USER GUIDE

FOR ADB STATISTICAL BUSINESS REGISTER

DECEMBER 2018

ASIAN DEVELOPMENT BANK

ADB

USER GUIDE

FOR ADB STATISTICAL BUSINESS REGISTER

DECEMBER 2018

ASIAN DEVELOPMENT BANK

ADB

Creative Commons Attribution 3.0 IGO license (CC BY 3.0 IGO)

© 2018 Asian Development Bank
6 ADB Avenue, Mandaluyong City, 1550 Metro Manila, Philippines
Tel +63 2 632 4444; Fax +63 2 636 2444
www.adb.org

Some rights reserved. Published in 2018.

ISBN 978-92-9261-386-0 (print), 978-92-9261-387-7 (electronic)
Publication Stock No. TIM189625-2
DOI: http://dx.doi.org/10.22617/TIM189625-2

The views expressed in this publication are those of the authors and do not necessarily reflect the views and policies of the Asian Development Bank (ADB) or its Board of Governors or the governments they represent.

ADB does not guarantee the accuracy of the data included in this publication and accepts no responsibility for any consequence of their use. The mention of specific companies or products of manufacturers does not imply that they are endorsed or recommended by ADB in preference to others of a similar nature that are not mentioned.

By making any designation of or reference to a particular territory or geographic area, or by using the term "country" in this document, ADB does not intend to make any judgments as to the legal or other status of any territory or area.

This work is available under the Creative Commons Attribution 3.0 IGO license (CC BY 3.0 IGO) https://creativecommons.org/licenses/by/3.0/igo/. By using the content of this publication, you agree to be bound by the terms of this license. For attribution, translations, adaptations, and permissions, please read the provisions and terms of use at https://www.adb.org/terms-use#openaccess.

This CC license does not apply to non-ADB copyright materials in this publication. If the material is attributed to another source, please contact the copyright owner or publisher of that source for permission to reproduce it. ADB cannot be held liable for any claims that arise as a result of your use of the material.

Please contact pubsmarketing@adb.org if you have questions or comments with respect to content, or if you wish to obtain copyright permission for your intended use that does not fall within these terms, or for permission to use the ADB logo.

Notes:
In this publication, "$" refers to United States dollars.
Corrigenda to ADB publications may be found at http://www.adb.org/publications/corrigenda.
Microsoft ® Windows, Microsoft ® Excel, Microsoft Server ® are registered trademarks of Microsoft Corporation.
MySQL is a registered trademark of MySQL AB in the United Stated, the European Union and other countries.
Laravel is a trademark of Taylor Otwell.
Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.
Apache, OpenOffice are registered trademarks of The Apache Software Foundation.
All other trademarks referenced herein are used solely for identification purposes and are the properties of their respective owners.
All names, businesses, and data presented in the ADB SBR system screenshots or mentioned in the publication are all randomly generated and fictitious. Any resemblance to actual persons, or businesses is purely coincidental.

Printed on recycled paper

Contents

FIGURES v
FOREWORD vi
ACKNOWLEDGMENTS vii
ABBREVIATIONS viii
PROJECT OVERVIEW
System Features ix
Uses of the Statistical Business Register x
Involvement of Partner Developing Member Countries xi

I. SYSTEM DEVELOPMENT

I.A. Project Management 1
I.B. Software Development 1
I.C. Solution Architecture 2
I.D. Database 3
I.E. Database Management System 4
I.F. Programming Requirements 4
I.G. IT Environments 5
I.H. Programming Languages 8
I.I. Relationships with Other Systems and Registers 10
I.J. Unique Identifiers 11
I.K. Tools and Software for Record Linkage 11
I.L. Job Scheduling Software 12

II. INSTALLATION GUIDE

II.A. Recommended Server Specifications 15
II.B. Recommended Network Specifications 15
II.C. Web Services
II.C.1. Install the Web Server 15
II.C.2. Back up the Configuration Files 19
II.C.3. Configure Apache 21
II.C.4. Configure PHP 22
II.C.5. Deploy the Application Code 24
II.C.6. Configure MySQL 24
II.C.7. Set Up the Application Database 26
II.D. Application Code
II.D.1. Extract the Source Code 32
II.D.2. Modify the .env File 33
II.D.3. Complete the Installation 37
II.E. Background Services 40

II.F. Back Up and Restore Data
II.F.1. Prepare the Backup Script 45
II.F.2. Schedule Regular Backups 46
II.F.3. Restore from a Backup 51

III. USER GUIDE

III.A. Account Types and Access Restrictions 55
III.B. Getting Started
III.B.1. Launch the Application 55
III.B.2. Create an Account 57
III.B.3. Log In 58
III.B.4. Explore the Home Page 59
III.B.5. Log Out 60
III.C. User Account
III.C.1. Update Profile 60
III.C.2. Change Password 61
III.C.3. Reset Password 61
III.C.4. Select the Display Language 61
III.D. Records
III.D.1. View Record Information 62
III.D.2. Create a New Record 63
III.D.3. Edit an Existing Record 66
III.D.4. Recall a Submitted Record 67
III.D.5. Review a Submitted Record 68
III.D.6. Delete an Approved Record 70
III.E. Import and Export
III.E.1. View Imports List 71
III.E.2. Import Records from File 72
III.E.3. Resolve Record Matches from Import 74
III.E.4. Export List of Establishments 76
III.E.5. Export All Establishment Records 77
III.E.6. Export Customized Records List 78
III.F. Settings
III.F.1. Manage System Users 80
III.F.2. Manage Data Sources 82
III.F.3. Manage Classifications 83
III.F.4. Manage Announcements 85
III.F.5. Manage Files 85
III.F.6. Manage Links 86
III.F.7. View Audit Log 87
III.F.8. Create a Backup of the Database 88
III.G. Support
III.G.1. View Help Guides 89
III.G.2. Contact Support 89

Figures

Figure 1.1	Traditional Waterfall Software Development Life Cycle	1
Figure 1.2	Software Development Life Cycle	2
Figure 1.3	Solution Architecture	3
Figure 1.4	Tabular Data versus Unstructured Data	4
Figure 1.5	Basic Flowchart Used for Automated and Manual Testing	5
Figure 1.6	WAMP Stack and LEMP Stack	6
Figure 1.7	Cloud Deployment	6
Figure 1.8	Network-Attached Storage	7
Figure 1.9	Advanced Backups	8
Figure 1.10	Programming Languages Used in the ADB Statistical Business Register	8
Figure 1.11	Versions of Laravel	9
Figure 1.12	Relationships with Other Systems and Registers	10
Figure 1.13	Establishment Identification Number	11

Foreword

The Asian Development Bank (ADB) has been supporting its developing member countries (DMCs) to modernize their business registers with a view to improving their business environment. Given the close link between business registers and statistical business registers (SBRs), and the urgency in Asia and the Pacific for statistical infrastructure development, there is a need and an opportunity to further support DMCs in establishing viable SBRs. In December 2013, recognizing the importance of SBRs in providing data for informed decision making, ADB initiated the regional capacity development technical assistance project R-CDTA 8594: Statistical Business Registers for Improved Information on Small, Medium-Sized, and Large Enterprises. In the ensuing 5 years, the project was implemented in Bhutan, Cambodia, the Lao People's Democratic Republic, Malaysia, and Sri Lanka.

R-CDTA 8594 was designed primarily to support the national statistical offices (NSOs) of Asia and the Pacific in instituting viable SBR systems. Developing such systems is a complex endeavor that involves deep understanding of the technical and technological issues as well as the practical applications of data derived therefrom. Hence, to facilitate the modernization of the statistical infrastructure in the region, ADB developed a reference SBR system that not only can be readily installed, customized, and operationalized but can also be adopted and built upon by the DMCs for their own larger data management objectives. The ADB SBR system is also adaptable to changes in technology and is compatible with various source databases such as those pertaining to taxes, company registries, and customs. To ensure the long-term viability of the SBR system and to build sustainable technical capacity in the subject matter, a series of training activities was conducted at the regional and country levels as part of the project.

The *User Guide for ADB Statistical Business Register* complements the capacity-building initiatives undertaken under R-CDTA 8594. It provides step-by-step instructions on how to set up, use, and maintain the ADB SBR suite besides outlining the key technical considerations that guided the development of the software solution by ADB. This publication is intended to serve as a primary reference not only for the R-CDTA participant countries but also for those looking to establish SBRs of their own using the software solution developed under this technical assistance.

I would like to thank and acknowledge all the consultants, experts from international organizations, the national implementing agencies, and other government entities for their contribution, cooperation, and dedication in making the ADB SBR project a success. We hope that this document would guide the DMCs in establishing viable SBRs and, thereby, contribute to the enhancement of their statistical capacity.

Yasuyuki Sawada
Chief Economist and Director General
Economic Research and Regional Cooperation Department
Asian Development Bank

Acknowledgments

The ADB SBR User Guide was prepared under the Asian Development Bank (ADB) regional capacity development technical assistance *(R-CDTA) 8594: Statistical Business Registers for Improved Information on Small, Medium-Sized, and Large Enterprises.*

ADB acknowledges the strong support and valuable contributions of international experts Michael Biddington (United Nations Economic and Social Commission for Asia and the Pacific) and Markie Muryawan (United Nations Statistics Division) throughout this project. We thank the management and staff of the national implementing agencies for their hard work and commitment to overcome a wide range of challenges and accomplish the project objectives in a timely and efficient manner. These national implementing agencies are the National Statistics Bureau, Bhutan; National Institute of Statistics, Cambodia; Lao Statistics Bureau, Lao People's Democratic Republic (Lao PDR); Department of Statistics, Malaysia; and Department of Census and Statistics, Sri Lanka.

We especially acknowledge the contributions of the project's national and alternate coordinators, and local information technology experts: Ugyen Norbu, Sonam Laendup and Sonam Zangmo (Bhutan); Has Bunton, Solorvak Khin and Somethea Buoy (Cambodia); Chanseleum Kongkeo, Vichai Darasavong, and Hoth Singnoy (Lao PDR); Siti Haslinda Mohd Din and Hartini Yaacob (Malaysia); and Amarajeewa Jinabandu Satharasinghe, Udaya Maheswaran and Saman Weerasooriya (Sri Lanka).

R-CDTA 8594 was implemented by the Development Economics and Indicators Division of the Economic Research and Regional Cooperation Department of ADB, under the overall coordination and supervision of Mahinthan Joseph Mariasingham, with technical assistance from Eileen Capilit, in collaboration with Nikko Angelo Antonio, Angela Mari Ferrer, Maria Denise Peña, Albert San Juan, and Irene Talam. Oth Marulou Gagni and Eric Suan provided administrative assistance. This user guide was published with the support of the ADB Department of Communications and Office of Administrative Services Logistics Management Unit (Printing).

Rana Hasan
Director
Economic Research and Regional Cooperation Department
Asian Development Bank

Abbreviations

ADB	Asian Development Bank
API	application programming interface
DBMS	database management system
DMC	developing member country
EIN	establishment identification number
IT	information technology
LAO PDR	Lao People's Democratic Republic
LAN	local area network
NAS	network-attached storage
PHP	Personal Home Page/Hypertext Preprocessor
RAID	Redundant Array of Independent Disks
SATA	Serial AT Attachment
SBR	Statistical Business Register
SQL	structured query language
VPN	virtual private network

PROJECT OVERVIEW

A statistical business register (SBR) is a regularly updated and structured database of business establishments in a particular territory.

Recognizing the importance of an SBR in providing information for evidence-based decision- and policy-making, the Asian Development Bank (ADB) implemented **R-CDTA 8594: Statistical Business Registers for Improved Information on Small, Medium-Sized, and Large Enterprises**.

The key components of the ADB R-CDTA 8594 project are the establishment and maintenance of SBR systems and databases in the national statistics offices of partner developing member countries (DMC). However, developing an SBR is a complex undertaking that requires comprehensive understanding of both conceptual and technical matters, as well as the intended practical uses and applications of the data derived from such a system. In this regard, ADB developed a prototype SBR that DMCs can easily adopt and modify for their own purposes or to address emerging requirements.

System Features

The ADB SBR system has the following key features and characteristics:

- organized storage of **historical and current information** on businesses for efficient retrieval of records;
- ability to integrate information from **multiple sources** to contribute to completeness and accuracy;
- highly **configurable import mechanism** to allow partial and complete datasets to efficiently enter the system;
- system-assisted but user-decided **matching of establishment information** to identify duplicates and link multiple record entries;
- generation of **per-establishment information according to customized parameters** such as standard classification systems (geographic, industry, and product), period covered, business sizes, or other variables;
- automated **system management tasks** (e.g., database backups, audit logs) to assist IT administrators in system maintenance;
- modular features to maintain relevance to the functions and responsibilities of users based on their account type; and
- built-in **assistive tools** (e.g., infotips, instructions) for key features and pages.

By making it easier to access and retrieve statistical information about businesses, the SBRs contribute to the formulation of evidence-based policies and program administration.

Uses of the ADB Statistical Business Register

Production of Business Statistics

Statistics about businesses can be extracted directly from the ADB SBR, provided that the data are of sufficiently good quality, which is only likely for well-developed and well-maintained SBRs. Such data outputs include sums, counts, and their time series to present growths, trends, and other patterns. These may be aggregated and disaggregated by geographic location, industry or economic activity, product, and other key parameters. Specific information such as business size, ownership, assets value, employment, and business-to-business affiliations are also available.

Multisource Linkage

All unique imported records will be assigned to a system-generated Establishment Identification Number (EIN), thus providing a mechanism for linking individual business data from different sources. The system also implements a matching algorithm to compare new records with existing entries and link records pertaining to the same business entity. This feature contributes to the completeness of the information of establishments that are already in the database. In addition, the SBR allows its system administrators to configure other identification systems (e.g. Tax ID) to be captured by the system and be used as reference to match SBR records with records from external data sources.

Sampling Frames for Surveys

The SBR can produce sampling frames for business surveys according to the specifications provided by the survey manager. The frame specification defines the reference period to which the frame should refer, the type of sampling unit, the population of units to be included, the data items that are required for sample selection, and (for the sampled units only) the contact data items required for conducting the survey. The substantial information available in the SBR allows for sampling frames that focus on specific subgroups of the entire population of businesses in the country. The integration of standard classification systems and other common business variables also contributes to the convenience of creating targeted sampling frames.

Sampling Frames for Partner Surveys

Statistics offices may partner with other government agencies, development organizations, and even private organizations and the academe that conduct surveys. Without compromising data security and privacy, the SBR can be used as the main tool to create the sampling frame.

Involvement of Partner Developing Member Countries

ADB and the DMCs of Bhutan, Cambodia, the Lao People's Democratic Republic (Lao PDR), Malaysia, and Sri Lanka were partners throughout the implementation of the project. ADB led in the development of the application, in close consultation with the statistics offices and key officials of the partner DMCs to ensure that the system would sufficiently address their needs. The ADB project team also facilitated trainings and similar activities to capacitate the technical staff of the partner countries. This allowed the partner DMCs to introduce their own system enhancements and modifications in the future to address emerging needs, even after the end of the ADB technical assistance (TA).

2014

30 June to 2 July | Colombo, Sri Lanka
Workshop on the overview of the TA project, its objectives, the SBR, the data requirements, regional work plans and timetables, and the development of a national work plan

3 to 4 July | Colombo, Sri Lanka
9 to 11 September | Vientiane, Lao PDR
17 to 19 September | Thimphu, Bhutan
15 to 17 October | Putrajaya, Malaysia
27 to 29 October| Phnom Penh, Cambodia
Country meetings to discuss the types of data to be shared by the partner DMCs with ADB as well as the planned activities to firm up the SBR strategic development plan for each country

2015

26 to 31 January | Malaysia
4 to 11 February | Malaysia
Knowledge sharing of good practices and the experiences of the Department of Statistics of Malaysia (DOSM) in the development, use, and maintenance of SBRs

17 to 23 September | Putrajaya, Malaysia
Training on the concepts, methods, IT requirements, and possible analyses resulting from linking SBR and trade statistics; conducted in collaboration with the DOSM and the United Nations Statistics Division

23 to 26 September | Kuala Lumpur, Malaysia
Training on the first ADB SBR prototype and discussion of various data integration methods into the ADB SBR

2016

13 to 20 January | Thimphu, Bhutan
16 to 20 May | Lao PDR
5 to 9 September | Sri Lanka
26 to 30 September | Cambodia
Workshops on the progress of the ADB SBR system development, application demonstration, and sharing of proposed improvements to the system

2 to 4 May | Bangkok, Thailand
Forum of experts and managers involved in economic statistics production to support national research on the quality of economic statistics production and statistical infrastructure, in coordination with the United Nations Economic and Social Commission for Asia and the Pacific (UNESCAP)

25 to 29 July | Thimphu, Bhutan
Technical support and training in preparation for the pilot launch of the ADB SBR, with the National Statistics Bureau of Bhutan

2017

23 to 27 January | Siem Reap, Cambodia
Workshop on the progress of the ADB SBR system development, application demonstration, and sharing of proposed improvements or enhancements to the system

23 to 24 March | Colombo, Sri Lanka
23 October to 3 November | Bhutan
Technical support and training in preparation for the pilot launch of the ADB SBR system, with the implementing agencies

8 to 10 November | Vientiane, Lao PDR
Technical discussion on data management and assessment of IT infrastructure, with the Lao Statistics Bureau

20 to 25 November | Bandar Seri, Brunei Darussalam
Training on MySQL, PHP, and Laravel to capacitate the SBR system administrators of the implementing agencies in troubleshooting the installation, configuration, and maintenance of the system; and in introducing future enhancements to the system

2018

19 to 23 March | Colombo, Sri Lanka
Technical discussion on data management and SBR system installation and configuration

16 May to 2 June | Phnom Penh, Cambodia
Technical support and training in SBR system installation, configuration, and maintenance, including Javascript, MySQL, and Laravel, and data preparation techniques, with the National Institute of Statistics, Cambodia

SYSTEM DEVELOPMENT

I.A. Project Management

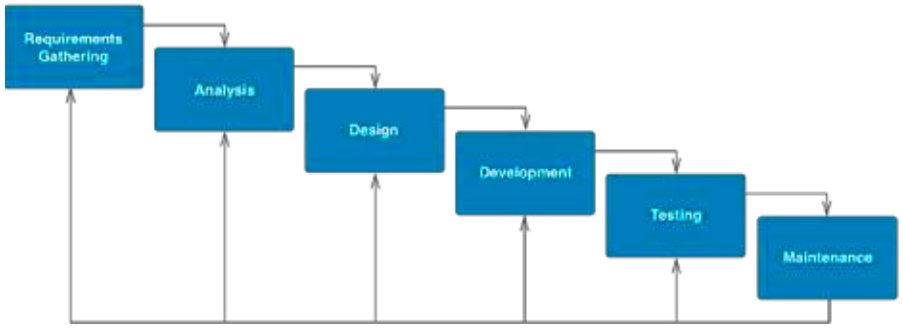
The project allocated enough time for proper conceptualization. The first stage focused on translating the initial requirements into actual application features in preparation for the initial product release. Afterwards, new input and user feedback were periodically received and evaluated as potential enhancements to the system. Prioritizing these proposed changes was critical to establish concrete work plans and time lines. The articulation and reinforcement of the project goals (versus the non-goals) were necessary to properly focus and scope the required development work.

