Carmona OCM - the software for automatic notifications or announcements by means of SIP phone lines

Table of contents.

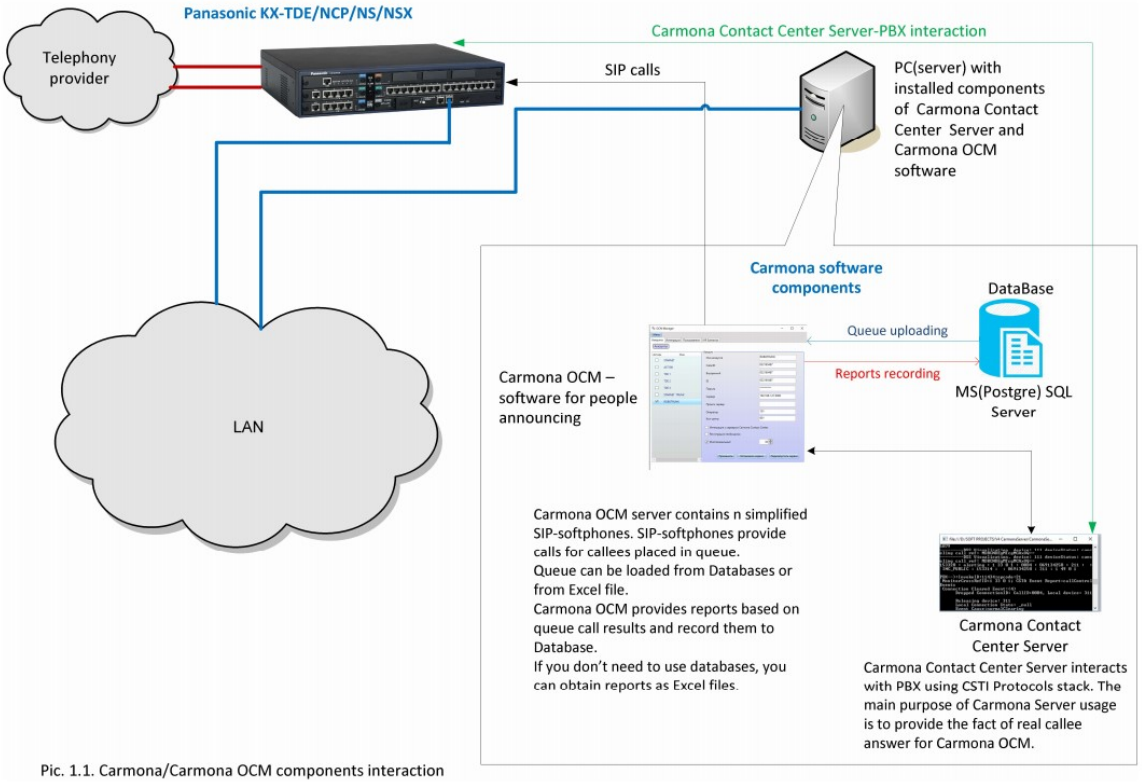
1. Intro

Carmona OCM (Carmona OCM DC) is the software the main purpose of which is sending notifications or announcements by means of phone lines. Carmona OCM uses SIP technology for announcing so it's required that you have some SIP accounts at your provider(s) in order to use the software.
Phone numbers of people you intend to announce should be placed into the queue as items with certain rules of call processing.
Carmona OCM components are:

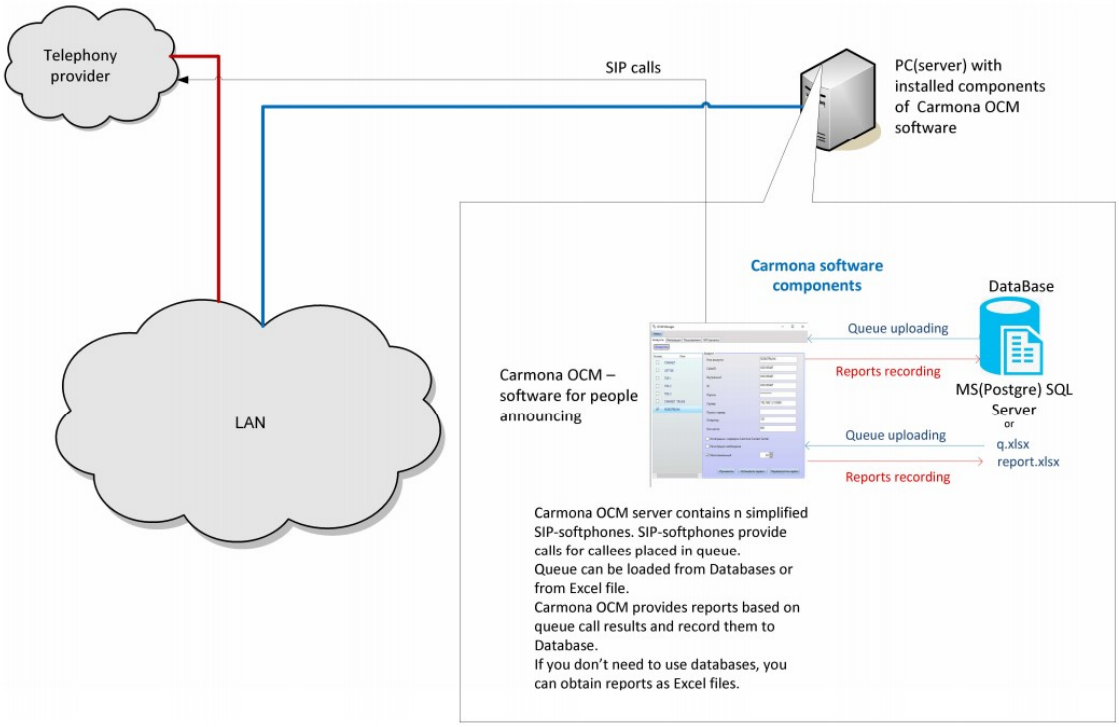
1. CarmonaOCMsvc(Carmona OCM server in future) – service running on server machine, the heart of the system.
2. OCM Manager – tool to configure system settings.
3. OCM Monitor – tool for queues control and queue progress monitoring.
4. MS SQL Server or PostgreSQL Server for entries and results storage – optional.

If you use Carmona OCM software behind Panasonic PBX (the latter looks as SIP provider from Carmona OCM point of view) you can use Carmona Contact Center Server (Carmona server in future) in order to start announcement correctly at moment when called party actually answers the call. If you are not using the Carmona server, then you need to purchase a license for the Carmona OCM DC product, which allows you to connect directly to the Panasonic PBX (for NS/NSX models, additional licenses for the PBX itself may be required).

You can see the interactions schemas of components on pictures below.



Pic. 1.1. Carmona/Carmona OCM components interaction



Pic. 1.2.Carmona OCM components interaction

Number of simultaneous calls is defined by purchased license (in case of Carmona server usage the license of the Carmona server should include this number too) and provided (by carrier) number of SIP accounts.

Minimum PC Requirements for Carmona OCM server.

Operation System: Windows 7, Windows 8.x, Windows 10 /Windows Server 2008 and higher.
64-bit OSs are preferable
Preinstalled framework: dotNet 4.6.1 or higher
RAM: 8 Gb and more
Hard Drive: 500 GB and more
Network Interface Card: 100Mbit
In case of Amazon Polly Text-To-Speech service usage – credentials for this service (AWSAccessKey, AWSSecretKey, provided in rootkey.csv file)

CPU:
Number of SIP Accounts <=10:
Intel i3-540 equivalent or greater.

Number of SIP Accounts > 10 and <= 50:
Intel i5-4590 equivalent or greater.

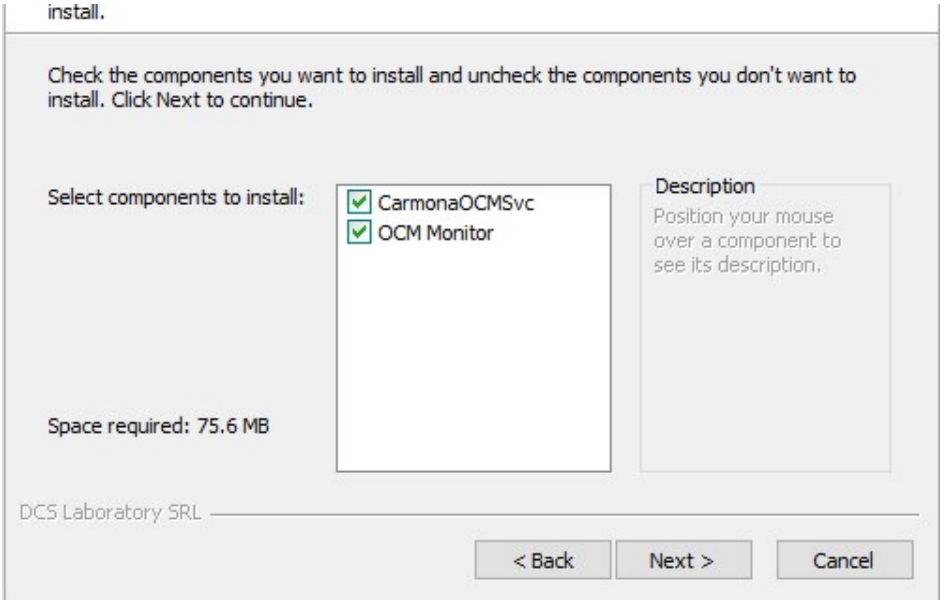
Number of SIP Accounts >50 and <=100:
Intel i7-8550U equivalent or greater.

2. Installation

To start the installation, run Carmona OCM 2.1.7.exe.

Follow the prompts of the installer.

On components selection page, select required components:



CarmonaOCMsvc component installs CarmonaOCMsvc service and OCM Manager application.

OCM Monitor component installs OCM Monitor application.

After installation is completed, you can run OCM Manager tool to configure your Carmona OCM system, if CarmonaOCMsvc component has been selected earlier.

Licensing

Perform following steps to apply the license that you purchased:

Stop CarmonaOCMsvc in Services of Computer Management window.

Navigate cursor on CarmonaOCMsvc and right-click on Properties.

Enter the activation key into Start Parameters field.

Ensure you have Internet connection on this machine.

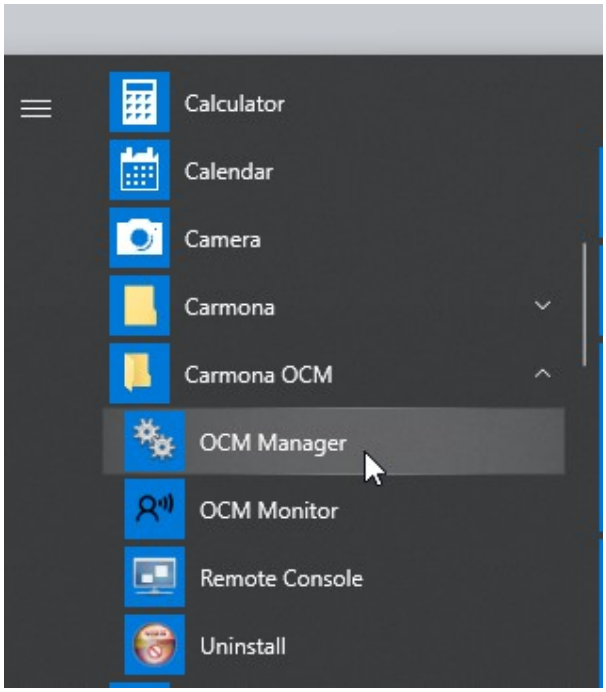
Press Start button in this window.

3. Configure Carmona OCM system

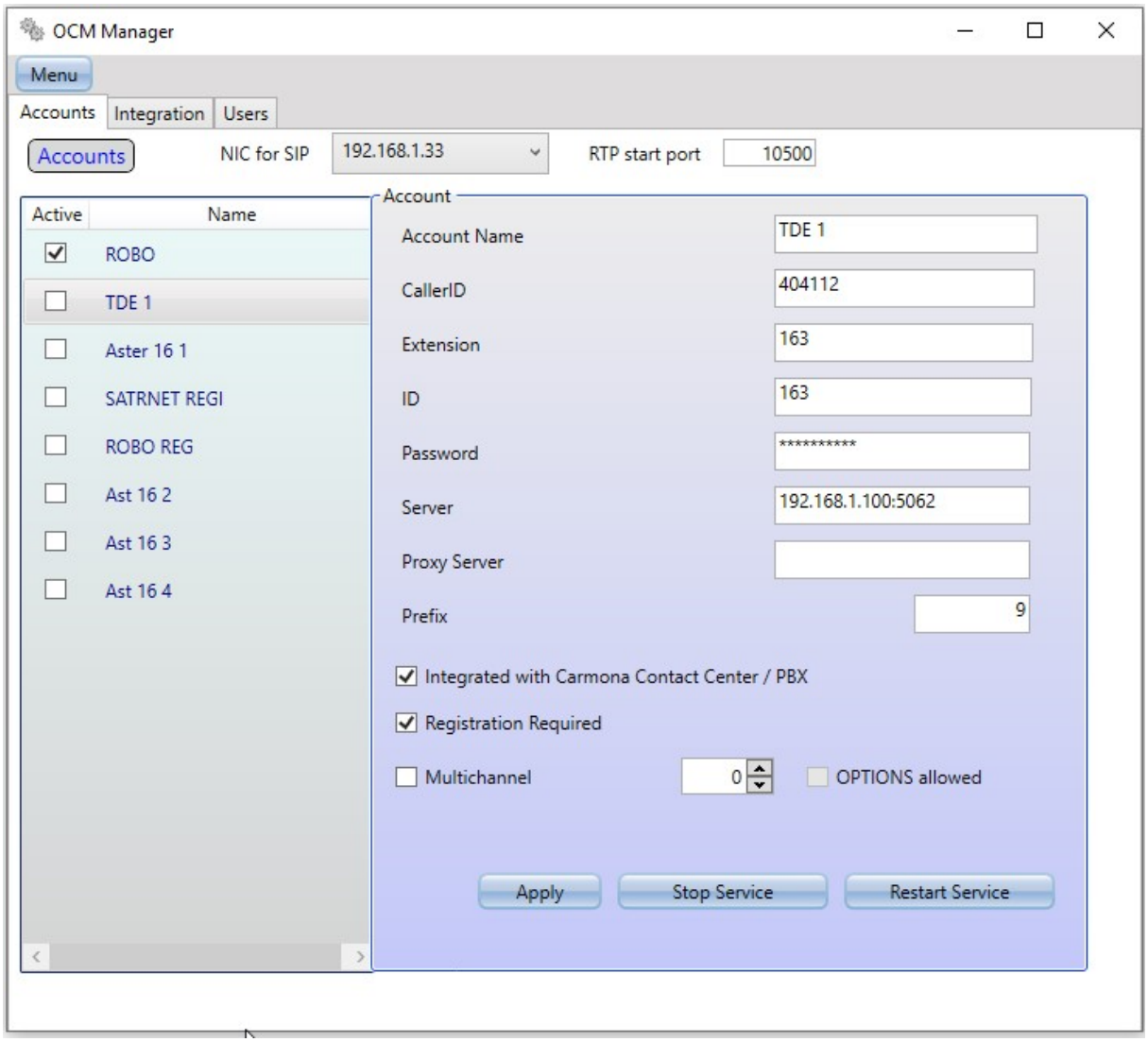
Configuration of Carmona OCM system consists of:

- Configuration of SIP accounts.
- Configuration of interaction parameters for OCM Monitor, Carmona Server, Text-To-Speech Engines, Databases in case any of these components is used/installed.
- Configuration of profiles for OCM Monitor users.

