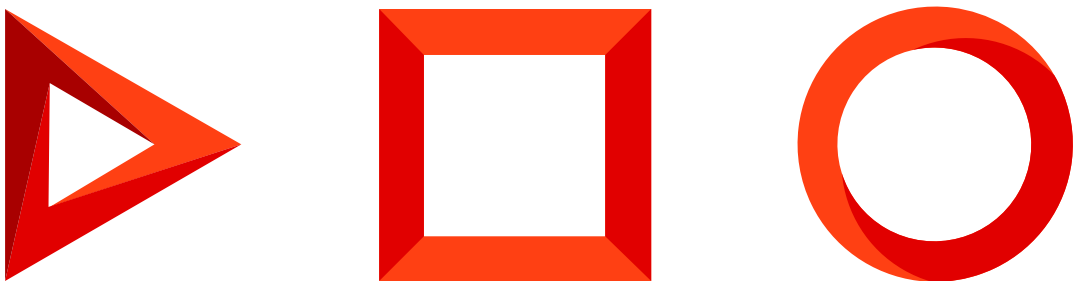
Creatio Academy

Asterisk

Set up integration with Asterisk

Version 8.0



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

© 2023 Creatio. All rights reserved.

Table of Contents

© 2023 Creatio. All rights reserved.

Set up integration with Asterisk

PRODUCTS: ALL CREATIO PRODUCTS

To set up an Asterisk integration, take the following steps:

1. Prepare Asterisk for integration. Read more >>>
2. Set up Creatio Messaging Service. Read more >>>
3. Set up the message exchange library. Read more >>>
4. Set up the Asterisk parameters. Read more >>>

In Creatio, the Asterisk integration functionality requires a separate license. You need to generate a license request, send it to our service team, upload the received license file into the system, and distribute the licenses among the users. Read more: Creatio licensing and Manage user licenses.

The integration is only possible if complete preliminary Asterisk setup was performed by the phone integration administrator.

Attention. If you set up the telephony for a Creatio production environment, deploy Creatio Messaging Service on a separate node rather than on the Creatio application server. To ensure the fault tolerance of your phone integration, we recommend setting up at least two nodes with Creatio Messaging Service, as well as a balancer that would redirect users in case of lost connection with one of the nodes.

1. Prepare Asterisk for integration

The AMI (Asterisk Manager Interface) interface is used to interact with Asterisk. Use AMI to connect to Asterisk servers, configure and manage client programs.

1. Create an AMI user for Creatio, specify the user's parameters in the “manager.conf” file, for example:

```
[ creatio ]

secret = creatio

deny=0.0.0.0/0.0.0.0

permit=0.0.0.0/0.0.0.0

read = system,call,log,verbose,command,agent,user,originate

write = system,call,log,verbose,command,agent,user,originate
```

Replace the “deny” and “permit” values with the corresponding addresses.

© 2023 Creatio. All rights reserved.

2. Check the parking feature activity and the “features.conf” file parameters, for example:

```
parkext => 700

parkpos => 701-720

context => parkedcalls

parkingtime => 45
```

Attention. The “parkingtime” value determines the maximum time a call can be on hold. Upon the expiry of this time the subscriber will resume the conversation with the agent. The “parkingtime” value should be sufficiently long to avoid early call return.

2. Set up Creatio Messaging Service (formerly Terrasoft Messaging Service)

The messaging service lets you connect Creatio to Asterisk via AMI protocol and distribute call events between Creatio users. Some of the settings differ depending on the platform on which the Asterisk telephony service is deployed – .NET Framework or .NET Core.

Set up Creatio Messaging Service on the .NET Framework platform

1. Before installing Creatio Messaging Service (CMS), make sure that your computer runtime environment has:
   - A .NET Framework package version 4.7.2 or later on the server where you are going to install Creatio Messaging Service.
   - At least 2 Gb of RAM and 20 Gb of free drive space.
2. Contact Creatio support to receive the messaging service installation files or download the files via the URL: Download Creatio Messaging Service. Unpack the archive to a folder to ensure a smooth installation. If you run the installation directly from the archive, the archiver application may interfere with the install wizard.