Each value-adding proposed change was plotted into the future release schedules, which detailed the indicative timelines for development and testing, as well as the responsibilities of each team member. Weekly status meetings were conducted to communicate the progress and setbacks encountered, including any deviations and adjustments to the plans.

I.B. Software Development

The project was initially developed in a traditional waterfall software development life cycle (SDLC). In a waterfall model, software development goes through a linear sequence of phases and the project only moves forward upon completion of the preceding phase. The rationale is that time spent on each phase in the cycle can reduce costs at the succeeding stages.

Figure 1.1: Traditional Waterfall Software Development Life Cycle



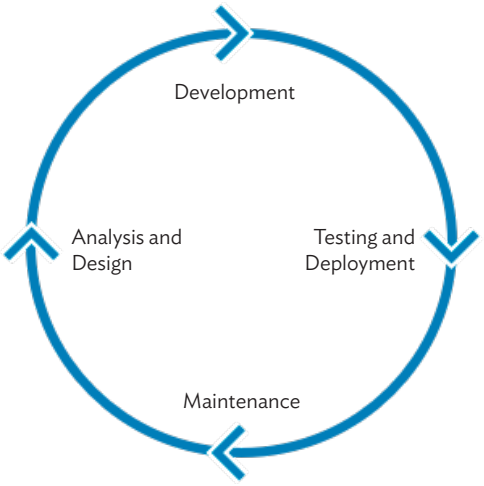
Source: S. L. Pfleeger and J. M. Atlee. 2012. Chapter 2: Modeling the Process and Life-Cycle. In *Software engineering theory and practice.* 4th ed. Johanneshov: Prentice-Hall International.

The project kicked off with requirements gathering, where the core variables and the process workflows were obtained from the users of the system and from existing reference materials regarding business registers. The developers analyzed these requirements and selected the best implementation option. The project then moved into the design phase, where the requirements were comprehensively documented and the overall architecture of the system ws finalized. Using the design documents as basis, the programmers worked on developing the initial version of the system. Once the system was fully developed with minimal deviations from the requirements, the ADB project team conducted in-house testing to ensure that all features and functionalities were working correctly.

Ideally, after a product release, the developers' focus would be primarily on optimization and maintenance. However, since the system was to be demonstrated and deployed to the five partner developing member countries (DMCs), the team considered adaptive methodologies so that the next iterations can accommodate additional requirements from the DMCs. To receive feedback from both internal and external sources throughout the phases of system development, the team observed a robust and consistent triage mechanism based on criticality and relevance.

An iterative software development life cycle was implemented after the first major release of the statistical business register (SBR) system to allow for smaller release cycles, thus enabling the team to adapt quickly to emerging requirements.

Figure 1.2: Software Development Lifecycle



Source: PSource: S. L. Pfleeger and J. M. Atlee. 2012. Chapter 2: Modeling the Process and Life-Cycle. In *Software engineering theory and practice*. 4th ed. Johanneshov: Prentice-Hall International.

I.C. Solution Architecture

The SBR system was designed as a web application that relies on a central web server. The database resides in this server, allowing multiple users to connect to the system at the same time.

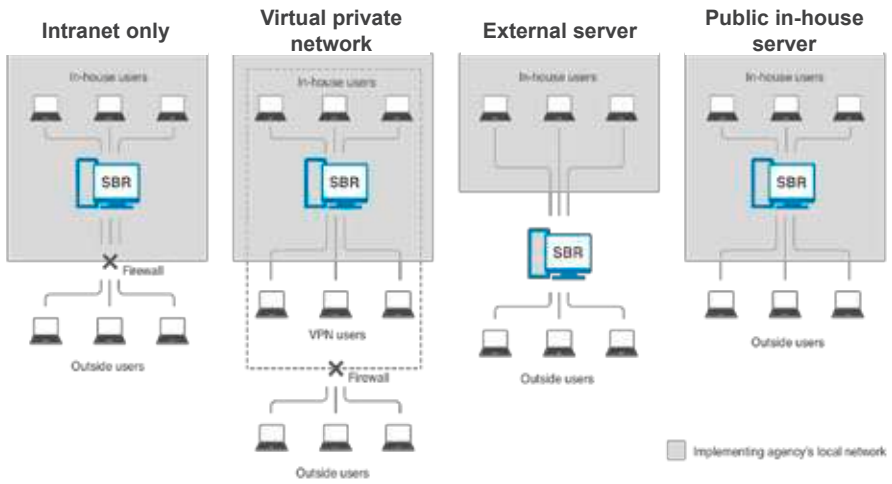
The underlying network infrastructure of SBR system can be chosen by the implementing agency, depending on their access requirements and data policies. For instance, the

implementing agency can choose whether to use an external web server or an in-house web server running inside the premises of their office.

In addition, they will have control over which computers are able to connect to the web server: in-house computers only within the same local network, in-house and external computers connected within the same virtual private network (VPN), or all computers connected using the public internet.

Since the records of the SBR system are stored in a standard relational database, the records can be accessed by external software that has been provided valid login credentials. This opens up the SBR data to widely-available reporting tools, statistical software, and even potential linkages with other databases.

Figure 1.3: Solution Architecture



Source: Authors.

I.D. Database

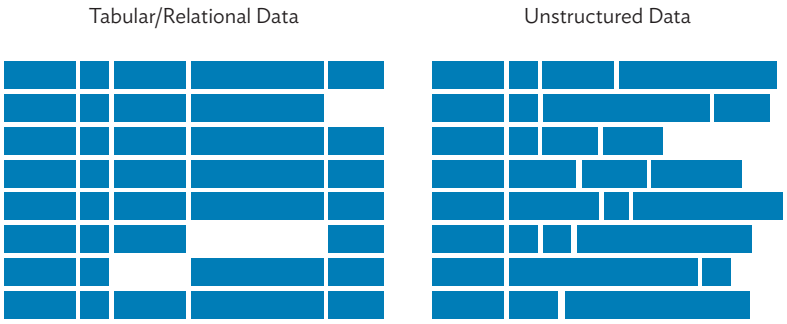
A relational database was chosen since its variables are well-defined and less likely to change. The team ensured that the data model captured all the establishment variables and carefully balanced database design best practices with practicality and usability so that users can easily understand how the data are organized should there be a need to manually dig into the data.

Web-based information systems commonly use relational databases, which are widely supported by website scripting languages. These databases are also compatible with data visualization and reporting tools. By linking to business intelligence applications, database users can generate summaries, dashboards, and charts that are frequently utilized by organizations.

Using a non-relational database (such as NoSQL) was considered early in the project. This is because non-relational databases do away with rigid tabular structures and instead accommodate data into flexible formats that depend on each of the inputted records. This means that this database type simply accepts any new record, whether or not it follows the format of the previous record. However, because of its flexible structure, the use of a non-

relational database may entail more complexity for performance optimization. Since the core variables collected by the SBR system are fairly consistent, there is no clear benefit to using a non-relational database for this purpose.

Figure 1.4: Tabular Data versus Unstructured Data



Source: Authors.

A database management system is a software system that uses a standard method of cataloging, retrieving, and running queries on data. The DBMS manages incoming data, organizes it, and provides ways for the data to be modified or extracted by users or other programs.

I.E. Database Management System

During the design and initial release, the system used MySQL, the de facto database management system (DBMS) for web applications. Subsequent deployments used MariaDB, which is a drop-in replacement for MySQL. This means that existing code that use MySQL will also accept MariaDB without any modification.

Both MySQL and MariaDB have widely available resources for web development. Both do not have software licensing costs, which benefits small agencies deploying the SBR system. However, MariaDB is more actively maintained by the open-source community compared to MySQL.

The ADB SBR system avoided the use of vendor-specific features to enable the implementing agencies of partner DMCs to consider other relational DBMS. Using such features would have required code modifications to ensure that all functionalities and system performance remain the same.

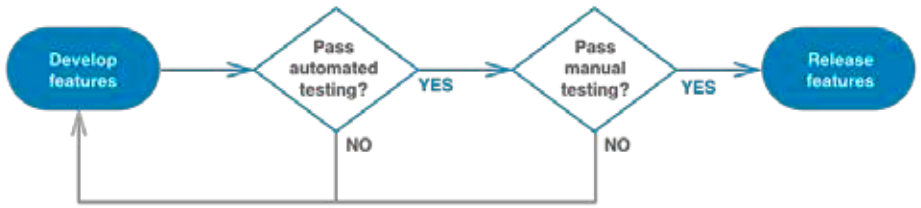
I.F. Programming Requirements

Developers. ADB's SBR development team consisted of four developers, with one dedicated to database development, and one dedicated to front-end development. All developers had extensive experience in deploying web-based information systems and expertise in the chosen programming languages (HTML, CSS, JavaScript, PHP, MySQL).

Testing. Manual testing was performed by both the developers and non-developer members of the team. The latter were crucial to obtain the perspective of a user. This allowed the team to revise the interface and workflows to be more user-friendly. The developers also wrote automated testing scripts, which allowed the team to ensure sustainable software quality by running known test cases repeatedly and consistently. Automated tests were run each time new features and fixes were added to the application

code to make sure that the existing features were not adversely affected by the modifications to the code.

Figure 1.5: Basic Flowchart Used for Automated and Manual Testing



Source: Authors.

Running automated tests made the application code a lot easier to maintain even as new features were introduced to the system over the development period. It also reduced the time needed to perform manual testing, and the time and effort needed to identify code regressions.

Version control. The team opted to use Git as their version control method. This version control software allowed multiple developers to work concurrently on the same code base. It kept track of code changes performed by each developer, and provided a facility to resolve any conflicts when developers worked on the exact same section of the code. In addition, by using a version control software, the team always had the option to undo changes and go back to an earlier version of the code.

The entire ADB SBR system comes from a single code base or repository. All previous versions are kept here and any future updates and releases will be added to this single repository. Rather than maintain separate versions of the ADB SBR system code for each country, the developers designed a universal system with a flexible architecture to support country-specific configurations and country-specific scenarios. Whenever the ADB SBR system is deployed, the country's developers can edit a single configuration file to control country-specific features.

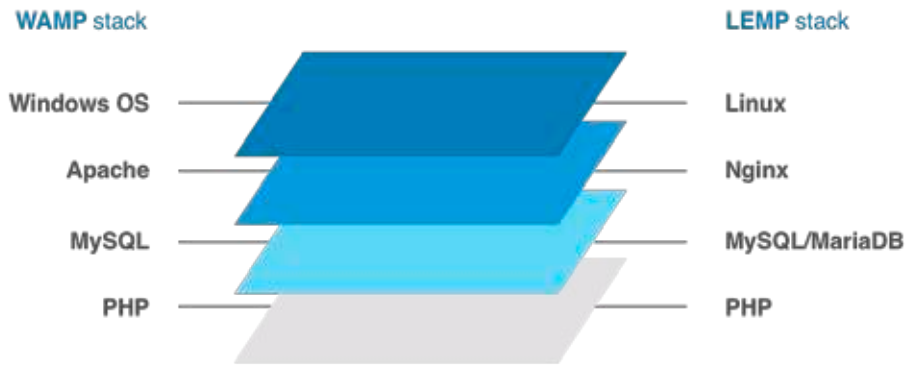
Maintaining a single codebase simplifies the delivery of updates to multiple deployments of the system across different countries. Any given code update would be automatically compatible and applicable to all countries. In contrast, branching off the code for country-specific modifications or one-off modifications would mean that code updates have to be rewritten, customized, and tested for a known branch of the code tree.

I.G. IT Environments

The system was designed to run on a **WAMP** stack, using the set of software comprised of the **W**indows operating system, **A**pache HTTPD, **M**ySQL or **M**ariaDB, and **P**HP. Alternatively, the system can also run on a **LEMP** stack, wherein the first two components are the **L**inux-based operating system and Nginx (pronounced as "**E**ngine X").

These are some of the most popular software suites used in operating web applications or web sites. Choosing one over the other largely depends on the availability of software licenses and technical skills of the administrators.

Figure 1.6: WAMP stack and LEMP stack



Source: Authors.

The SBR system is a web application that can be run on Windows or Linux-based operating systems.

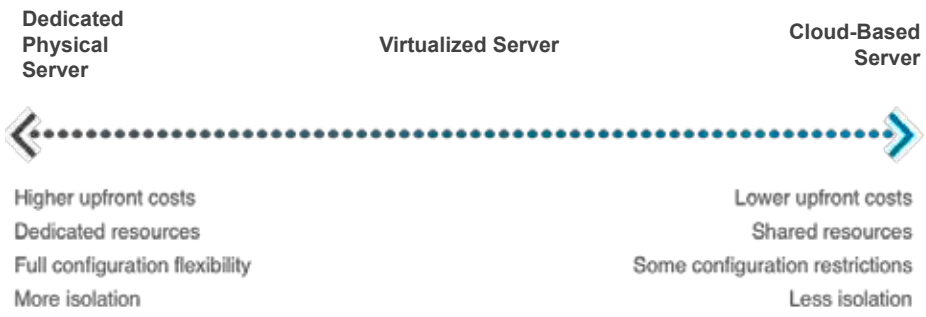
Use of existing infrastructure. Simpler deployments can reside on existing physical workstations, and are therefore recommended if less than a million records are expected for each year's worth of data.

Dedicated SBR server. Typically recommended deployments will require a dedicated server (tower or rack server) on modern hardware. The server should provide enough storage for long-term use and enough processing power and memory to perform aggregate operations on millions of records. The server should have at least one reliable gigabit Ethernet connection for the high data transfer rates needed when importing or exporting large datasets from the system.

Public cloud deployment. More technologically advanced agencies may take advantage of cloud-based solutions that require minimal or no initial capital outlay. Instead, the resources needed would be acquired through a subscription-type of arrangement. Public cloud deployment involves purchasing virtual units of computing resources on the cloud from commercial providers.

During a technical workshop, some country participants expressed interest in cloud-based solutions for development initiatives, as introduced by United Nations Statistics Division's Markie Muryawan. Some issues on privacy and security were raised, but with the proper implementation of the appropriate security practices, a public cloud application may involve less risk than a poorly-secured application in a private environment.

Figure 1.7: Cloud Deployment



Source: Authors.

Private cloud and virtualized servers. Some governments have institutionalized private clouds—networked hardware that combines their processing power, storage, and memory into a combined pool. Single-purpose virtual servers, such as that for the ADB SBR system, be created on this private cloud, utilizing only the resources necessary, while any unused power or memory is automatically diverted for other virtual servers to use. However, utilizing shared infrastructure entails arrangements for data ownership, operating and licensing costs, and requires procedures, protocols, and responsibilities, especially if the shared infrastructure is managed by another government ministry or department.

Backup strategy. Regardless of the hardware infrastructure chosen, any SBR system deployment geared for long-term benefit should have the sufficient backup strategy in case of server hardware or software failures. The ADB SBR system does not have any unique backup strategy requirements that are different from those of any typical information system on the same scale.

Ideally, backups should be in multiple tiers, depending on the available resources. The easiest method is backing up the data to the same machine and/or hard disk, as long as it is outside of the reach of the ADB SBR system. This allows the administrator to easily restore the data to an earlier consistent state, should some glitch cause database corruption. However, any hardware failure affecting the ADB SBR system will most likely affect the backups as well.

Thus, it is widely practiced to keep the backups on hardware not residing internally on the server. This can be through external USB backup disks, or on network-attached storage (NAS).

Simple NAS devices allow external storage to be accessible over network connections. Storage disks are expected to fail over time, and manually maintaining multiple backup disks can be tedious and prone to human error. Advanced NAS devices can house and manage multiple disks when configured as a redundant array of independent disks (RAID). Certain RAID options tolerate the failure of one storage disk without any loss of data. Once the single faulty disk is replaced, the NAS can automatically recreate and write the lost data back onto the new disk. It is ideally recommended that these backup devices would be situated in a separate room, or entirely off-site.

NAS units can come in the form of small stand-alone appliance, or rack-mounted servers. Source: synology.com

Figure 1.8: Network-attached Storage

Source: SEA PC Magazine. 2018. *The Best NAS (Network Attached Storage) of 2018.* sea.pcmag.com/network-attached-storage.

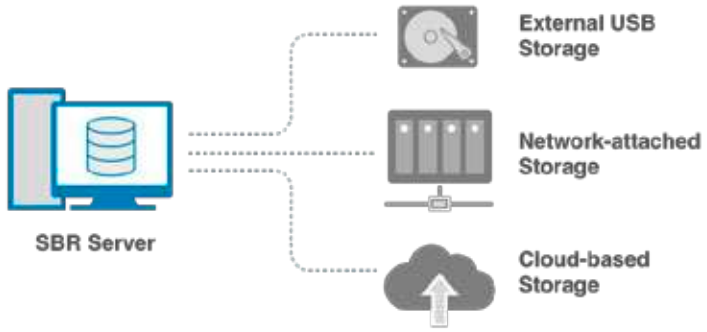
Automated backups. Automated backup scripts are ideal to have, but are only possible only after the implementing agency has allocated resources for backup disks. It also

requires designated personnel. Automation may be done by creating specialized scripts, or through special-purpose utility software.

Manual backups. In lieu of automation, manual backups and archiving at regular, frequent can intervals are just as adequate. Automation works best with NAS units, as they can reside elsewhere and do not require that they be physically attached to the server.

