select **Show paired**. When this is selected, call requests and calls will be shown on one line with begin time, end time and duration.

Note that in the **Activity** log it is possible that a **Disconnected** call is shown without a matching **Connected** entry. Under certain circumstances a call may, as far as the AlphaCom is concerned, get disconnected before it was ever connected. This can for instance be the case for a call to an intercom that is set to Private mode and which times out or is cancelled before the call has been accepted.

Click **Export** to export logs to a comma delimited text file. This file can be imported into Excel for further investigation. Snapshots can be exported to a specified directory.

Not every operator may be able to view or export all information; this will depend on the access rights that have been assigned to the particular operator.

4.18 Sounds

It is possible that the client is configured to emit sounds to draw the attention of the operator to an event, for instance an incoming call. It may be that the sound is considered annoying. The operator can switch off the sound by clicking this icon  located at the right bottom of the screen. Only the currently audible sound will be switched off. New events will be annunciated.

A Device Types & States

The following device types and their states are defined.

AlphaCom node	Node up Node down	
AlphaCom station	Idle Active Custom 1 Connected to other party Connected to this operator Call request low priority Call request high priority Error Active custom 2	
RCI	Idle Active	
RCO	Idle Active	
IP camera	Not viewed Viewing	
VS-Recorder server	Disconnected Connected	
VS-Recorder camera	Not viewed Viewing	
General Purpose Device	State 0 State 1 State 2 State 3 State 4 State 5 State 6 State 7	

B Filter Sets Message Types

AlphaCom node	Node state
AlphaCom station	On Call Request Add On Call Request Remove On Connect On Disconnect On Error On CustomParams
RCI	RCI Change State
RCO	RCO Change State
IP camera	Not viewed Viewing
VS-Recorder server	VS-Recorder connect state
VS-Recorder camera	Not viewed Viewing
General Purpose Device	GPD Change State

C SQL Server

SQL Server supports 2 types of Authentication - see also https://msdn.microsoft.com/en-us/library/ms144284.aspx:

- Windows authentication
- SQL Server authentication

In the Vingtor-Stentofon Operator Config Tool, it is possible to select either mode:

- Select **OS Authentication** for Windows authentication – note that Vingtor-Stentofon Operator uses the **NT AUTHORITY\SYSTEM** account
- Deselect **OS Authentication** for SQL Server authentication and enter a user name/password for an existing SQL server user with the appropriate rights.

Windows authentication is always available. SQL Server authentication can only be used if the SQL server instance is set up to support **mixed mode**. It will normally be the IT department that will decide which method should be used.

The selected user must have the following properties set for proper operation:

- Database owner for the defined database
- **View any definition**

Use **Microsoft SQL Server Management Studio** (freely available from Microsoft) to set the selected user as database owner

- Start Management Studio and connect to the SQL server instance
- Navigate to **Security**/**Logins**
- Select the appropriate user, right click and select **Properties**
- Select **User Mapping**
- In the top section of the dialog select the appropriate database
- In the bottom section of the dialog select **db_owner**, leave all other selections as they were
- Click OK and leave Management Studio

To set **View any definition**:

- Start Management Studio and connect to the SQL server instance
- Navigate to **Security**/**Logins**
- Select the appropriate user, right click and select **Properties**
- Select **Securables**
- In the bottom section of the dialog select the TAB **Explicit**
- Enable the checkbox on the intersection of the column **Grant** and the row **View any definition**
- Click OK and leave Management Studio

D General Purpose Devices

D.1 General

General Purpose Devices are devices of which the state can be set in two ways:

- From a script
- Via an HTTP GET to the GPD server implemented in the Vingtor-Stentofon Operator server

The GPD server controls general Purpose Devices, which is part of the Vingtor-Stentofon Operator server. HTTP GET commands can be sent to the host address of the Vingtor-Stentofon Operator server - the port number can be configured, default = 19803.

There are three HTTP GET commands:

getState
setStateSingle
setStateMulti

D.2 getState

http://Vingtor-Stentofon Operator_host_address:GPD_port/getState/GPDName

GPDName can be:

- Source Name
- ID

D.3 setStateSingle

http://Vingtor-Stentofon
Operator_host_address:GPD_port/setStateSingle/GPDName/state_id/Set_Reset/Exclusive

- state_id - A GPD has 8 state_id's (0-7), each of which can be true or false
- Set_Reset – true or false, depending what state the PPD state_id should be set to
- Exclusive – if Exclusive=true and Set_Reset=true, then all other state_id's for the GPD will be set to false

D.4 setStateMulti

http://Vingtor-Stentofon Operator_host_address:GPD_port/setStateMulti/GPDName/xxxxxxxx

- xxxxxxxx – bitmap representation of the state_id's; state_id's will be true or false depending on the value '1' or '0' in the bitmap'; state_id 0 is left-most bit

D.5 Response

The response from the GPD server to each command provides the GPD ID, the state of each state_id in binary representation (xxxxxxxx) and the device name.

Example:

request: http://localhost:19803/setStateMulti/GPD2/00101000
response: GPD:9156fd8f-cd70-4e27-bca8-a9eadbbdab63;00101000;DeviceName

E Scripts

E.1 General

Scripting is implemented in the .NET C# language 4.0. Scripting is client-side and scripts are run by the client application only. Commands can change the state of remote devices (stations, RCI/O, GPD...) or interact with application objects, UI components (maps, views) or the operating system.

Scripting is used in the five parts of the client application:

- Custom script commands (script sets) – popup commands available in maps, device list or some other locations where a device is shown or listed
- Custom message triggers (filter sets) – scripts performed as a reaction on incoming messages.
- Button matrix commands
- Scripts attached to tabbed pages
- Startup script at startup of the operational mode

Each script is launched in a separate thread so the running of a script does not stop the application UI or message processing. However, each command that requires an interaction with the UI has to be synchronized with the main thread.