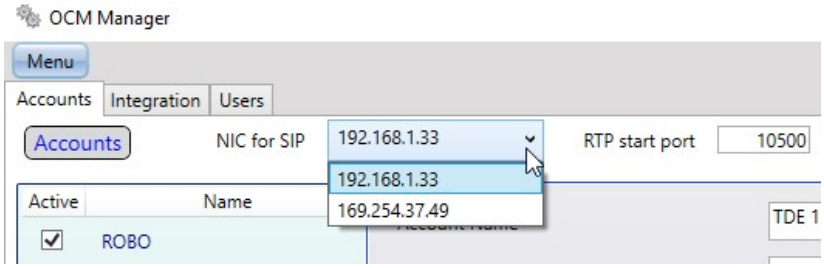
To start the configuration, run OCM Manager



OCM Manager will open with the following interface:



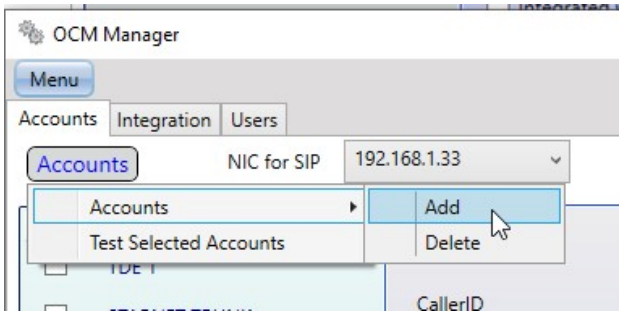
If computer when Carmona OCM is installed has more than one network interface you can define the required for SIP connection interface here:



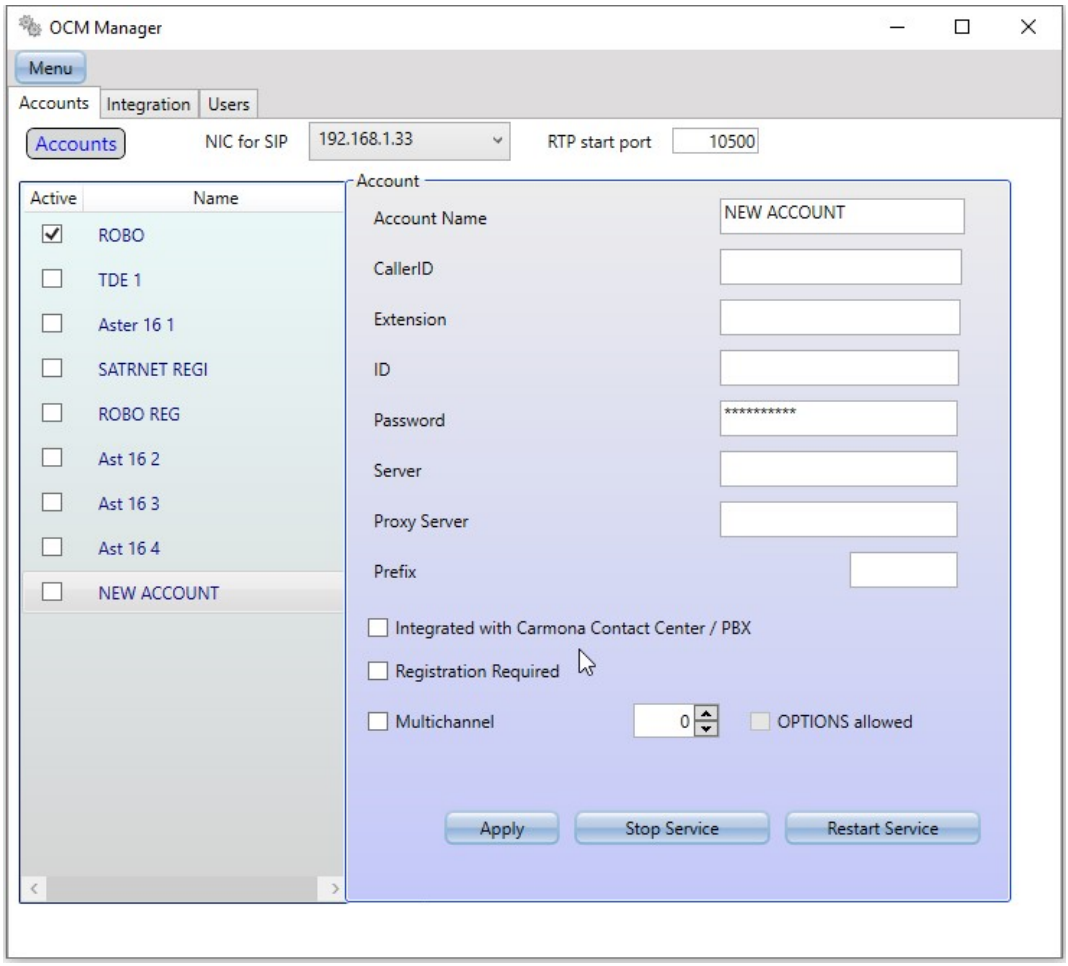
Also you can define the initial port (listen) for RTP:

3.1. Configure SIP accounts

Press **[Accounts]** button, then press *Accounts*, then press *Add*.



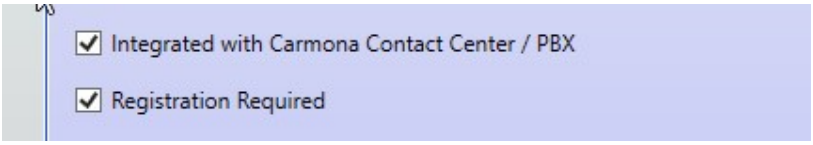
Now you can add SIP account parameters:



Fill in all required fields on this form. If your SIP server uses port other than 5060, specify it explicitly, for example: sipmaster.com:5088, 192.168.1.45:5084

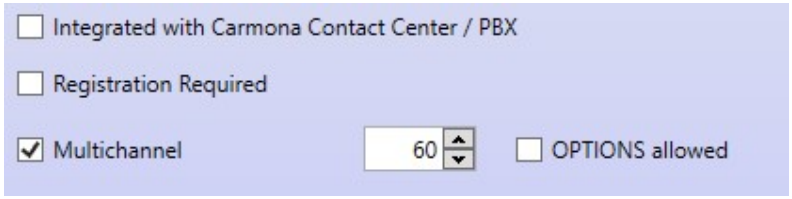
Each account can use its own prefix. If items of queue also use prefixes, prefixes will be inserted in the following order: Account prefix + Item prefix + Item phone number

If you use Panasonic PBX SIP-EXTs, enable following options:

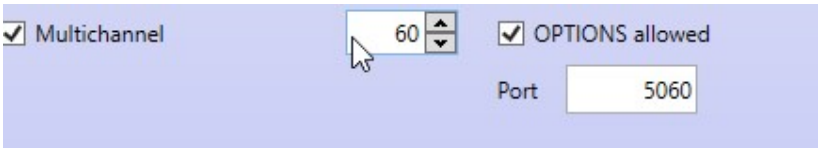


and set the CarmonaCC/PBX server connection parameters on **Integration** tab(see below).

If your carrier provides accounts without registration you can use Multichannel mode for base account. Disable *Registration Required* option, enable *Multichannel* option and set the number of additional channels:



You can allow OPTIONS and define the port for basic channel, on which it will be listen OPTIONS messages. In this case basic channel will be activated on application/service start.



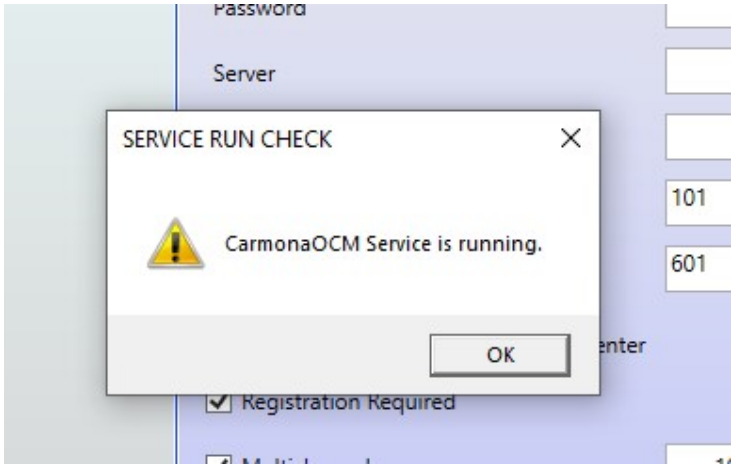
Create all required SIP accounts this way.

For each created account, mark it as *Active* if you are going to use the account. Otherwise it will not be used by system in queue progress.

Don’t forget to press **[Apply]** button when changes are made.

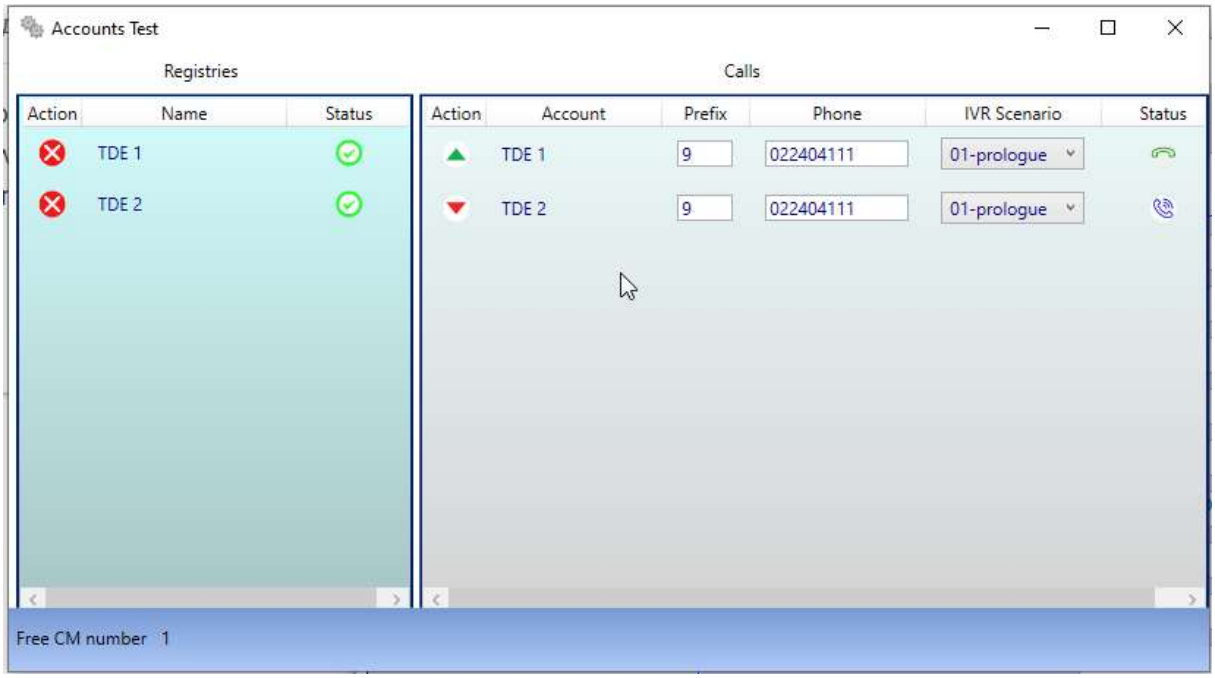
You can run the accounts test for all accounts marked as *Active*. Press **[Accounts]** button, then *Test Selected Account*.

The accounts test can be performed only if CarmonaOCMsvc is not running. You will get the warning if it is running.



Return to OCM Manager and press **[Stop Service]** button. Now you are able to perform accounts test. After you are completed the test, press **[Restart Service]** button.

In the accounts test window, you have to enter the real phone number, prefix and select scenario name. If there are no scenarios created yet, only the “default” scenario can be used.



Click green triangle ▲ in *Action* column to start a call. To end the call click ▼ red triangle.

Column *Status* of *Call* group reflects the call state:

- idle(on hook)
- ringing
- talk

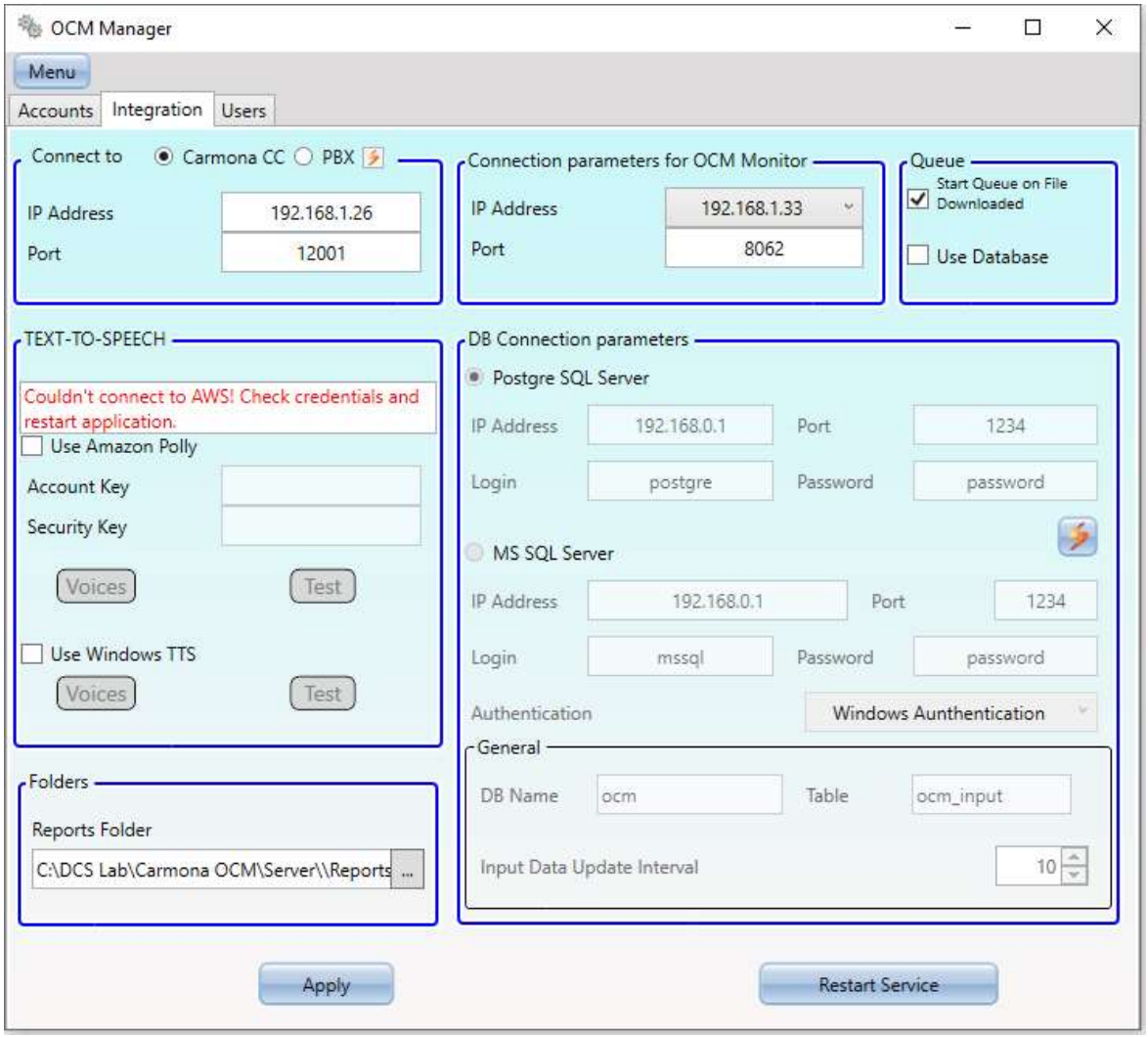
Column *Status* of *Registries* group reflects the registration status:

- unregistered
- registering
- registered

Don’t forget to Restart Service after you completed the tests.

3.2. Configure integration parameters

To configure integration parameters, click on **Integration** tab.



Parameter groups description:

Connect to Carmona CC / PBX parameters.

Parameters of this group are used for connection to Carmona Contact Center Server or directly to PBX. You have to set parameters if you are using SIP accounts on Panasonic PBX.

- IP Address – IP address used by the server where Carmona Contact Center Server is installed or PBX main IP-address.
- Port – port which Carmona Contact Center Server uses or PBX CTI port.

If you select the direct connection to PBX, you can test connection pressing button ⚡

Direct connection to PBX is available only if you've bought the Carmona OCM DC license.

Connection parameters for OCM Monitor

Parameters of this group are used to allow OCM Monitor applications to be connected to Carmona OCM Server,

- Server IP – IP address of the machine which Carmona OCM Server should use.
- Port – port which Carmona OCM Server should use.

  This port should be allowed in the firewall rules for incoming connections.

Queue

Parameters of this group are used to allow queue to start automatically when queue is uploaded(from any source) and use databases for queue processing.

- Start Queue on File Downloaded – check this field to make system start queue immediately when queue is uploaded, no matter whether is it uploaded from file or from database. If this field is not checked, the one way to start queue is via OCM Monitor.
- Use Database – check this field if it's required to use database for queue uploading. Database tables are described below. If this field is not checked, queue could be loaded from .xlsx or .csv files. File contexts are described below.