Advanced backups. More elaborate backup strategies involve ensuring backups are performed on different kinds of physical media (e.g., magnetic, solid state, optical, tape, etc.) and on both online and offline locations (i.e., hot and cold backups).

Figure 1.9: Advanced Backups



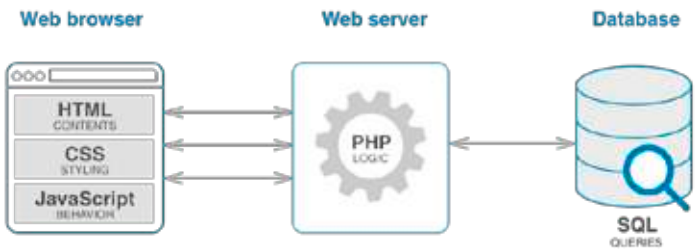
Source: Authors.

As another option, cloud infrastructure companies offer backup solutions that employ different techniques without any intervention whatsoever from the SBR system administrator.

I.H. Programming Languages

The ADB SBR system is coded extensively with the popular programming languages for web development.

Figure 1.10: Programming Languages used in ADB SBR



Front-end	Hypertext Markup Language (HTML) version 5 Cascading Stylesheets (CSS) JavaScript
Back-end	Hypertext Preprocessor (PHP) version 7 Structured Query Language (SQL)
Code libraries and frameworks	Bulma 0.6.1, jQuery 3, and SCSS for front-end Laravel Framework 5.5 LTS for PHP-based back-end

Source: Authors.

The ADB SBR system's interface is in the form of an HTML5 webpage, with styling information written in CSS. These webpages are dynamically generated by the web server based on the application logic, which is written in PHP.

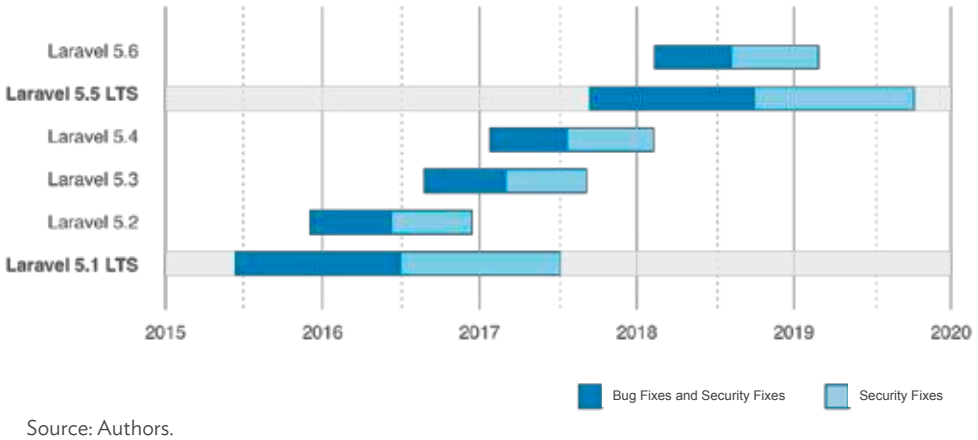
Code libraries are used in designing the front-end user interface to speed up overall development time while improving interface usability and aesthetics. These allow for an intuitive design using modern interface conventions. Users will find that the navigation, buttons, links, form controls, and other interface elements feel familiar because they look and behave similarly as those used in the latest web applications and social media interfaces. The development team avoided the use of uncommon or unusual interface interaction models as these may be detrimental to the usability of the system.

Back-end development frameworks also significantly decrease the total development time. Including these frameworks in the ADB SBR application allowed the team to reuse tried and tested code. The maintainers of these open-source frameworks (e.g. Laravel framework) versions introduce breaking changes that may detrimentally affect existing code of applications built on the frameworks. This requires some adjustment on the part of the developers when updating a framework to newer versions. If not carefully performed and extensively retested, upgrading to newer versions may introduce new bugs.

Over time, the open-source community stops providing bug fixes and security patches for older versions of the framework, and instead works on delivering these for the newer versions. For this reason, some community-developed frameworks designate certain versions as Long Term Support (LTS) releases, wherein they commit to providing patches and fixes for an extended period of time (e.g. 18 months versus a typical 6-month period) even as newer versions of the framework are available. These updates do not contain new features which may be a source of new bugs, as they are meant to fix bugs and security holes that are already present in the particular version. This is practical when long-term reliability and stability of software quality, and ease of maintenance are preferred over cutting-edge features and enhancements.

As mentioned, Git was chosen by the developers for code management and version control. A private repository is hosted on a popular online platform, to which individual developers synchronize the code they develop on their machines on a regular basis. Conflicts arising from merging code are resolved early on.

Figure 1.11: Versions of Laravel



Source: Authors.

I.I. Relationships With Other Systems and Registers

The ADB SBR system relies on a batch import facility to source data in bulk. This is typically an Excel or CSV file prepared through manual encoding or through data generation from another system (which may require some transformation). Future integration with other systems may come in either of two forms:

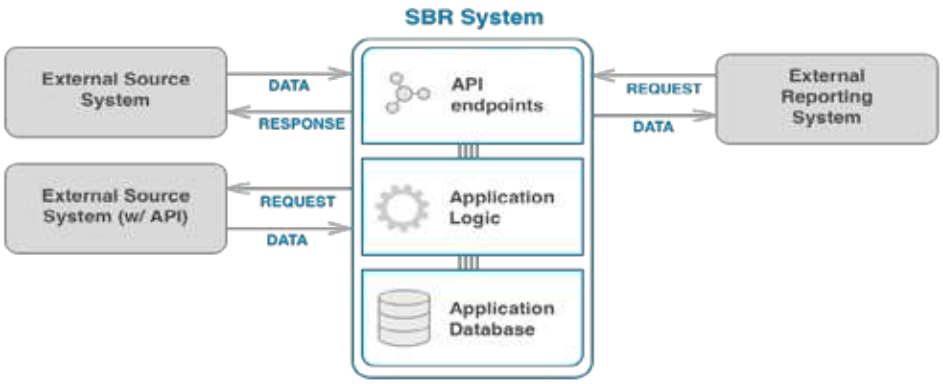
1. ***Direct Database Connection***
   The relational data model employed in the database is highly semantic and adheres to modern data structure conventions. The column names are descriptive, and the relationships between tables are intuitive. Consideration was given to avoid making the database application independent.

   Developers of external applications that aims to import the data managed by the ADB SBR system may easily do so by connecting directly to the database using standard database connection protocols. However, it is preferable to do so using a read-only connection or read-only database credentials in order to preserve data integrity. Alternatively, Extract Transform Load (ETL) tools and/or processes may be used to create mirrors or data silos from the master ADB SBR database, which the developers and their applications may freely manipulate.

2. ***Application Programming Interface (APIs)***
   APIs in software are special functions designed to allow developers of third-party systems to integrate their own applications and perform predefined routines or request specific information in a securely-controlled manner. Requests sent to APIs must adhere to a specified format, which is usually detailed in a provided API Reference document. APIs may use security protocols that require credential information contained within the requests in order to make sure the external systems are authorized to make such requests. APIs are ideally made to be application-agnostic, such that they are not dependent on a specific external application.

   Overall, APIs facilitate explicitly designed interaction with external systems, all while still maintaining secure and controlled access over actions performed, information manipulated or requested.

Figure 1.12: Relationships with Other Systems and Registers



Source: Authors.

Through the use of APIs, the ADB SBR system can be developed further to import and digest data obtained from source systems, and can provide its own data for use of third-party reporting or transactional systems.

For instances wherein an external source system initiates the process and pushes the data to the ADB SBR system, the latter must have API endpoints ready to accommodate the importation. On the other hand, if the ADB SBR system will initiate the process, it must connect to the external source system's API and adhere to its format.

When external applications need highly specific packets of information on a single establishment or record, an API is highly recommended as against the alternative of mirroring or exporting the entire database, which is too cumbersome for certain purposes.

I.J. Unique Identifiers

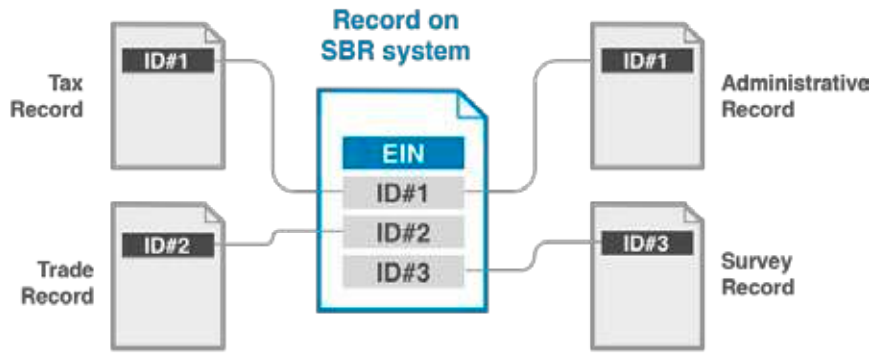
The ADB SBR system relies extensively on a single unique identifier for the basic statistical unit, the Establishment Identification Number (EIN). During record entry, the EIN is requested from the encoder. If not provided, the system searches for similar existing records. If a matching establishment is found, the new record is automatically linked to the EIN. Otherwise, the system creates a new EIN using an incrementing generated number, and assigns this to the new record.

Furthermore, the developed ADB SBR system allows the input of several existing identifiers used by other source systems (e.g. tax identification or business registration numbers), albeit on an optional basis as they may not always be applicable to all records. These identifiers may determine duplicates or missing links between records within the system. They may also facilitate future linkages of entities on external systems.

During missions and workshops, the ADB SBR team encouraged the implementing agencies to institutionalize a unique identifier that fits the system requirements.

If the identifier is to be co-owned with other agencies, the definition of an establishment or statistical unit must be made fully compatible among all co-owners.

Figure 1.13: Establishment Identification Number



Source: Authors.

I.K. Tools and Software for Record Linkage

The ADB SBR system uses a probabilistic or deterministic matching scheme to link records from existing sources. Parameters for this algorithm depends on the reliability and authoritativeness of each of the datasets being matched, and the functionalities developed are easily configurable for tweaking (such as for confidence intervals) to get optimal results.

The system implements the following to ensure optimal results for data matching:

- ***Initial input***
  Upon manual record creation or import, the system scans the identifier (e.g., tax identification) for possible matches already existing in the database.

- ***Regular scanning of duplicates***
  At regular intervals, the system scans for possible duplicate records within a calendar year.

- ***Regular scanning of unlinked records***
  At regular intervals, the system scans for establishment entries (each with one or more records) that may refer to the same establishment or statistical unit, but due to some variations, are captured by the system as separate entities. Linking them entails merging them into one EIN for a unified set of records for a more complete picture of the establishment's historical information.

As the latter two scenarios entail matching one record with all the others within a subset of data, the processing requirements for these operations can possibly scale exponentially with the record count. As such, the subset to which each record will be compared against must be highly segmented. Additionally, the tasks should be performed at off-peak hours in order to avoid concurrency problems and performance bottlenecks.

I.L. Job Scheduling Software

The system does not require any sophisticated scheduling software to run special operations. Regularly-scheduled tasks necessary to operate the system are written on the system's code itself. The system was developed using the Laravel framework, which took care of a lot of logic that was necessary to set up scheduled tasks.

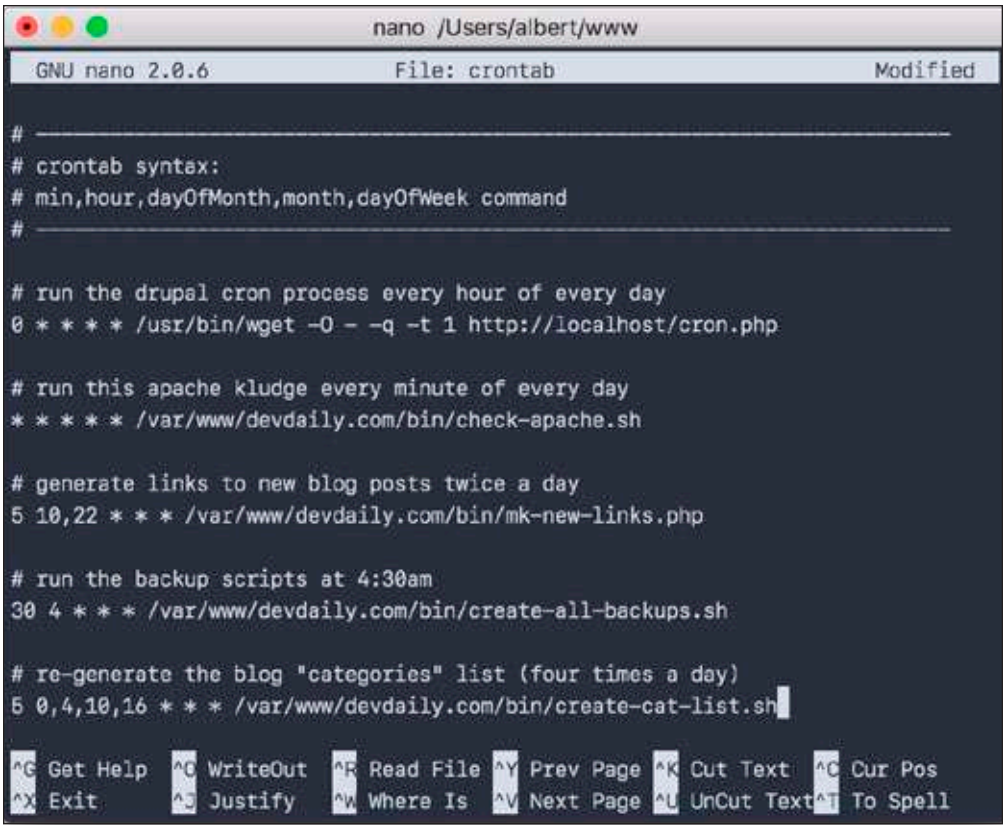
If necessary, the system can be configured to take over some of the administrator's workload by scheduling automated backups and maintenance activities.

Aside from tasks executed at regular intervals, some user-initiated tasks can be triggered to execute in the background. When these tasks are triggered, each task and its pertinent details are stored in a database table, and form a queue of jobs. Special system processes called "queue workers" regularly query this table for any jobs to be executed. Tasks are then initiated on a first-in first-out (FIFO) basis, and each task is removed from the table once initiated.

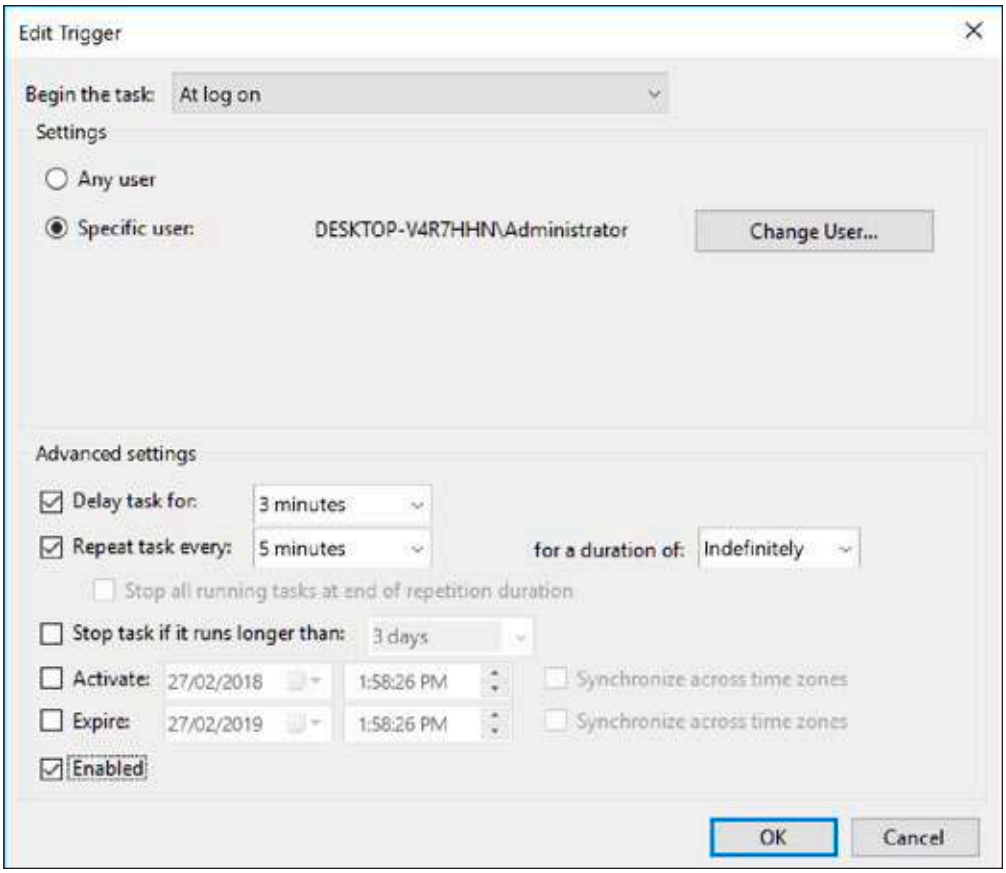
This is particularly important for long-running tasks, such as data import, file processing, backup, and export. If not handled properly, tasks that take a few seconds to complete will make the system appear unresponsive, while even longer tasks (taking 30 seconds or more) will fail as they will exceed the system's allotted time for a single page request. Separating these tasks from the normal flow of the user's session allows the server to continue handling the user's succeeding requests, while the requested task is delegated to be run by a queue worker in the background.

The web server software alone is unable to schedule tasks and manage job queues. For this, the server's operating system is needed to invoke the tasks. Job queues, on the other hand, require one or several job workers that are on standby, waiting on the database for tasks to come in. It is necessary to ensure that at any point in time, there is at least one worker process running for each job queue.

Although lacking a graphical user interface, cron's robust scheduling ensures that tasks are run on schedule for as long as the machine is operational.



The Edit Trigger windows of the built-in Windows Task Scheduler



For deployments on Linux-based servers, a native task scheduling system called "cron" is available. Cron is particularly useful for triggering scheduled tasks. A popular software package called Supervisor can be used to ensure job workers run at all times.

