User and DB Manager

Release Note 1.0.0

User and DB Manager Release Note 1.0.0

Copyright © …... to be added

TRADEMARKS:   to be added

DISCLAIMER to be added

DOCUMENT No.:

Contents

1. [Introduction 5](#_Toc482295968)

[1.1 Purpose 5](#_Toc482295969)

[1.2 Scope 5](#_Toc482295970)

1. [Release Description for Release No 1.0.0 6](#_Toc482295971)

[2.1 Release Specifications 6](#_Toc482295972)

[2.1.1 System Requirements 6](#_Toc482295973)

[2.1.2 Software Requirements 6](#_Toc482295974)

[2.2 Release Features 8](#_Toc482295975)

[2.3 Release Restrictions 8](#_Toc482295976)

[2.4 Freeware Used 8](#_Toc482295977)

1. [Compilation and Installation of User and DB Manager 9](#_Toc482295978)

[3.1 Product Packaging 9](#_Toc482295979)

[3.1.1 Binary code 9](#_Toc482295980)

[3.1.2 Installation procedure 9](#_Toc482295981)

[3.2 Instruction to update file (user\_db\_mgr\_config.conf): 10](#_Toc482295982)

[3.2.1 Testing with test\_db\_manager program: 11](#_Toc482295983)

[3.2.2 List of commands used by test\_db\_manager: 11](#_Toc482295984)

[3.3 Source code 12](#_Toc482295985)

[3.4.1 Compilation Procedure 12](#_Toc482295986)

[3.4.2 User & DB Manager Source File Structure: 13](#_Toc482295987)

[3.4.3 Source File Description: 13](#_Toc482295988)

[3.5 JSON Command Format Examples: 14](#_Toc482295989)

Chapter

# Introduction

## 1.1 Purpose

This document provides release-specific information for User and DB manager tool and testing procedure to test the User and DB manager program.

## 1.2 Scope

The scope of this document is to detail information about the features of User and DB manager tool, compilation of source code and installation.

Chapter

# Description

This chapter describes the system requirements, software requirements and features of the User and DB manager.

## 2.1 Release Specifications

### 2.1.1 System Requirements

User and DB manager is supported on Ubuntu different versions The following table lists the system requirements according to the OS on which User and DB manager has been tested.

| OS | Requirements |
| --- | --- |
| Linux | Computer System with Ubuntu 12.04 installed as OS. |
| Linux | Computer System with Virtual box 5.01 with Ubuntu 16.04 installed as OS.  **(Picarro PI2000)** |

### 2.1.2 Software Requirements

The following freeware software libraries are required to implement the features of User and DB manager:

|  |  |  |  |
| --- | --- | --- | --- |
| S.N. | Library Name | Version | Usage in Software Module |
| 1 | Zero MQ | 4.2.2 | ZMQ sever |
| 2 | JSON C Parser | 0.12.99 | JSON Parser |
| 3 | SQLITE3 | 3.7.15.2 | DB Management |
| 4 | LIBHARU | 2.4.0 | PDF Generator |
| 5 | GNUPLOT | 5.0 | Graph Generator Utility |
| 6 | LIBGD | 3.0 | Graphics |
| 7 | LIBPNG | 1.2.4 | PNG Image |
| 8 | LIBZ | 1.2.3 | Libpng dependency |

### 2.2 Release Features

The major features provided in the release 1.0.0 of the User and DB manager are listed below:

* User management in database table.
* Surrogate gas table and calibration table database management.
* Report generation with surrogate gas table and plotting X-Y linear scatter graph.

The following JSON request command has been handled in the User and DB manager.:

* createValidationReport
* loginUser
* logoutuser
* getCurrentUser
* addUser
* deleteUser
* listAllUsers

### 2.3 Release Restrictions

The current release has the following restrictions:

* Configuration file **(user\_db\_mgr\_config.conf)** should be filled properly before executing the user\_db\_manager program.
* Directory should exist which would be mentioned in configuration file.
* Invalid JSON command format are not executed by the User and DB manager.
* Only single user can login at a time.
* To generate validation report user should be logged-in.
* Latest directory folder name should be passed in createValidationReport command.

Chapter

# Compilation and Installation

## 3.1 Product Packaging

### 3.1.1 Binary code

The following table contains the details of the Binary code.

|  |  |  |  |
| --- | --- | --- | --- |
| Filename | Description | Size | Date and Time stamp |
| *user\_db\_manager\_bin\_1.0.0.tar.gz* | Contains Binary package for Linux. | 6,847 KB | May 11, 2017, 6:00:56 PM |
| *user\_db\_install.sh* | Installation script for Application. | 2 KB | May 11, 2017, 6:00:56 PM |