TEXT-TO-SPEECH

Parameters of this group are used to allow text-to-speech engines in some scenarios. You can use Amazon Polly TTS service or Microsoft TTS engine(and Google TTS service in future).

- Use Amazon Polly – check this field if you are going to use Amazon Polly TTS service. If you check this field you have to enter AWS Account Key and AWS Secret Key, which you get after creating AWS account.
- Use Windows TTS - check this field if you are going to use Microsoft TTS engine.

Checking the fields provides all available voice lists in both cases. For test of voices press **[Voices]** button, select the voices and then press **[Test]** button. After that you can test the voices or make .wav files in **Voice Adjustment** window.

Uncheck both fields if are not going to use any TTS engines.

Folders
You can define the Reports folder here.

DB Connection parameters

Parameters of this group are used to set database connection parameters in case you are going to use databases for queue processing.

- PostgreSQL Server – check this field if you are going to use PostgreSQL Server.
- MS SQL Server – check this field if you are going to use MS SQL Server.

Enter required parameters in the fields.

In the *General* parameters group enter the required parameters. These parameters are applicable for both types of database servers.

DB Name - database name that will be used for work. Default value: ocm

Table – name of the table which should be used for queue items input. Default value: ocm_input.

Input Data Update Interval – interval in seconds, which will be used for new or append queue items lookup in database.

⚡ - use this button to check database connection(don't forget to first apply the changes).

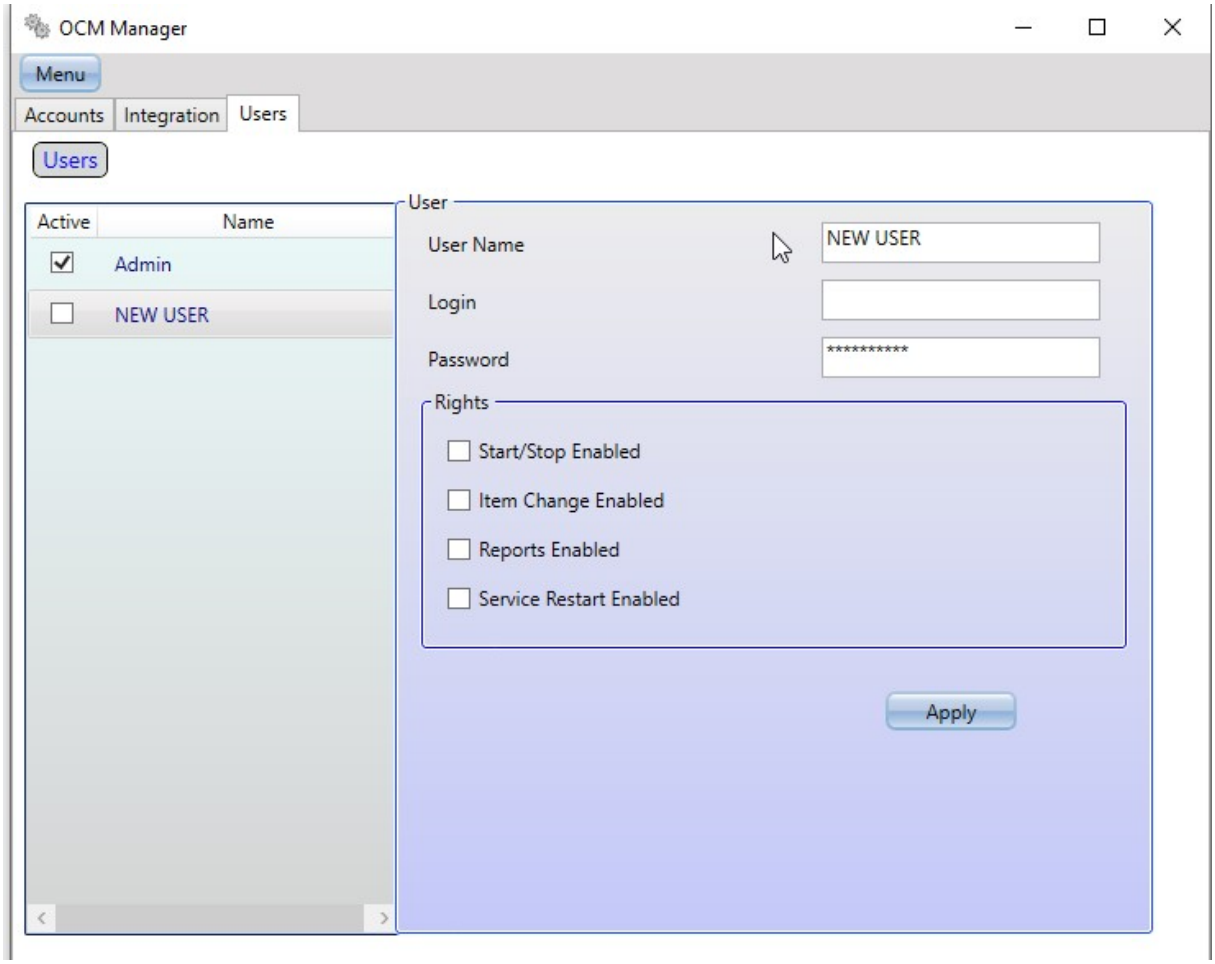
Table Naming Restriction: queue result report should be recorded to a table with name equal to input table name, but with "input" replaced by "report". Example: input table name is "org_ocm_input", the report table name should be "org_ocm_report". The input table name should contain "input" anyway. Please pay attention to this while designing your database!

All required CREATE scripts for both database types with default values are presented in the Appendix 3.

Press **[Apply]** button to make you changes effective. All parameters entered on this tab require service restart – use **[Restart Service]** button.

3.3. Configure User profiles

To configure user profiles select **Users** tab.



Press **[Users]** button, then press *Add* to create new user profile.

Fill in all required fields. Then press **[Apply]** button.

Parameters entered on this tab do not require service restart.

After changes are made, OCM Monitor application users are able to connect to Carmona OCM server with saved login/password. Their possible actions are defined by Rights (permissions) parameters.

4. Queues – compose and run.

The main object of Carmona OCM work is a queue, or a list of items containing phone numbers, which should be called and the set of rules describing how to do it.

Each item Contains the following parameters of announce process:

- Called person name
- Phone number(s) - that's what CarmonaOCM will use to dial. There can be several phone numbers for one queue item.
- Item priority – priority of the call.
- Call ringing duration – duration of the call if the called person doesn't answer.
- Number of call attempts before notification is considered as unsuccessful
- Pause time before next call attempt
- Scenario – messages and actions executed when the called person answers. See Scenario sub-chapter.
- Start time - time when item should be started
- Record required – whether the answer would be recorded or not.
- Confirmation – allow the called person to confirm, by pressing the confirmation button (see button sequence), that he (she) has listened the message.
- Minimum listened percent – minimum listened time (in percent of total message duration) for the call to be considered as successful. Not applicable if confirmation used.
- Prefix – this parameter is applicable if CarmonaOCM is connected to Panasonic PBX and means the prefix of CO Line access.

Queue can be created in following ways.

- Create items in Excel file(xlsx). See Appendix 1 for detailed files description.
- Create items in character-separated values file (csv). See Appendix 1 for detailed file description.
- Create items in database. See Appendix 2 for detailed database tables description.
- Create items with OCM Monitor application (see OCM Monitor Usage chapter).

After queue is composed it has to be uploaded to the Carmona OCM for execution.

Queue can be uploaded by following ways:

- Copy Excel file to the specified folder with the specified file name (folder: YourInstallationFolder\Server\Queue(default: C:\DCS Lab\Carmona OCM\Server\Queue), file name q.xlsx).
- Copy character-separated values file to the specified folder with the specified file name(folder: YourInstallationFolder\Server\Queue(default: C:\DCS Lab\Carmona OCM\Server\Queue), file name q.csv).
- Read input table from database.
- Upload queue by OCM Monitor application (see OCM Monitor Usage chapter).

After queue is uploaded to the system it should be started for execution.

Queue can be started in following ways:

- Automatic start
- Manual start by OCM Monitor application (see OCM Monitor Usage chapter).

Automatic start is available if Integration->Queue->Start Automatically On File Downloaded parameter is enabled in configuration.

Automatic start occurs when:

- File q.xlsx in folder YourInstallationFolder\Server\Queue (default: C:\DCS Lab\Carmona OCM\Server\Queue) is created or changed.
- File q.csv in folder YourInstallationFolder\Server\Queue (default: C:\DCS Lab\Carmona OCM\Server\Queue) is created or changed.
- Database input table contains records with State column value = 1. See Appendix 2.
- Queue is uploaded via OCM Monitor application (see OCM Monitor Usage chapter).

Each start of new queue stops and clears the previous running queue, if latter is present.

4.1. Scenario

The mandatory element of each queue item is Scenario.

Scenario is an object, describing the messages that have to be played to called person, the actions defined by buttons pressed by called person, and the order of execution of those messages and actions.

All scenario configurations are presented in XML files which system uses during queue execution.

All scenario files have .scn extension and are located in ...\Server\Scenarios folder.

Scenarios can be created in following ways:

- With OCM Monitor applications (see OCM Monitor Usage chapter).
- Simple (play message, then wait confirmation button) scenarios can be created automatically during queue uploading.

For simple scenario creation it's enough to put the next data into the queue files:

name of the message sound file with extension wav ("greetings.wav", for example) into column "SCENARIO"

confirmation button symbol ("#", for example) into column CONFIRMBUTTON, if confirmation is required.

or the same data into database input table fields "Scenario" and "ConfirmButton" correspondingly, if database used.

Specified message sound files should be in ...\Server\Sounds prior to uploading the queue.

During queue uploading the system will try to find specified sound file and create scenario file named the same as sound file, and place it to ...\Server\Scenarios folder. If file doesn't exist or its format is incorrect, the system will create the item, but will mark it as incorrect.

Of course, you can put an existing scenario file name to "SCENARIO"("Scenario") column/field. Do not use extension in this case! For example, put "greetings", if you already have scenario "greetings.scn" in ...\Server\Scenarios folder.

If you use advanced scenarios, do not use extension for scenario name data in the queue files or database tables.

4.2. Reports

If you don't use databases, reports will be created as Excel files each time the queue is completed. Report files are located in the Reports folder which you've defined in the **Integration->Folders**.

Also, if connected user (OCM Monitor application) has permission for Reports, the report file will be send to the user PC. User can find it in the directory of OCM Monitor installation, in folder /Reports.

If you use databases, reports will be stored in database report table each time the queue is completed. You can use your own Report Builder for report visualization.

See Appendix 1, A1.3 Report file description and Appendix 2 A2.2. Report table description.

5. OCM Monitor

OCM Monitor is the application for queue creation, control and monitoring, as well as for scenario creation and editing.
For queue control and monitoring, OCM monitor should be connected to Carmona OCM Server.

Starting to describe OCM Monitor, we need to take a look at system file synchronization. As described in “4. Queue-compose and run” chapter, the queues use scenarios for the called persons notifying. Some scenarios can use sound files for notifying the called persons, others can use text-to-speech engines.
Carmona OCM server stores all required files in its own Scenarios/Sounds folders.
Each OCM Monitor stores the copies of server files in its own folders.
OCM Monitor can add or edit scenarios and sound files. Each time when scenario is added or edited, OCM Monitor starts file synchronization in order to update files on server and on other OCM Monitors connected to system.

If OCM Monitor is not connected to Carmona OCM Server at the moment when scenario or sound files are created or edited, those created or edited files will be stored and synchronization will start at next connection.

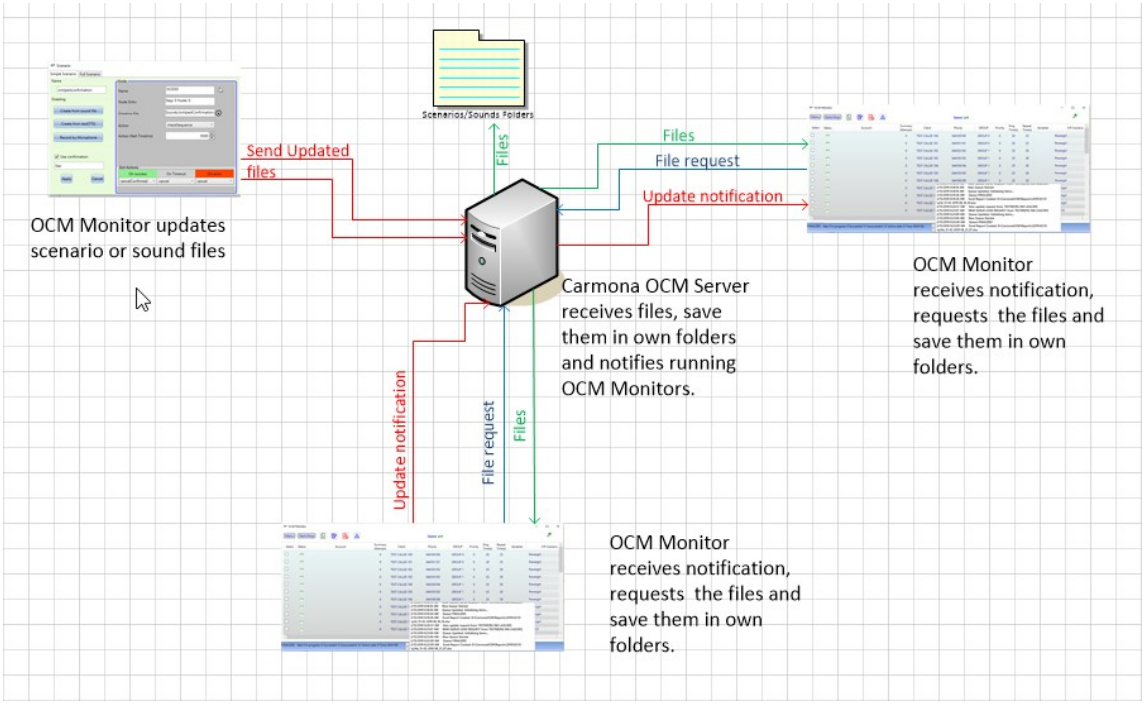


Fig. 5.1. In system file synchronization.

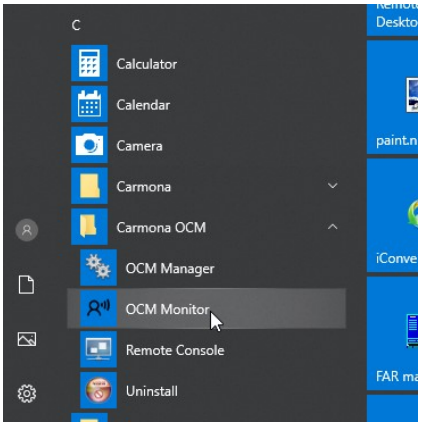
After scenario or sounds files changes are applied, all changes applied to current running queue items for the next call, if the items use these scenarios/sound files.

So, file synchronization allows you to prepare the scenarios and queues without connecting to the server.