Note: all methods in the global App object, which interact with the application GUI (Log, Notify, Alert etc.), are synchronized with the UI and they can be used directly in the script code. All programming rules for the C# language apply as normal.

Wrapping code {	
App.STE("$C L101");	Code written by user in the script editor
}	

E.2 The App object instance

The App object is the object that offers methods and data fields (parameters) related to Vingtor-Stentofon Operator states and entities at the time the script is launched.

E.2.1 STE

Definition

```
bool STE(string aCommand);
```

Example

```
bool res = App.STE("$CALL L101 L102");
```

Information

Sends the command to the AlphaCom exchange through the Vingtor-Stentofon Operator server. Returns true when the command is handled successfully.

E.2.2 Notify

Definition

```
void Notify(string text);
```

Example

```
App.Notify("Show this text");
```

Information

Show notify toast strip on the bottom side of the main form.

E.2.3 Alert

Definition

```
void Alert(string text);
```

Example

```
App.Alert("My message");
```

Information

Show alert strip in the bottom right corner.

E.2.4 Log

Definition

```
void Log(string text, LogLevel infoLevel, bool showInJournal, bool saveToDatabase, bool resendToOthers);
```

Example

```
App.Log("Text", true, true);
App.Log("Text", LogLevel.Warning, true, true, true);
```

Information

infoLevel: LogLevel.Info, LogLevel.Warning, LogLevel.Alarm, LogLevel.Error

Show text information in the onscreen Journal – showInJournal=true

Log text information in the database – saveToDatabase=true

Resend the text to all other Vingtor-Stentofon Operator operators – resendToOthers=true

E.2.5 Delay

Definition

```
void Delay(int milliseconds);
```

Example

```
App.Delay(1000);
```

Information

Delays specifies the number of milliseconds, range 0 – 10000.

E.2.6 PlaySound

Definition

```
void PlaySound(string fileName, bool synchronized);
```

Example

```
App.PlaySound(@".\sounds\correct.wav", true);
```

Information

Plays the defined sound file. When a path is not specified, the default path will be used. The default path is the 'Sound' subdirectory in the Vingtor-Stentofon Operator client installation directory.

synchronized=true: sound is played synchronized with the script execution; the script execution halts while the sound is playing and continues when the sound playback has finished

synchronized=false: playing is started and the script continues with the next instruction

E.2.7 PlaySoundLoop

Definition

```
void PlaySoundLoop(string aFileName, bool aSynchronized, int aRepetition);
```

Example

```
App.PlaySoundLoop(@".\sounds\correct.wav", true, 10);
```

Information

Plays the defined sound file number of aRepetition times. If aRepetition=0, then the file will be repeated infinitely.

When a path is not specified, the default path will be used. The default path is the 'Sound' subdirectory in the Vingtor-Stentofon Operator client installation directory.

synchronized=true: sound is played synchronized with the script execution; the script execution halts while the sound is playing and continues when the sound playback has finished

synchronized=false: playing is started and the script continues with the next instruction

E.2.8 StopSound

Definition

```
void StopSound();
```

Example

```
App.StopSound();
```

Information

The **StopSound** command tries to stop all sounds immediately. If **synchronized=true** is used in the command to start the sound, the sound may not stop until the file sound has finished playing.

A **Stop Sound Button** is available in the bottom right corner of the Vingtor-Stentofon Operator client GUI. This button can also be used to stop sounds.

E.2.9 Exec

Definition

```
void Exec(string command, bool synchronized);
void Exec(string command, string params, bool synchronized);
```

Example

```
App.Exec("notepad.exe", true);
App.Exec("notepad.exe", "myfile.txt", true);
```

Information

Executes an external command.

synchronized=true: Command execution is synchronized with the script execution; the script execution halts while the command is running and continuous when the command is finished

synchronized=false: command execution is started and the script continues with the next instruction

E.2.10 Shell

Definition

```
void Shell(string command, string params = null);
```

Example

```
App.Shell(@"http://google.com");
```

Information

Executes an Operating System (Windows) shell command.

E.2.11 Call

Definition

```
void Call(string filename, string entryName);
```

Example

```
App.Call(@".\scripts\ScriptStandard.avs", "Cancel");
```

The content of the script ‘ScriptStandard.avs’:

```
public void Call()
{
        App.STE("@U%NNOP $CALL L%DIROP L%DIRA");
}


public void Cancel()
{
        App.STE("@U%NNOP $C L%DIROP");
}
```

Information

Executes a Vingtor-Stentofon Operator script from an external filename.

Note: calling between external scripts is limited (maximum 3 deep).

E.2.12 ShowMapActive

Definition

```
void ShowMapActive(string mapName);
```

Example

```
App.ShowMapActive("Floor 12");
```

Information

Shows the specified map in the Active map pane. The mapName parameter must be exactly (but not case sensitive) a map caption defined in the project.

Note: if no map of the specified name is found or no Active map pane is specified in the current layout, the command has no effect.

E.2.13 ShowMapFollow

Definition

```
void ShowMapFollow(string mapName);
```

Example

```
App.ShowMapFollow("Floor 12 ");
```

Information

Shows the specified map in the Follow map pane. The mapName parameter must be exactly (but not case sensitive) a map caption defined in the project.

Note: if no map of the specified name is found or no Follow map pane is specified in the current layout, the command has no effect.

E.2.14 ShowMapRequest

Definition

```
void ShowMapRequest(string mapName, int viewId);
```

Example

```
App.ShowMapRequest("Floor 12", 20);
```

Information

Shows the specified map in the on request map view. The mapName parameter must be exactly (but not case sensitive) a map caption defined in the project.

Shows a map in OnRequest pane. When viewId = 0 (or specified view is not found), the first allocated on request MapView is used.

