

# REXYGEN Runtime Core

## User guide

REX Controls s.r.o.

Version 3.0.1

2024-03-01

Plzeň (Pilsen), Czech Republic

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Configuration</b>	<b>3</b>
2.1	Location of Configuration Files . . . . .	3
2.2	Description of Configuration Options . . . . .	4
2.3	RexCore Parameters . . . . .	6
<b>3</b>	<b>System Log</b>	<b>7</b>
3.1	Sources and Severities . . . . .	7
<b>4</b>	<b>Web Interface</b>	<b>8</b>
4.1	Available services . . . . .	8
<b>5</b>	<b>Authentication</b>	<b>9</b>
5.1	Users and roles . . . . .	9
5.2	Using unix accounts . . . . .	10
<b>6</b>	<b>Security</b>	<b>11</b>
	<b>Bibliography</b>	<b>13</b>

# Chapter 1

## Introduction

The most important part of the REXYGEN system is the real-time core contained in a software component called **RexCore**. The **RexCore** is an independent process in the case of a *Windows* or a *GNU/Linux* operating systems, a group of real-time tasks in the case of a hard real-time operating system or a whole system in the case of a microcontroller but it is always the software component where a control algorithm developed by a user runs and from which a device is being handled and which initiates a communication with all other devices in the field that are involved in the control. It also provides a detailed diagnostics or various communication protocols for human-machine interface.

This document contains all the information for configuration of the **RexCore** software component on Windows or GNU/Linux operating systems that is necessary to put the **RexCore** into operation or commissioning. A developer of a control algorithm should read [1]. A user that is not yet familiar with the REXYGEN system should read any of getting-started tutorials, for example [2], [3], [4].

It should be noted that although a big effort has been put into making the **RexCore** as unified as possible on all supported platforms, it is principally not possible to provide absolutely the same functionality and performance on all devices and environments. A user should consider the targeting platform very carefully and eventually consult his needs with the REX Controls company.

## Chapter 2

# Configuration

The RexCore itself requires only a very limited interaction with a user. In fact it needs no interaction at all when a standard installation or distributing channel is used and a default configuration is sufficient for a user. It is however often necessary to adjust the default configuration or secure the control system when commissioning or sometimes even during the development.

### 2.1 Location of Configuration Files

All files that are required for the proper operation of RexCore are located in a *configuration directory*. The *configuration directory* is `/rex/rexcore` in the case of GNU/Linux or in the `REX Controls` directory in the *All users profile* in the case of Windows, which should be `C:\ProgramData\REX Controls\REX<version>\RexCore` in most cases.

Following files may be present in the *configuration directory*:

- `license.txt` – A file containing licensing keys for the device. It may be placed manually by a user to the device, but using a REXYGEN Studio or REXYGEN DWM tools for licensing operations is most common.
- `exec.rex` – A binary file with the control algorithm configuration. A user should not touch the file unless he or she knows exactly what to do.
- `hmi.rex` – A binary file with the Human-Machine Interface configuration. A user should not touch the file unless he or she knows exactly what to do.
- `auth.rex` – A binary file with the configuration of users, groups and permissions. A user should not touch the file unless he or she knows exactly what to do.
- `rexcore.cfg` – A textual file for RexCore with all configuration options that are available and which may be adjusted by a user. A detailed options description is provided in section [2.2](#).

## 2.2 Description of Configuration Options

A configuration of RexCore is done by modifying configuration options in the `rexcore.cfg` file. Each configuration option should be on a single separate line and should have a format `option=value` with no additional spaces or tabs. A following table contains descriptions of all configuration options.

<b>Option</b>	<b>Default value</b>	<b>Description</b>
<code>server.tcp</code>	<code>:43981</code>	Address and port on which a diagnostic subsystem of RexCore should listen for incoming TCP connections. Set to "disabled" to disable the service. A wildcard address is used when none is specified which means that listening is performed on all network interfaces.
<code>server.ssl</code>	<code>:43997</code>	Address and port on which a diagnostic subsystem of RexCore should listen for incoming SSL connections. Set to "disabled" to disable the service. A wildcard address is used when none is specified which means that listening is performed on all network interfaces.
<code>server.http</code>	<code>:8008</code>	Address and port on which an integrated web server should listen for incoming HTTP connections. Set to "disabled" to disable the service. A wildcard address is used when none is specified which means that listening is performed on all network interfaces.
<code>server.https</code>	<code>:8009</code>	Address and port on which an integrated web server should listen for incoming HTTPS connections. Set to "disabled" to disable the service. A wildcard address is used when none is specified which means that listening is performed on all network interfaces.
<code>auth.enabled</code>	<code>1</code>	Enable/disable authentication subsystem. User has to be authenticated by a user name and password if the authentication subsystem is enabled. No authentication is required if disabled.
<code>auth.allowsystem</code>	<code>1</code>	Enables or disables the possibility of logging into the REXYGEN system using unix accounts. This option is available on GNU/Linux systems only. See section 5.2 for more information.