   Attention. Deploy CMS on the server connected to both the Creatio application server and the PBX. Read more: Telephony integration basics.
3. Run the Creatio Messaging Service Install.msi file on the machine intended as the message exchange server and proceed with the installation.
4. Make sure that the “TerrasoftMessagingService” service is running in the Windows Services application. If the “TerrasoftMessagingService” service is not running, start it manually.
5. Open the folder with the service files: ~\BPMonline Messaging Service. Specify the following parameters for Asterisk connector in the “Terrasoft.Messaging.Service.exe.config” configuration file:

© 2023 Creatio. All rights reserved.

```
<asterisk filePath="" url="Asterisk_name_or_server_address" port="Asterisk_server_port" userN
```

See the **list of Asterisk connector parameters** in the following table.

Parameter caption	Parameter function
FilePath	Use this parameter for diagnostics of the system. It allows to repeat a set of events from the file. The default value should be empty.
URL	Asterisk server IP address parameter.
protocol	The parameter enables selecting the protocol type: SIP or PJSIP. Contact your PBX administrator to find out the needed protocol type.
Port	AMI protocol port. By default, “5038.”
OriginateContext	The command is used to initialize the call from Creatio phone number. The parameter contains the caption of the context from which the call will be made to the user phone number. The default value for FreePBX is “from-internal”
parkingLotContext	The context for call initialization to receive parking line. The default value is “originateContext.”
AutoPauseOnCommutationStart	The checkbox is used for the correct work with Asterisk queues. If the checkbox is selected, the system will put the agent on a pause in all queues after answering the call. The feature is used to avoid the second call during the handling of the first one and/or putting the first call on hold.
queueExtensionFormat	The call channel format while receiving the call from the queue. If using LocalChannel, the default value in FreePBX is “Local/{0}@from-queue.”
sendRingStartedOnRingingState	The checkbox stands for the correct handling of the call from the queue. If this checkbox is selected, the system will display the call with the user after receiving the “NewState” event with the Ringing parameter by AMI. The default value is “On.”
traceQueuesState	This setting is used to define the agent's status in the queues. If the agent receives the second call from the queue while handling the first one in Creatio, it is used for debugging. Information about the status of agents is being written to the log file of the connector. The default value is “Off.”

© 2023 Creatio. All rights reserved.

Configuration example:

```
<asterisk   filePath=" "url="10.0.15.185" port="5038"

userName="bpm" secret="bpm" originateContext="from-internal"

parkingLotContext="from-internal" autoPauseOnCommutationStart="true"

queueExtensionFormat="Local/{0}@from-queue/n" asyncOriginate="true"

sendRingStartedOnRingingState="true" traceQueuesState="false"

protocol="SIP/" packetInfoConfig=""/>
```

6. Test the phone integration.

Note. Follow this instruction if you need to update Creatio Messaging Service.

Set up Creatio Messaging Service on the .NET Core platform

Attention. You can set up Messaging Host on the .NET Core platform for Creatio version 7.16.3 and later.

1. Install Docker. Installing Docker on Linux platforms is covered in the Docker guide. Run the “docker --version” command on Linux machine to verify the installed Docker version.
2. Install Docker Compose. Installing Docker Compose for Linux OS is covered in the Docker documentation. Run the “docker-compose --version” command on Linux machine to verify the installed Docker Compose version.
3. Install and set up the Docker Compose components. The container of the messaging service is deployed via the Docker Compose utility. Download the archive. Unzip the archive with the configuration files and scripts to a custom directory, for example, /opt/messaging.host.

Note. The configuration files contain all necessary default settings for a Linux based server.