E.2.15 ViewCamera

Definition

```
void ViewCamera(string IdOrName, int viewId);
```

Example

```
App.ViewCamera("DeviceName", 15);
```

Information

Show a camera in the defined tile view pane. View ID should be set in the layout definition.

E.2.16 ViewMap

Definition

```
void ViewMap(string mapName, int viewId, bool reset);
```

Example

```
App.ViewMap("DeviceName", 15, true);
```

Information

Show map in the tile view pane.

Reset - true: fit map into view, false: keep previous map viewport.

E.2.17 ViewDevice

Definition

```
void ViewDevice(string IdOrName, int viewId, bool animate);
```

Example

```
App.ViewDevice("%ID", 15, true);
```

Show specified device in its map in the tile view pane and focuses it

E.2.18 ViewPicture

Definition

```
void ViewPicture(string aLocation, int aViewId);

void ViewPicture(string aLocation, int aViewId, Color aColor);

void ViewPicture(string aLocation, int aViewId, string aHexaColor);
```

Example

```
App.ViewPicture(@"c:\Data\MyPictures\Photo1.jpg", 15);
App.ViewPicture(@"c:\Data\MyPictures\Photo1.jpg", 15, Color.Blue);
App.ViewPicture("http:\\www.somesite.com\Photo1.jpg", 15, "#FFCC66");
```

License restriction

With a Basic license, this function will not be called and an error message will be raised in the System log

E.2.19 ViewDefaultPicture

Definition

```
void ViewDefaultPicture(int aViewId);
```

Example

```
App.ViewDefaultPicture(15);
```

Displays the picture as defined as **Image** or **Image location** for the Tile view with ID=aViewID.

License restriction

With a Basic license, this function will not be called and an error message will be raised in the System log

E.2.20 ViewClose

Definition

```
void ViewClose(int view Id);
```

Example

```
App.ViewClose(15);
```

Information

Close specified view.

E.2.21 ShowDevice

Definition

```
void ShowDevice(string aName, int aViewId);
void ShowDevice(string aName, int aViewId, bool aZoom, bool aAnimate);
```

Example

```
App.ShowDevice("Main Entrance", 20);
App.ShowDevice("Back door", 20, true, true);
App.ShowDevice("Reception", 155, true, false);
```

Information

Shows device (ID or name) in requested map view ID with or without zoom. If the aZoom parameter is true, it is possible to animate the zoom.

As zoom factor the global Map zoom factor will be used, see paragraph 3.12.3.

E.2.22 SetButtonState

Definition

```
void SetButtonState(string buttonId, bool toggle);
```

Example

```
App.SetButtonState("ButtonID", true);
```

Information

Set button matrix button state.

E.2.23 SetButtonLEDState

Definition (1)

```
void SetButtonLEDState(string buttonId, int ledId, int ledState);
```

Example

```
App.SetButtonLEDState("ButtonID", 0, 0);
App.SetButtonLEDState("ButtonID", 0, 0);
```

Information

Set button matrix button's LED state.

- Led id: 1 or 2 (left, right).
- Led state: 0= off, 1= slow, 2= fast, 3= on.

The above method uses the default LED colors

Definition (2): App.SetButtonLEDState – overloaded

```
void SetButtonLEDState(string aButtonId, LEDKind aLedKind, LEDState aLedState, Color aColor,
Color aAlterColor);
```

```
public enum LEDKind
```

```
{
    Left,
    Right
}

public enum LEDState
{
    Off,
    Slow,
    Fast,
    On
}
```

When Color.Transparent is used as the aColor or aAlterColor parameter, the default LED color will be used.

The AlterColor is used for Slow and Fast LED state.

Example 1 - set MyButton, left LED to slow flashing Red/Yellow

```
App.SetButtonLEDState("MyButton", LEDKind.Left, LEDState.Slow, Color.Red, Color.Yellow);
```

Example 2 - set MyButton, left LED to OFF state

```
App.SetButtonLEDState("MyButton", LEDKind.Left, LEDState.Off, Color.Transparent,
Color.Transparent);
```

E.2.24 DispWrite

Definition (1)

```
void DispWrite(int aDisplayId, int aLineNr, string aText);
```

Example

Uses font and colors as defined for the display pane

```
App.DispWrite(10, 1, "This is line 1 in display pane with ID=10");
```

Definition (2)

```
void DispWrite(int aDisplayId, int aLineNr, string aText, Color aFontColor, int aFontSize, FontStyle
aFontStyle = FontStyle.regular, string aFontName = “”);
```

Example

```
App.DispWrite(1, 1, "Script launched", Color.Red, 24, FontStyle.Bold,
"Verdana");
```

Note: It is possible to use mini-markup/text as well, see
https://www.devcomponents.com/kb2/?p=515

Example

```
App.DispWrite(1, 1, "<div align='center' valign='center'
padding='4,4,4,4'>This <font color='white'><b>is</b></font> line
1</div>");
```

Information

Writes to lines in a display pane.

E.2.25 SetGPDState

Definition

```
void SetGPDState(string IdOrName, int stateIndex, bool value, bool exclusive );
```

Example

```
App.SetGPDState("%ID", 2, false, true);
```

Information

Set GPD state.

Index: 0..7

Exclusive: true = when value is true, all other states are set to false

E.2.26 SetGPDStates

Definition

```
void SetGPDStates();
```

Example

```
App.SetGPDStates("ID_or_name", new bool[8]);
```

Information

Set GPD states at all.

E.2.27 ResetGPD

Definition

```
void ResetGPD();
```

Example

```
App.ResetGPD("ID_or_name");
```

Information

Set all GPD states to false.