<code>auth.allowroot</code>	1	Enables or disables of logging into the REXY-GEN system as an <code>admin</code> using the unix root account. This option is available on GNU/Linux systems only. See section 5.2 for more information.
<code>auth.super.enabled</code>	0	Enables or disables <i>super</i> accounts. See section 5.1 for more information.
<code>auth.operator.enabled</code>	0	Enables or disables <i>operator</i> accounts. See section 5.1 for more information.
<code>auth.guest.enabled</code>	0	Enables or disables <i>guest</i> accounts. See section 5.1 for more information.
<code>log.file.enabled</code>	0	Enables or disables logging into file defined by the option <code>log.file</code> .
<code>log.file</code>	<code>rexcore.log</code>	RexCore log file where all diagnostic messages from the system log are stored if the option <code>log.file.enabled</code> is set.
<code>hmi.file</code>	<code>hmi.rex</code>	HMI file
<code>auth.file</code>	<code>auth.rex</code>	Authentication database file
<code>exec.file</code>	<code>exec.rex</code>	Algorithm configuration file
<code>hmi.path</code>	<code>../www/hmi</code>	A path for additional/static HMI files. The path is absolute or relative to RexCore configuration directory.
<code>archive.path</code>	<code>../arc</code>	A path for binary files with archive data. The path is absolute or relative to RexCore configuration directory.
<code>data.path</code>	<code>../data</code>	A path for blocks and drivers data files. The path is absolute or relative to RexCore configuration directory.
<code>web.webroot</code>	<code>../www</code>	A path for document root of static web files that are persistent on target. The path is absolute or relative to RexCore configuration directory.
<code>perm.file</code>	<code>permem.dat</code>	Persistent memory file
<code>perm.file.size</code>	10240	Size of persistent memory file
<code>perm.file.period</code>	300	Save period of persistent memory file
<code>rexcore.cert</code>	<code>rexcore.cer</code>	RexCore certificate file used for HTTP or diagnostic connections over SSL. The file should be in PEM format.
<code>rexcore.privkey</code>	<code>rexcore.key</code>	RexCore private key file used for HTTP or diagnostic connections over SSL. The file should be in PEM format.

<code>dgn.messages</code>	<code>0xc333370</code>	Diagnostic messages that are shown in system log. User should not set the option directly but rather use REXYGEN Studio to configure the system log. See chapter 3 for more information.
---------------------------	------------------------	--

## 2.3 RexCore Parameters

In the file `/etc/rexcore/rexcore.conf`, it is possible to modify the `OPTIONS` variable to change the parameters passed to RexCore during startup. The list of parameters is provided in the table below.

Parameter	Description
<code>-v</code>	Print RexCore version.
<code>-h, -?</code>	Print help.
<code>-d</code>	Daemonize RexCore.
<code>-k</code>	Terminate running RexCore instance.
<code>-x</code>	Force remove pid file.
<code>-m</code>	Lock all mapped memory to RAM.
<code>-p &lt;file&gt;</code>	Use specified file as pid file.
<code>-c &lt;file&gt;</code>	RexCore configuration file.
<code>-l</code>	Enable logging into console.
<code>-i &lt;file&gt;</code>	Set CPU affinity for RT threads. Default is 0.
<code>-j &lt;file&gt;</code>	Set CPU affinity for diagnostics threads. Default is 0.
<code>-u</code>	Use <code>~/.</code> as the main configuration folder.
<code>-z &lt;opt&gt;</code>	Set configuration option as if set in configuration file.

The command `runrex` can be used to manually launch RexCore from the command line. Output from RexCore will be redirected to standard output. The command `runrex -?` will display the help for all parameters.

## Chapter 3

# System Log

The RexCore has an integrated system log to which all diagnostic messages are stored. Diagnostic messages are errors, warnings or informational messages that may have impact on execution of a control algorithm.

Each message that is written into the system log has an information about a date and time when it was written and flags that indicate a source subsystem and a severity of the message.

### 3.1 Sources and Severities

Following sources of diagnostic messages exists in the REXYGEN system:

- OS - operating system abstraction layer,
- CORE - real-time core,
- DIAG - diagnostic subsystem,
- BLOCK - function block algorithms,
- IODRV - input/output of communication drivers.