For deployments on Windows-based servers, the built-in Task Scheduler application is employed to trigger tasks. The Task Scheduler is used to start job worker processes every few minutes, but is configured to not run a new instance if an existing instance is already running. This is not an ideal use case of the Task Scheduler, and while it may be somewhat tedious to set up, it is sufficient.

INSTALLATION GUIDE

II.A. Recommended Server Specifications

- A physical, virtual machine, or cloud-based server
- A network interface capable of at least 100 Mbps
- For a Linux-based server:
  - Ubuntu, latest LTS release
  - At least 2 GB RAM
  - At least 50 GB storage
  - At least 2 CPU cores or threads
- For a Windows-based server:
  - Windows Server 2012 R2 or newer
  - At least 8 GB RAM
  - At least 80 GB storage
  - At least 4 CPU cores or threads
- Keyboard, mouse, and monitor are all optional if server can be managed remotely

II.B. Recommended Network Specifications

- Secure local area network (LAN) capable of assigning static IP addresses
- LAN components (e.g. routers, switches and cables) capable of at least 100 Mbps (i.e. cat5 or better)
- Operational firewall (either virtual or physical)
- Connected to the internet (if ADB SBR access will be allowed through internet)

II.C. Web Services

II.C.1. Install the Web Server

Developed as a web application, the ADB SBR system requires an **operational web server** with the following **web services** configured and running properly:

- Apache or Nginx
- PHP 7.1 or higher, with the following extensions:
  - OpenSSL PHP Extension
  - PDO PHP Extension (Mbstring PHP Extension)
  - Tokenizer PHP Extension
  - XML PHP Extension

- MySQL version 5.7 or higher, or MariaDB 10.2 or higher
- Node.js version 8.9 or higher
- Git version 2.13 or higher

All the required web services will be installed on Windows using the **Laragon 3.1** installer (laragon-wamp.exe). Laragon combines Apache, PHP, MySQL, Node.js, and Git into a single installer and provides a convenient way to start and stop the web services. It also installs some helpful applications, such as Notepad++, HeidiSQL, and CMDER.

The same web services may also be installed individually without the use of Laragon.

Assumption

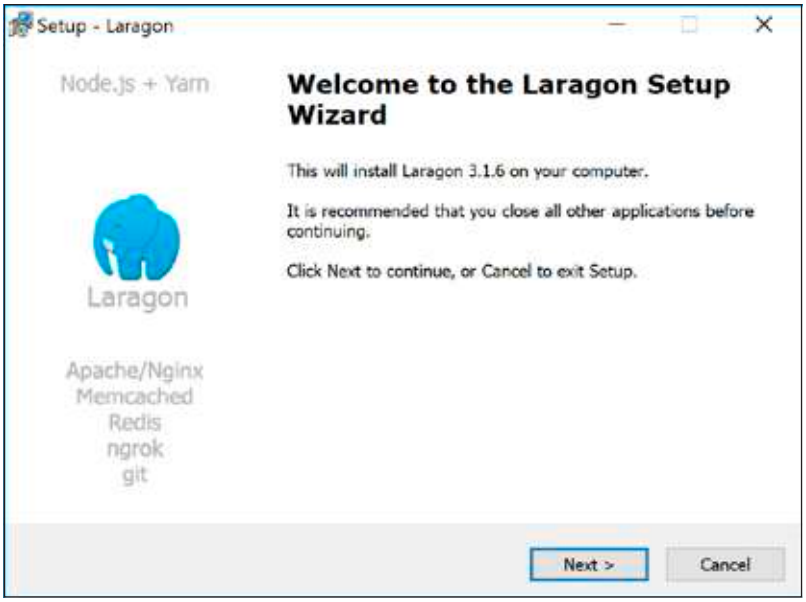
The ADB SBR IT Administrator is logged into Windows with administrator privileges.

Related links

https://laragon.org
https://www.heidisql.com/
https://notepad-plus-plus.org/
https://cmder.net

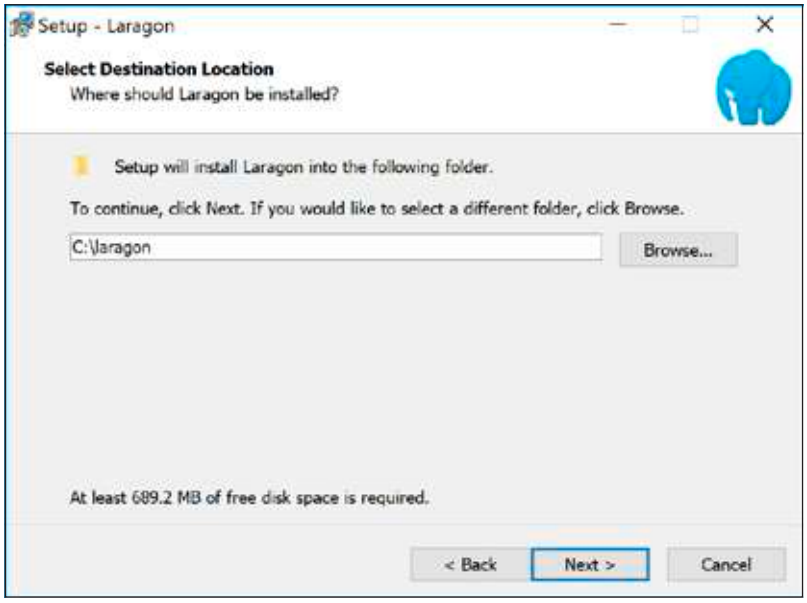
STEP 1

Run the Laragon installer (**laragon-wamp.exe**). Click **Next** to begin the installation.



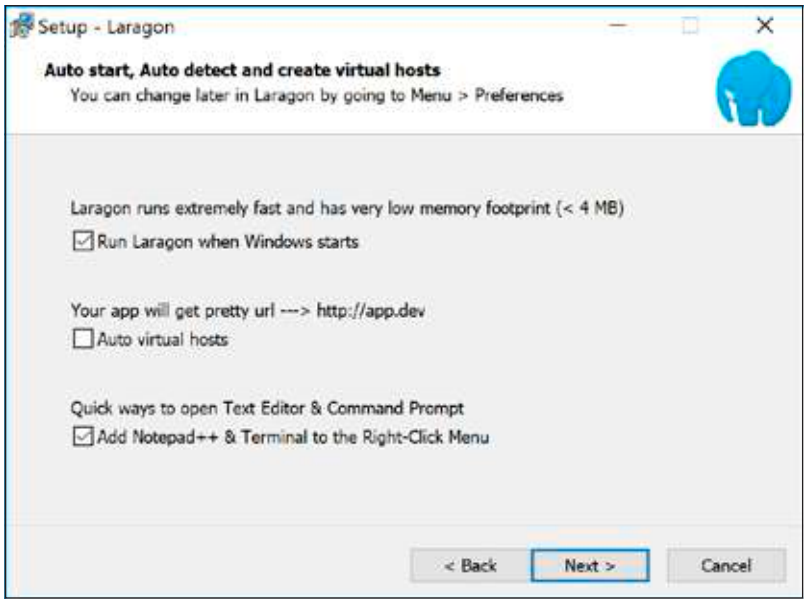
STEP 2

Set the installation directory to **C:\laragon**, and click **Next**.



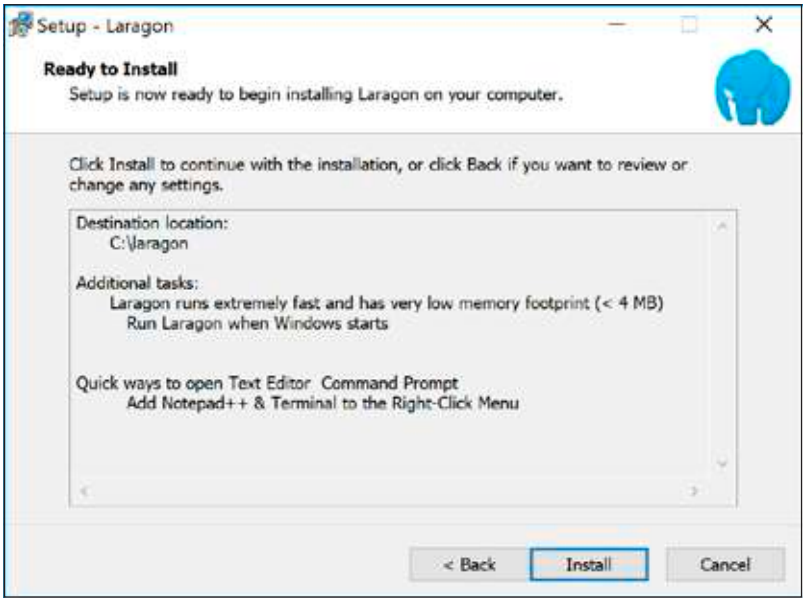
STEP 3

Select or deselect the options as follows, and click **Next**.



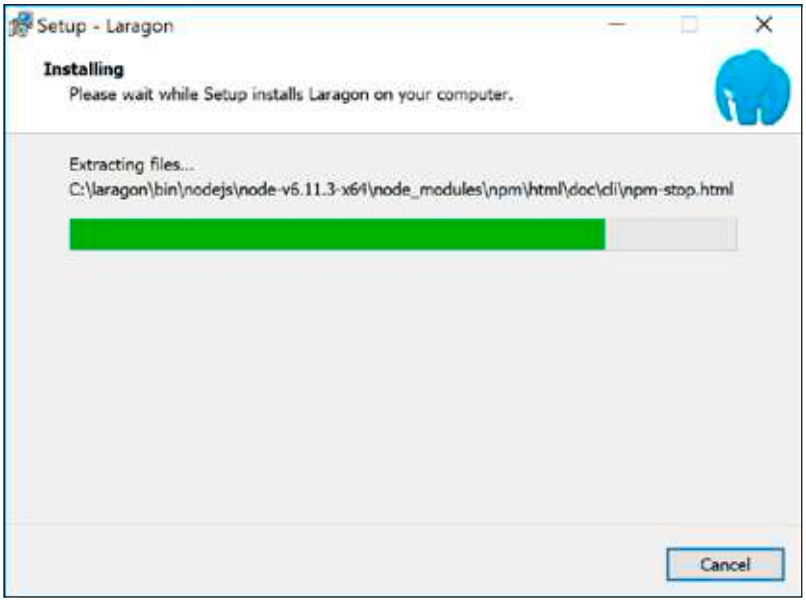
STEP 4

In the next window, click **Install**.



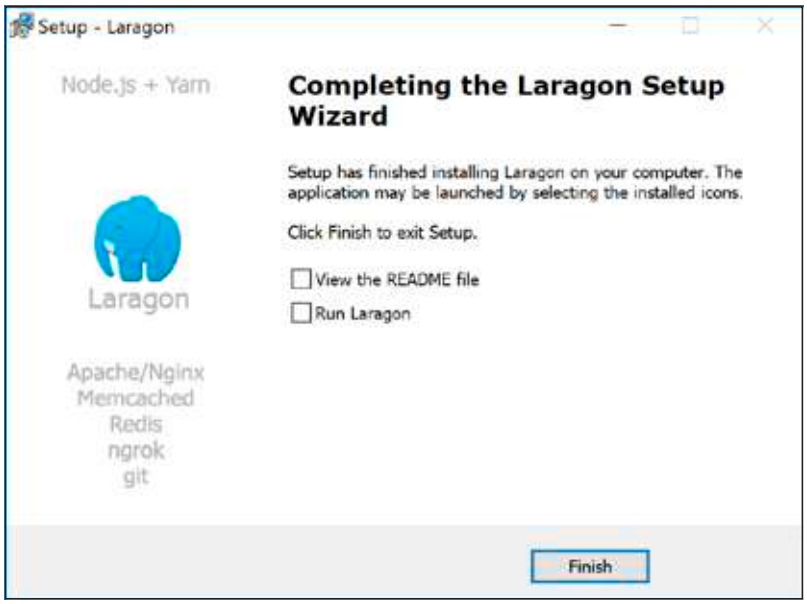
STEP 5

Wait for the program to complete installation.



STEP 6

In the next window, deselect the *Run Laragon* option, and click **Finish**. Once installed, Laragon will be accessible through the Windows Start Menu or through the created desktop shortcut.



II.C.2. Back up the Configuration Files

The newly installed web services have configuration files that need to be modified. As good practice, a copy of the original set of configuration files should be kept in the same directory and renamed with the suffix **.backup**, although a different naming scheme may be used.

Assumptions

- Laragon and the web services are stopped and are not running.
- The file explorer is set to show the extensions of all files.

List of configuration files to back-up

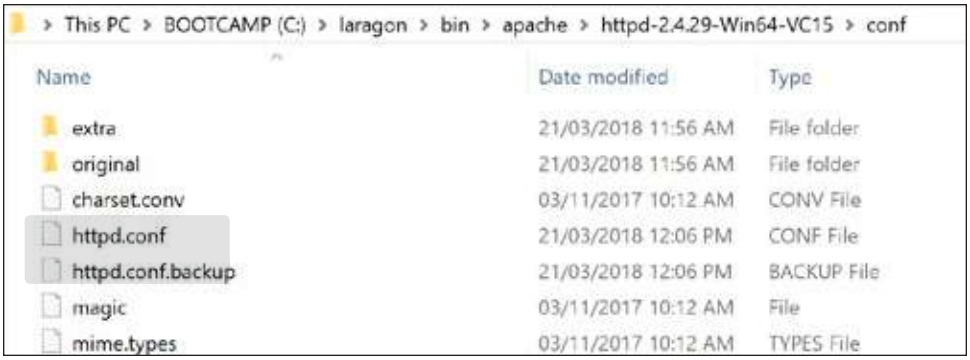
C:\laragon\bin\apache\<apache version code>\conf\httpd.conf
C:\laragon\bin\php\<php version code>\php.ini
C:\laragon\bin\php\<php version code>\php.ini-production
C:\laragon\bin\mysql\<mysql version code>\my.ini

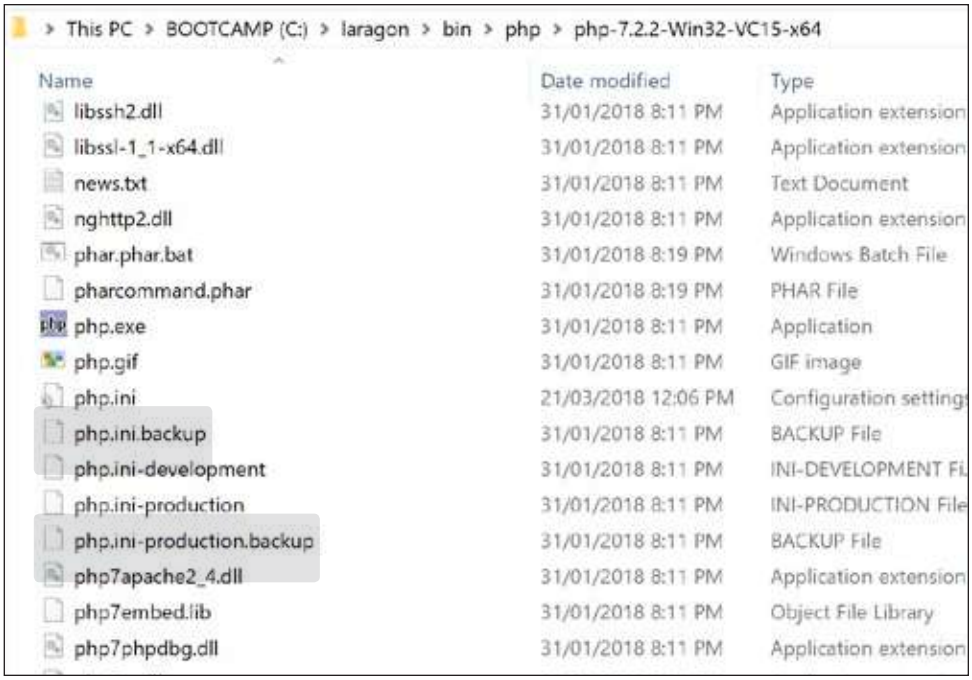
STEP 1

Locate each of the configuration files from within the default installation directory of Laragon (C:\laragon).

STEP 2

Create a copy of each of the files and **rename the copies** based on the selected naming scheme.





II.C.3. Configure Apache

The Apache HTTP Server is configured by placing directives in plain text configuration files. The main configuration file is commonly named **httpd.conf.**

Assumptions

- Laragon and the web services are stopped and are not running.
- The file explorer is set to show the extensions of all files.
- **A backup of C:\laragon\bin\apache\<apache version code>\conf\ httpd.conf has been made.**

Related link

https://httpd.apache.org/docs/current/configuring.html

STEP 1

Locate the file **httpd.conf** from within the installation directory of Laragon, C:\laragon\bin\apache\<apache version code>\conf. Open the file using a **text editor** directory of Laragon (C:\laragon).

STEP 2

Locate the lines pertaining to Listen, and set to **Listen 80**.

```
57  #Listen 12.34.56.78:80
58  Listen 80
```

STEP 3

Locate the block pertaining to LoadModule, and remove the hash or number sign (#) from the lines containing the terms **deflate_module, expires_module, filter_module, and headers_module**.

```
109  LoadModule deflate_module modules/mod_deflate.so
110  LoadModule dir_module modules/mod_dir.so
111  #LoadModule dumpio_module modules/mod_dumpio.so
112  LoadModule env_module modules/mod_env.so
113  LoadModule expires_module modules/mod_expires.so
114  #LoadModule ext_filter_module modules/mod_ext_filter.so
115  #LoadModule file_cache_module modules/mod_file_cache.so
116  LoadModule filter_module modules/mod_filter.so
117  #LoadModule http2_module modules/mod_http2.so
118  LoadModule headers_module modules/mod_headers.so
```

STEP 4

Locate the lines that contain the main server block, which starts with **<Directory "C:\laragon\www">**.

STEP 5

Edit the text entries within the main server block, as follows:

- Options Indexes FollowSymLinks to **Options FollowSymLinks Includes ExecCGI**
- AllowOverride None to **AllowOverride All**

```
247 DocumentRoot "C:/laragon/www"
248 <Directory "C:/laragon/www">
249     #
250     # Possible values for the Options directive are
251     # or any combination of:
252     #   Indexes Includes FollowSymLinks SymLinksifOw
253     #
254     # Note that "MultiViews" must be named *explicit
255     # doesn't give it to you.
256     #
257     # The Options directive is both complicated and
258     # http://httpd.apache.org/docs/2.4/mod/core.html
259     # for more information.
260     #
261     Options FollowSymLinks Includes ExecCGI
262
263     #
264     # AllowOverride controls what directives may be
265     # It can be "All", "None", or any combination of
266     #   AllowOverride FileInfo AuthConfig Limit
267     #
268     AllowOverride All
269
```

STEP 6

Save and close the **httpd.conf** file.