E.2.28 GetDeviceState

Definition

```
int GetDeviceState(string aIdOrName);
```

Example

```
int st = App.GetDeviceState("GPD1001");

bool[] s = new bool[8];
s[0] = (st & 0x1) != 0;
s[1] = (st & 0x2) != 0;
s[2] = (st & 0x4) != 0;
s[3] = (st & 0x8) != 0;
s[4] = (st & 0x10) != 0;
s[5] = (st & 0x20) != 0;
s[6] = (st & 0x40) != 0;
s[7] = (st & 0x80) != 0;
```

Information

GetDeviceState returns an integer type value. States are not exclusive for all device types, GPD and station devices can have more than one state; use an &-operation to filter on a specific state as shown in the example above.

E.2.29 GetStationState

Definition

```
int GetStationState(int aNode, string aDirNo);
```

Example

```
int st = App.GetStationState(1, "101");

bool[] s = new bool[8];
s[0] = (st & 0x1) != 0;
s[1] = (st & 0x2) != 0;
s[2] = (st & 0x4) != 0;
s[3] = (st & 0x8) != 0;
s[4] = (st & 0x10) != 0;
s[5] = (st & 0x20) != 0;
s[6] = (st & 0x40) != 0;
s[7] = (st & 0x80) != 0;
```

Information

GetStationState returns an integer type value. Station devices can have more than one state; use an &-operation to filter on a specific state as shown in the example above.

E.2.30 GetRCIState

Definition

```
int GetRCIState(int aNode, int aRCINr);
```

Example

```
int st = App.GetRCIState(1, "RCI1");
```

Information

GetRCIState returns an integer type value which has either value 0 (RCI inactive) or 1 (RCI active).

E.2.31 Get RCOState

Definition

```
int GetRCOState(int aNode, int aRCONr);
```

Example

```
int st = App.GetRCOState(1, "RCO1");
```

Information

GetRCOState returns an integer type value which has either value 0 (RCO inactive) or 1 (RCO active).

E.2.32 SetGlobal

Definition

```
bool SetGlobal(string aName, object aValue);
```

Example

```
App.SetGlobal("Test", "This text is now stored in variable Test");
```

Information

SetGlobal stores a value in the location with name defined by aName. If aName does not exist yet, it will be created; otherwise, the old value will be overwritten. The value is of type **object**, which means that any kind of variable can be stored. It is up to the programmer to know what type was stored and do the appropriate type conversion when the value is retrieved.

The variable is global to the local client. The value survives script end and can be used to pass values between scripts, or to store a value which can be used the next time the same script is started.

E.2.33 GetGlobal

Definition

```
object GetGlobal(string aName, object aDefault);
```

Example

```
string s = (string)App.GetGlobal("Test", "(?)");
```

Information

The above example returns the value stored in Test as a string if the variable Test exists, otherwise it will return a string with value (?).

E.2.34 GlobalExists

Definition

```
bool GlobalExists(string aName);
```

Example

```
if (App.GlobalExists("TEST"))
{
        App.Alert("TEST already exists!");
}
else
{
        App.Alert("TEST does not exist!");
}
```

Information

GlobalExists returns **true** if the variable already exists, otherwise it will return **false**.

E.2.35 DeleteGlobal

Definition

```
bool DeleteGlobal(string aName);
```

Example

```
App.DeleteGlobal("TEST");
```

Information

DeleteGlobal deletes the previously defined variable.

E.2.36 ActivateForm

Definition

```
void ActivateForm(string aFormTitle, bool aValue);
```

Example

```
App.ActivateForm("MyForm", true);
```

Information

ActivateForm activates (shows) or deactivates (hides) the form, but only when the form-flag Use pane active flags is enabled and the (internal) activity of the relevant functional controls allow a change of visibility state, see paragraph 3.9.3.

E.2.37 SelectLayout

Definition

```
void SelectLayout(string LayoutDefCaption);
```

Example

```
App.SelectLayout("Layout_1");
```

Information

Replaces the Layout currently in use with the specified layout.

License restriction

With a Basic license, this function will not be called and an error message will be raised in the System log

E.2.38 SelectStartupLayout

Definition

```
void SelectStartupLayout();
```

Example

```
App.SelectStartupLayout();
```

Information

Replaces the Layout currently in use with the layout as selected when the client started in Operational mode.

License restriction

With a Basic license, this function will not be called and an error message will be raised in the System log

E.3 Runtime parameters

E.3.1 Defined runtime parameters

%DIROP
DIROP – type string
Directory number of the Vingtor-Stentofon Operator position related intercom unit

%DIRA
DIRA – type string
Directory number related to an intercom device in the GUI
'A-subscriber' in data protocol messages

%DIRB
DIRB – type string
‘Related to’ directory number in event handler
‘B-subscriber’ in data protocol messages

%NNOP
NNOP – type int
Node number associated with %DIROP

%NNA
NNA – type int
Node number associated with %DIRA

%NNB
NNB – type int
Node number associated with %DIRB

%CAPTION
CAPTION – type string
Device name – Note, CAPTION is now only available for backward compatibility with scripts which have been written previously; it is advised to use ACAPTION, BCAPTION and OPCAPTION instead

%ACAPTION
ACAPTION – type string
Device name; available in cases where also DIRA is passed as parameter

%BCAPTION
BCAPTION – type string
Device name; available in cases where also DIRB is passed as parameter

%OPCAPTION
OPCAPTION – type string
Device name; available in cases where also DIROP is passed as parameter

%DURATION
DURATION – type int
Duration of the call request or call

%TAG
TAG – type int
TAG number (Mail Tag or Connection Reference)

%PRIO
PRIO – type int
Call request priority (as defined in the AlphaCom exchange)

%CALLMODE
0 – No Access
1 – Call in Open
2 – Private ringing mode
3 – Camp-on busy mode
Note: %CALLMODE is added to the **On Call Request Add** filter; this is because private ringing and camp-on busy calls can be converted to a low priority call request

%CALLPRIORITY
A station Call Priority as passed by the CALL_STATUS_BC message
Note: %CALLMODE is added to the **On Call Request Add** filter; this is because private ringing and camp-on busy calls can be converted to a low priority call request

