

Semaphore Software LLC



ParseLogsToDB

User's Manual

November 11, 2025

Table of Contents

1	Introduction	3
2	Requirements.....	3
3	Platform Support	3
4	Initial Setup	3
4.1	Obtain and Unpack the Software.....	3
4.2	Verify Functionality	3
5	Program Overview	4
6	Information Commands.....	4
6.1	help	4
6.2	version.....	4
7	Core Commands.....	5
7.1	check-new	5
7.2	deactivate.....	5
7.3	license	5
7.4	process	6
8	JSON File Format and Options	7
8.1.1	Required Keys	7
8.1.2	Optional Keys	7
8.1.3	Optional Email Keys	7
8.1.4	Template Config File	9
8.1.5	JSON File Guidelines.....	9
9	Log Files and Execution Report Email	10

1 Introduction

ParseLogsToDB is a command-line interface (CLI) application for parsing signal logs and inserting the information into a perpetual database table. The application uses a configurable JSON file to provide the required configuration.

2 Requirements

- Operating System: Windows 8/10/11.
- Database: An existing database in which the program can perform CRUD operations.
- License: Valid ParseLogsToDB license key.
- Disk Space: Around 175 MB.
- Installation: No installer. Unzip files and run the executable.

3 Platform Support

ParseLogsToDB can parse logs from the following platforms and log types.

1. ElectroLogIXS
 - a. Crossing Events
 - b. Data

4 Initial Setup

4.1 Obtain and Unpack the Software

ParseLogsToDB will be provided in a ZIP file containing a Revision History, End User License Agreement, this User's Manual, and the files needed to run the ParseLogsToDB executable.

Unpack the contents of the ZIP file to an empty folder (recommended to be named ParseLogsToDB).

Review the End User License Agreement, the entirety of the User's Manual, and other documentation before proceeding.

4.2 Verify Functionality

Open a command prompt window (Windows key **"cmd"** + Enter) and change directory to the folder that contains the ParseLogsToDB executable (**cd "[YOUR_FILE_PATH]"**).

Enter the command **ParseLogsToDB --version** and confirm the current version number is displayed.

```
C:\[YOUR_FILE_PATH]>ParseLogsToDB --version
ParseLogsToDB.exe, version 1.0.0
```

5 Program Overview

ParseLogsToDB is used to parse previously downloaded signal log files and insert them into a database table based on a configuration provided through a JSON file. All program functionality is controlled through a CLI (Command Line Interface) using commands whose definitions or usage are provided below as well as through the `--help` flag that may be placed after any of the commands to provide usage details.

The program requires a valid license key which must be provided through the `license` command. Once input, the key will be retained indefinitely. The license key can be removed from a device by using the `deactivate` command, which will free up the license for use on a different device.

The user can check if there is an updated version of ParseLogsToDB available by using the `check-new` command.

The main program function is invoked by using the `process` command in conjunction with the name of a JSON configuration file.

The details for usage of all these commands can be found in their own section of this manual, or as previously mentioned by appending the `--help` flag behind any of these commands in the CLI.

6 Information Commands

6.1 help

The `--help` flag will display information and list all available commands for whatever it is appended to. In the case of using `--help` after invoking the CLI directly, information will be provided about the core commands available.

```
C:\[YOUR_FILE_PATH]>ParseLogsToDB --help
Usage: ParseLogsToDB.exe [OPTIONS] COMMAND [ARGS]...

Options:
  --version  Show the version and exit.
  --help     Show this message and exit.

Commands:
  check-new
  deactivate
  license
  process
```

6.2 version

The `--version` flag will display the version of ParseLogsToDB being run. It cannot be called with any other commands.

7 Core Commands

7.1 check-new

Check if a newer version of ParseLogsToDB is available.

```
C:\[YOUR_FILE_PATH]>ParseLogsToDB check-new
New version check initiated.
Application is currently up-to-date.
```

7.2 deactivate

Deactivate the license on the current machine to allow use on another machine.

Options:

- **--force** – Skip confirmation prompt and force deactivation

```
C:\[YOUR_FILE_PATH]>ParseLogsToDB deactivate

=====
LICENSE DEACTIVATION WARNING
=====
This will permanently remove the license from this machine.
After deactivation:
  • All license files will be deleted
  • You will need to re-enter your license key to use this application
  • The license can be activated on a different machine
=====

Do you want to proceed with deactivation? [y/N]: y
Initiating license deactivation...
The deactivation was successful.
```

7.3 license

Add and validate a license key.

```
C:\[YOUR_FILE_PATH]>ParseLogsToDB license ABCDE-01234-FGHIJ-56789
Adding license with key: *****-*****-*****-56789
License ACTIVE. Expiration: Valid forever!
```

7.4 process

Parses logs using information from the provided configuration JSON file into a database table.

```
C:\[YOUR_FILE_PATH]>ParseLogsToDB process [PATH_OF_CONFIG_FILE.JSON]  
License ACTIVE. Expiration: Valid forever!  
Processed '1.1.1.1 Location Name Download Date.log': 111999 records
```

8 JSON File Format and Options

The configuration file must be of the JSON format (ending with a .json suffix) and follow the [JSON format](#). The file must contain all the required key value pairs and may contain some or all of the optional key value pairs to supplement functionality. The keys and the value types are explained below with an example of the content of a JSON file in the [Template Config File](#) section.