## 3.1.2 Installation procedure

1. Extract the **user\_db\_manager\_bin\_1.0.0.tar**.gz file.
2. Open the file **user\_db\_mgr\_config.conf** and add the path of ppm data files in this config file.
3. Follow the instruction given in next heading (3.2) to update the **user\_db\_mgr\_config.conf** file.
4. The config file (user\_db\_mgr\_config.conf) is provided for the flexibility of report generation.
5. Run the script **user\_db\_install.sh.**
6. Restart the system.
7. On system boot up **user\_db\_manager** program would be running in background.
8. Program running can be checked with command:

**ps -ax | grep user\_db\_manager**

1. Now run the client utility **(./test\_db\_manager)** to communicate with User and DB manager program.

## 3.2 Instruction to update config file (user\_db\_mgr\_config.conf):

1. Do not add blank space/tab at the beginning of any line.
2. Always start a comment with # initials.
3. Config file must be filled with valid values for first four keys: FILE\_PATH, HEADER, FILE\_PREFIX, FILE\_COUNT.

**KEY DEFINITIONS:**

* **FILE\_PATH**:

Path to the calibration data files before the date folder name. Date folder name would be passed as parameter in careaValidationReport command.

* **HEADER:**

This is the header name of GAS for which validation report would be generated. This can be changed to H2O or H2O2 etc.

* **FILE\_PREFIX:**

This is the prefix to add in the file name while creating the .dat file names E.g. NBDS2039-20170222-ch4\_100ppm.dat.

* **FILE\_COUNT:**

This count will contain the number of nominal concentration defined in this configuration file. This will be used to plot X-axis data points on the graph. The concentration key=value pair and FILE\_COUNT should be matched.

* **KEY=VALUE pair:**

This will define nominal concentration as a KEY=VALUE pair. The KEY would be used to create/compare with the file name available in the calibration file folder.

**Example:**

* + If ppm data files for CH4 gas are in directory /home/ch4-ppm-calibration//20170305/ • Add the config file keys as:
  + FILE\_PATH=/home/ch4-ppm-calibration/
  + HEADER=CH4
  + FILE\_PREFIX=NBDS2039-
  + FILE\_COUNT=4
  + zero\_air=0.0
  + 2ppm=2.058
  + 10ppm=10
  + 100ppm=100.2
  + Pass the latest validation directory name (20170305) as json request from test\_db\_manager application in createValidationReport command. E.g.: {\"command\": \"createValidationReport\", \"validationTime\": \"170424 02:58\", \"validationDirName\": \"20170222\"}

## 3.2.1 Testing with test\_db\_manager program:

* Run client program (test\_db\_manager) on the terminal
* Run List user.
* Run ADD user command.
* Run List user.
* Run Login user.
* Run Get current user.
* Run Create validation report.
* Run Logout user.
* Run Get current user.
* Check the DB file for updates.

**Path of generated database file and copied user\_db\_mgr\_config.conf file:**

* /use/local/bin/user\_db\_config/user\_db\_manager.db
* /use/local/bin/user\_db\_config/ user\_db\_mgr\_config.conf

**Path of generated report:**

* + /home/ch4-ppm-calibration/Report/<creation date>-Validation.pdf

e.g. May-11-2017-Validation.pdf

## 3.2.2 List of commands used by test\_db\_manager:

**1. Add User Commands:**

{\"command\": \"addUser\", \"username\": \"user1\", \"role\": \"admin\", \"password\": \"password\"} {\"command\": \"addUser\", \"username\": \"user2\", \"role\": \"operator\", \"password\": \"password\"} {\"command\": \"addUser\", \"username\": \"user3\", \"role\": \"operator\", \"password\": \"password\"} {\"command\": \"addUser\", \"username\": \"user4\", \"role\": \"operator\", \"password\": \"password\"}

**2. DELETE Commands:**

{\"command\": \"deleteUser\", \"username\": \"user1\", \"role\": \"admin\"}

{\"command\": \"deleteUser\", \"username\": \"user2\", \"role\": \"operator\"}

{\"command\": \"deleteUser\", \"username\": \"user3\", \"role\": \"operator\"}

{\"command\": \"deleteUser\", \"username\": \"user4\", \"role\": \"operator\"}

**3. Login Commands:**

{\"command\": \"loginUser\", \"username\": \"user1\", \"password\": \"password\"}

{\"command\": \"loginUser\", \"username\": \"user2\", \"password\": \"password\"}

{\"command\": \"loginUser\", \"username\": \"user3\", \"password\": \"password\"}

{\"command\": \"loginUser\", \"username\": \"user4\", \"password\": \"password\"}

**4. List User Command:**

{\"command\": \"listAllUsers\"}

**5. Log Out User Command:**

command\": \"logoutUser\"}

**6. Current user Command:**

{\"command\": \"getCurrentUser\"}

**7. Report Generation Command:**

{\"command\": \"createValidationReport\", \"validationTime\": \"170424 02:58\", \"validationDirName\": \"20170222\"}