%RCO
RCO – type int
RCO number

%RCOCAPTION
RCO – type string
Device name; available in cases where also RCO is passed as parameter

%RCI
RCI – type int
RCI number

%RCICAPTION
RCI – type string
Device name; available in cases where also RCI is passed as parameter

%CHG
CHG – type int
State change, 1 for on, 0 for off, coded as UINT1

%BUTTONID
BUTTONID – type string
ID associated with button in a button matrix

%ERROR
ERROR – type bool

%ID
ID – type string
DeviceID : GUID
DeviceID : Source name

%CP1
CP1 – type int
Value of Customer Parameter1
%CP2, %CP3, %CP4, %CP5 are value for Customer Parameter2-5

%CSP1
CSP1 – type string
Value of Customer String Parameter1
%CSP2, %CSP3, %CSP4, %CSP5 are value for Customer String Parameter2-5

%MSG
MSG – type int
Value of the MessageID of the underlying ACDP-message from AlphaCom to Vingtor-Stentofon Operator

%STATES
STATES = type bool
Holds an array of booleans related to the states of a GPD

```
bool[] gpd = (bool[])Av.Params.GetValue<bool[]>("STATES", null);

if (gpd != null)
{

      string v1;

      v1 = string.Format("gpd[0] = {0}", gpd[0]);
      Av.Log(v1, true, false);

      v1 = string.Format("gpd[1] = {0}", gpd[1]);
      Av.Log(v1, true, false);

      v1 = string.Format("gpd[2] = {0}", gpd[2]);
      Av.Log(v1, true, false);

}
```

E.3.2 Parameter substitution

%-type parameters are substituted via text substitution before the script is running.

Assume that a call request has been made from the station with directory number 111.

The value of %DIRA for the On Call Request Add filter message is 111.

A script statement as below to flash a LED on a DAK button in a button matrix

```
App.SetButtonLEDState("DAK%DIRA",LEDKind.Left, LEDState.Slow,
Color.Blue, Color.White);
```

will in fact be replaced by

```
App.SetButtonLEDState("DAK111",LEDKind.Left, LEDState.Slow,
Color.Blue, Color.White);
```

Alternatively, the actual value of a parameter can also be evaluated during run-time of the script as each fired script has its own instance of parameter values accessible through the App.Params object. In that case, the name of the parameter without the %-prefix is used as the identifier. The App.Params.Get method will return the correct value.

```
string dircr = App.Params.Get("DIRA", "");
App.SetButtonLEDState("DAK"+dircr, LEDKind.Left, LEDState.Slow,
Color.Blue, Color.White);
```

Note that these values can be added or modified during the period the script is running and that therefore the actual value of the parameter may not be the same as what it was when the script was started.

```
string App.Params.Get("stringTypeParameter", "DefaultString");
int App.Params.Get("intTypeParameter", DefaultValue);
bool App.Params.Get("boolTypeParameter", Default_true_or_false);
```

E.3.3 Available parameter

It is possible to retrieve the list of available parameters:

```
string App.Params.GetParamList("; ");
```

When in doubt which parameters are available at the moment a script is triggered, it is possible to add these lines at the start of the script:

```
string s = App.Params.GetParamList("; ");

App.Log(s, true, false);
```

When the script is triggered, a list of available parameters will now be shown in the onscreen journal.

E.4 Sample scripts

E.4.1 Accept a call request, make a call

```
App.STE("@U%NNOP $CALL L%DIROP L%DIRA");
```

E.4.2 Cancel a call

```
App.STE("@U%NNOP $C L%DIROP");
```

E.4.3 Place a call back into the call queue

```
App.STE("@U%NNOP $C L%DIROP");
App.STE("@U%NNA $SM L%DIRA L%DIROP U120 NM18 GV U1");
```

E.4.4 Open a door during conversation

```
string dirop = App.Params.Get("DIROP", "");
string dira = App.Params.Get("DIRA", "");
int nnop = App.Params.Get("NNOP", 0);
int nna = App.Params.Get("NNA", 0);


if ((dira==dirop) && (nna==nnop))
{
  App.STE ("@U%NNB $ER U2 U0 U1 U1 L%DIRB U3 L(%NNOP)%DIROP");
  App.Delay (3500);
  App.STE ("@U%NNB $ER U2 U0 U0 U1 L%DIRB U3 L(%NNOP)%DIROP");
}
else
{
  App.STE ("@U%NNA $ER U2 U0 U1 U1 L%DIRA U3 L(%NNOP)%DIROP");
  App.Delay (3500);
  App.STE ("@U%NNA $ER U2 U0 U0 U1 L%DIRA U3 L(%NNOP)%DIROP");
}
```

E.4.5 DAK keys, LED activation

On an incoming call request, it is possible to flash a LED on a DAK key in a button matrix.

Call request from a substation, for instance 111.

Define the DAK keys in the Button matrix to have a **Button ID** related to the directory number of the associated intercom station (DAKdirnr), in this case **DAK111**.

The remote station has a **Filter set** assigned to it (RemoteFilter).

The script for **OnCallRequestAdd**:

```
App.SetButtonLEDState("DAK%DIRA", LEDKind.Left, LEDState.Slow,
Color.Blue, Color.White);
```

The script for **OnCallRequestRemove**:

```
App.SetButtonLEDState("DAK%DIRA", LEDKind.Left, LEDState.Off,
Color.Transparent, Color.Transparent);
```

The operator station has a **Filter set** assigned to it (OperatorFilter).