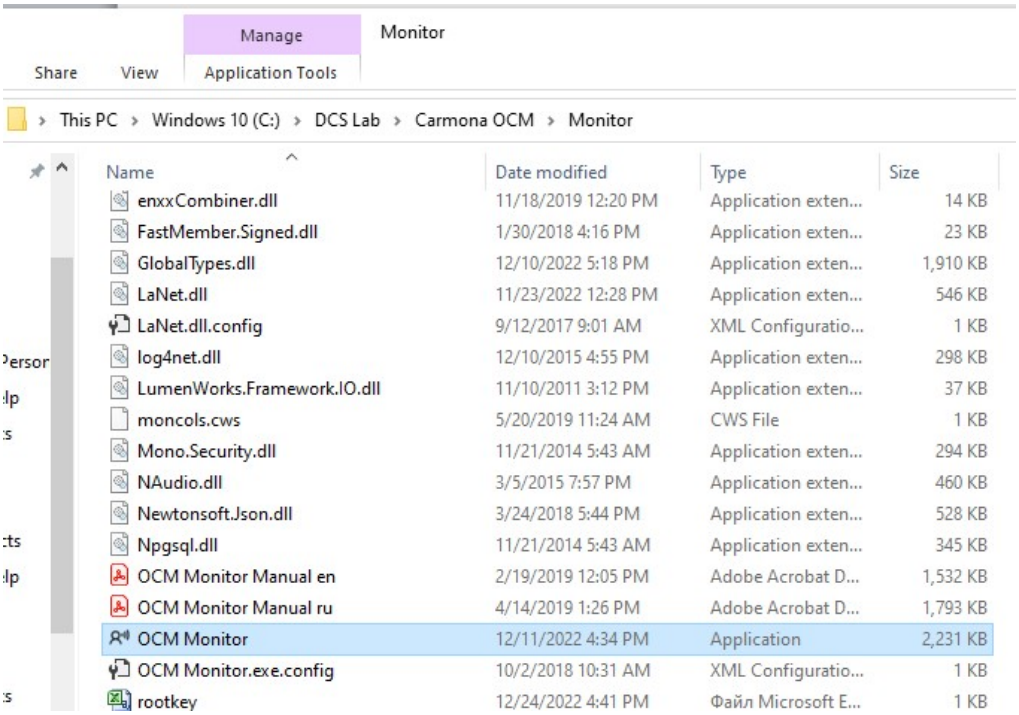
5.1. OCM Monitor usage

To open OCM Monitor, double-click on the shortcut on the desktop

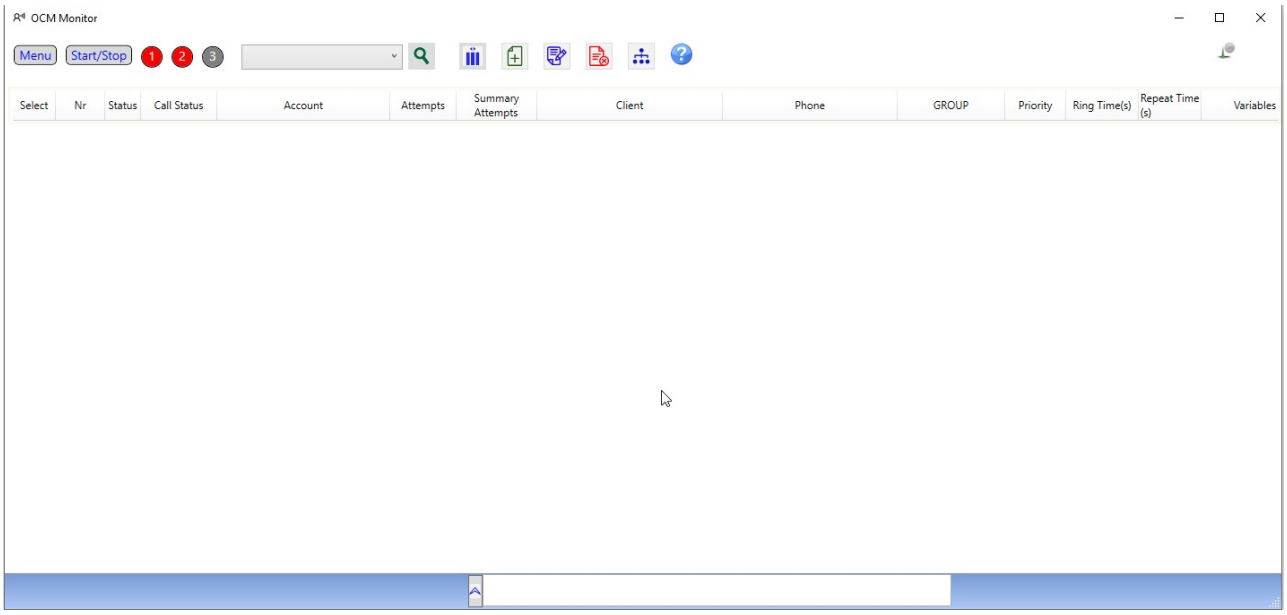
or start it from Program Menu.



or start application from folder directly:

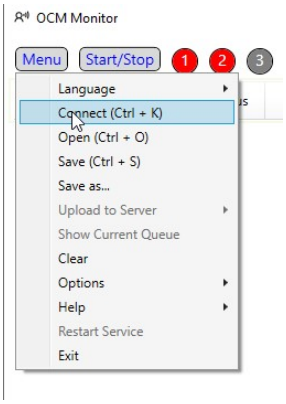


OCM Monitor opens with following interface:

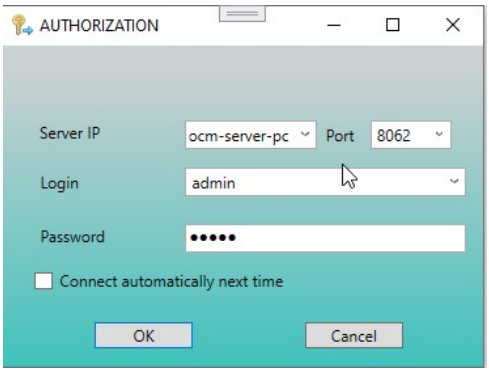


5.1.1. Connecting to Carmona OCM Server

In order to connect to Carmona OCM Server, press **[Menu]** button, then select *Connect* (or press Ctrl + K).



The AUTHORIZATION window will be opened.

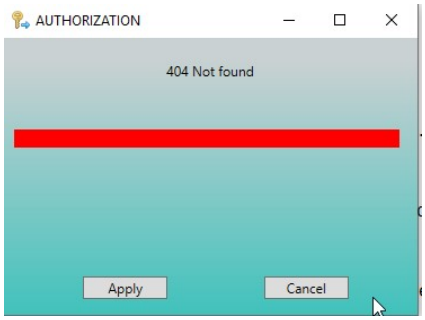


Enter server IP or host name, port number (default 8022), login and password. Enable option *Connect automatically next time*, if you prefer to skip the authorization next time.

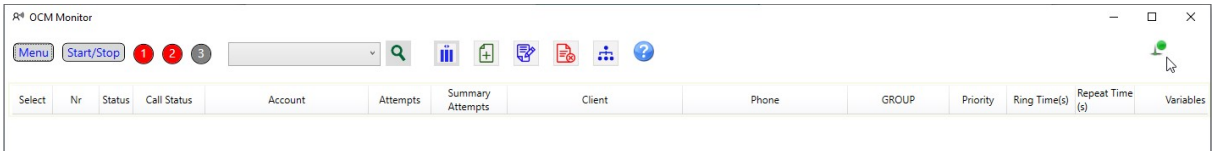
If authorization succeeded, connection parameters will be stored and can be use next time, except the password.

If option *Connect automatically next time* is enabled, next time you will be connected without this window showing.

If connection attempt failed you will be informed about the reason of failure:



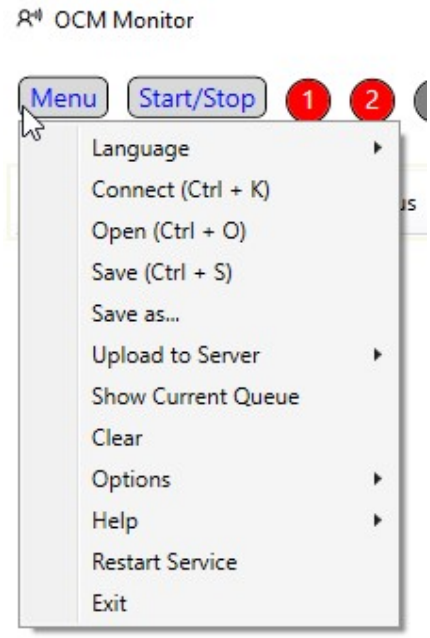
Connection state is indicated by green connection state icon in the right upper corner of interface. Connection events are logged in the log frame in the central lower part of the screen.



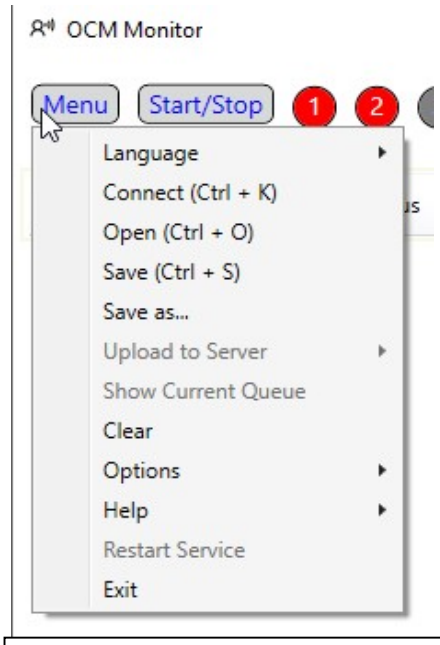
WARNING! In current version the only one user can be connected at time.

5.1.2. Menu items

After you pressed **[Menu]** button, the following items are available:



When you are connected to server



When you are not connected to server

Items description:

Language – allows you to select the interface language(English, Russian and Romanian are available at the moment).

Connect – closes current connection and opens AUTHORIZATION window.

Open – runs “Open file” dialog and creates the queue from the selected Excel or csv file.

Save – saves opened file.

Save as – allows you to save opened file with another name or path.

Upload to server – allows you to upload the queue from selected in “Open file” dialog file to server. The queue items can replace current queue running on server, or be appended to it, and updated queue will start immediately.

Show current queue – displays running queue items on the screen. If queue is already finalized, snapshot of the last queue will be displayed.

Clear – if queue is not running, you can clear displayed queue items.

Options – set the required option value here.

- Insert leading ‘0’ – fixes the problem when Excel casts text field with phone number with leading zero to numeric field without leading zero. If option is set, leading zero will be inserted to phone numbers, if they not start with zero.
- Use ffmpeg convert to wav – allows you to use sound files with formats different from wav, PCM-mono, 16 bit, 8000. The ffmpeg library will convert those files to required format online. Available formats: "wav", "wma", "mp3", "ogg", "oga", "mogg", "aac", "aif", "cda", "m4a", "m4b", "opus", "gsm", "flac", "weba", "sln".
- Strategy – allows you to define how the queue will pass through the items.
  Maximum reach – all items will pass through down to the last element, and only then repeat timers will be checked.
  the Priority of repeat timer – the queue will return to upper item and start the next attempt when item’s repeat timer expired.
  Maximum reach (reset repeat timer) - all items will pass through down to the last element, and when it will reached, the queue returns to upper item and start the next attempt immediately, even the repeat timer not expired yet.

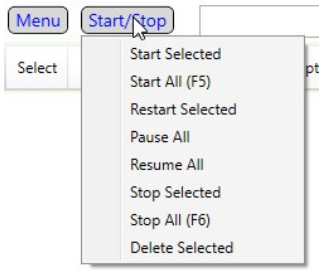
Help - allows you to view user manual, information about the program and visit the website of the company-developer.

Restart service – force restart of service. This option is disabled if user has no permission for this action.

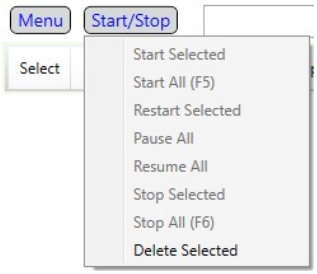
Exit – closes the OCM Monitor.

5.1.3. Start/Stop Operations

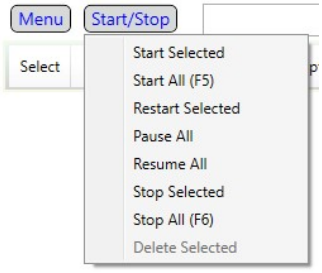
After you pressed **[Start/Stop]** button, the following items are available:



When you are connected and have all permissions



When you are not connected or you have no permission for Start/Stop operations



When you are connected but you have no permission to change the items

Items description:

Start Selected – allows you to start only selected items. If items have been already stopped, they will be started with current attempt counters. If queue is not created yet, the new one will be created for selected items only.

Start all – allows you to start all items. The new queue will be created and run.

Restart selected – allows you to restart selected items even when the queue is already finalized or selected items are stopped. Attempt counters will be set to initial values.

Pause all – allows you to put the queue on hold. Running calls will continue, and will be paused only after they completed.

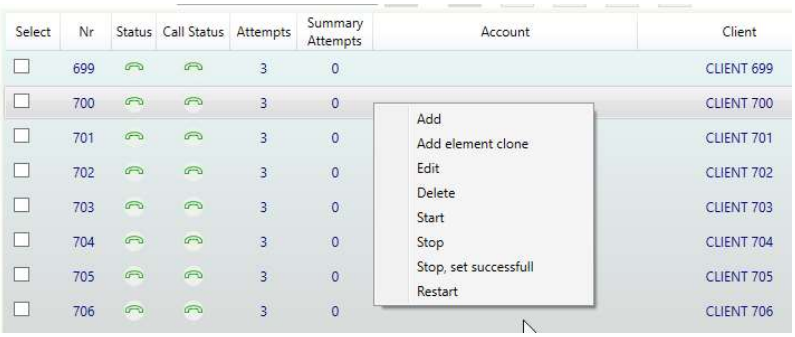
Resume all – allows you to continue paused queue.

Stop selected – allows you to stop selected items. Items will be stopped and removed from the queue, but you can start them later keeping the current attempt counters without restarting the whole queue until queue is not completely finalized.

Stop all – allows you to stop the queue. The queue will be finalized and the snapshot of it will be shown on the screen.

Delete selected – allows you to delete selected items. If queue is running, items will be stopped and removed from queue immediately.

Start, Stop and Restart operations for selected item also are available in item context menu (right-click on selected item).



5.1.4. Icons

The following icons are located in the upper part of interface:

- Search… - allows you to filter items by phone number, client name, or by item statuses.

- Columns Visibility – select columns to show/hide.

- Add item – allows you to add an item. If queue is running, the item will be appended to the end of queue.

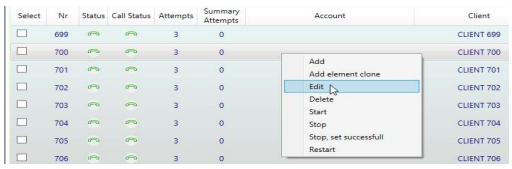
- Edit item – allows you to edit selected item.

- Delete item – allows you do delete selected item(s).

- IVR Scenario – allows you to create the new or edit existing scenario (see Scenario chapter).

- Information – displays some useful information: connection status, license status, last used SIP accounts.

Add, edit and delete operations also available in item context menu (right-click on selected item).



5.1.5. Queue state and events online information

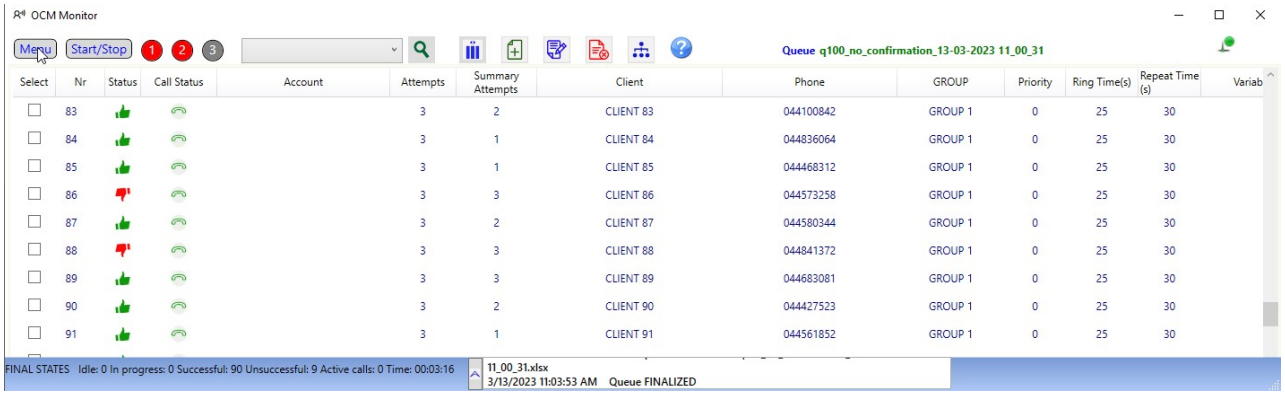
OCM Monitor interface has some info elements.

The queue name info is located in the upper central part of interface.

Connection status indicator is located in the upper right part of interface.

Info panel for current queue state and item result counters is located in the lower left part of interface.

The log frame is located in the lower central part of interface.



Lower left info panel presents the following information:

Queue state – current state of queue: IDLE, IN_PROGRESS, PAUSED, FINAL STATES.

Idle items counter – number of not started items.

In progress items counter – number of items in progress.