### 3.3 Source code

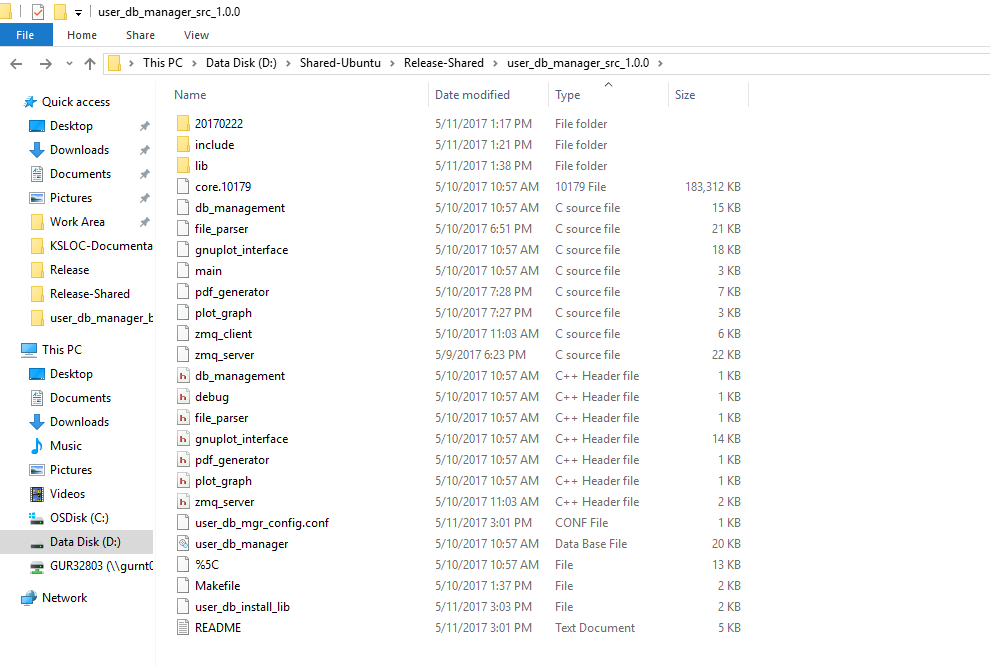
The following table contains the details of the Source code.

|  |  |  |  |
| --- | --- | --- | --- |
| Filename | Description | Size (Bytes) | Date and Time stamp |
| user\_db\_manager\_src\_1.0.0.tar.gz | User and DB Manager 1.0.0 source code | 11290 KB | May 11, 2017, 6:00:56 PM |
| *user\_db\_install\_lib.sh* | Script for copying libraries needed for compilation. | 2 KB | May 11, 2017, 6:00:56 PM |

## 3.4.1 Compilation Procedure

1. Extract the **user\_db\_manager\_src\_1.0.0.tar**.gz file.
2. Open the file **user\_db\_mgr\_config.conf** and add the path of ppm data files in this config file.
3. Follow the instruction to update the **user\_db\_mgr\_config.conf** file.
4. This config (user\_db\_mgr\_config.conf) file is provided for the flexibility of report generation.
5. Run the script **user\_db\_install\_lib.sh.**
6. Now build the directory with following commands:
   1. **make clean**
   2. **make**
   3. **make install**
7. Restart the system.
8. On system boot up **user\_db\_manager** program would be running in background.
9. Program running can be check with command:
   1. **ps -ax | grep user\_db\_manager**
10. Now run the client utility **(./test\_db\_manager)** to communicate with User and DM manager program.

## 3.4.2 User & DB Manager Source File Structure:



## 3.4.3 Source File Description:

|  |  |  |
| --- | --- | --- |
| S.N. | Source File Name | Description |
| **1** | main.c | Contains the main entry point function for User and DB Manager program. Create the server thread program. |
| **2** | zmq\_server.c | Creates the ZMQ server program and parse the json request commands and execute them. |
| **3** | zmq\_client.c | Creates the ZMQ client program and send the json request commands to server program. |
| **4** | db\_management.c | Creates and executes database queries. |
| **5** | file\_parser.c | Parse the config file, extract the calibration data values and perform calculation on those values to fill in the database file. |
| **6** | gnuplot\_interface.c | Run as utility program to plot the X-Y graph. |
| **7** | pdf\_generator.c | Generates the final PDF report file. |
| **8** | plot\_graph.c | Generate the X-Y graph PNG file that further used by the pdf generator. |

## 3.5 JSON Command Format Examples:

**Add User Commands:**

{\"command\": \"addUser\", \"username\": \"user1\", \"role\": \"admin\", \"password\": \"password\"}

**DELETE Commands:**

{\"command\": \"deleteUser\", \"username\": \"user1\", \"role\": \"admin\"}

**Login Commands:**

{\"command\": \"loginUser\", \"username\": \"user1\", \"password\": \"password\"}

**List User Command:**

{\"command\": \"listAllUsers\"}

**Log Out User Command:**

command\": \"logoutUser\"}

**Current user Command:**

{\"command\": \"getCurrentUser\"}

**Report Generation Command:**

{\"command\": \"createValidationReport\", \"validationTime\": \"170424 02:58\", \"validationDirName\": \"20170222\"}