The script for **OnConnect**:

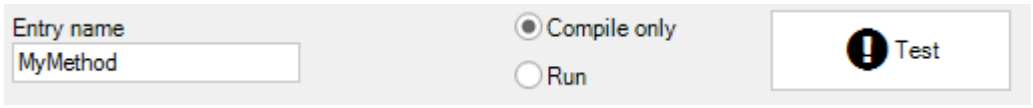
```
App.SetButtonLEDState("DAK%DIRB", LEDKind.Left, LEDState.On,
Color.Green, Color.Transparent);
```

The script for **CallDisconnect**:

```
App.SetButtonLEDState("DAK%DIRB", LEDKind.Left, LEDState.Off,
Color.Transparent, Color.Transparent);
```

E.5 Subroutines

It is possible to write a script with subroutines. Be certain to redefine the name of the entry method in the field 'Entry name' below the script editor area.



Wrapping code {	

private void AlertEx(string text)
{
App.Alert("My new text is: " + text);
}
 	
// main method
public void MyMethod()
{
App.SetButtonLEDState("DAK%DIRA",LEDKind.Left,
LEDState.Slow, Color.Blue, Color.White);
 	
AlertEx("LED is flashing Blue");
}
	Code written by user in the script editor

}	

E.6 Using your own library

It is possible to use a custom .NET library.

Assume a sample library called MyCustomLib.dll with class Calculator with the method Sum:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace MyCustomLib
{
	public class Calculator
	{
		public int Sum (int A, int B)
		{
			return A + B;
		}
	}
}
```

Example of Vingtor-Stentofon Operator script using the MyCustomLibrary:

```
// library file
string fn = @".\scripts\MyCustomLib.dll";

// load assembly
System.Reflection.Assembly asm = System.Reflection.Assembly.LoadFrom(fn);
if (asm == null)
{
	App.Notify("Cannot load library");
	return;
}

// get custom class
Type type = asm.GetType("MyCustomLib.Calculator");
if (type == null)
{
	App.Notify("Required class not found");
	return;
}

// get custom method
System.Reflection.MethodInfo method = type.GetMethod("Sum");
if (method == null)
{
	App.Notify("Required method not found");
	return;
}

// create class instance
object classInstance = Activator.CreateInstance(type, null);
```

```
// prepare parameters
System.Reflection.ParameterInfo[] parameters = method.GetParameters();
object[] parametersArray = new object[] { 2, 5 };

// call method and show result
int result = (int)method.Invoke(classInstance, parametersArray);
App.Notify("Result is: " + result.ToString());
```

E.7 Script files

E.7.1 General

It is possible to define script files and execute the scripts in such a file using the Call method, see paragraph E.2.11. Also, see paragraph E.7.2 for an example of such a script file and how to use it.

E.7.2 Standard script file

A standard script file is supplied: **ScriptStandard.avs** is located in **C:\Program Files (x86)\Vingtor Stentofon\Vingtor-Stentofon Operator\Vingtor-Stentofon Operator Client\scripts**, assuming the default installation location is selected.

The contents of the file:

```
public void Call()
{
        App.STE("@U%NNOP $CALL L%DIROP L%DIRA");
}


public void Cancel()
{
        App.STE("@U%NNOP $C L%DIROP");
}


public void PlaceBackInQ()
{
        App.STE("@U%NNOP $C L%DIROP");
        App.STE("@U%NNA $SM L%DIRA L%DIROP U120 NM18 GV U1");
}

public void OpenDoor()
{
    string dirop = App.Params.Get("DIROP", "");
    string dira = App.Params.Get("DIRA", "");
    int nnop = App.Params.Get("NNOP", 0);
    int nna = App.Params.Get("NNA", 0);


    if ((dira==dirop) && (nna==nnop))
    {
      App.STE ("@U%NNB $ER U2 U0 U1 U1 L%DIRB U3 L(%NNOP)%DIROP");
      App.Delay (3500);
      App.STE ("@U%NNB $ER U2 U0 U0 U1 L%DIRB U3 L(%NNOP)%DIROP");
    }
    else
```

```
    {
      App.STE ("@U%NNA $ER U2 U0 U1 U1 L%DIRA U3 L(%NNOP)%DIROP");
      App.Delay (3500);
      App.STE ("@U%NNA $ER U2 U0 U0 U1 L%DIRA U3 L(%NNOP)%DIROP");
    }
}
```

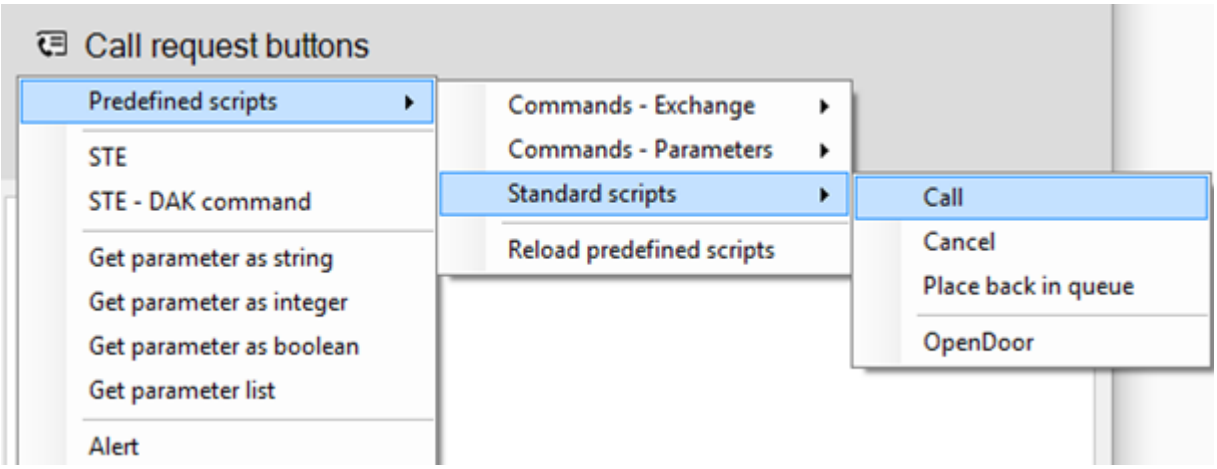
The functions can be called in this way:

```
App.Call(@".\scripts\ScriptStandard.avs", "Call");
App.Call(@".\scripts\ScriptStandard.avs", "Cancel");
App.Call(@".\scripts\ScriptStandard.avs", "PlaceBackInQ");
App.Call(@".\scripts\ScriptStandard.avs", "OpenDoor");
```

E.7.3 Predefined scripts

Scripts in script files can be used as described in paragraph E.7.2. It is also possible to import the text of the body of a function that is defined in a script file directly.

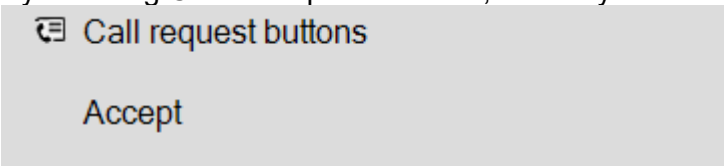
Right click in the script edit area. A list of available scripts will be displayed.



Standard scripts refers to the file **ScriptStandard.avs**. It shows in the menu as Standard script because the menu text is defined inside the file as following:
/// av.group = Standard scripts

The script function names are used in the lower level menu unless defined differently. The in the file **ScriptStandard.avs** there is a function **PlaceBackInQ** but for the purpose of the menu this is defined to be listed as **Place back in Queue**:
/// av.caption = Place back in queue

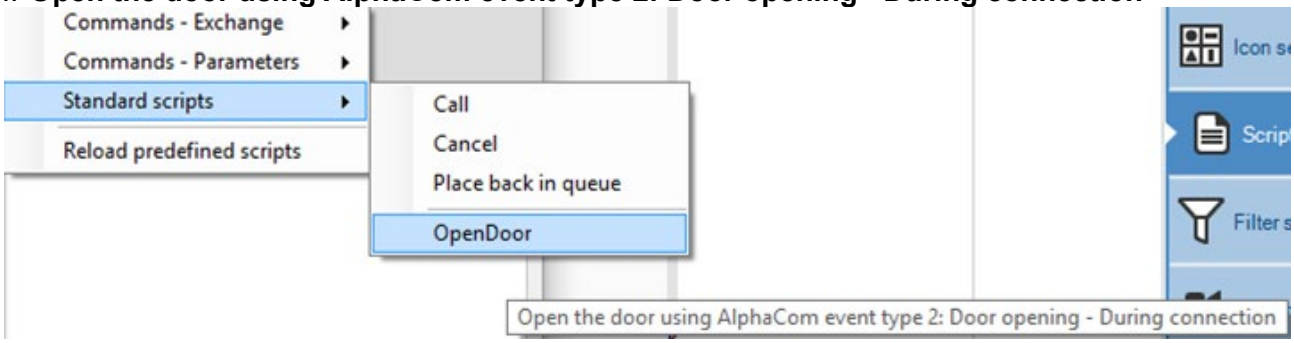
By selecting Call in the picture above, the body text of that function is placed in the script edit area.