II.C.4. Configure PHP

The behavior of PHP is changed by editing its main configuration file named **php.ini**, which is read when PHP starts up. This php.ini file contains several parameters followed by an equals symbol (=) and their corresponding values. There must be a space before and after the equals symbol.

STEP 3

Rename the file php.ini-production to php.ini.

STEP 4

Open the new php.ini file using a **text editor**.

STEP 5

Edit the text entries, as follows:
(i) post_max_size = 8M to **post_max_size = 300M**
(ii) upload_max_filesize = 2M to **upload_max_filesize = 300M**

```
668 ; Maximum size of POST data that PHP will accept.
669 ; Its value may be 0 to disable the limit. It is igno
670 ; is disabled through enable_post_data_reading.
671 ; http://php.net/post-max-size
672 post_max_size = 300M
```

```
822  ; Maximum allowed size for uploaded files.
823  ; http://php.net/upload-max-filesize
824  upload_max_filesize = 300M
```

STEP 6

Save and close the php.ini file.

Assumptions

- The web services were installed in the default installation directory using Laragon 3.1.
- Laragon and the web services are stopped and are not running.
- The file explorer is set to show the extensions of all files.
- **A backup of C:\laragon\bin\php\<php version code>\php.ini has been made.**

Related link

http://php.net/manual/en/configuration.file.php

II.C.5. Deploy the Application Code

Once the web services are configured and running, the latest application code for the system must be copied to the proper web root directory.

> **Assumption**
>
> A working copy of the ZIP file sbr-mysql.zip is available for use.

STEP 1

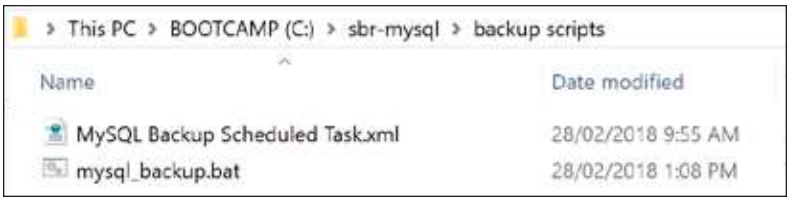
Extract the contents of the file **sbr-mysql.zip** to C:\ to create the following directories:

- **C:\sbr-mysql\backup data**
- **C:\sbr-mysql\backup scripts**
- **C:\sbr-mysql\log file**

STEP 2

Confirm that the following files are included inside the directory C:\sbr-mysql\backup scripts:

- **mysql_backup.bat**
- **MySQL Backup Scheduled Task.xml**



II.C.6. Configure MySQL

MySQL reads startup options from option files. Option files (also called configuration files) set commonly used parameters, so that they need not be specified in the command line each time a program is run. The main option file for MySQL is **my.ini.**

The parameters are grouped together and applied to their corresponding modules. The group header is denoted by the name enclosed in brackets (e.g. [mysqld] is the group header for the mysqld server).

Each parameter is followed by an equals symbol (=) and then its corresponding value. There must be no space before and after the equals symbol. And aside from the newline (line break), there must be no spaces or extra characters after the assigned values.

Assumptions

- The web services were installed using Laragon 3.1 in the default installation directory.
- Laragon and the web services are stopped and are not running.
- The file explorer is set to show the extensions of all files.
- **A backup of C:\laragon\bin\mysql\<mysql version code>\my.ini has been made.**

Related link

https://dev.mysql.com/doc/refman/5.7/en/option-files.html

STEP 1

Locate the file **my.ini** from within the installation directory of Laragon, C:\laragon\bin\mysql\<mysql version code>. Open the file using a **text editor.**

STEP 2

Under the group header **[mysqld]**, edit / add the text entries, as indicated.

```
[mysqld]
port=3306
socket=/tmp/mysql.sock
key_buffer_size=256M
max_allowed_packet=512M
table_open_cache=256
sort_buffer_size=16M
read_buffer_size=8M
read_rnd_buffer_size=8M
myisam_sort_buffer_size=256M
thread_cache_size=16
query_cache_size=128M
```

```
query_cache_type=ON
query_cache_limit=32M
slow_query_log=1
slow_query_log_file="C:/sbr-mysql/log file/slow-query.log"
long_query_time=1
log_queries_not_using_index=1
```

STEP 3

Save and close the my.ini file.

II.C.7. Set Up the Application Database

The default installation of MySQL or MariaDB only has a single user with no password set. The administrator must create a database and a user account to be used by the ADB SBR system. It is good practice to use this account for the web application. The administrator may also create additional accounts for other system users, although each user is encouraged to create his or her account following the procedure described in ***Section III.B.2. Create an account.***

Assumption

- The web services were installed using Laragon 3.1 in the default installation directory.
- **The configuration files MySQL or MariaDB have been modified from their defaults.**
- **The default user for MySQL or MariaDB is root with no password set.**
- **HeidiSQL was installed during the Laragon installation.**

Related link

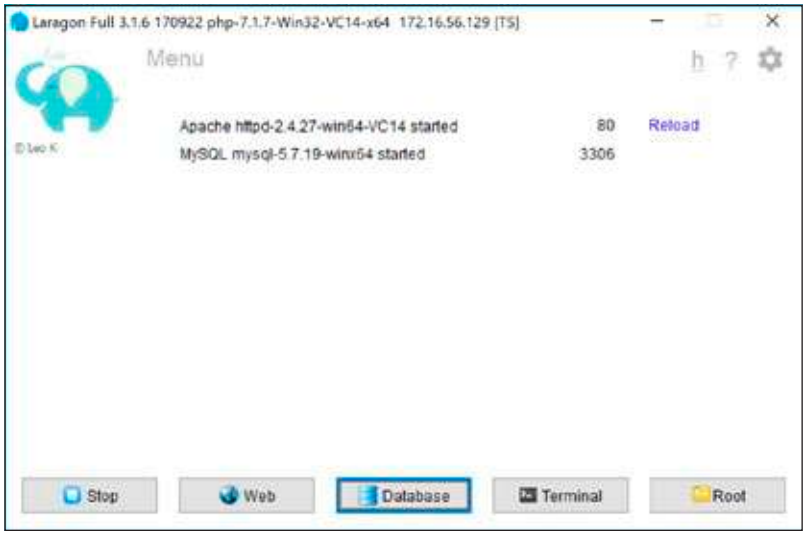
https://www.heidisql.com/

STEP 1

Launch **Laragon** using the desktop shortcut or the Start Menu item.

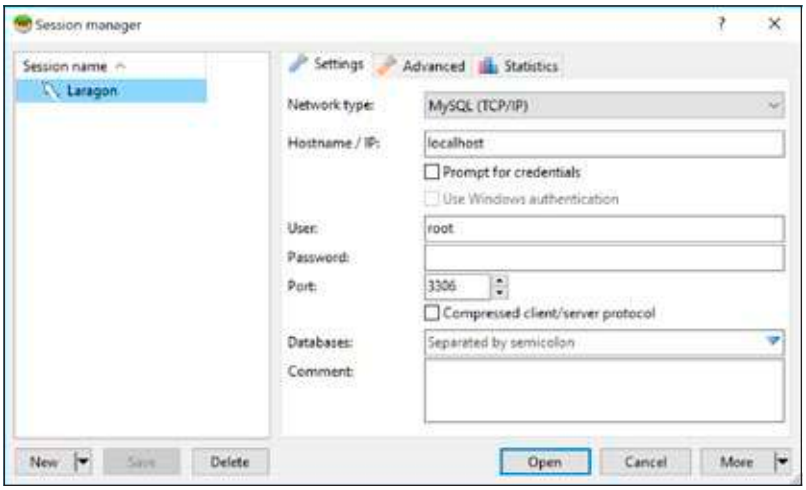
STEP 2

In the Laragon window, click **Database** to open HeidiSQL.



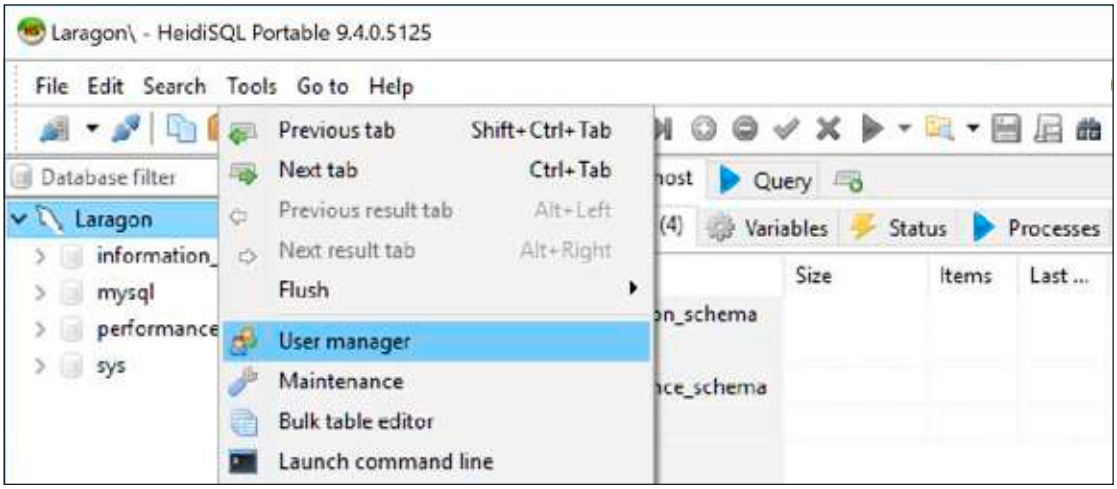
STEP 3

In the Session Manager window, select **Laragon** from the session list on the left pane and click **Open**. This will launch HeidiSQL and connect to the database using the default user (root), password ((empty)), and port (3306).



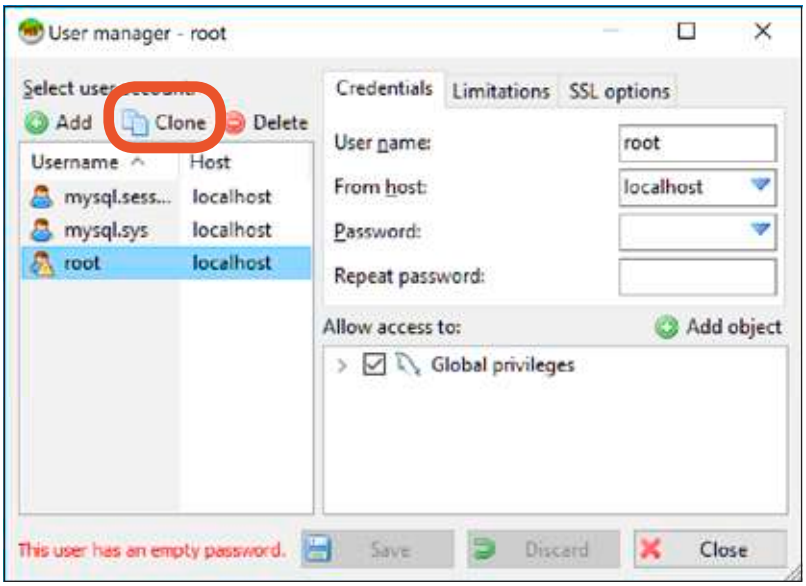
STEP 4

In the HeidiSQL window menu bar, navigate to and click **Tools > User manager.**



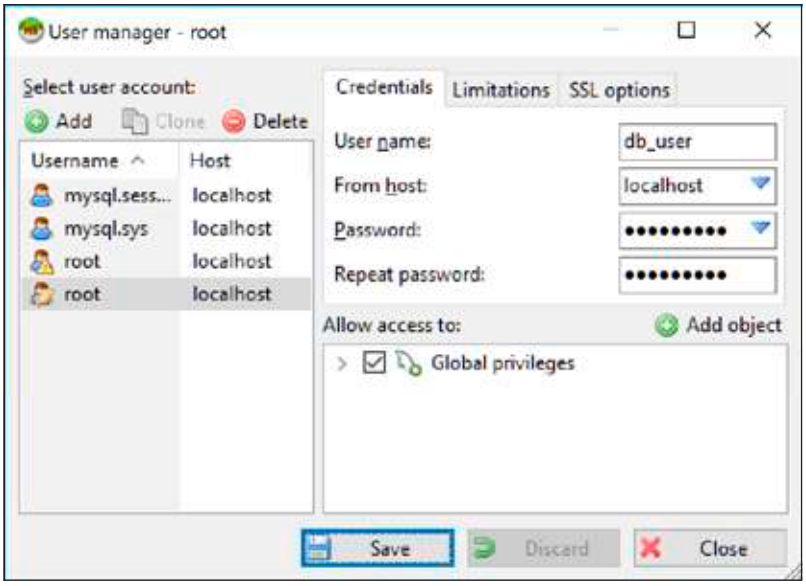
STEP 5

On the left pane, select **root** from the list of usernames and click **Clone.**



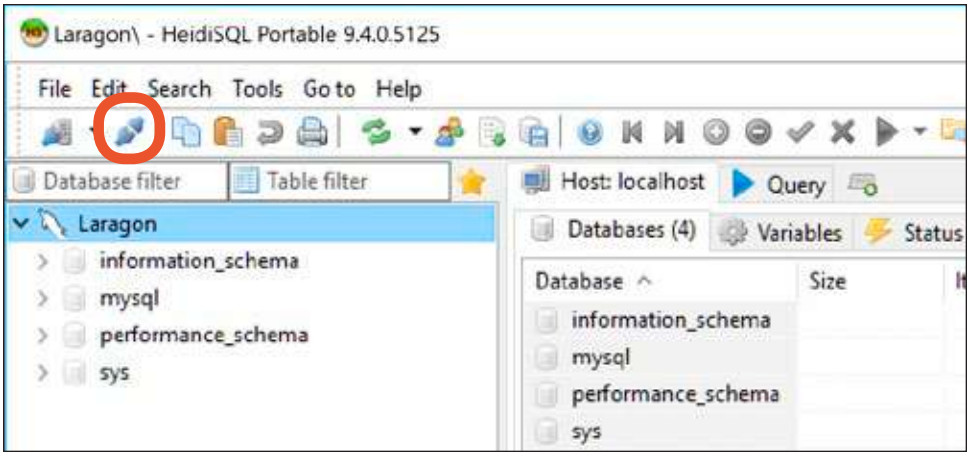
STEP 6

Select the **newly created account**, and specify a **new username and password**. **Take note of the new account details as this will be used later to configure the web application.** Click **Save**, then close the window.



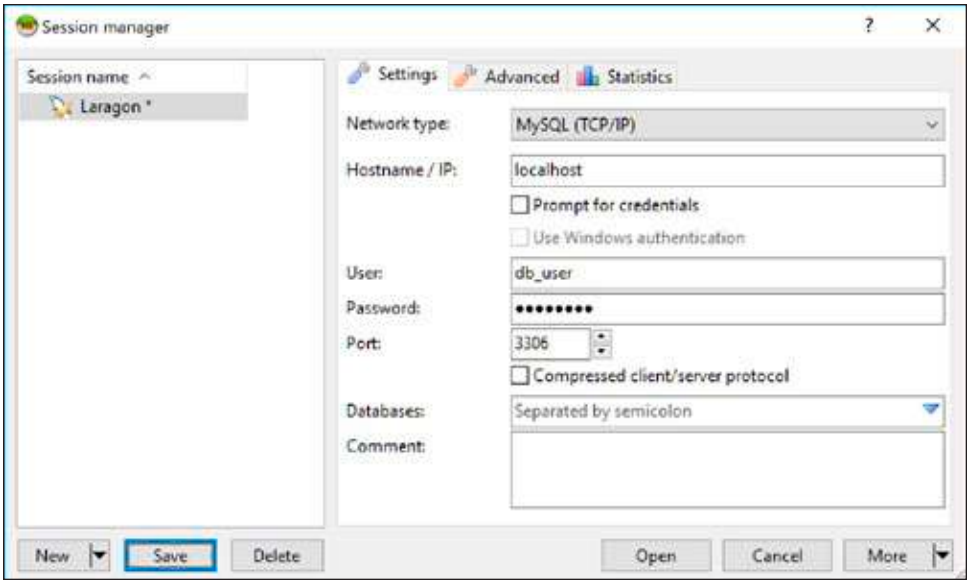
STEP 7

Back in the main window, click the **Disconnect** icon from the toolbar.



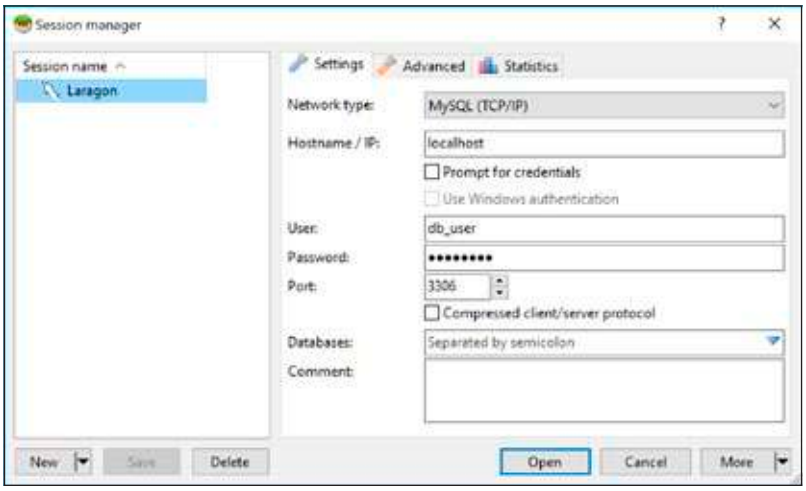
STEP 8

In the Session Manager, enter the **username and password** specified from Step 6 and click **Save**.