The archive structure of the configuration files and scripts:

```
/etc/

...\appsettings.json – service configuration

...\nlog.config – setup of the service log level
```

© 2023 Creatio. All rights reserved.

```
docker-compose.yml – utility configuratiton Docker Compose

.env – file containing the environment variables to run the components
```

4. Using the Linux terminal, go to the /docker-compose catalog of the unzipped archive, for example, /opt/messaging.host/docker-compose.
5. Run the “sudo docker-compose pull” command in the terminal. Wait until the download of the required service component images from the Docker Hub is complete.

Attention. If the server Internet access is restricted, download the needed images manually to a machine with free access (see the docker-compose.yml configuration file) and transfer the images as files to the target machine using the sudo docker export and sudo docker import commands.

6. Specify the following parameters for Asterisk connector in the “etc/appsettings.json” configuration file:

```
{
"url": "Asterisk_server_name_or_address",

"port": "Asterisk_server_port",

"userName": "Asterisk login",

"secret": "Asterisk password",

"originateContext": "Outbound context",

"parkingLotContext": "Parking context",

"autoPauseOnCommutationStart": "true",

"queueExtensionFormat": "Local/{0}@from-queue/n",

"asyncOriginate": "true",

"sendRingStartedOnRingingState": "true",

"traceQueuesState": "false",

"protocol": "The used protocol type – SIP/ or PJSIP/",

"packetInfoConfig": "Additional package parameters for processing in configuration",

"filePath": ""
}
```

7. Run the sudo docker-compose up -d command to launch the service. A “logs” folder will be created in the

© 2023 Creatio. All rights reserved.

current catalog.

See the **list of Asterisk connector parameters** in the following table.

Parameter caption	Parameter function
FilePath	Use this parameter for diagnostics of the system. It allows to repeat a set of events from the file. The default value should be empty.
URL	Asterisk server IP address parameter.
protocol	The parameter enables selecting the protocol type: SIP or PJSIP. Contact your PBX administrator to find out the needed protocol type.
Port	AMI protocol port. By default, “5038.”
OriginateContext	The command is used to initialize the call from Creatio phone number. The parameter contains the caption of the context from which the call will be made to the user phone number. The default value for FreePBX is “from-internal”
parkingLotContext	The context for call initialization to receive parking line. The default value is “originateContext.”
AutoPauseOnCommutationStart	The checkbox is used for the correct work with Asterisk queues. If the checkbox is selected, the system will put the agent on a pause in all queues after answering the call. The feature is used to avoid the second call during the handling of the first one and/or putting the first call on hold.
queueExtensionFormat	The call channel format while receiving the call from the queue. If using LocalChannel, the default value in FreePBX is “Local/{0}@from-queue.”
sendRingStartedOnRingingState	The checkbox stands for the correct handling of the call from the queue. If this checkbox is selected, the system will display the call with the user after receiving the “NewState” event with the Ringing parameter by AMI. The default value is “On.”
traceQueuesState	This setting is used to define the agent's status in the queues. If the agent receives the second call from the queue while handling the first one in Creatio, it is used for debugging. Information about the status of agents is being written to the log file of the connector. The default value is “Off.”

Configuration example:

© 2023 Creatio. All rights reserved.

```
"asterisk": {
 "url": "Asterisk_server_name_or_address",
 "port": "Asterisk_server_port",
 "userName": "Asterisk login",
 "secret": "Asterisk password",
 "originateContext": "Outbound context",
 "parkingLotContext": "Parking context",
 "autoPauseOnCommutationStart": "true",
 "queueExtensionFormat": "Local/{0}@from-queue/n",
 "asyncOriginate": "true",
 "sendRingStartedOnRingingState": "true",
 "traceQueuesState": "false",
 "protocol": "The used protocol type – SIP/ or PJSIP/",
 "packetInfoConfig": "Additional package parameters for processing in configuration",
 "filePath": ""
 }
```

8. Test the phone integration.

3. Set up the message exchange library

Message exchange library selection and setup is performed once by the system administrator. Some of the settings differ depending on the platform that the Asterisk telephony service is deployed on – .NET Framework or .NET Core.

