



User Manual and Integration Documentation

Contents

1. Product Overview
2. System Requirements and Installation
3. Basic Concept and Operating Modes
4. Recognition Modes and Profiles
5. Local REST API
6. Processing Single Images
7. Batch Processing
8. RTSP and Multi-Camera Operation
9. Configuration File
10. Performance and System Sizing
11. Privacy and Security
12. Licensing and Operation
13. Troubleshooting and Diagnostics
14. Integration Examples
15. Planning Typical Installations
16. Checklists and FAQ

Note: Endpoints and options may be available depending on the license, platform, and installed product edition.

1. Product Overview

CAR-READER®-DLE is a locally operated runtime for automatically recognizing vehicle license plates in images and camera streams. Processing takes place on the customer system. No cloud connection is required for recognition.

1.1 Typical Use Cases

- Access and barrier control
- Parking and parking-space management
- Site access and logistics facilities
- Visitor and fleet management
- OEM solutions for cameras, embedded systems, and industry software
- Local video and security systems

1.2 Product Features

- Local REST API for easy integration
- Single-image, batch, and RTSP processing
- Support for multiple cameras and lanes
- Optimized parallel processing
- Configurable parallel processing
- Profiles for international and two-line license plates
- Optional annotated result images
- Privacy-friendly defaults without permanent storage of uploaded images
- Windows, Linux x64, and Linux ARM64 operation, depending on the package

1.3 Basic Principle

An application sends an image or connects a camera to the runtime. The runtime returns a structured JSON result containing the recognized text, formatted representation, confidence, profile, and layout information. The calling application then decides on barrier operation, logging, database matching, or other business logic.

2. System Requirements and Installation

2.1 Supported Platforms

Platform	Typical Use
Windows x64	Workstation, server, control room, integration into Windows applications
Linux x64	Server, appliance, container-oriented system integration
Linux ARM64	Edge processing for one or a few cameras

2.2 Hardware Recommendations

Scenario	Recommendation
Single camera / occasional images	Modern multi-core processor; at least 8 GB RAM recommended
4 cameras, event-driven	Modern office PC, preferably with 8 or more logical processors and 16 GB RAM
8 cameras, event-driven	High-performance multi-core PC, 16 GB RAM or more; load test recommended
Large installations	High-performance server or multiple runtime instances; project-specific sizing

Achievable performance depends on the processor, image resolution, profile, number of simultaneous events, and required latency. A load test should be performed on the target hardware before production deployment.

2.3 Installation

Windows

Run CAR-READER-DLE_Runtime_Setup.exe as administrator.

Linux

```
sudo bash ./CAR-READER-DLE_Runtime_Setup.run
```

After installation, the runtime runs as a local service by default or is controlled using the supplied start/stop tools.

2.4 Verify Operation

```
curl -s http://127.0.0.1:5099/health
```

A successful response contains the runtime status and basic operating information.

3. Basic Concept and Operating Modes

3.1 Single-Image Mode

An image is evaluated immediately. This mode is suitable for applications that already generate camera images or snapshots.

3.2 Batch Mode

A local folder is processed. This mode is suitable for testing, migrations, quality checks, and larger image collections on the same computer.

3.3 RTSP/Live Mode

The runtime connects to an RTSP stream, keeps current frames available, and can analyze images manually or in response to events. Multiple cameras are managed separately using unique camera_id or lane_id values.

3.4 Single and Multiple License Plates

In standard mode, the best license plate in an image is returned. In multi-plate mode, several visible license plates can be recognized and returned as an ordered list. This mode is suitable, for example, for parking-area overviews or scenes containing several vehicles.

4. Recognition Modes and Profiles

4.1 Profile Overview

Profile	Use	Recommendation
international_latn	General mode for Latin-script license plates	Good default choice for a known, predominantly single-line license plate population
auto	Automatic selection of suitable format and layout logic	Recommended for mixed countries, Dutch plates, or a mix of single- and two-line plates
de	Germany	When only German license plates are expected
at	Austria	When only Austrian license plates are expected
ch	Switzerland	When only Swiss license plates are expected
nl	Netherlands	For Dutch formatting with multiple separators
two_line	Two-line license plates	When the camera predominantly captures two-line license plates

4.2 Selection Guide

- Known single-line license plate format: international_latin or the appropriate country profile.
- Mixed countries and layouts: auto.
- Predominantly two-line license plates: two_line.
- Dutch license plates: nl or auto.
- Highest throughput: choose the most specific profile possible when the application area is clearly defined.