8.1.1 Required Keys

The JSON file must contain these keys (case-sensitive) with a matching data value:

- **log_path** – The path of a folder containing the log files which are to be parsed. The complete path should be provided including escape characters.
 - Example: `"C:\\Users\\name\\sub_folder_name\\test_logs_folder"`
- **logs_type** – One of the following allowable log types. Because different log types are placed in different database table configurations, only one log type may be processed during each run of the program. Separate configuration files may be set up and run at different times for different log types.
 - `"data"`
 - `"crossing_system_event"`
- **database_name** – The name of the database in which the table will be stored.
- **table_name** – The name of the table in which the data will be stored.
- **server_driver** – The server type and driver combination.
 - Example: `"mssql+pyodbc"`
- **server_path** – The path at which to access the SQL database server
 - Example: `"localhost\\SQLEXPRESS"`
- **server_query** – Any additional server connection parameters that may be required to successfully access the database server.
 - Example:

```
{
  "driver": "ODBC Driver 18 for SQL Server",
  "TrustServerCertificate": "Yes",
  "Trusted_Connection": "Yes"
}
```

8.1.2 Optional Keys

- **server_username** – Username for server login, if required.
- **server_password** – Password for server login, if required.

8.1.3 Optional Email Keys

These keys are associated with the option to receive script execution reports via email after the execution of any **process** command.

8.1.3.1 Universal Email Keys

These keys are always required to access the email execution report functionality.

- **email_backend** – There are different methods of sending email, and this parameter must be set to determine which should be used. If this parameter is present and configured, an error message will be provided for any missing support parameters for the selected backend. Backend options are:
 - **smtp**
 - **mailtrap**
- **email_from_addr** – The address from which the email will originate.
- **email_to_addrs** – A list of email addresses to which the email will be sent.

8.1.3.2 SMTP Email Keys

These keys are required when using the SMTP backend.

- **email_pw** – The password for the email address from which the email will originate.
- **smtp_server** – The address of the SMTP server that will send the email.
- **smtp_port** – An optional key to configure the SMTP port (defaults to 587 if unused).

8.1.3.3 Mailtrap Email Keys

These keys are required when using the Mailtrap backend.

- **mailtrap_token** – A personal Mailtrap token to utilize the service for the **email_from_addr** provided.

8.1.4 Template Config File

```
{
  "_comment": "This is a template config file. Copy and fill in
your values.",

  "logs_path": "/path/to/your/logs",
  "logs_type": "data",
  "database_name": "YourDatabaseName",
  "table_name": "YourTableName",
  "server_driver": "mssql+pyodbc",
  "server_path": "localhost\\SQLEXPRESS",
  "server_query": {
    "driver": "ODBC Driver 18 for SQL Server",
    "TrustServerCertificate": "Yes",
    "Trusted_Connection": "Yes"
  },

  "_email_comment": "Optional: Email notifications - choose ONE
backend below",

  "_smtp_option": "Option 1: SMTP",
  "email_backend": "smtp",
  "email_from_addr": "sender@yourdomain.com",
  "email_to_addrs": ["recipient@example.com"],
  "email_pw": "YOUR_EMAIL_PASSWORD",
  "smtp_server": "smtp.yourdomain.com",
  "smtp_port": 587,

  "_mailtrap_option": "Option 2: Mailtrap",
  "email_backend": "mailtrap",
  "email_from_addr": "sender@yourdomain.com",
  "email_to_addrs": ["recipient@example.com"],
  "mailtrap_token": "YOUR_MAILTRAP_TOKEN"
}
```

8.1.5 JSON File Guidelines

- Keys are case-sensitive and must match exactly.
- Empty values are allowed for optional fields.
- Extra or unexpected keys will be ignored.

9 Log Files and Execution Report Email

ParseLogsToDB keeps a running log of execution success or failure. This log is saved in the installation folder as **debug.log**. This file contains detailed execution information for each time the CLI is run and includes error traces for troubleshooting. The developer may request this file be provided in instances where additional troubleshooting assistance is needed.

Because the user may be interested in receiving information about the success or failure of each run, the option is given to have the program send an Execution Report via email following the completion of the program execution. The email sent provides only information for the immediately preceding run of the program.