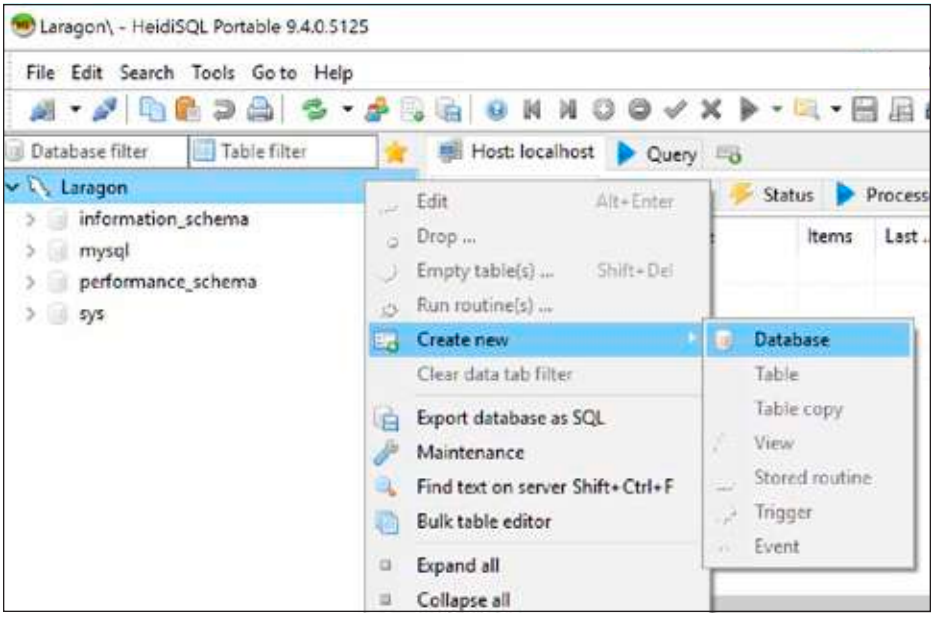
STEP 9

Click **Open** to connect to the database again.



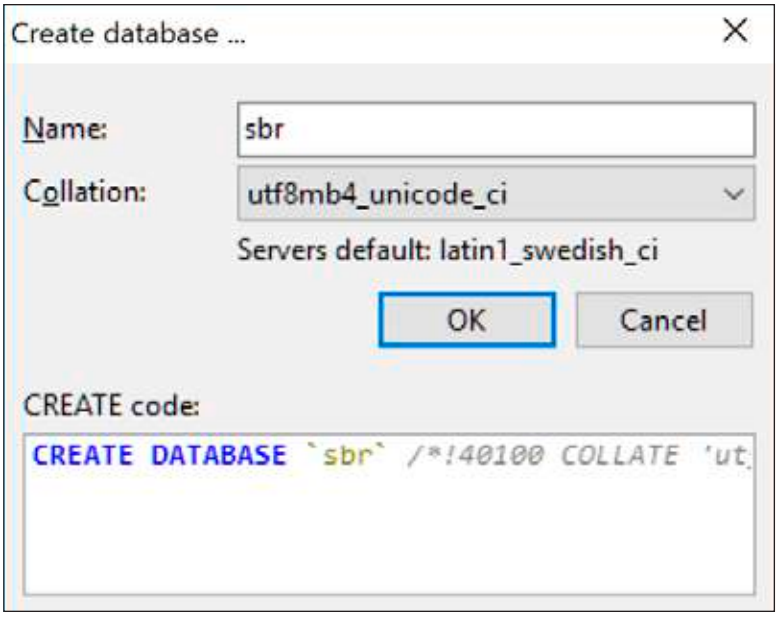
STEP 10

In the main window, right click on **Laragon** from the left pane. Select **Create new > Database.**



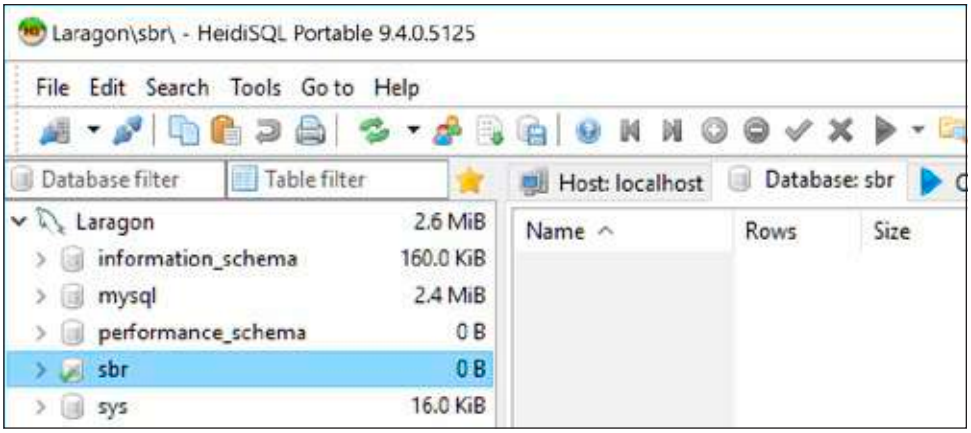
STEP 11

Specify a **name for the ADB SBR database** (e.g. "sbr") and select **utf8mb4_unicode_ci** from the Collation dropdown options. Click **OK**.



STEP 12

Back in the main window, check to **confirm** if the created database is listed under Laragon on the left pane.



II.D. Application Code

II.D.1. Extract the Source Code

The main application source code is within a single ZIP file (**sbr.zip**) that must be decompressed and placed in the web root directory of the web server.

Assumption

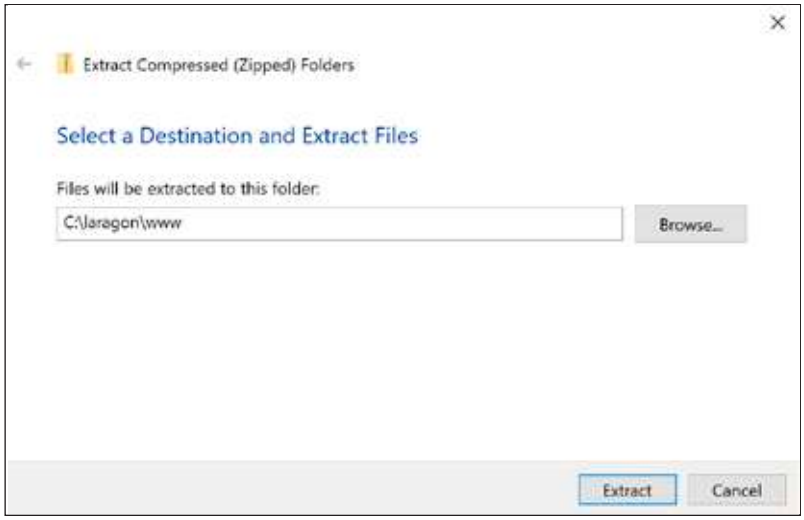
A working copy of the ZIP file sbr.zip is available.

STEP 1

Browse through the directory of the web root folder, **C:\laragon\www**, and **delete** its contents.

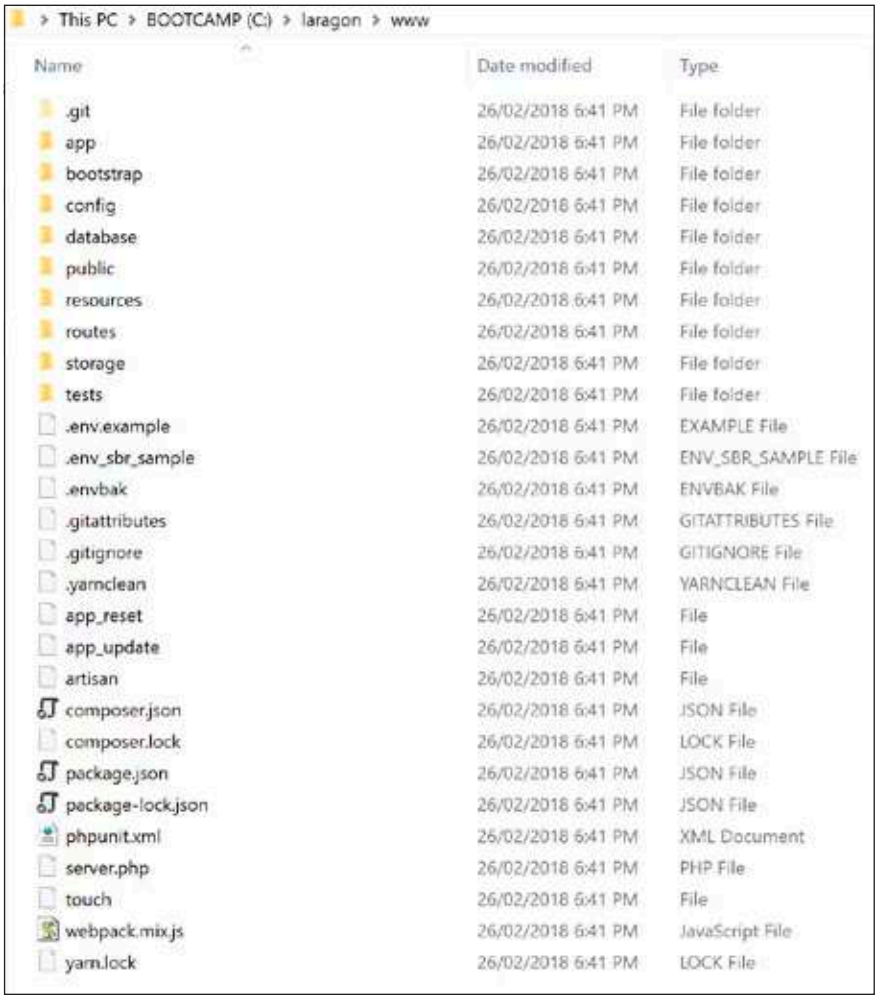
STEP 2

Extract the contents of the **sbr.zip** file into this directory.



STEP 3

After the extraction, **confirm** that the contents of the web root folder C:\laragon\www are similar to the screenshot below.



II.D.2. Modify the .env File

The ADB SBR system has the **.env configuration file** which contains settings on how to connect to the database and email server, as well as how to apply some country-specific text labels and business rules within the web application.

The .env file contains several parameters, and each parameter is followed by an equals (=) symbol and its assigned value. The assigned value is enclosed in quotation marks (“assigned value”).

Some of the parameter values must be modified to reflect country-specific setup and business logic. In other cases, new parameters must be added, while strictly following the same format described. Each parameter must be in its own line (i.e. two or more parameters cannot be in the same line).

Assumption

- The .env file is in the web root folder of Laragon, C:\laragon\www
- A valid email account is designated as the sender of ADB SBR system-generated notifications. It is recommended that this not be the same email account of the administrator.

Parameters syntax

- No spaces before and after the equals symbol separating the parameter and its value.
- No spaces or extra characters after the assigned value.
  Format: **PARAMETER=VALUE**
  Sample: **DB_DATABASE="sbr"**

STEP 1

Browse through the directory of the web root folder, C:\laragon\www, and open the **.env file** using a **text editor.**

STEP 2

Modify the **database parameter values** as follows:

```
 8 DB_CONNECTION=mysql
 9 DB_HOST=127.0.0.1
10 DB_PORT=3306
11 DB_DATABASE=sbr
12 DB_USERNAME=db_user
13 DB_PASSWORD=password
14 DB_DUMP_BINARY_PATH="C:\laragon\bin\mysql\mysql-5.7.19-winx64\bin"
```

STEP 3

Modify the **mail parameter values** using the information of the designated email account to be used and its email server settings. The example provided uses a Gmail account and, therefore, uses settings for Gmail.

```
22 MAIL_DRIVER=smtp
23 MAIL_HOST=smtp.mailtrap.io
24 MAIL_PORT=587
25 MAIL_USERNAME=noreply@sbrdomain.com
26 MAIL_PASSWORD=7459a6417316ca
27 MAIL_ENCRYPTION=TLS
28 MAIL_FROM_ADDRESS=noreply@sbrdomain.com
29 MAIL_FROM_NAME="SBR Administrator"
```

Alternatively, the following parameter values can be used if the designated email account's service provider is Yahoo or Outlook.

Yahoo
MAIL_HOST=smtp.mail.yahoo.com
MAIL_USERNAME=sbr.noreply@yahoo.com
Outlook
MAIL_HOST=smtp-mail.outlook.com
MAIL_USERNAME=sbr.noreply@outlook.com

STEP 4

Modify the **country-specific parameter values** to configure certain text labels and rules on system use.

STEP 5

Save and close the .env file.

Database parameters

DB_DATABASE
Name of the MySQL database which holds the tables for the ADB SBR system. The name specified in Step 11 of ***Section II.C.7. Set up the application database*** must be used.

DB_DUMP_BINARY_PATH
Location of the mysqldump executable, which is used by the system for generating database backups. It may change depending on the MySQL version used.

DB_USERNAME
MySQL account to be used by the ADB SBR system to connect to the MySQL database. The name specified in Step 6 of ***Section II.C.7. Set up the application database*** must be used.

Mail parameters

MAIL_ENCRYPTION
Encryption protocol required by the SMTP server; may be set to null to disable

MAIL_FROM_ADDRESS
Complete email address of the account

MAIL_FROM_NAME
Name to be displayed in the From section of emails received from the email account

MAIL_HOST
SMTP server hostname or IP address

MAIL_PORT
SMTP port number

MAIL_USERNAME
Email address of the account to be used

MAIL_PASSWORD
Password associated with the email account to be used

Country-specific parameters

SBR_ACTIVITY_LEVELS =Section,Division,Group,Class,Subclass
Comma-separated economic activity classification levels, starting from the largest grouping

SBR_ACTIVITY_LOCALIZED=false
Set to "true" if the economic activity classification has an available alternate translation

SBR_CLASSIFICATIONS_DUMP=classifications-PHL.sql
The SQL file which will insert the data into the classification tables

SBR_COUNTRY_CCA3=PHL
Official three-letter country code

SBR_CONTACT_EMAIL=sbr.noreply@gmail.com
Email address of the SBR administrator which will be visible on the Contact Support page

SBR_CONTACT_PHONE="+639999999999"
Contact number of the SBR administrator will be visible on the Contact Support page

SBR_ENABLED_LOCALES=en,ph
Official two-letter language codes for the available language options

SBR_LOCATION_BOTTOM_UP=true
Set to "true" if full location should be displayed starting from the smallest grouping up to the largest grouping
Set to false to display in reverse

SBR_LOCATION_LEVELS ="State,District,Township,Town/VT,Ward/Village"
Comma-separated location classification levels, starting from the largest grouping

SBR_LOCATION_LOCALIZED=true
Set to "true" if the location classification has an available alternate translation

SBR_LOCATION_PADDING=0
Minimum number of characters for the location code
The system pads the location codes with trailing zeroes until the code reaches the minimum length

SBR_MAX_ADMINISTRATORS=1
Maximum number of administrator accounts in the ADB SBR system

SBR_MAX_SUPERVISORS=2
Maximum number of supervisor accounts in the ADB SBR system

SBR_MIN_SUPERVISORS=1
Minimum number of supervisor accounts in the ADB SBR system

SBR_PRODUCT_LEVELS =Section,Division,Group,Class,Subclass
Comma-separated product classification levels, starting from the largest grouping

SBR_PRODUCT_LOCALIZED=false
Set to "true" if the product classification has an available alternate translation

SBR_SEED_CLASSIFICATIONS=true
Set to "true" if the classification tables (location, economic activity, product) are populated with a prepared SQL file

SBR_SEED_RECORDS_COUNT=10000
The number of dummy records that will be inserted to the system

SBR_SEED_RECORDS=false
Set to "true" if the system will be populated with dummy records
Useful only for testing

SBR_SEED_SOURCES=false
Set to "true" if the system will be populated with dummy data sources for records

II.D.3. Complete the Installation

A **shell script** is used to execute the following steps needed by the system:
(i) Download additional files from the internet
(ii) Create the database tables for the ADB SBR system
(iii) Pre-populate the reference tables inside the database
(iv) Create the necessary subdirectories and optimize the asset files inside the www directory

Assumption

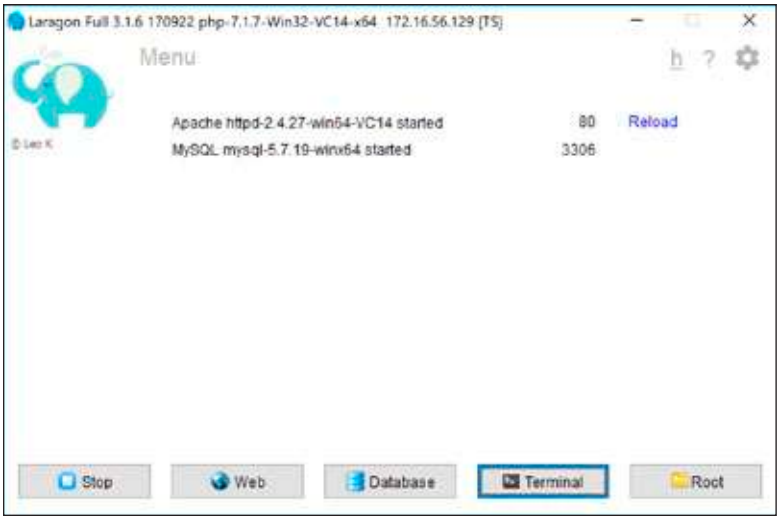
- The server has a secure and reliable internet connection.
- Composer, Node.js, and Git were included in the installation of Laragon.

STEP 1

Launch **Laragon** using the desktop shortcut or the Start Menu item.

STEP 2

In the Laragon window, click **Terminal**.

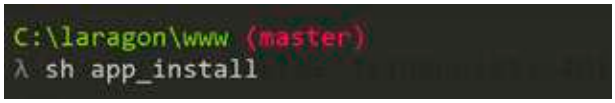


STEP 3

Type **cd C:\laragon\www** and press **Enter**.

STEP 4

Type **sh app_install** and press **Enter**.