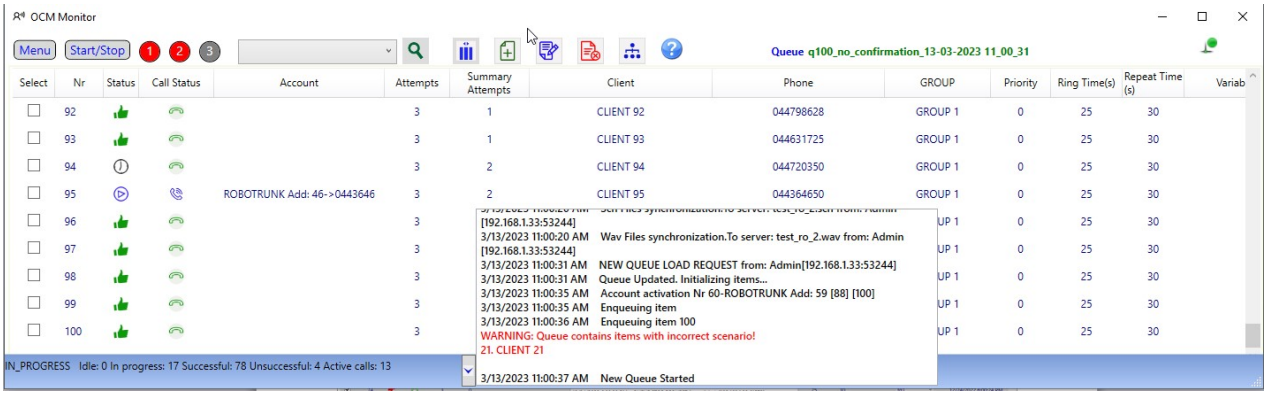
Successful items counter – number of items with successful notification result.

Unsuccessful items counter - number of items with unsuccessful notification result

Finalized queue execution time – time of queue execution. It appears only after the queue is finalized.

Each user action related to the queue or scenario/sound files is shown in the log frame as well as system events.

The log frame can be expanded or collapsed using the arrow button.



5.1.6. Quick start

You can define three queue files for quick start. In the top part of interface three buttons located for this:

To set the queue file for button or clear button setting, point the cursor on button and press right mouse button.

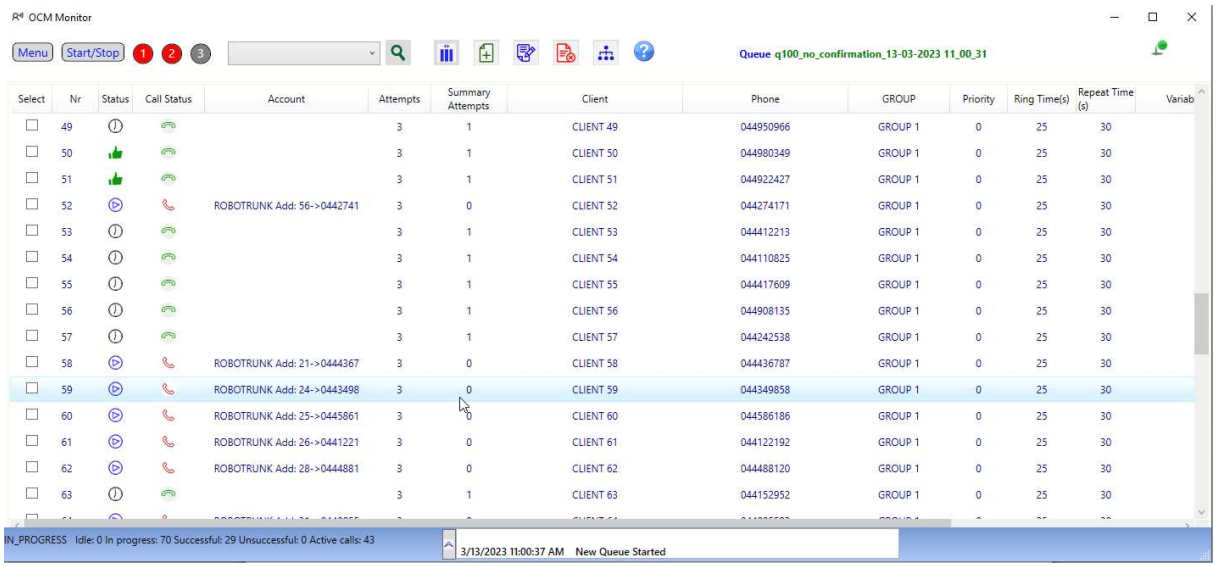
If button is set the file path will show, and the button color is red. Not set buttons are grey.

You can quick start by pressing the button or pressing Ctrl + F1, F2 or F3 correspondingly.

The queue will start immediately if there is no running queue, or you get the question whether to stop the running queue.

5.2. Queue monitoring

You can monitor the running queue item states in the central part of the interface.



Item state is described in the columns:

Select - shows the selected items. Click on column header will inverse the selection. Click with right mouse button will reset selection.

Status - item status, graphically presented by following icons:

 - IDLE, item not started.

 - IN PROGRESS, item person is currently being called.

 - PAUSED, item is waiting for the next attempt.

 - READY, item is ready to make next attempt, but it's waiting for free SIP account.

 - SUCCESSFUL, item person has been notified.

 - UNSUCCESSFUL, item person has not been notified after all attempts.

 - Scenario cannot be executed due to some errors with sound or scenario file(file not found or file format is incorrect, etc).

Account – shows used SIP account when call is being executed.

Summary attempts – shows summary attempts for all item phone numbers.

Client – shows the name of called person.

Phone – shows the item phone number(s).

GROUP – shows the item group name.

Priority – shows the item priority.

Ring Time – shows the maximum duration during which the called person has not yet answered the call.

Repeat Time - shows the time between attempts.

Variables – shows the variables that should be used in the scenario.

IVR Scenario – shows the name of scenario for the item.

Date/Time – shows the time when the item was started.

Attempts – shows how much attempts is left before item will be stopped.

Call Status – item call status, graphically presented by following icons:

 - idle, no calls are being executed

 - called person is being alerting

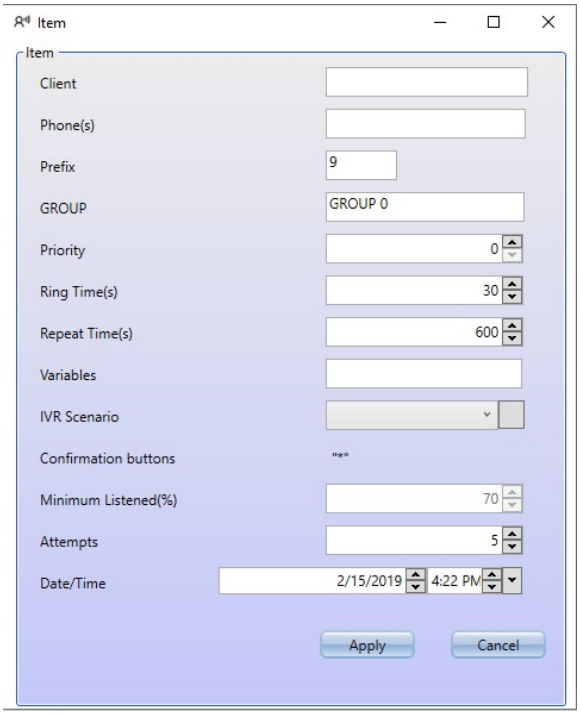
 - called person has answered, and listens the message now

Items updates each time when changes in item or call statuses occurred.

5.3. Create and Edit Items

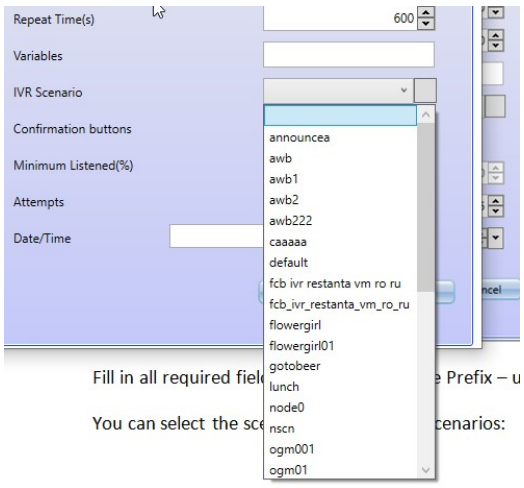
You can create a new item clicking on Add icon or selecting *Add* item from the item context menu.

After you perform one of these operations the following interface will appears:



Fill in all required fields. Be careful with the Prefix – use it only if it's actually needed.

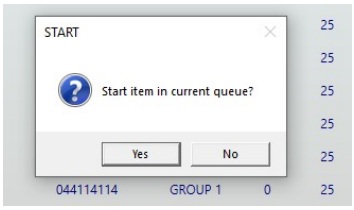
You can select the scenario from existing scenarios:



If the scenario uses confirmation, the *Minimum Listened(%)* option becomes unavailable, and, of course, will not be used by the scenario.

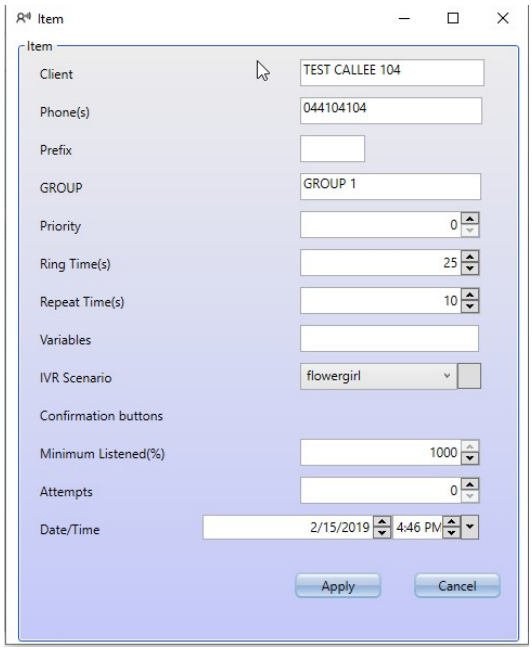
You can go to **Scenario** interface clicking the scenario button:

After all fields are filled in, press **[Apply]** button. You will be prompted to save changes to the file.
If queue is running, you will be prompted to start new item in the running queue:



If you will select **[Yes]**, the created item will be appended to the running queue.

Similarly, you can edit the selected item. The difference is that Item Interface will be opened with selected item data:



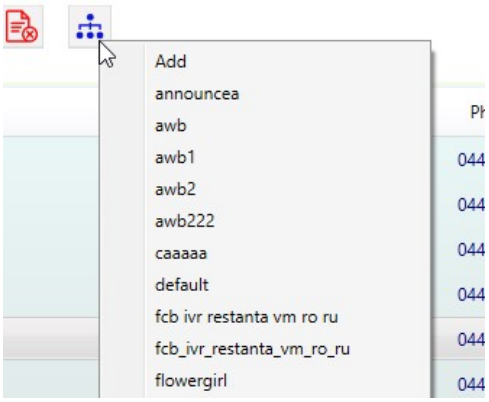
After **[Apply]** button is pressed, you will be prompted to save changes to the file. If queue is running, the changes will be applied when current call is completed.

5.4. Create and edit scenario

Scenario interface can be opened by clicking on IVR Scenario icon or from Item interface using scenario edit button.

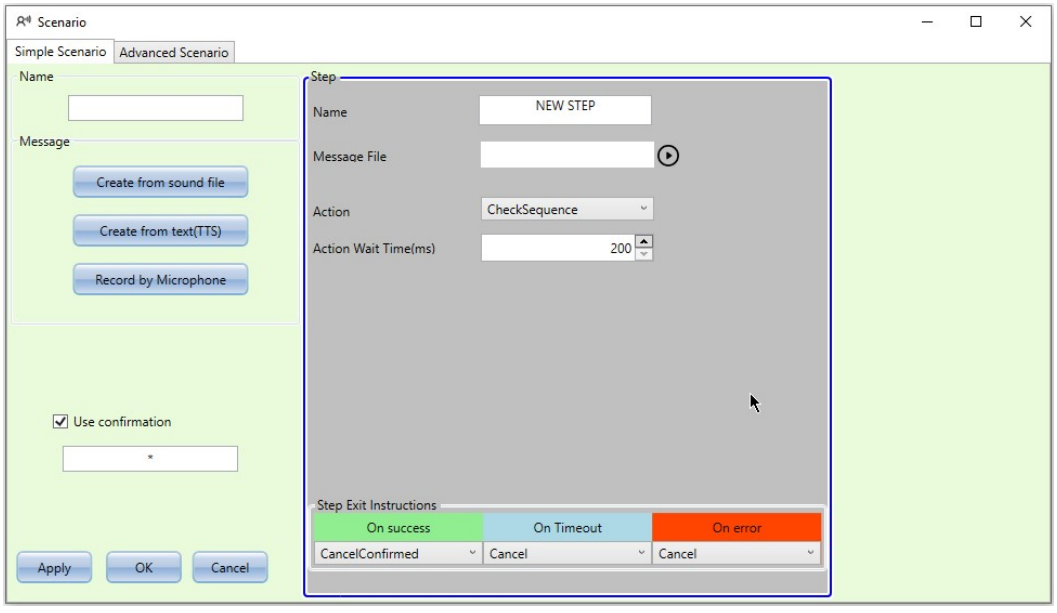
From item interface:

From IVR Scenario icon:



You can select *Add* to create new scenario or edit an existing one.

The **Scenario** interface will be opened:



If you select to edit an existing scenario, all the fields will be filled in with the scenario data.

5.4.1. Simple scenario

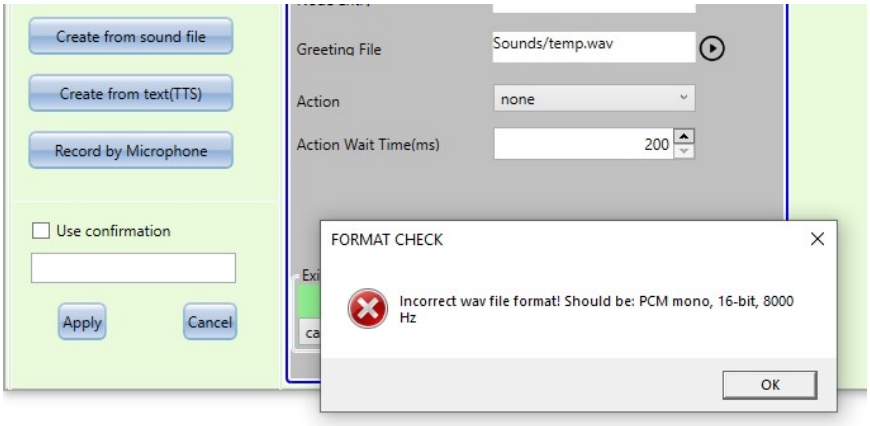
Use **Simple Scenario** tab to create or edit a scenario based on sound (wav) file. Simple scenario uses only one message and only one action, usually the confirmation action.

First you need to define the sound file. You can do it in the following ways:

- Select existing sound file
- Create sound file using text-to-speech engines
- Record sound from microphone

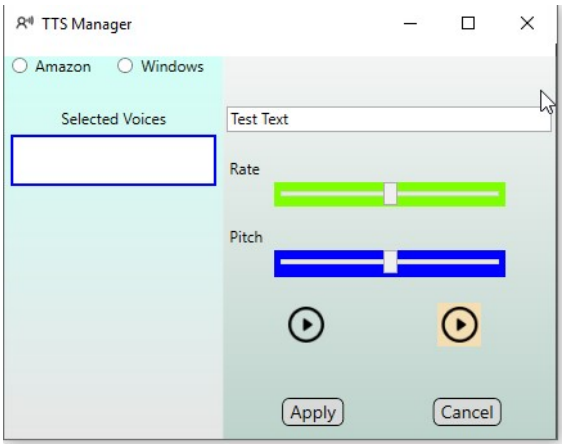
To select an existing file, press **[Create from sound file]** button. You will be prompted to select the file.