4.3 Formatted and Raw Text

The result can contain both normalized text without separators and the formatted representation. Example: text = BMS1234, formatted_text = B-MS1234. Integrations should generally use normalized text for comparisons and formatted text for display purposes.

4.4 Confidence

Confidence is a value between 0 and 1. Higher values indicate greater recognition certainty. An appropriate threshold depends on image quality and the business process. For automatic barrier decisions, matching against an allowlist of license plates and project-specific validation are also recommended.

5. Local REST API

5.1 Base Address

```
http://127.0.0.1:5099
```

By default, the API is accessible only locally. Remote access should be enabled deliberately and protected with an API token and appropriate network safeguards.

5.2 Authentication

When an API token is enabled, it is sent in one of the following headers:

```
X-CAR-READER®-DLE-Token: <TOKEN>
```

or

```
Authorization: Bearer <TOKEN>
```

5.3 Public Endpoints

Method	Endpoint	Purpose
GET	/health	Runtime and license status
POST	/api/single	Recognize an image
POST	/api/batch	Process a local image folder
GET	/api/license/status	License status
GET	/api/runtime/config	Effective public configuration
GET	/api/runtime/capabilities	Available features and profiles
GET	/api/diagnostics/package	Generate a support package
GET	/shutdown	Shut down the runtime gracefully
POST	/api/rtsp/connect	Connect an RTSP camera
GET	/api/rtsp/status	Read camera status
GET	/api/rtsp/frame	Retrieve the current preview image
POST	/api/rtsp/lane_roi	Set the ROI
POST	/api/rtsp/analyze	Analyze the current image or trigger event
POST	/api/rtsp/disconnect	Disconnect camera(s)
GET	/api/rtsp/cameras	List registered cameras

5.4 HTTP Status Codes

Status	Meaning	Recommended Action
200	Request successful	Evaluate the result
400	Invalid request or no usable result	Check the request and input data
401/403	Authentication failed or access is not permitted	Check the token and access configuration
409	Maximum number of cameras reached	Disconnect a camera or adjust max_cameras
429	An analysis is already running for the same lane	Discard the frame during live video; do not resend it immediately
503	Inference queue full	Wait briefly and retry with backoff

6. Processing Single Images

6.1 Minimal Request

```
curl -s -X POST "http://127.0.0.1:5099/api/single" \  
-F "image=@C:/images/license-plate.jpg" \  
-F "plate_profile=international_latin"
```

6.2 Important Request Fields

Field	Type	Description
image	File, required	JPEG, PNG, BMP, or WebP
plate_profile	Text, optional	Recognition profile; default from configuration
country_profile	Text, optional	Alternative country specification
lane_id / camera_id	Text, optional	Stable identifier for lane/frame protection
use_gap_reconstruction	Boolean, optional	Formatting/separator reconstruction
use_gap_ranker	Boolean, optional	Advanced separator selection

6.3 Typical Response

```
{  
  "ok": true,  
  "text": "BMS1234",  
  "formatted_text": "B-MS1234",  
  "confidence": 0.9931,  
  "plate_layout": "single_line",  
  "line_count": 1,  
  "separator_positions": [1],  
  "country_profile": "INTERNATIONAL_LATIN",  
  "profile": "international_latin",  
  "lane_id": "lane-01"  
}
```

6.4 Multiple Lanes

```
curl -s -X POST "http://127.0.0.1:5099/api/single" \  
-F "image=@lane-01.jpg" \  
-F "lane_id=lane-01" \  
-F "plate_profile=auto"
```

Always use the same stable identifier for each camera or lane. This allows the runtime to discard outdated subsequent frames from the same lane in a controlled manner while other lanes are processed in parallel.