Following severities are distinguished:

- Error,
- Warning,
- Information,
- Verbose information.

See [5] for more information on how to configure the system log.



## Chapter 4

# Web Interface

Starting with version 2.50 of the REXYGEN system, a highly optimized web server is integrated directly into RexCore. The server listens on port 8008 for HTTP connection and on port 8009 for HTTPS connections by default. A certificate and a private key must be installed to enable HTTPS connections. See chapter 6 for more information.

### 4.1 Available services

Following services are handled by the integrated web server:

- sending of static file content from the executive configuration files,
- sending of static file content from the directory specified by the `web.webroot` configuration option,
- providing dynamic content over the REST API,
- providing dynamic content over a WebSocket connection,
- handling diagnostic connections over WebSocket.

See [6] and [7] for more information.

## Chapter 5

# Authentication

An authentication subsystem has been integrated into the version 2.50 of the REXYGEN system. Users, roles and permissions of the roles are defined by the authentication subsystem. The authentication subsystem may also be attached to standard accounts from `passwd/shadow` files in the GNU/Linux environment. An engine for PAM and LDAP authentication mechanisms is planned but not yet supported. The user account configuration interface is available in REXYGEN Studio and REXYGEN DWM.

### 5.1 Users and roles

Following roles and permissions are defined:

- *admin* - full permission set, a user with this role may perform any operation and has full control over the target device.
- *super* - a user with this role can not perform any operation that has an impact on the target device ie. reboot the device, set system clock etc., but still may modify the running executive,
- *operator* - a user with this role can not change the running executive but may read and change signal values,
- *guest* - a user with this role can only read signal values and can not make any modifications.

Only the *admin* role is enabled and the roles *super*, *operator* and *guest* are disabled by default. A user may enable those roles by setting options `auth.super.enabled`, `auth.operator.enabled` and `auth.guest.enabled`. For each role a user account with the respective name is present.

The authentication subsystem is enabled by default. A user may disable the authentication by the option `auth.enabled`. The admin has an empty password by default and a user should always set the password after the first login.

A user must provide authentication credentials (name and password) before connecting to the target device from REXYGEN Studio, REXYGEN DWM or REXYGEN HMI Designer or when using the integrated web interface. An error is returned and the connection is closed if invalid credentials are submitted.

## 5.2 Using unix accounts

In the GNU/Linux environment, the authentication subsystem may be attached to standard accounts defined by passwd/shadow files by enabling configuration option `auth.allowsystem`. To put a user into a specific role, following system group must be present and corresponding user account must a member of that group:

- *admin* - `rex-admin`,
- *super* - `rex-super`,
- *operator* - `rex-operator`,
- *guest* - `rex-guest`.

In the GNU/Linux environment, a root system account may also be bound to the admin user of the REXYGEN system by enabling configuration option `auth.allowroot`.

*Example:* To bound a "pi" system account with the "admin" role of the REXYGEN system in the Raspberry Pi device, just create a system group "rex-admin" and add the user "pi" as a member with a shell command "`usermod -a -G rex-admin pi`". It is then possible to login into the REXYGEN system using the user name "pi". The password may be changed with a shell command "`passwd`".

## Chapter 6

# Security

A SSL protocol should always be the choice when communicating with the target device over Internet or untrusted network. A self-signed certificate is automatically created during the installation process on both GNU/Linux and Windows machines. If a self-signed certificate is sufficient, there is nothing else that has to be done.

If a custom certificate and a private key is available, a user has to put them into the RexCore configuration directory. A file with the certificate should be named `rexcore.cer` and a file with the private key should be named `rexcore.key`. Both files should be in the PEM file format. The RexCore service should be restarted every time a file with certificate or private key is changed.

# List of Figures

# Bibliography

- [1] REX Controls s.r.o.. *Function blocks of REXYGEN – reference manual*, 2020. →.
- [2] REX Controls s.r.o.. *Getting started with REXYGEN*, 2020. →.
- [3] REX Controls s.r.o.. *Getting started with REXYGEN on Raspberry Pi*, 2020. →.
- [4] REX Controls s.r.o.. *Getting started with REXYGEN on Debian*, 2020. →.
- [5] REX Controls s.r.o.. *REXYGEN Studio – User manual*, 2020. →.
- [6] REX Controls s.r.o.. *REST API of REXYGEN – Reference Manual*, 2020. →.
- [7] REX Controls s.r.o.. *REXYGEN HMI – User manual*, 2020. →.