If file that you have selected has incorrect format, a warning message will appear:



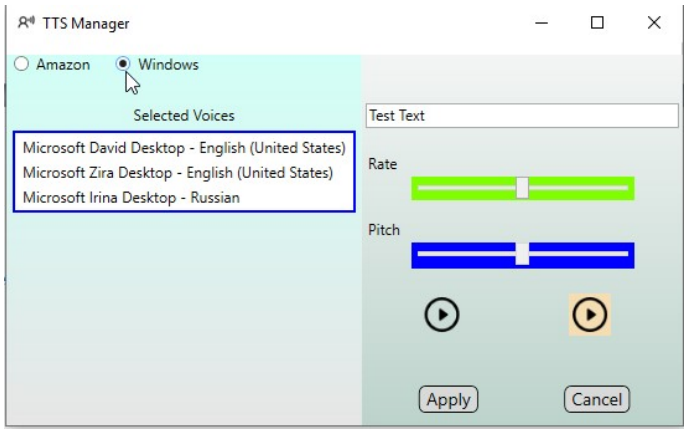
To create sound file using text-to-speech engines, press **[Create from text(TTS)]** button.

TTS Manager interface will be opened:

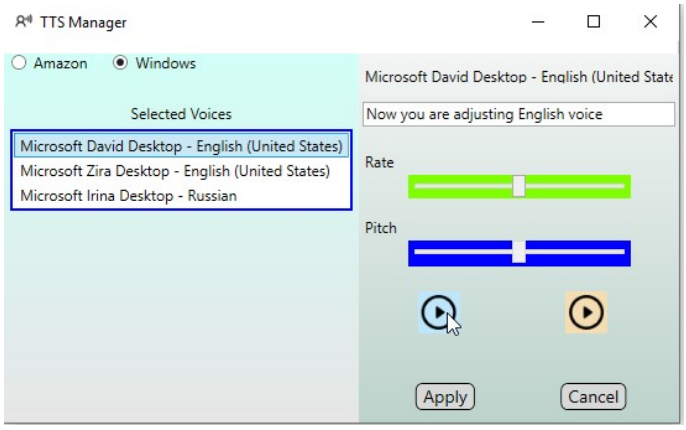


Select engine that you going to use: Amazon Polly or Microsoft TTS engine.

The voice list will be published.

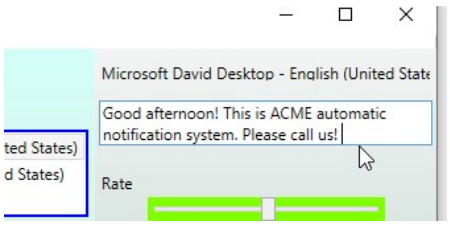


Select the voice and test it.



If you use Windows (Microsoft) TTS, you can adjust the rate of the voice (not available for Amazon Polly).

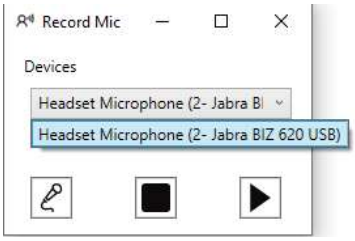
Write the required phrase in text field:



After you have adjusted and tested the voice, press **[Apply]** button.

You will return to **Scenario** interface.

To record sound file using a microphone, press **[Record by Microphone]** button. The **Record Mic** interface will be opened. Select the microphone from the available device list.



Then press button. Three seconds countdown will be started, afterwards, say your phrase while “Record” is on the screen.

Stop the record by button. Now you can play the record using button.

If the record is correct, close the window and return to **Scenario** interface.

Enter the scenario name in the *Name* field.

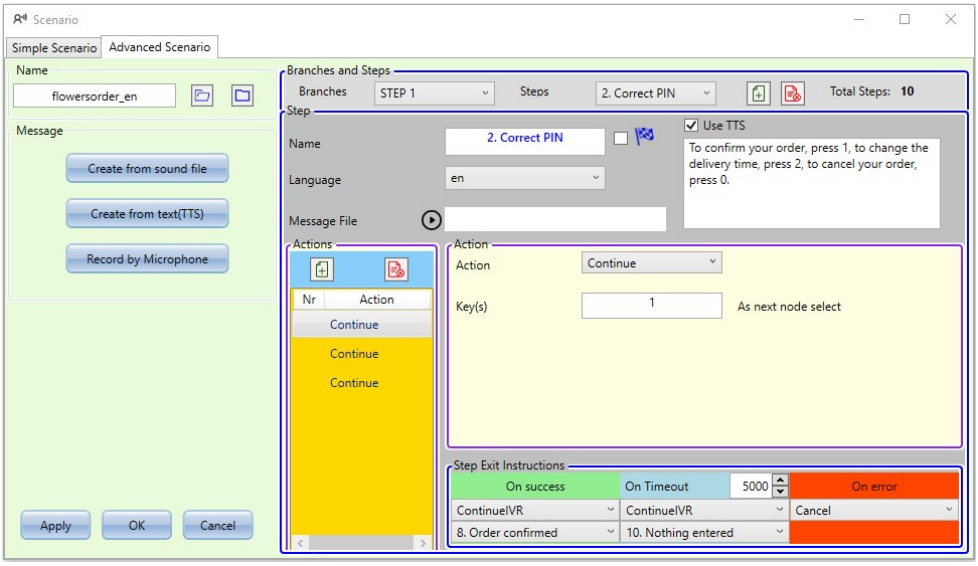
If it is required that the called person has to confirm that the notification is received, check the *Use confirmation* option and enter confirmation button or confirmation button sequence(separated by comma) in the field:

The *Action* will be changed from: “none to “checkSequense” automatically and *Action Wait Time* will be changed to 5 seconds(5000 ms). *Action Wait Time* determines the maximum time between the end of the message and the first confirmation button entered, as well as between buttons entered.

Press **[Apply button]**. File synchronization will start immediately.

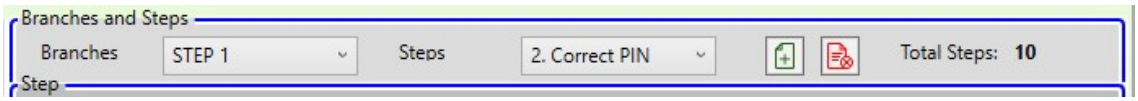
5.4.2. Advanced scenario

Use **Advanced Scenario** tab to create or edit a more complicated scenarios. Advanced scenarios allow you to use various interaction schemes between system and called person.

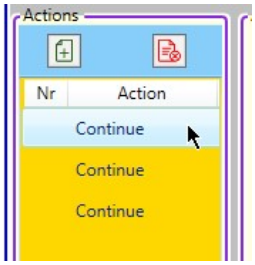


5.4.2.1. Branches and Steps

Advanced scenario consists of Branches [1], each of which consists of Steps.

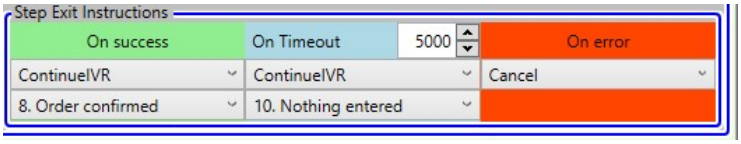


Each step can contain the **Actions**, which can process the pressed by called person buttons or play preset in the item variables to called person.



Each step can step over to another step or cancel scenario execution.

Each step can have three different exit points: **On Success, On Timer and On Error**.



On Timer exit point exists in all cases, **On Success** and **On Error** exit points are available if **Actions** are used.

Exit point tells the system what to do when the current step is over. It consists of instruction and optionally next step.

If step doesn't contain action, you can use TTS message for greeting and define the sound file as well. The latter will play after greeting completed, restarting **OnExpired** timer.

5.4.2.2. Actions

Action is the function that processes the pressed buttons (or "keys"), or plays variable to called person, or makes another action.

There are defined different action types in the system depending of their functionality.

Actions **Continue**, **CheckSequence** verify are the pressed buttons matching to predefined in action sequence or not and puts the entered buttons into reports.

Action **CheckPIN** waits for the button defined as "*End of Entering*", then check if the entered button sequence is matching to predefined user PIN [2].

Action **CollectKeysTillEndKey** waits for the button dedicated as "*End of Entering*", then puts the collection into reports.

Action **CollectKeys** collects the predefined number of entered buttons and puts the collection into reports.

Action **PlayVariable** plays the predefined variable to called person and doesn't process any buttons.

Action **Transfer** transfers the call to the predefined extension at your sip server. Action will try to make the attended transfer during time, set for **OnExpired** exit point. If transfer target not answered, **OnExpired** exit point will be used for further behavior of scenario. If transfer target answered, the **OnSuccess** exit point will be used. If transfer target is busy the **OnError** exit point will be used.

Action **Record** allows to record customer message. Record will placed in the directory of installation to folder '\CustomerMessages' on server. Assume that this action is the final action of scenario, so set for all exit pointers the CancelConfirmed instruction to avoid call repeating.

If action processed entered buttons successfully (button sequence matches to sequence predefined in action or number of entered buttons reaches the limit in **CollectKeys** action or entered buttons matches to predefined user PIN in **CheckPIN** action), the **On Success** exit point should be used to define the further behavior of scenario.

If action processed entered buttons unsuccessfully (button sequence doesn't match to predefined in action sequence or number of entered buttons is over the limit in **CollectKeysTillEndKey** or **CheckPIN** actions or entered buttons doesn't match to predefined user PIN in **CheckPIN** action), the **On Error** exit point should be used to define the further behavior of scenario.

If buttons are not completely entered during predefined in **On Timer** exit point wait time, the **On Timer** exit point should be used to define next step or scenario canceling.

You can use several **Continue** or **CheckSequence** actions in each step in order to process different customer choices. The **On Success** exit point will be defined for each action in step.

The only one **CollectKeys**, **CollectKeysTillEndKey**, **CheckPIN** and **PlayVariable** action can be used in each step.

5.4.2.3. Step exit points

Step exit points define further scenario behavior when the step is completed.

Each exit point has the instruction and, depending of it, can have the next destination (another step).

The following instructions are available:

Cancel – cancels scenario execution.
CancelConfirmed – cancels scenario execution and informs the system that item for which scenario has been executed can be considered as successful.
CancelCollected – cancels scenario execution and informs the system that some keys have been collected.
ContinueIVR – informs the system that scenario has to be continued. You must define the next step of scenario in this case. The step selection combo box is available when this instruction is selected.

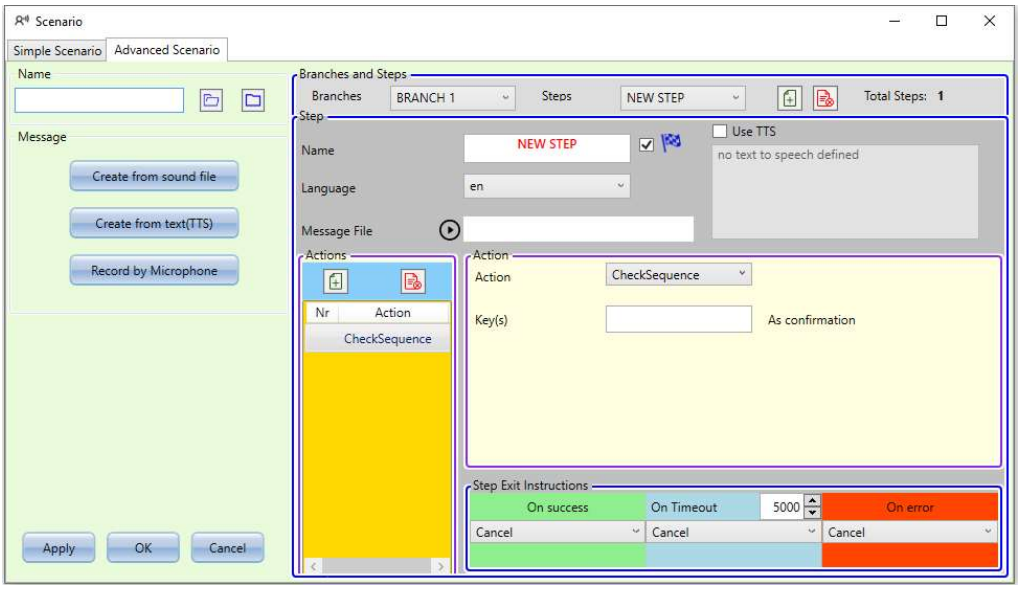
If you don't use text-to-speech engine in the steps, you can define the message sound file for each step in the same way as in the **Simple Scenario** section.

If you use text-to-speech engine in the steps, you can define the language of message for each step. Remember that voices for each language should be defined in the system settings (use **OCM Manager** Application).

[1] - only one Branch can be used in current version
[2] – PIN can be defined in "customerKeys.txt" file or in "ocm_userkeys" database table. System tries to validate entered PIN in both ways, checking database first. See descriptions of "ocm_userkeys" database table and "customerKeys.txt" file in the Appendix 4 of "Carmona OCM General en.pdf".

5.4.2.4. Create or edit the Advanced Scenario

When you select **Advanced Scenario** tab, the following interface will appear:



The scenario name field, the **[Open]** and **[New]** buttons are located in the upper left part of interface. If you are came to the scenario window with selected scenario, all the fields will be filled with data of selected scenario. You can open another scenario using **[Open]** button or create new scenario using **[New]** button.

You can select the scenario branches and steps in the upper part of *Branches and Steps* parameters group.

You can add the new step using **[Add]** button or delete the new step using **[Delete]** button. Any step but only one can be marked as start step by flag check.

You can define the step name in the Name field, select the language for text-to-speech engine, play the message sound file (if text-to-speech is not used), and define the text-to-speech usage and message text in the *Step parameters* group.

You can add or delete the action in the *Actions* parameter group, using the **[Add]** or **[Delete]** buttons correspondingly.

You can define action parameters in the *Actions* parameters group.

There are the different parameters for the action types.

For the **CheckSequence** and **Continue** action types you can define the key or key sequence, which called person should enter to complete the action. If key sequence is used, enter the keys separated by comma in the Key(s) field.

For the **CheckPIN** and **CollectKeysTillEndKey** action types you can define the key, which should be used as the end of keys entering.

For the **CheckPIN**, **CollectKeys** and **CollectKeysTillEndKey** action types you can define the maximum number of entered keys.

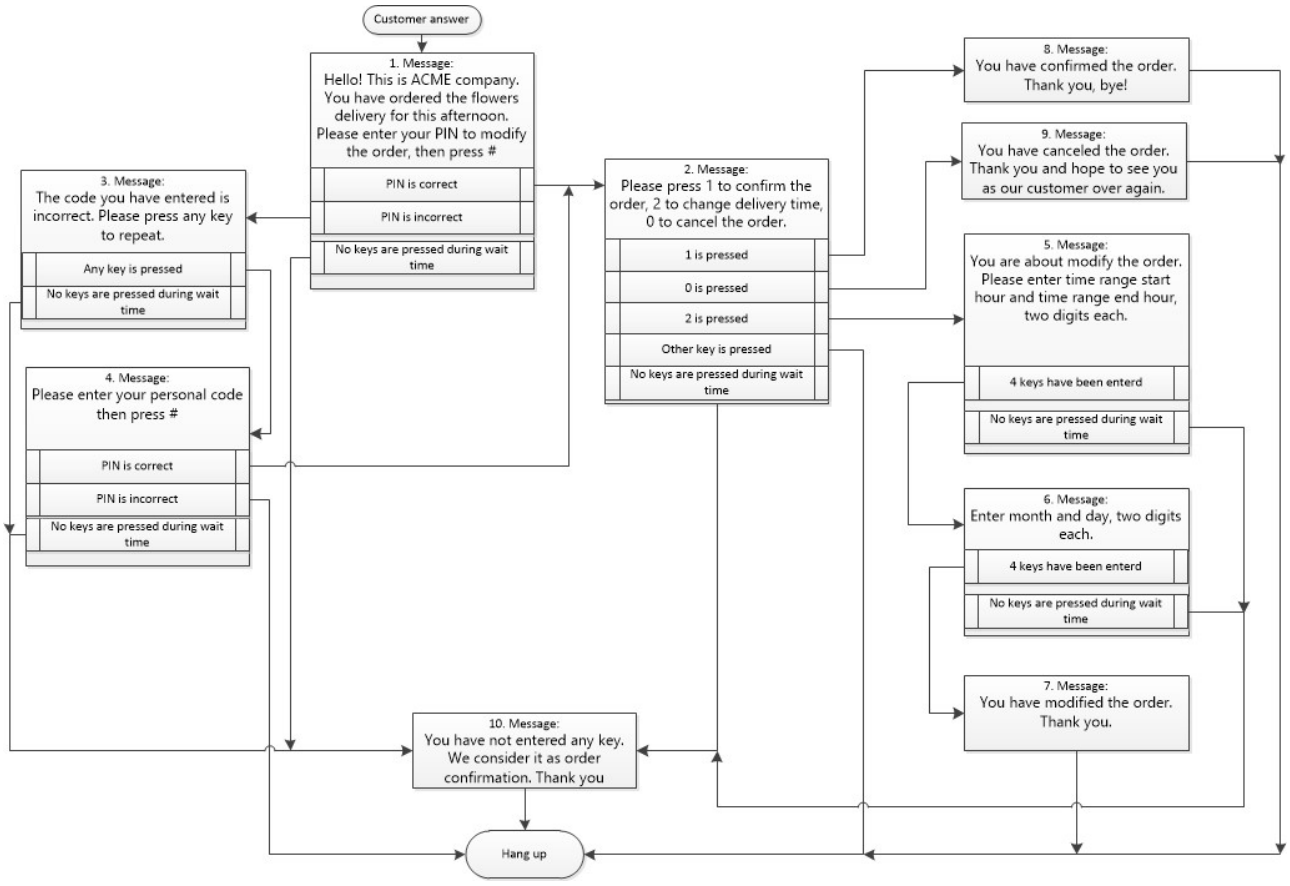
For the **PlayVariable** action types you can define the variable, which has to be played, the text-to-speech usage, and the sex (gender) of variable object (this parameter is important if non English language is used in the step). The variable values should be defined in the Variables field of item (you can set them in the Excel or csv file, in the database or in the **Item** window).