7. Batch Processing

7.1 Example

```
curl -s -X POST "http://127.0.0.1:5099/api/batch" \  
-H "Content-Type: application/json" \  
-d '{"folder_path":"D:/Images/Test","plate_profile":"auto"}'
```

7.2 Typical Options

Option	Description
folder_path	Local folder on the runtime computer
plate_profile	Profile for all images
use_gap_reconstruction	Reconstruct formatted license plates
use_gap_ranker	Advanced separator selection
save_non_matches	Export non-matching results, if enabled
export_char_boxes	Export character-box annotations, if required
allow_fast_path	Allow quality-preserving fast paths

Batch processing operates on local files. For remote systems, images should be transferred via `/api/single` or copied to the runtime computer beforehand.

8. RTSP and Multi-Camera Operation

8.1 Connect a Camera

```
curl -s -X POST "http://127.0.0.1:5099/api/rtsp/connect" \  
-H "Content-Type: application/json" \  
-d '{"camera_id": "lane-01", "profile_name": "Entrance 1", "rtsp_url": "rtsp://camera/stream"}'
```

8.2 Provide Credentials Separately

```
{  
  "camera_id": "lane-01",  
  "rtsp_url": "rtsp://camera/stream",  
  "username": "USER",  
  "password": "PASSWORD"  
}
```

Credentials should not be written to publicly accessible scripts, source code, or logs.

8.3 Analyze a Camera

```
curl -s -X POST "http://127.0.0.1:5099/api/rtsp/analyze" \  
-H "Content-Type: application/json" \  
-d '{"camera_id": "lane-01", "auto_trigger": true, "plate_profile": "international_latin"}'
```

8.4 List Cameras and Check Status

```
curl -s "http://127.0.0.1:5099/api/rtsp/cameras"  
curl -s "http://127.0.0.1:5099/api/rtsp/status?camera_id=lane-01"
```

8.5 Use an ROI

A Region of Interest (ROI) limits the relevant roadway area. This reduces irrelevant image regions and allows live analysis to operate more efficiently and reliably. The ROI should fully cover the expected license plate position and include a sufficient margin.

8.6 Event-Driven Operation

For live video, full recognition should not be performed on every video frame. A preliminary check for motion or license plate events is recommended. While a lane is already being analyzed, additional frames from the same lane can be discarded. This keeps the queue current and enables faster responses to new vehicles.

9. Configuration File

9.1 Basic Rules

- Back up the configuration before making changes.
- The JSON syntax must remain valid.
- Changes generally take effect after the runtime is restarted.
- Change only documented customer-relevant options.
- After performance changes, check the health endpoint and run a load test.

9.2 Main Sections

Section	Purpose
recognition	Default profile, minimum length, multiple license plates
grouping	Separator and formatting logic
output	Return annotations and detailed data
security	Local/remote access and API protection
live_trigger	Camera triggering and maximum camera count
deployment_profile	Description of the intended deployment profile

9.3 recognition

Option	Default/Example	Description
plate_profile	international_latin	Default profile when the request does not specify one
min_plate_length	2	Minimum accepted text length
preserve_umlauts	true	Preserve umlauts in the output
restore_known_umlaut_prefixes	true	Restore known prefixes for readability
multi_plate.enabled	false	Enable multiple license plates
multi_plate.mode	single_best / all_visible	Best license plate or all visible license plates
multi_plate.max_plates	8	Maximum number of returned license plates
multi_plate.min_ocr_confidence	0.5	Minimum confidence for multi-plate results

9.4 grouping

Option	Description
enabled	Enable formatted output and separator reconstruction
use_ranker	Improve the selection of plausible separator positions
separator	Default separator, usually a hyphen
multi_separator_enabled	Allow multiple separators, e.g. for the Netherlands
country_profile_formatting_enabled	Apply country-specific formatting

9.5 output

Option	Recommendation
include_annotated_image	Enable only when the image is actually required
include_plate_details	Usually true for integrations
include_image_payloads	Set to false for bandwidth or privacy requirements

9.6 security

Option	Description
allow_remote_clients	Deliberately enable or disable remote access
require_api_token_for_remote	Always recommended as true for remote access
restrict_runtime_acl	Restrict runtime files to authorized accounts
delete_secure_session_on_shutdown	Remove temporary runtime data on shutdown