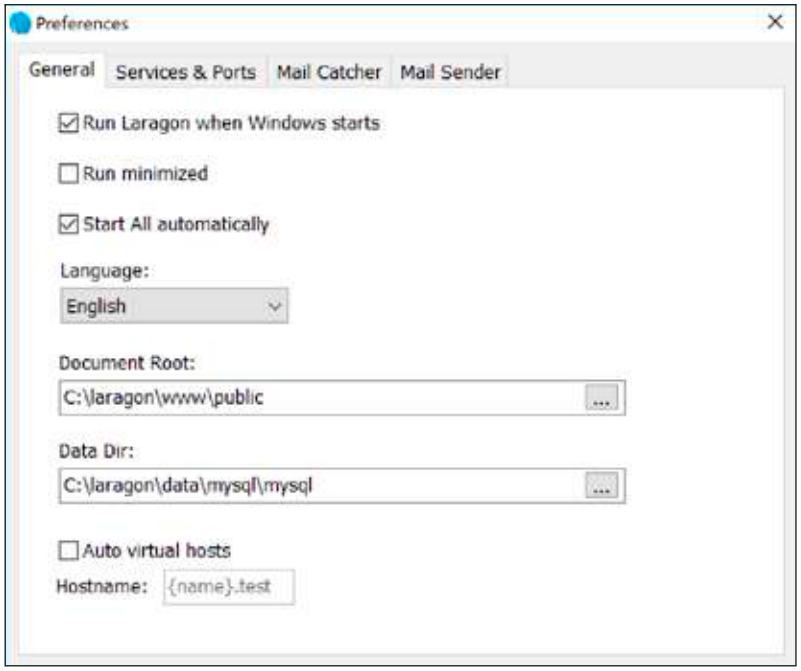
This command executes the shell script to perform the additional steps outlined at the beginning of this section. Wait for the shell script to **complete**.

STEP 5

Back in the Laragon window, click the **gear icon** and select **Preferences**.

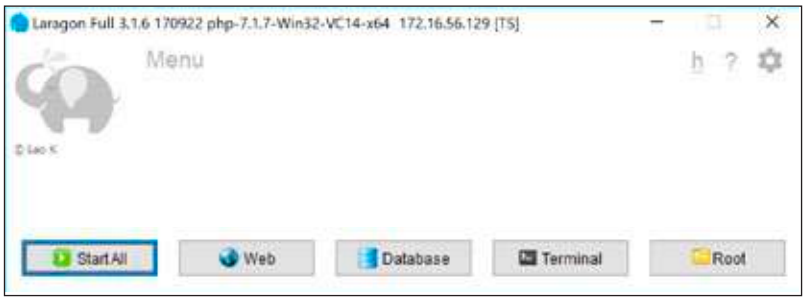
STEP 6

Set Document Root to **C:\laragon\www\public**, and click **Close (X)**.



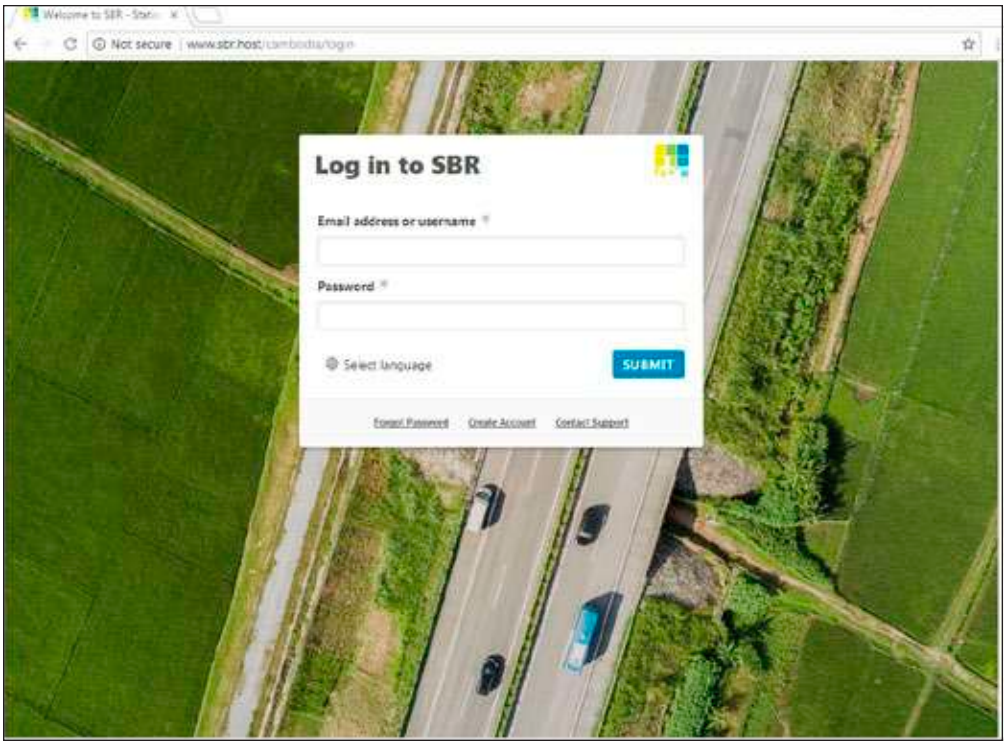
STEP 7

Back in the Laragon window, click **Start All**. Wait for the web server services to finish starting up.



STEP 8

Launch an **internet browser** and type in the URL **http://localhost/**. Press **Enter**. If the ADB SBR login page loads properly, then the system is successfully configured to be accessible.



II.E. Background Services

The ADB SBR system performs certain functions in the background, so that its users are able to change pages without halting the long-running processes. The administrator must use Windows Task Scheduler to ensure that these functions are continuously running in the background and will restart after a failure.

Assumption

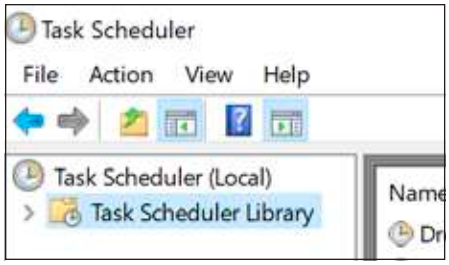
- The Windows operating system installed has Task Scheduler.
- Working copies of the following XML files are available:backup scripts.
  - SBR Service - Batch.xml
  - SBR Service - Default.xml
  - SBR Service - Export.xml
  - SBR Service - Scheduler.xml

STEP 1

Launch **Task Scheduler** using the desktop shortcut or the Start Menu item.

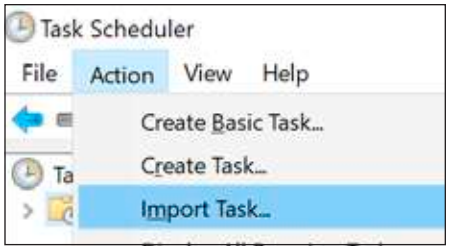
STEP 2

In the Task Scheduler window, select **Task Scheduler Library** from the left pane to display all the tasks that are configured in the server.



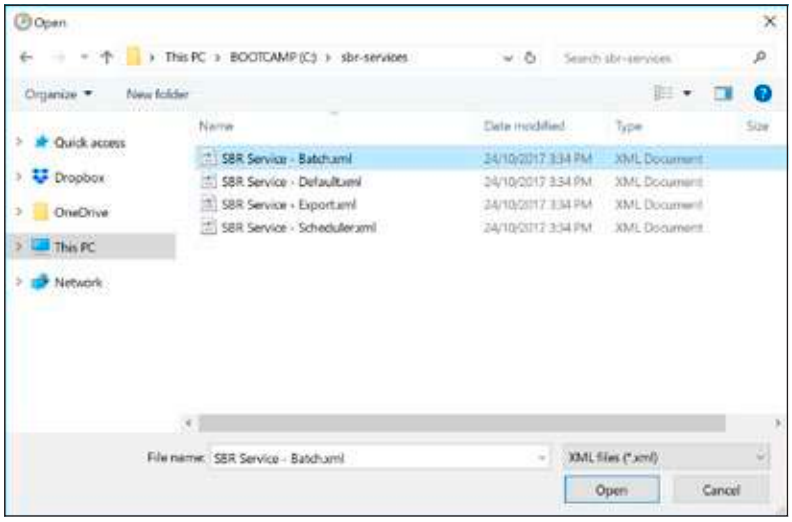
STEP 3

From the menu bar, select **Action > Import Task.**



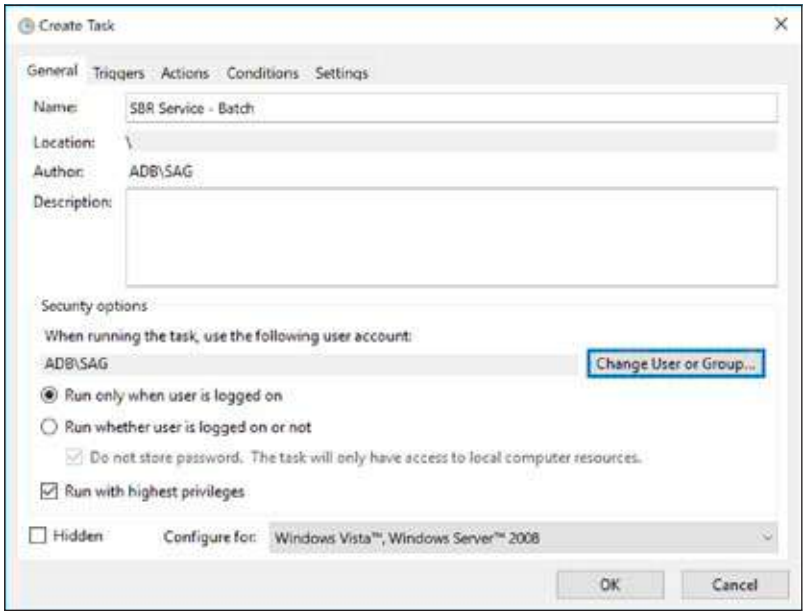
STEP 4

Select any of the XML files from within the directory **C:\laragon\services,** and click **Open**. A **Create Task** window opens.



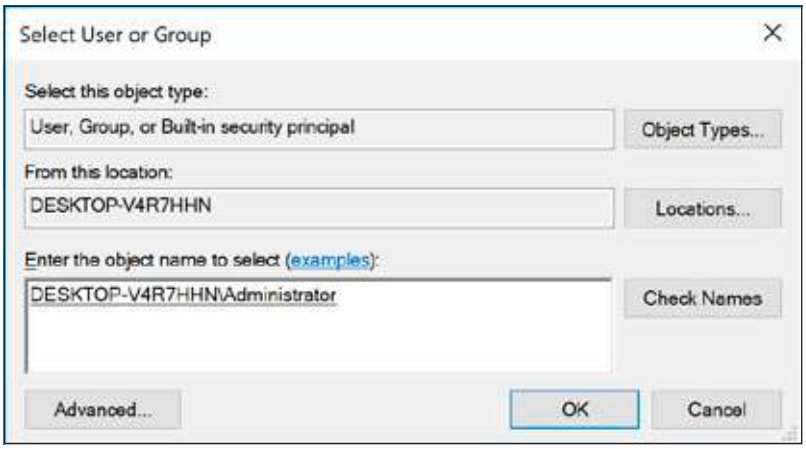
STEP 5

Under the **General** tab of the Create Task window, click **Change User or Group**. A **Select User** or **Group** window opens.



STEP 6

For the Enter the object name to select field, type in the **Windows administrator account name** (the same account used to run Laragon) after the backslash (\ ).



STEP 7

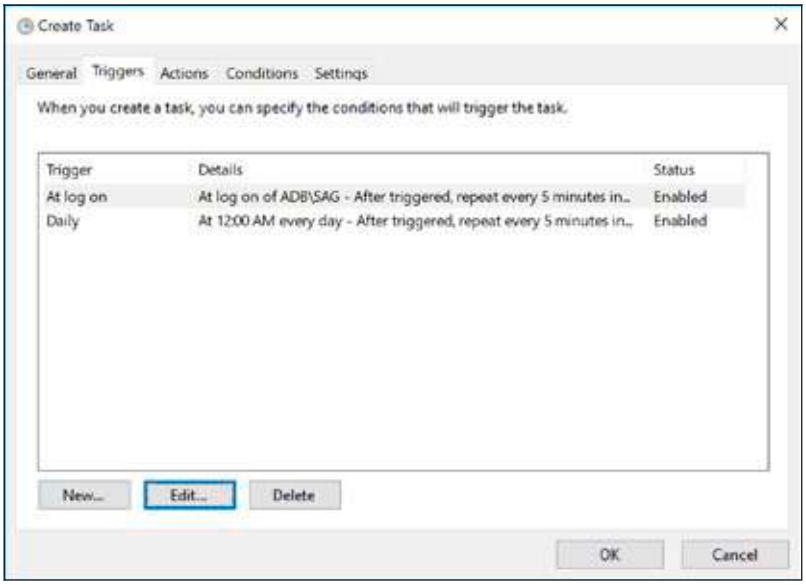
Click on **Check Names** to verify the account name.

STEP 8

Click **OK**.

STEP 9

Under the **Triggers** tab of the Create Task window, select **At log on** from the list and click **Edit**. An Edit Trigger window opens.

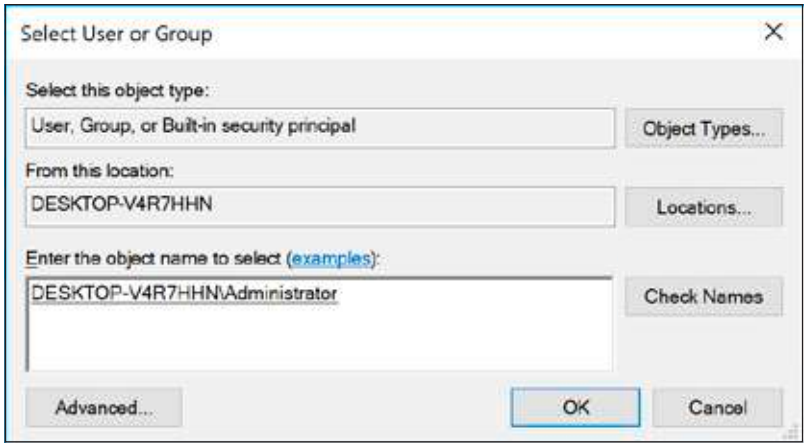


STEP 10

Click **Change User**. A Select User or Group window opens.

STEP 11

For the Enter the object name to select field, type in the **Windows administrator account name** (same account used to run Laragon) after the backslash (\).



STEP 12

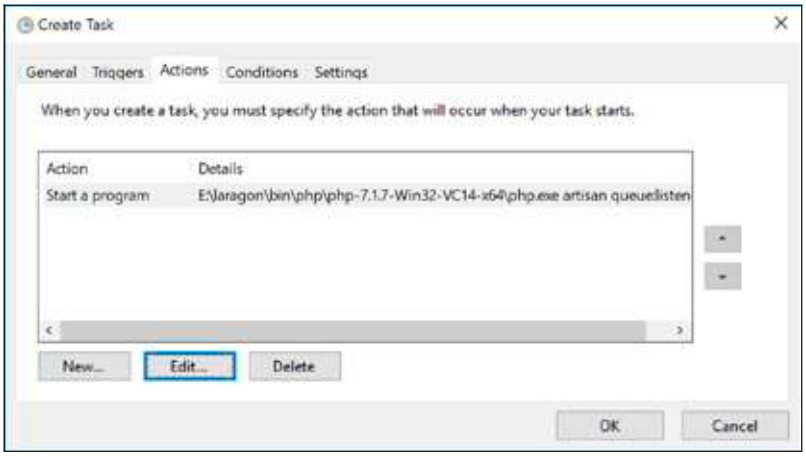
Click on **Check Names** to verify the account name.

STEP 13

Click **OK**.

STEP 14

Under the **Actions** tab of the Create Task window, select **Start a program** from the list and click **Edit**. An Edit Action window opens.



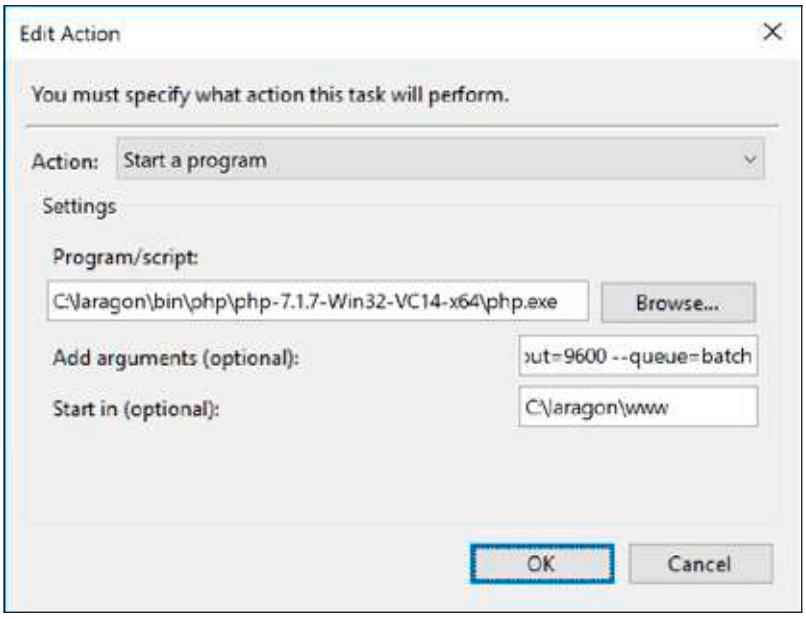
STEP 15

Modify the different settings, as follows:

(i) For Action, select **Start a program**

(ii) For Program/script, click Browse to locate and select the file **php.exe** from the directory **C:\laragon\bin\php\<php version code>**

(iii) For Start in, specify **C:\laragon\www**



STEP 16

Click **OK**.