```
Av.STE ("@%NNOP $CALL L%DIROP L%DIRA");
```

The Vingtor-Stentofon Operator client on start of the software reads predefined scripts. It is possible to add script files that are defined later by clicking **Reload predefined scripts**.

Comments in the file are shown as tool tip:
// Open the door using AlphaCom event type 2: Door opening - During connection



The separator between **Place back in Queue** and **OpenDoor** is because it is defined as part of the function for **OpenDoor**:
// av.separator

F Reusing Some of the Programming

There are several ways to reuse programming between projects.

F.1 SQL Database

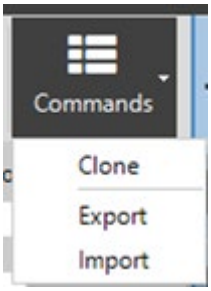
Define a new project database

- Add all the programming which is expected to be used in every future project, for instance
  - Icon sets
  - Script sets
  - Filter sets
  - Layout definitions
  - Camera templates
  - Button matrixes
- Open the SQL server instance using SQL Management Tools (a free tool from Microsoft), and select the defined database
- Right-click the database
- Select 'Tasks/Backup…'
- Backup the database and keep it safe
- For the next project, import the database on the new project PC and use it as a starting point.

F.2 Export, Import and Clone

It is also possible to select **Icon sets**, **Script sets**, **Filter sets**, **Camera templates**, **Button matrices** and **Layout definitions** and export them to XML files. These XML files can be imported again into a new project.

- Select the entity to be exported
- Click on the **Commands** button in the menu at the top of the list



- Select **Export** and select the location to save the exported information in
- In a new project, click on Commands > Import and select the file to import

Clone will simply make a copy and put it directly in the list with the same name and the extension **(clone)**. This makes it easy if for instance a new script set needs to be made which only differs from an already in the project existing script set.

- Select the script set
- Clone it
- Edit it (change the name and make changes to the scripts as required)

G Defaults

When a new database is defined in the Vingtor-Stentofon Operator Server Configuration Tool, the following items are preloaded into the database. Note that these will be visible as normal, and can therefore be edited or deleted as required:

- Icon sets
  - Intercom_Iconset_1
  - Intercom_Iconset_2
  - Intercom_Iconset_3
  - Turbine_Iconset_1
  - Turbine_Iconset_2
  - RCO_Iconset_1
  - RCO_Iconset_2
  - RCI_Iconset_1
- Script sets
  - Intercom_Scriptset_Generic
  - Intercom_Scriptset_OpenDoor
  - Intercom_Scriptset_CallInPrivate
  - Intercom_Scriptset_BackInQ
- Filter sets
  - DAKConnect_Filterset
  - DAKCR_Filterset
- Camera templates
  - TCIV
  - CP-CAM
  - AXIS
- Button matrices
  - Keyboard
  - PressToTalk
  - DAKPanel1
  - DAKPanel2
- Layout definitions
  - Layout_1
  - Layout_2
  - Layout_3
  - Layout_4
  - Layout_NoVideo

H Functionality and licenses

This appendix lists the functionality, which is available in Vingtor-Stentofon Operator, depending on the available license.

Enhanced license	Basic license	Description
Layout can have multiple forms	Only 1 form can be defined in a layout	A form is the placeholder for all visual elements. A form can be subdivided using horizontal and vertical splitters.
Assign forms to specific monitors, connect as many monitors to the client PC as physically supported	Only 1 monitor is supported, the default monitor as defined by Windows	
Multiple map-views in a form	Single map-view in the form	A map view is where maps are placed. Maps can contain device icons which are shown in their different statuses
Device list	Device list	A device list lists all defined device. In case of zoning, only devices related to the accessible zones are listed
Call request list	Call request list	The call request list lists all active call requests (in the zone)
Active map selector	Active map selector	An active map is a map with at least one active (non-idle) device; This list lists all active maps (in the zone)
Map selector	Map selector	Lists all maps (in the zone)
Active call request pane	Active call request pane	Select a call request in the call request list and move it to the Active call request pane; this pane is not strictly required, as all information is available in the call request list

Enhanced license	Basic license	Description
Scriptset	Scriptset	Use script sets to attach functionality to devices (state-dependent); invoked on (double-)clicking devices
Filterset	Filterset	Use filter sets to attach functionality to devices (state-dependent); invoked on data received
Established call pane	Established call pane	Holds information about the call between the ‘associated station’ and the other party in the call
Button matrix	Button matrix is not supported	Define buttons and attach an action on push and/or release
Journal (onscreen log)	Journal (onscreen log)	List logging information on the screen; provides a simple IM system between operators
Multiple Tile views Tile views support drag and drop of a camera from the device list Tile view supports cameras, maps and image files	Single Tile view Tile view does not support drag and drop Script command ViewCamera is not supported Tile view does not support an Image file	A tile view can show a video stream, a map or an image. A map is shown including device icons, but does not react to mouse clicks A tile can be assigned to a call request list or an Established call pane to show video related to the call request or call
Grid view	Grid view is not supported	A grid view is a grid of tiles. A grid view supports video streams and maps. A grid view does not have an ID and can therefore not be used as destination in scripts
Hide own station on a map	Hide own station on a map is not supported	Hide the associated station from the map it is placed in

Enhanced license	Basic license	Description
Associate up to 4 cameras with an intercom station	Associate up to 4 cameras with an intercom station	All cameras are shown in a tile view when the station is in call request or call mode
Activity log, in which it is possible to log all call and call request activity	Activity log, in which it is possible to log all call and call request activity	All call requests and calls are stored in the log; The log is part of the SQL database file
System log, in which system errors and customized information is logged	System log, in which system errors and customized information is logged	System logs and custom logs are stored in this log; the log is part of the SQL database
Export of System log and Activity log	Export of System log and Activity log is not supported	Export the logged information to a .csv-file
Link maps in a hierarchical chain	Link maps in a hierarchical chain is not supported	Make overview maps and define areas which are linked to underlying maps; A defined area becomes active when an underlying map has an active icon
Zoning, define devices and maps as part of a zone, and assign zones to profiles	Zoning is not supported; note that in configuration mode it is possible to define zones, but it is not possible to assign a zone to a user profile	Define which maps and devices are visible to an operator
Tabbed pages	Tabbed pages	Defined a tabbed panel, and place tabbed pages in the panel; a way of saving desktop space
Change the active layout from a script	Change the active layout from a script is not supported; in operational mode, the function SelectLayout will not be called and an error will be raised in the System log	Change the layout which is visible on the fly; useful in emergency circumstances when the layout should have as few elements as possible for easy operation

Enhanced license	Basic license	Description
Scripting: • Scripts sets • Filter sets • Button matrices • Tabbed pages: on selection and deselection of a tabbed page	Scripting: • Scripts sets • Filter sets	Use scripts to customize and extend the functionality of Vingtor-Stentofon Operator
Close form on no activity	Close form on no activity is not supported	Close and open a form automatically on map, call request, call or camera becoming active, see paragraph 3.9.3 for details
Display pane		The display panel allows displays with a programmable number of lines to be defined
Snapshot functionality from video streams is supported in • Established call pane • Active call request pane • Tile view • Grid view		Possibility to take a snapshot of the video view, which will be stored as a jpg-file

I Languages

Zenitel can provide you with the tools to translate the Vingtor-Stentofon Operator Client GUI text - please send a mail to CustomerService@zenitel.com. Once done, Zenitel will convert the translated text into a resource file. The resource file will be available to all Zenitel customers.

As from the next release of Vingtor-Stentofon Operator, the resource file will be part of the Vingtor-Stentofon Operator installation package. Before a release, the resource file can be manually added to the Vingtor-Stentofon Operator Client installation directory, by default:

C:\Program Files (x86)\Vingtor Stentofon\Vingtor-Stentofon Operator\Vingtor-Stentofon Operator Client.

The new language will be available for selection as soon as Vingtor-Stentofon Operator Client is restarted.

The WEEE Directive does not legislate that Zenitel, as a 'producer', shall collect 'end of life' WEEE.

This 'end of life' WEEE should be recycled appropriately by the owner who should use proper treatment and recycling measures. It should not be disposed to landfill.

Many electrical items that we throw away can be repaired or recycled. Recycling items helps to save our natural finite resources and also reduces the environmental and health risks associated with sending electrical goods to landfill.

Under the WEEE Regulations, all new electrical goods should now be marked with the crossed-out wheeled bin symbol shown.

Goods are marked with this symbol to show that they were produced after 13th August 2005, and should be disposed of separately from normal household waste so that they can be recycled.