For the **Transfer** action type you can define the extension at your sip server as transfer target, the time to wait the answer, and the number of transfer attempts. When attempts exceeded the exit instruction for **OnExpired** exit point will be force set to Cancel and scenario will be canceled.

You can define the required exit instructions and the next step jumps in the *Step Exit Instructions* parameters group. Also you can define the timeout for waiting of button pressing.

5.4.2.5. The Advanced Scenario creation example

Suppose we need to remind to our customers, that they have ordered the flowers delivery. We need to know if they are still ready to receive the order, or they are intent to cancel the order, or they wish to change the delivery date and time. The algorithm of dialog will be the following:



We have to create 10 steps. As we can see the steps No 7, 8, 9, 10 doesn't provide key check, so we don't need to use any actions in this steps.

The steps No 1 and 4 have to check PIN entry, so we need to use CheckPIN action in these steps.

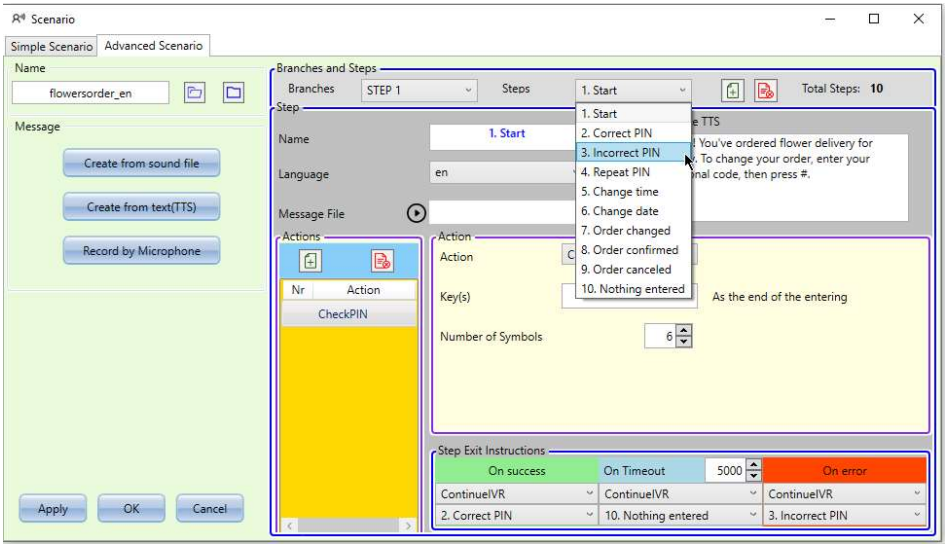
In the step No 2 we have to check three possible key entries, so we need to use three CheckSequenece (or Continue) actions with different parameter Key(s) in this step.

In the step No 3 we have to check any key entry, so we need to use one CheckSequenece (or Continue) action in this step.

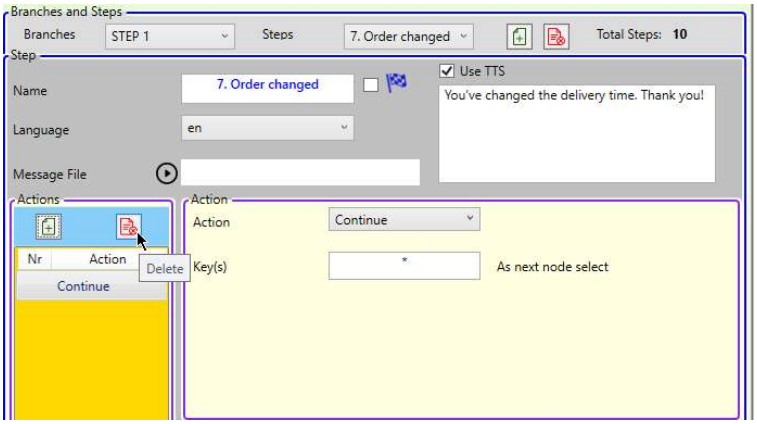
In the step No 5 and 6 we have to collect four keys in each, so we need to use CollectKeys action in these steps.

We will use text-to-speech conversion for all steps.

Create 10 steps with friendly names using **[Add]** button.

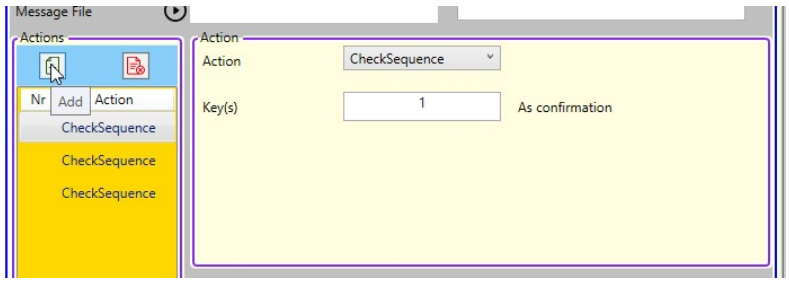


For steps 7, 8, 9, 10 delete default provided actions, using **[Delete]** button.

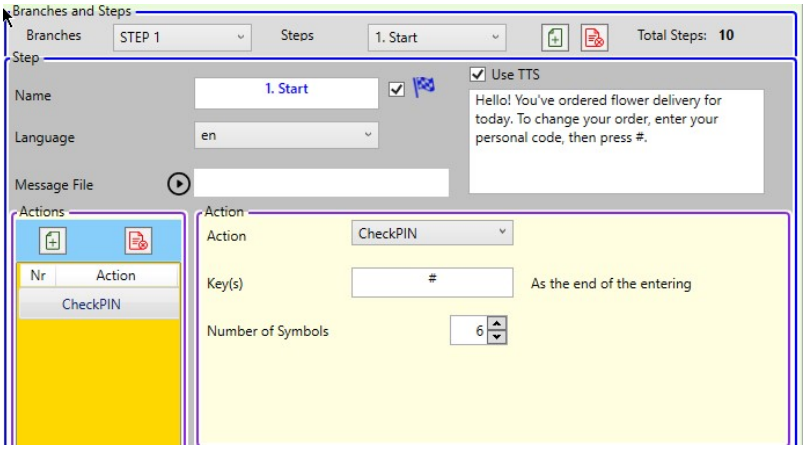


. . .

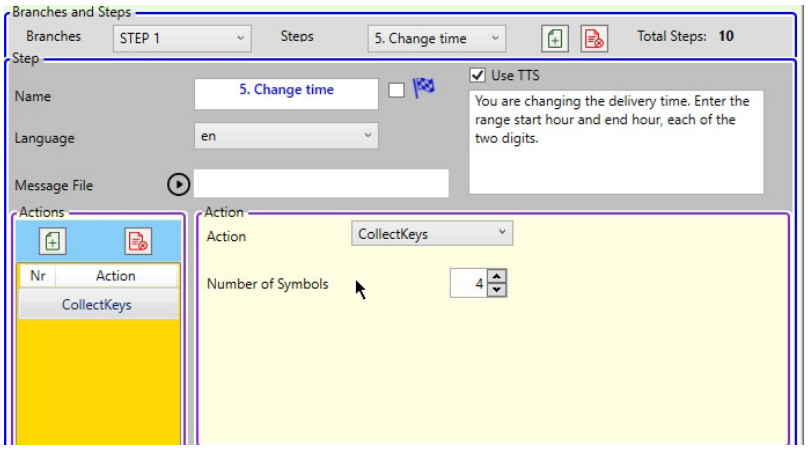
For step 2 add two actions in addition to the default provided one, using **[Add]** button, and configure them.



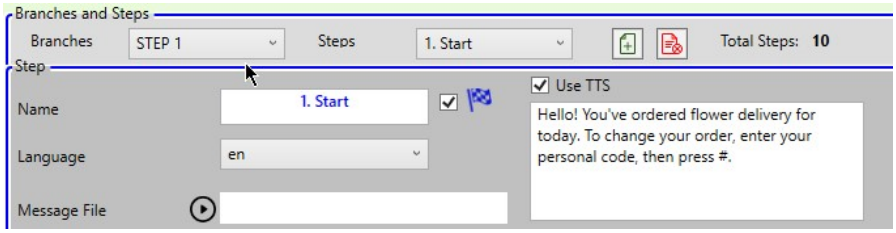
For steps 1, 3, 4, 5, 6 configure default provided actions.



. . .

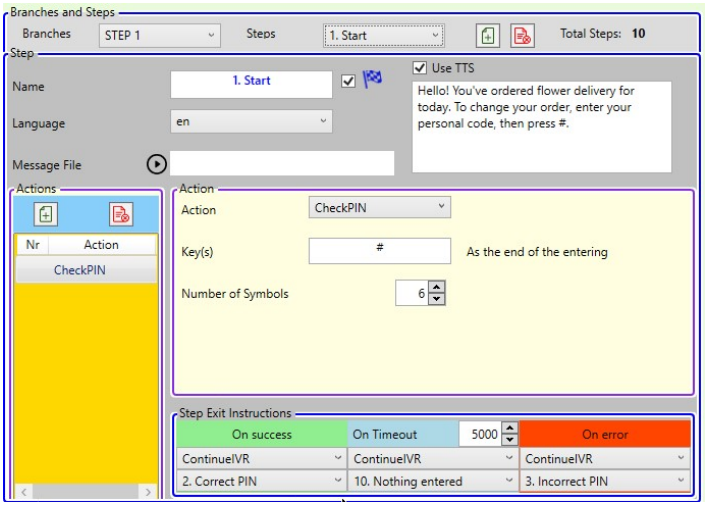


Check *Use TTS*, select the language and type the required messages in *Use TTS* text field for each step.

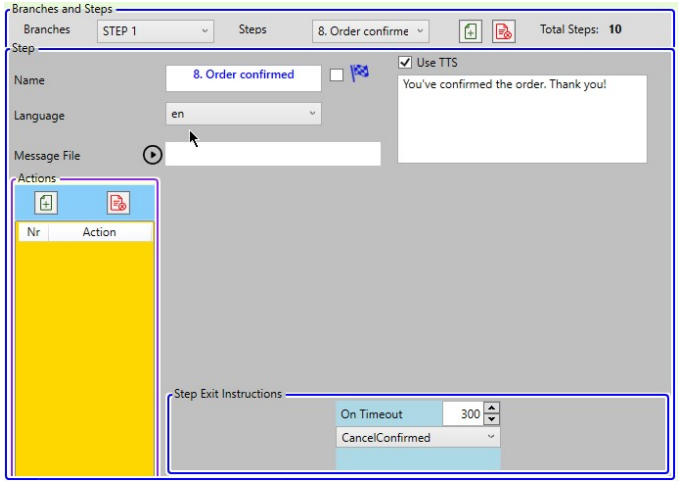


If you need to make some delay before message playing use blanks before text.

Configure step jumps in the *Step Exit Instructions* parameters group and adjust wait time for each step. Use minimal value (200 ms) for the steps where key check is not required (no action used).



...



5.4.2.6. Variables usage

Using variables allows you to convey to the called subscriber some personal data - amounts, dates, phrases.

Variables are set in the *Variables parameter* for each element of the queue.

Each variable should start with the % symbol, then follow it number (optional), followed by a letter meaning type of variable, then the value of the variable should follow.

The following types of variables are applicable:

N (numeric) - number

D (date) – date

T (time) - time

S (string) – phrase

$ (monetary) - numeric monetary

If the number of the variable is not specified, then the scenario will be search the variable in accordance with its order number of all variables of the queue item.

Example of variable parameter in a queue item: %1N225642%7N1500 in the Item window:

in the Excel file:

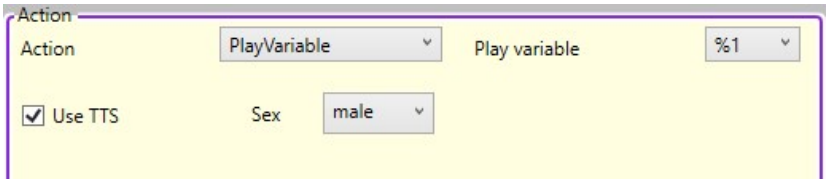
C	D	E
GROUP	PRIORITY	VARIABLESSTRING
GROUP 0	0	%1N225642%7N1500

in the database table:

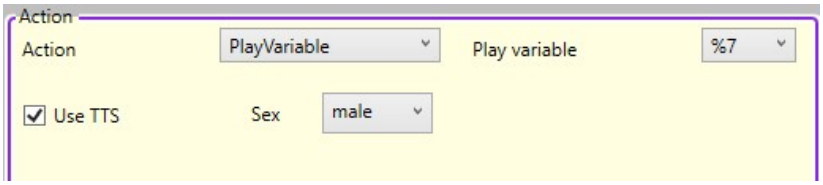
M - dbo.ocm_input

callee	calleeid	queue	phones	qroupname	priority	variables	prefix	ringtime
TEST CALLEE 100	0	fcb1002	044100100	GROUP 0	0	N225642%7N1500		20
TEST CALLEE 101	0	fcb1002	044101101	GROUP 0	0			20

While executing the scenario in the step where the **PlayVariable** action is used, the specified variable number will be searched for in the *Variables parameter* of the queue element and, if found, the value of the variable will be played back.

1. 

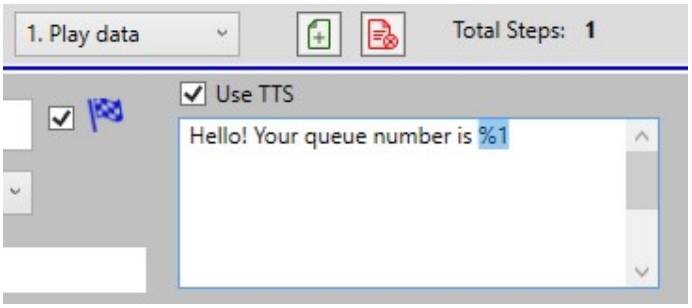
In this step, the number 225642 will be announced to the called party.

2. 

In this step, the number 1500 will be announced to the called party.

If a record without a number is used: %N223564%N800, at step 1 there will be number 223564 announced, and at step 2 nothing will be announced, because the record contains only two variables (the seventh variable is absent).

Also, if text-to-speech conversion for step is used, you can put the variable directly in *UseTTS* text:



In this case try to avoid symbols ‘%’ as part of message text.

It should be noted that the use of variables is very simple when text-to-speech conversion is used and is quite difficult if prepared sound fragments are used to combine the “number” sound file. In this case, you need to record audio fragments for all possible numbers, tens, hundreds, thousands, etc. in all applicable grammatical cases, genders and numbers.

6. Apendix 1. Description of queue files

A1.1. Excel queue file description

File is used as input data source for queue creation.
You have to use xlsx (Excel 2007 or higher) Excel files only.
The first row of the file must only consist of column names.
The file should have the following columns. Columns with mandatory values are in bold.