Set up the library on the .NET Framework platform

1. Open the system designer by clicking  in the top right corner of the application window.
2. Click “System settings” in the “System setup” block.
3. In the “Default messages exchange library” (“SysMsgLib” code) system setting, set the default value to “Telephony integration library based on Asterisk (AMI) protocol.”

© 2023 Creatio. All rights reserved.

4. In the “Message exchange server” (“SysMsgServerNode” code) system setting, select the connection parameters of the system messages service. In the [*Default value*] field, specify the message exchange network address in the following format: “ws://[*server*]:2013” for sites served over HTTP or “wss://[*server*]:2013” for sites served over HTTPS, where:

   - [*server*] – domain name of the server that hosts the message exchange service. Using IP addresses or “localhost” is not recommended.
   - “2013” – the port used by default for connecting to the messaging service. You can change the port number in the “Terrasoft.Messaging.Service.exe.config” file. Using “localhost” is not recommended, as it may cause errors when connecting to the phone integration server. When using a wss connection, make sure that server address matches the address in the SSL certificate.

   Note. If your website is served over HTTPS and secure (WSS) connection is used for WebSockets, you will need to install a security certificate on the message exchange server and specify it in the configuration files of the message service. For more information about the setup process, contact Creatio technical support at support@creatio.com.

Set up the library on the .NET Core platform

Attention. You can set up Messaging Host on the .NET Core platform for Creatio version 7.16.3 and up.

1. Open the system designer by clicking  in the top right corner of the application window.
2. Click “System settings” in the “System setup” block.
3. In the “Default messages exchange library” (“SysMsgLib” code) system setting, set the default value to “Phone integration library based on Asterisk (AMI) protocol.”
4. In the “Message exchange server” (“SysMsgServerNode” code) system setting, select the connection parameters of the message service. In the [*Default value*] field, specify the message exchange network address in the following format: “http://[*server*]:2013” for sites served over HTTP or “https://[*server*]:2014” for sites served over HTTPS, where:

   - [*server*] – domain name of the server that hosts the message exchange service. Using IP addresses or “localhost” is not recommended.
   - “2013 or 2014” – the port used by default for connecting to the messaging service. You can change the port number in the “docker-compose.yml” configuration file. Using “localhost” is not recommended, as it may cause errors when connecting to the phone integration server. When using an HTTPS connection, make sure that server address matches the address in the SSL certificate.

   Note. If your website is served over HTTPS and secure connection is used for WebSockets, you will need to install a security certificate on the message exchange server and specify it in the docker-compose configuration. For more information about the setup process, contact Creatio technical support.

© 2023 Creatio. All rights reserved.

4. Set up the Asterisk parameters

These settings should be applied to every Creatio user who received Asterisk integration license. To do this:

1. Open the user profile page by clicking the [*Profile*] image button on the main page of the application.
2. Click the [*Call Center parameters setup*] button.
3. On the displayed page, fill out the required fields:

   a. [*Disable call Center integration*] – this checkbox allows you to disable Creatio integration with the phone integration. The call button will not be displayed on the communication panel of the application.

   b. [*Number*] – Asterisk user line number. It matches the phone number by default. For example, to track the SIP/305 user line, specify the “305” value, and to track the SIP/office line, specify the “office” value.

   > **Attention.** A separate line is used for each user. It is not recommended to specify the same line for several users, as it may cause errors.

   c. [*Outgoing call context*] – specify the outgoing call context if it differs from the system outgoing call context specified in the “Terrasoft.Messaging.Service.exe.config” file for this particular user.

   d. [*Enable debugging*] – this checkbox allows you to display troubleshooting information within the browser console. This troubleshooting information can be used when the phone integration runs into problems and the customer addresses the service team.

4. Click [*Save*].
5. Refresh the browser page to apply the changes.

© 2023 Creatio. All rights reserved.