9.7 live_trigger

Option	Description
enabled	Enable live triggering
mode	Runtime trigger strategy
max_cameras	Maximum number of registered RTSP cameras
idle_rearm_seconds	Time until the system is ready again after an event
duplicate_cooldown_seconds	Temporarily suppress repeated results
recent_frame_buffer_size	Number of recently retained frames
partial_plate_reject_enabled	Reduce incomplete license plate events

The additional live-trigger values are used to fine-tune specific camera scenarios. Changes should be tested with real streams. For typical installations, the supplied default values are a good starting point.

10. Performance and Presets

10.1 Important Principles

- The Runtime manages parallelism, resource limits, and image preparation automatically..
- For live video, run full recognition in response to events, not for every video frame.
- A suitable ROI reduces irrelevant image areas and stabilizes live processing.
- Use an appropriate country or layout profile for a known plate population than auto.
- Request annotations, JPEG, and Base64 image data only when actually required.

10.2 System Sizing

For typical installations with four to eight cameras, the video frame rate is not the decisive factor; the number of simultaneous vehicle events is. Four cameras at 25 FPS each do not automatically generate 100 OCR jobs per second. Full recognition should be triggered only by an appropriate event.

Situation	Required Average Throughput
4 events in 2 seconds	2 recognitions/s
4 events in 1 second	4 recognitions/s
8 events in 2 seconds	4 recognitions/s
8 events in 4 seconds	2 recognitions/s

10.7 Evaluating a Load Test

- successful equals the requested number
- failed = 0
- no HTTP 503 responses during normal target operation
- HTTP 429 may be normal when intentionally dropping frames from the same lane
- p95 latency is within the process target
- the queue does not remain permanently full
- sufficient memory remains available and the system does not swap heavily

11. Privacy and Security

11.1 Privacy-Friendly Default Configuration

The default configuration can discard uploaded and annotated images, exclude license plate text from customer logs, and create diagnostic packages without images. Operators are nevertheless responsible for legal compliance, informing data subjects, retention periods, and access controls.

11.2 Recommended Measures

- Make the API available locally or within a protected network wherever possible.
- Enable remote access only with a token and firewall rules.
- Store license plates only when required by the business process.
- Limit retention periods.
- Assign access and administrative rights according to the principle of least privilege.
- Manage RTSP credentials securely.
- Handle support packages according to company policy before sharing them.

11.3 Network Operation

For external clients, the runtime should not be exposed directly to the internet without protection. An internal network, VPN, or upstream reverse proxy with TLS and access control is recommended.

12. Licensing and Operation

12.1 Check License Status

```
curl -s http://127.0.0.1:5099/api/license/status
```

The endpoint indicates whether the installed license is valid and which enabled features are available.

12.2 Hardware or System Changes

A license may be bound to the licensed installation. Before replacing hardware, reinstalling the operating system, or migrating, coordinate the license transfer with the provider.

12.3 Updates

- Back up the configuration before updates.
- Review the release notes included with the customer package.
- After an update, run health, license, and recognition tests.
- If performance has changed, retest the preset on the target hardware.

13. Troubleshooting and Diagnostics

13.1 Health Endpoint

```
curl -s http://127.0.0.1:5099/health
```

Important public fields are status, license_ok, uptime_seconds, rtsp_camera_count, and rtsp_connected_count.

13.2 Typical Issues

Symptom	Possible Cause	Action
Runtime not accessible	Service not started or incorrect port	Check the start tool/service and firewall
401/403	Token is missing or incorrect	Check the header and configuration
429 LANE_BUSY_FRAME_DROPPED	A job is already running for the same lane	Normal for live video; send a current frame later
503 INFERENCE_QUEUE_FULL	Processing capacity is temporarily saturated	Retry after retry_after_seconds with backoff or reduce load
RTSP not connected	URL, credentials, or network problem	Check the stream with a player, and verify the network and credentials
High latency	Many simultaneous events, large images, or the auto profile	Review ROI, image size, profile, and target hardware
High RAM usage	Many cameras, large images, or high event density	Reduce load or provide more memory

13.3 Support Package

```
curl -s http://127.0.0.1:5099/api/diagnostics/package
```

The support package contains customer-specific operating information for troubleshooting. Whether images are included depends on the privacy configuration.