CALLEENAME	**Name of person (mandatory).**
ITEMPHONESSTRING	**Person phones (mandatory). If several phones are used, they should be separated by colon.**
GROUP	Group Name (not used in current version).
PRIORITY	Execution priority: from 0 to 9, where 0 is the highest (default value: 5).
VARIABLESSTRING	Variables string.
PREFIX	CO Line access prefix.
RINGTIME	Ring time, seconds. Default value: 25.
REPEATPAUSETIME	Time between attempts, seconds. Default value: 30.
SCENARIONAME	**Scenario Name. You can use scenario name as well as sound file name**.* If you are going to use sound file name, you must use extension .wav(for example: debit.wav), sound file must be in PCM Mono, 16 bit, 8000 Hz format if option *Use ffmpeg convert to* wav (see 5.1.2) is switched off.
TRIES	Number of attempts. Default value: 3.
MINLISTENEDPERCENT	Minimal percentage of message duration in order to consider the message as listened . Default value: 80. Not applicable when confirmation is used.
CONFIRMBUTTON	Button or button sequence that must be entered for message confirmation. Values: 0-9, A,B,C,D, *, #.
QUEUEITEMSTARTSTR	Notification process start time. Default value: file uploading time(immediate start).

A.1.2. Character-separated-values (CSV) files description

File is used as input data source for queue creation.
You can only use comma as delimiter.
The first row of the file must only consist of field headers.
The file should have the following fields. Mandatory fields are in bold.

CALLEENAME	**Name of person (mandatory).**
ITEMPHONESSTRING	**Person phones (mandatory). If several phones are used, they should be separated by colon.**
GROUP	Group Name (not used in current version).
PRIORITY	Execution priority: from 0 to 9, where 0 is the highest (default value: 5).
VARIABLESSTRING	Variables string.
PREFIX	CO Line access prefix.
RINGTIME	Ring time, seconds. Default value: 25.
REPEATPAUSETIME	Time between attempts, seconds. Default value: 30.
SCENARIONAME	**Scenario Name. You can use scenario name as well as sound file name.*** If you are going to use sound file name, you must use extension .wav(for example: debit.wav), sound file must be in PCM Mono, 16 bit, 8000 Hz format if option *Use ffmpeg convert to* wav (see 5.1.2) is switched off.
TRIES	Number of attempts. Default value: 3.
MINLISTENEDPERCENT	Minimal percentage of message duration in order to consider the message as listened. Default value: 80. Not applicable when confirmation used.
CONFIRMBUTTON	Button or button sequence that must be entered for message confirmation. Values: 0-9, A,B,C,D, *, #.
QUEUEITEMSTARTSTR	Notification process start time. Default value: file uploading time(immediate start).
RECORD	Is record of person's reply is required. 1 – yes, 0 or empty – not.

* When you use sound file name as scenario name the scenario will be created by system at the moment the file is uploaded. Scenario message will use this sound file and scenario action will be based on CONFIRMBUTTON and MINLISTENEDPERCENT parameters.

A1.3. Report file description

File (.xlsx) is designed to display queue execution results and is created by system each time when queue is finalized. Workbook contains four worksheets:

Queue
User Actions
Items
Calls

Worksheet '**Queue'** contains the following columns:

Timestamp – queue action timestamp.
Queue – queue name.
Action – action(started, stopped).
Total Items – number of items in queue.
Successful – number of successful items.
Unsuccessful – number of unsuccessful items.
Total time – summary time of queue work.
Nr of used accounts – number of used accounts.

Worksheet '**User Actions'** contains the following columns:
Timestamp – user action timestamp.
User – user name.
User IP – user ip address.
Action – action that user took.
Affected items – list of items, affected by action.

Worksheet '**Items'** contains the following columns:

Callee - Called person name.
Enqueue Time – item enqueue time.
Start Time – item start time.
Group – item group.
Priority – item priority.
Phones – all item phones.
Answered Phone – last answered phone if answered, items phones if not.
Answer Time – last call answer time.
Attempts – number of attempts made.
Result – item execution result(successful, unsuccessful, etc).
Listened Time, ms – last answered call talk duration.
Call State by Attempts – list of attempts with call state and timestamp.
Error Code – error code for items which couldn't be processed correctly.
Confirmed – indicates whether message was confirmed or not.
Entered buttons - list of entered buttons for each attempt

Collected Keys – keys, collected in **CollectKeys** or **CollectKeysTillEndKey** actions
Transferred – was call transferred or not.
Transfer Destination – transfer destination, if call was transferred.
Record File – path to last recorded call conversation.

Worksheet '**Calls'** contains the following columns:

Call Start Time – call start time.
Callee - Called person.
Phone – phone used for the call.
SIP Account – SIP account, used for this call.
Attempt Nr – attempt number for call.
Call Result – call result (Answered, Request Timeout, etc).
Answer Time – call answer time.
Listened Time, ms – last answered call talk duration.
Confirmed – indicates whether message was confirmed or not.
Transferred – was call transferred or not.
Transfer Destination – transfer destination, if call was transferred.
Record File – path to last recorded call conversation.

After queue is finalized, the report file is written to the following directory on server:
Directory\YYYY\MM\DD\
having as a name: queuename_DD-MM-YYYY HH_mm_ss.xlsx
where Directory is the folder selected in OCM Manager-Integration-Reports Folder,
YYYY – year, MM – month, DD – day, HH – hour, mm – minute, ss - second of queue compose time.

If conversation records have been made, they are located in the directory on server:
Directory\YYYY\MM\DD\queuename_DD-MM-YYYY HH_mm_ss_Records
having as a name: XXXXXXXXX_DD-MM-YYYY HH_mm_ss.wav
where XXXXXXXXX – is phone number, Directory – see above,
YYYY – year, MM – month, DD – day, HH – hour, mm – minute, ss - second of call answer time.

7. Apendix 2. Description of queue database tables

A2.1. Input table description

Table name should contain “input” in the name.
Table should have the following columns. Columns on bold are mandatory.

1. Id –primary key. Do not enter any values here.
2. **Callee -** Name of the person (mandatory).
3. **Phones** - Person phones (mandatory) in national or international format. If several phones are used, separate them using comma.
4. Prefix - CO Line access prefix.
5. **Scenario** - Scenario Name. You can use scenario name as well as sound file name.* If you are going to use sound file name, you must use extension .wav(for example: debit.wav). Sound file must be in PCM Mono, 16 bit, 8000 Hz format if option *Use ffmpeg convert to* wav (see 5.1.2) is switched off.
6. **State** – new item property. If equal to 1, system will enqueuer the item after table is read and will change this property to 0(item enqueued).
7. CalleeId – Your database Person ID. This value will be present in the report table, so you can use it for report building.
8. Queue – queue name. If not defined, system will set it to: QND_DD_MM_YYYY HH_MM_SS.
9. GroupName - Group name (not used in current version).
10. Priority - Execution priority: from 0 to 9, where 0 is the highest (default value: 5).
11. Variables – variables.
12. Ringtime - Ring time, seconds. Default value: 25.
13. Pausetime - Time between attempts, seconds. Default value: 30.
14. Attempts - number of attempt. Default value: 3.
15. ListenPercents - Minimal percent of message duration to consider the message has been listened . Default value: 80. Has no effect when confirmation used.
16. ConfirmButton - Button or button sequence that has to be entered for message listened confirmation. Values: 0-9, A,B,C,D, *, #.
17. ItemStart - Notification process start time. Default value: file uploading time(immediate start).
18. IsAppended – if set to 1, item should be appended to the running queue, without restarting it.
19. IsRecordRequired – if set to 1, system will record the person’s conversation.

The system tracks changes in the database in accordance with the interval specified in the settings: [Integration -> Input Data Update Interval]. All items that have State property equal to 1, will be enqueued. So you have to set State property for all records just once. If next database table reading attempt will find the records with State equal to 1, system will reset the current queue and compose the new one.
If new items are enqueued, calls begin five seconds after composing the queue in the system.

* When you use sound file name as scenario name the scenario will be created by system at the moment of database reading. Scenario message will use this sound file, and scenario action will be based on CONFIRMBUTTON and MINLISTENEDPERCENT parameters.

A2.2. Report table description

Table name should contain “report” in the name.
Table should have the following columns.

1: Id - primary key
2: Callee - Name of the person (mandatory).
3: CalleeId - Your database Person ID.
4: Queue - queue Name. If not defined in the input table, system sets it to:

QND_DD_MM_YYYY HH_MM_SS.

5: QueueStart - Queue start time.
6: QueueEnd - Queue end time.
7: QueueItems – nu,ber of items enqueued.
8: EnqueueTime - Item enqueue time.
9: StartTime - Last call start time.
10: AnswerTime - Last call answer time.
11: AnsweredPhone - Last answered phone.
12: Attempts - The number of attempts made.
13: State - Item or queue state. For item: 0 - idle, 2 - successful, 3 – unsuccessful. For queue: 1 – queue execution is started, 200 – queue is finalized, 500 - error.
14: ListenedTime_ms - Listened duration for last message in milliseconds.
15: ErrorCode - Error code.
16: IsConfirmed - Called person confirmed that the message is listened using preset buttons.
17: RecordFile - Person conversation record file name.
18: EnteredButtons - Buttons which called person entered (for the last call).
19: IsTransfered - Last call was transferred.
20: TransferDest - Last call transfer destination.
21: QueueId - Queue ID. Value is generated each time when the queue is created.
22: ItemId - Item ID. Value is generated each time when the queue is created.
23: IsNewState - Property that indicates whether a new item or queue state is occurred.
24. CollectedKeys – entered by customer key sequences, referenced to the steps and actions.

All records which contain non-empty Guid(not equal to 00000000-0000-0000-0000-000000000000) in field QueueId, are related to queue states.
All records which contain empty Guid(equal to 00000000-0000-0000-0000-000000000000) in field QueueId, are related to item states.

8. Appendix 3. Database table CREATE scripts

A3.1 Database table scripts MS SQL Server 2008 R2 or higher

Table ocm_input.

```
USE [OCM]
GO
/****** Object:  Table [dbo].[ocm_input]    Script Date: 01/30/2019 16:07:57 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ocm_input](
	[Id] [bigint] IDENTITY(1,1) NOT NULL,
	[Callee] [nvarchar](150) NULL,
	[CalleeId] [bigint] NULL,
	[Queue] [nvarchar](100) NULL,
	[Phones] [varchar](75) NULL,
	[GroupName] [nvarchar](50) NULL,
	[Priority] [int] NULL,
	[Variables] [varchar](50) NULL,
	[Prefix] [varchar](20) NULL,
	[RingTime] [int] NULL,
	[PauseTime] [int] NULL,
	[Scenario] [nchar](50) NULL,
	[Attempts] [int] NULL,
	[ListenPercents] [int] NULL,
	[ConfirmButton] [varchar](50) NULL,
	[ItemStart] [datetime] NULL,
	[IsRecordRequired] [bit] NULL,
	[IsAppend] [bit] NULL,
	[State] [smallint] NULL,
 CONSTRAINT [PK_ocm_input] PRIMARY KEY CLUSTERED
(
	[id] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
ALTER TABLE [dbo].[ocm_input] ADD  CONSTRAINT [DF_ocm_input_State]  DEFAULT ((2))
FOR [State]
GO
```

Table ocm_report.

```
USE [OCM]
GO
/****** Object:  Table [dbo].[ocm_report]    Script Date: 01/30/2019 16:08:21 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ocm_report](
	[Id] [bigint] IDENTITY(1,1) NOT NULL,
	[Callee] [nvarchar](150) NULL,
	[CalleeId] [bigint] NULL,
	[Queue] [nvarchar](100) NULL,
	[QueueStart] [datetime] NULL,
	[QueueEnd] [datetime] NULL,
	[QueueItems] [int] NULL,
	[EnqueueTime] [datetime] NULL,
	[StartTime] [datetime] NULL,
	[AnswerTime] [datetime] NULL,
	[AnsweredPhone] [varchar](15) NULL,
	[Attempts] [int] NULL,
	[State] [int] NULL,
	[ListenedTime_ms] [int] NULL,
	[ErrorCode] [int] NULL,
	[IsConfirmed] [bit] NULL,
	[[RecordFile] [nvarchar](200) NULL,
	[EnteredButtons] [varchar](100) NULL,
	[IsTransfered] [bit] NULL,
	[TransferDest] [varchar](15) NULL,
	[QueueId] [varchar](50) NULL,
	[ItemId] [varchar](50) NULL,
	[IsNewState] [bit] NULL,
	[CollectedKeys] [varchar(MAX)] NULL
 CONSTRAINT [PK_ocm_report] PRIMARY KEY CLUSTERED
(
	[id] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
```

A3.2. Database table scripts for Postgre SQL Server 9.5 or higher

Table ocm_input.

```
CREATE SEQUENCE public.ocminput_id_seq
  INCREMENT 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  START 1
  CACHE 1;
ALTER TABLE public.ocminput_id_seq
  OWNER TO postgres;
CREATE TABLE public.Ocm_Input
(
        "Id" bigint NOT NULL DEFAULT nextval('ocminput_id_seq'::regclass),
        "Callee" text NULL,
        "CalleeId" bigint NULL,
        "Queue" text NULL,
        "Phones" text NULL,
        "GroupName" text NULL,
        "Priority" int NULL,
        "Variables" text NULL,
        "Prefix" text NULL,
        "RingTime" int NULL,
        "PauseTime" int NULL,
        "Scenario" text NULL,
        "Attempts" int NULL,
        "ListenPercents" int NULL,
        "ConfirmButton" text NULL,
        "ItemStart" timestamp without time zone NULL,
        "IsRecordRequired" int NULL,
        "State" int NULL,
        "IsAppend " int NULL,
 CONSTRAINT Ocm_Input_pk PRIMARY KEY ("Id")
 )
WITH (
  OIDS=FALSE
);
ALTER TABLE public.Ocm_Input
  OWNER TO postgres;
```

Table ocm_report.

```
CREATE SEQUENCE public.ocmreport_id_seq
  INCREMENT 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  START 1
  CACHE 1;
ALTER TABLE public.ocmreport_id_seq
  OWNER TO postgres;
CREATE TABLE public.Ocm_Report
(
        "Id" bigint NOT NULL DEFAULT nextval('ocmreport_id_seq'::regclass),
        "Callee" text NULL,
        "CalleeId" bigint NULL,
        "Queue" text NULL,
        "QueueStart" timestamp without time zone NULL,
        "QueueEnd" timestamp without time zone NULL,
        "QueueItems" int NULL,
        "EnqueueTime" timestamp without time zone NULL,
        "StartTime" timestamp without time zone NULL,
        "AnswerTime" timestamp without time zone NULL,
        "AnsweredPhone" text NULL,
        "Attempts" int NULL,
        "State" int NULL,
        "ListenedTime_ms" int NULL,
        "ErrorCode" int NULL,
        "IsConfirmed" int NULL,
        "RecordFile" text NULL,
        "EnteredButtons" text NULL,
        "IsTransfered" int NULL,
        "TransferDest" text NULL,
        "QueueId" text NULL,
        "ItemId" text NULL,
        "IsNewState" int NULL,
        “CollectedKeys” text NULL,
 CONSTRAINT Ocm_Report_pk PRIMARY KEY ("Id")
 )
WITH (
  OIDS=FALSE
);
ALTER TABLE public.Ocm_Report
  OWNER TO postgres;
```

9. Appendix 4. File description and database table for PINs

A file with client PINs is a text file with the name “customerKeys.txt”, which should be located in the same folder as is the CarmonaOCM.exe file (CarmonaOCM server installation folder).

Each line of the file contains the client's phone number and PIN, separated by commas. Example:

044222116,444444
044222112,22335578

The table of the database for storing PINs of clients has the name “ocm_userkeys” and contains three fields:

Id – primary key

userid - field with the client's phone number

keys - field with a client's PIN

Table CREATE script for **MS SQL** Server 2008 R2 and higher:

```
USE [OCM]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ocm_userkeys](
[id] [bigint] IDENTITY(1,1) NOT NULL,
[userid] [nvarchar](50) NULL,
[keys] [nvarchar](50) NULL,
CONSTRAINT [PK_ocm_userkeys] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Table CREATE script for **Postgre SQL** 9.5 and higher:

```
CREATE SEQUENCE public.Ocm_userkeys_id_seq
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 1
CACHE 1;
ALTER TABLE public.Ocm_userkeys _id_seq
OWNER TO postgres;
CREATE TABLE public.Ocm_userkeys
(
"Id" bigint NOT NULL DEFAULT nextval('Ocm_userkeys_id_seq'::regclass),
"Userid " text NULL,
"Keys " text NULL,
CONSTRAINT Ocm_userkeys _pk PRIMARY KEY ("Id")
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.Ocm_userkeys
OWNER TO postgres;
```