14. Integration Examples

14.1 Python

```
import requests

with open("license-plate.jpg", "rb") as image_file:
    response = requests.post(
        "http://127.0.0.1:5099/api/single",
        files={"image": image_file},
        data={
            "plate_profile": "international_latin",
            "lane_id": "lane-01"
        },
        timeout=30
    )
response.raise_for_status()
result = response.json()
print(result.get("formatted_text"), result.get("confidence"))
```

14.2 C#

```
using var client = new HttpClient();
using var content = new MultipartFormDataContent();
content.Add(new ByteArrayContent(File.ReadAllBytes("license-plate.jpg")),
    "image", "license-plate.jpg");
content.Add(new StringContent("international_latin"), "plate_profile");
content.Add(new StringContent("lane-01"), "lane_id");

var response = await client.PostAsync(
    "http://127.0.0.1:5099/api/single", content);
response.EnsureSuccessStatusCode();
var json = await response.Content.ReadAsStringAsync();
```

14.3 JavaScript

```
const form = new FormData();
form.append("image", file);
form.append("plate_profile", "auto");
form.append("lane_id", "lane-01");

const response = await fetch("http://127.0.0.1:5099/api/single", {
  method: "POST",
  body: form
});
const result = await response.json();
```

14.4 Retry Behavior

For HTTP 503, the client should retry after a short, increasing delay. For HTTP 429 and live frames, the discarded old frame should normally not be sent again; use the next current frame instead.

15. Planning Typical Installations

15.1 One Camera

- Use a stable camera position, suitable ROI, and specific profile
- Use a specific profile
- Optimize the ROI and image crop
- Enable annotations only for the user interface

15.2 Four Cameras

- Test simultaneous events from all cameras in a burst test
- Assign each camera a stable camera_id
- Use event-driven analysis
- Perform a burst test with four simultaneous requests

15.3 Eight Cameras

- Test the target hardware with realistic eight-camera load peaks
- Test 4 x 2 only when sufficient CPU resources and RAM are available

15.4 Special Case: Up to 30 Cameras

A large number of connected cameras is possible when full recognition is triggered only by events. For continuously high event density, plan for multiple Runtime instances or additional systems.

15.5 Acceptance Test

1. Check the health and license status.
2. Test at least 100 representative images for each camera scenario.
3. Simulate a four-/eight-lane burst.
4. Test camera operation over several hours.
5. Document p95 latency, queue behavior, error codes, and RAM usage.
6. Approve the privacy and retention settings.

16. Checklists and FAQ

16.1 Quick-Start Checklist

- Runtime installed and started
- /health returns status ok
- License status is valid
- Test image processed successfully via /api/single
- Appropriate profile selected
- Privacy settings reviewed
- Camera IDs are unique
- Performance preset tested on the target hardware
- Client handles 429 and 503 correctly
- Support process documented

16.2 Frequently Asked Questions

Does the runtime need to be connected to the internet?

No. License plate recognition operates locally and offline. Network access may be required for camera streams or communication with the customer application.

Can the runtime process multiple cameras in parallel?

Yes. Multiple cameras can operate in parallel. Appropriate system sizing depends on image resolution, event density, profile, and target hardware.

Why are frames discarded with HTTP 429 during live video?

If an analysis is already running for the same lane, an older subsequent frame may be deliberately discarded. This keeps processing current and prevents the queue from filling with outdated frames.

Which profile is best?

For a clearly defined license plate population, a specific country or layout profile is often the most efficient. For mixed license plates, auto is the most convenient choice.

Can I receive annotated images?

Yes, provided image output is enabled and requested. For backend-only integrations, annotations should remain disabled for performance and privacy reasons.

How many cameras can an office PC support?

Typical installations with four to eight event-driven cameras are possible on suitable modern office PCs. A reliable assessment requires testing on the specific system.