STEP 17

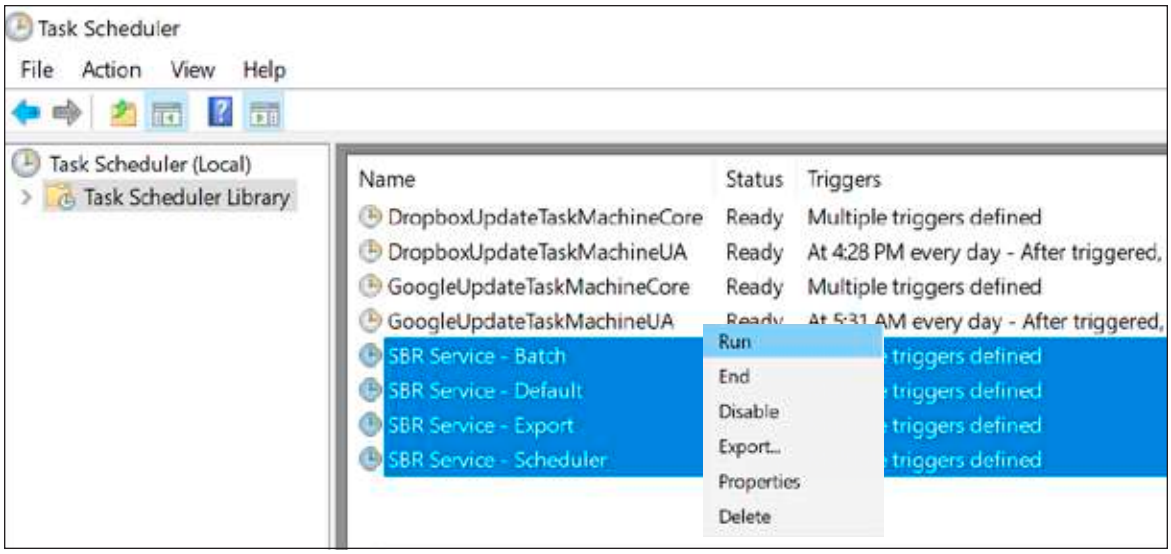
Back in the Create Task window, click **OK** to finish the task import.

STEP 18

Repeat steps 3 to 15 for the other XML files until all four have been completed. No particular order is required.

STEP 19

Once all four XML files are imported, **select all** of them from the Task Scheduler list. Right click and select **Run**.



II.F. Backup and Restore Data

II.F.1. Prepare the Backup Script

Database backups are performed using built-in functions of the MySQL server. A backup script is prepared to setup scheduled or periodic backups. The backup command is wrapped in this script where all the parameters are already set. The backup scripts generate an .SQL file containing the entire database structure, including all its tables and data.

Assumption

- The following directories are created, from ***Section II.C.5. Deploy the application code:***
  - C:\sbr-mysql\backup data
  - C:\sbr-mysql\backup scripts
- A working copy of the .BAT script mysql_backup.bat is inside the directory C:\sbr-mysql\backup scripts.

STEP 1

Browse through the directory C:\sbr-mysql\backup scripts and open the .BAT script **mysql_backup.bat** using a **text editor.**

STEP 2

Edit the following parameters using the specific values entered in ***Section II.C.7. Set up the application database.***

- MYSQL_DATABASE: **Database name** (from step 11)
- MYSQL_USER: **Username** (from step 6)
- MYSQL_PASSWORD: **Password** (from step 6)

```
1  rem set variables for mysql executable path and credentials
2  SET MYSQL_PATH=C:\laragon\bin\mysql\mysql-5.7.19-winx64\bin
3  SET MYSQL_USER=db_user
4  SET MYSQL_PASSWORD=db_password
5  SET MYSQL_DATABASE=sbr
```

Also, check if the following parameters have the correct values:

- MYSQL_PATH (bin folder): **C:\laragon\bin\mysql\<mysql version code>\bin**
- BACKUP_PATH (output folder): **C:\sbr-mysql\backup data**

```
15 rem set variables for the backup folder and name
16 SET BACKUP_PATH=C:\sbr-mysql\backup data
17 SET BACKUP_PREFIX=sbr_backup
18 SET BACKUP_SUFFIX=%YYYY%%MM%%DD%_%HH%%MI%%SS%
```

STEP 3

Save and close the file.

STEP 4

Execute the .BAT script, by double-clicking on the file. This creates a backup of the database, an .SQL file in C:\sbr-mysql\backup data, indicating the current date.

II.F.2. Schedule Regular Backups

Windows Task Scheduler is used for the automatic and periodic backups.

Assumption

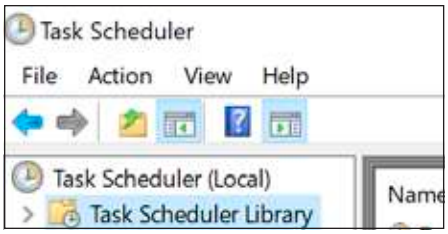
- The backup script is configured properly, from ***Section II.F.1. Prepare the backup script.***
- A working copy of the .XML file MySQL Backup Scheduled Task.xml is inside the directory C:\sbr-mysql\backup scripts.

STEP 1

Launch **Task Scheduler** using the desktop shortcut or the Start Menu item.

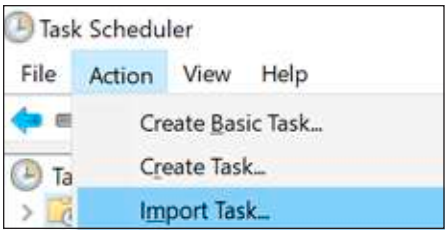
STEP 2

In the Task Scheduler window, select **Task Scheduler Library** from the left pane to display all the tasks that are configured in the server.



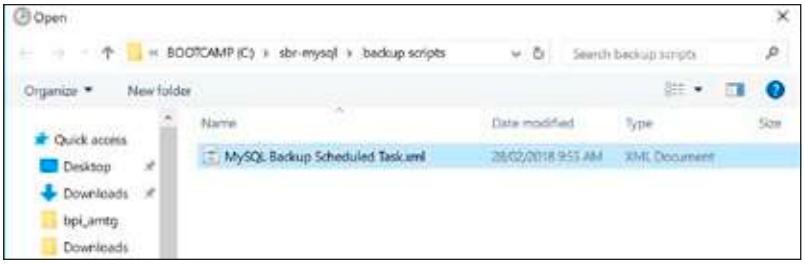
STEP 3

From the menu bar, select **Action > Import Task.**



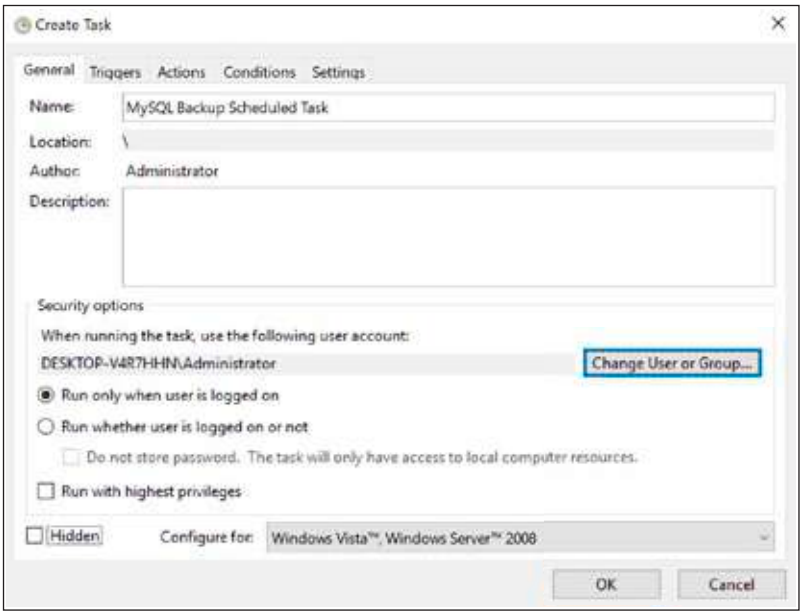
STEP 4

Select the .XML file **MySQL Backup Scheduled Task.xml** from within the directory C:\sbr-mysql\backup scripts, and click **Open**. A Create Task window opens.



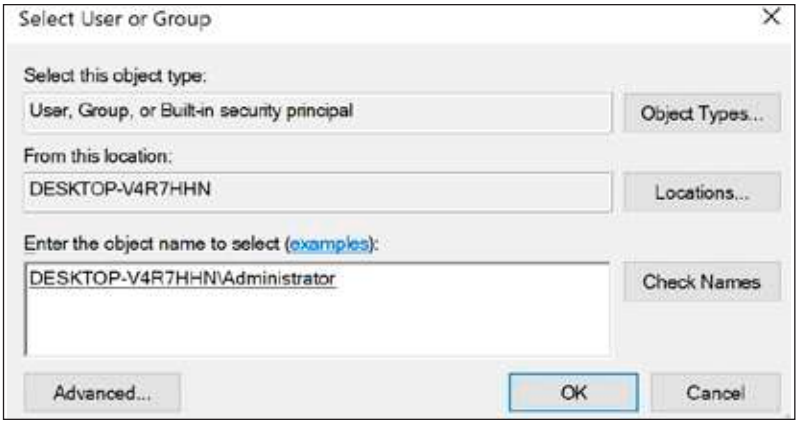
STEP 5

Under the **General** tab of the Create Task window, click **Change User or Group**. A Select User or Group window opens.



STEP 6

For the Enter the object name to select field, type in the **Windows administrator account name** (same account used to run Laragon) after the backslash (\ ).



STEP 7

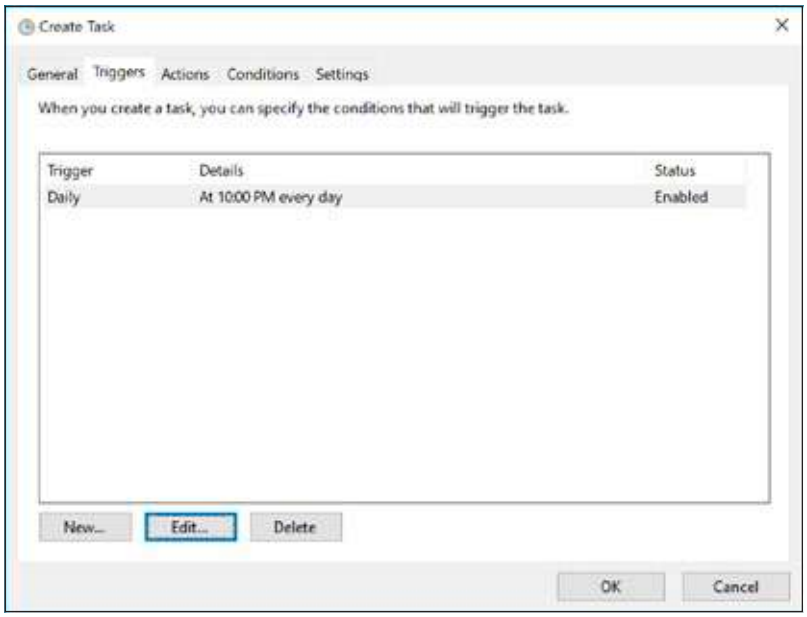
Click on **Check Names** to verify the account name.

STEP 8

Click **OK**.

STEP 9

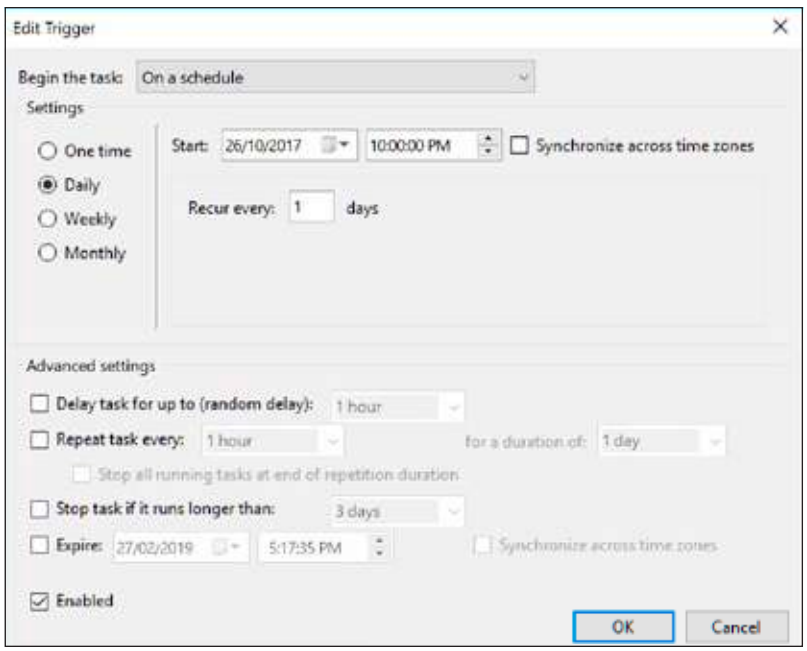
Under the **Triggers** tab of the Create Task window, select **Daily** from the list and click **Edit**. A New Trigger window opens.



STEP 10

Modify the different settings, as follows:

- For *Begin* the task, select **On a schedule**
- Set to **Daily**
- Set a **start time**
- Set to recur **everyday** (i.e. set the value to 1)
- Under Advanced settings, ensure that **Enabled** is or checked

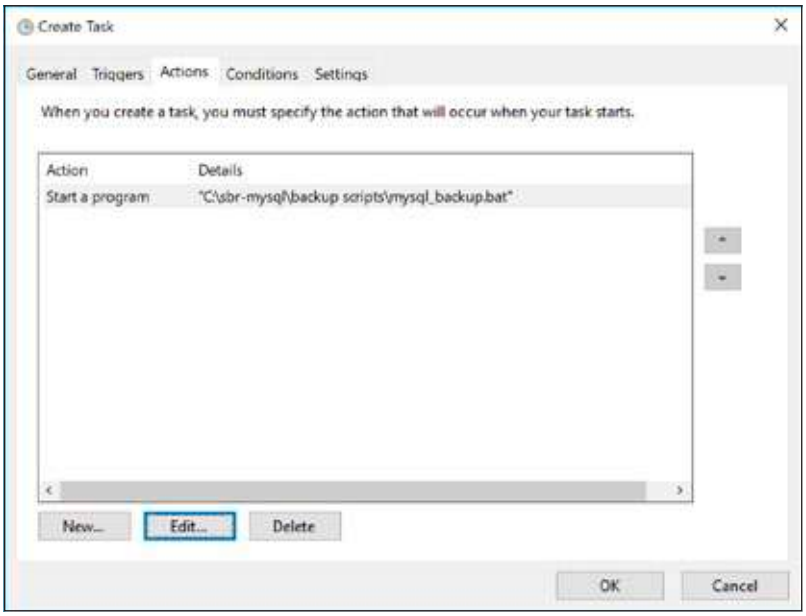


STEP 11

Click **OK**.

STEP 12

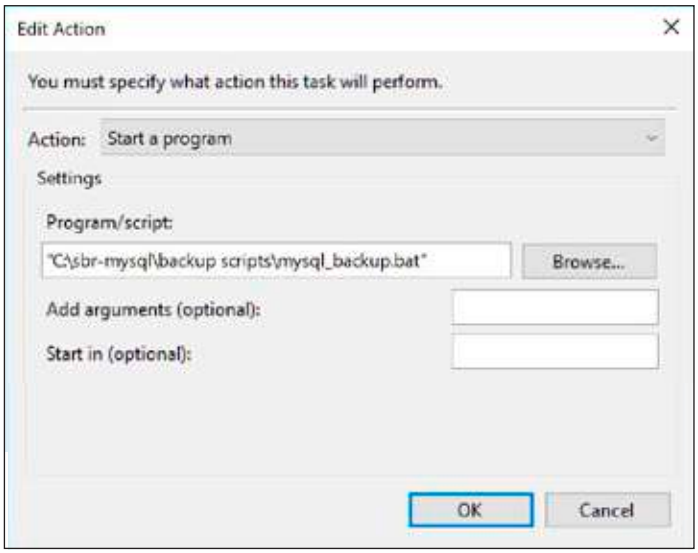
Under the **Actions** tab of the Create Task window, select **Start a program** from the list and click **Edit**. A New Action window opens.



STEP 13

Modify the different settings, as follows:

- For Action, select **Start a program**
- For *Program / script*, click Browse to locate and select the file **mysql_backup.bat** from the directory C:\sbr-mysql\backup scripts



STEP 14

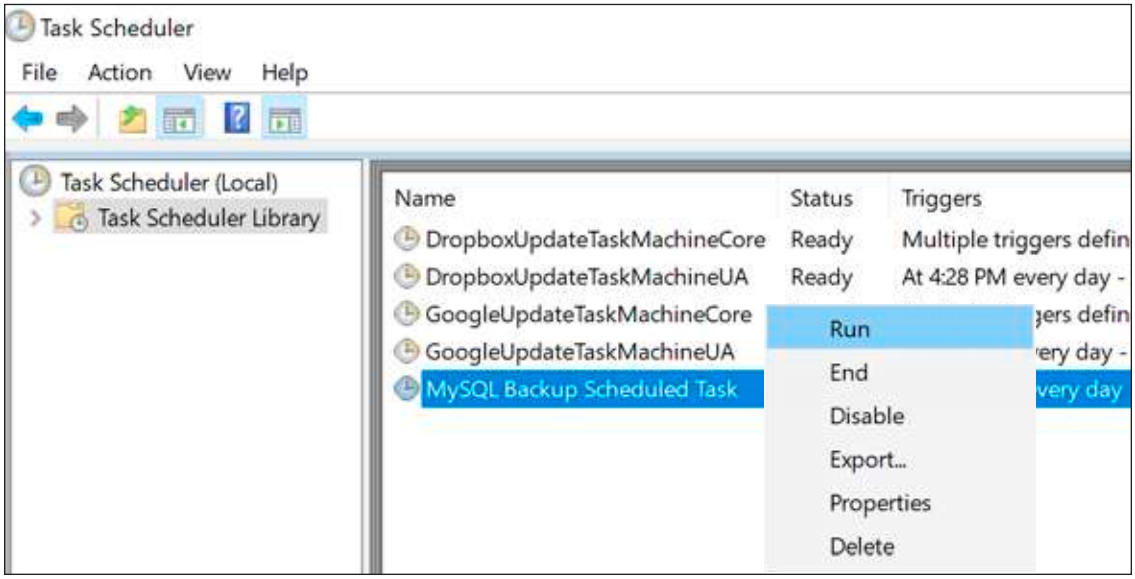
Click **OK**.

STEP 15

Click **OK** in the Create Task window to finish the task import.

STEP 16

To test, select **MySQL Backup Scheduled Task** from the Task Scheduler list. Right click and select **Run**. This creates a backup of the database, an .SQL file in C:\sbr-mysql\backup data, indicating the current date.



II.F.3. Restore from a Backup

An .SQL backup file contains the entire database structure, including all its tables and data at the time the backup was created. Performing a restore simply requires running the .SQL script inside any MySQL client application such as HeidiSQL.

Assumption

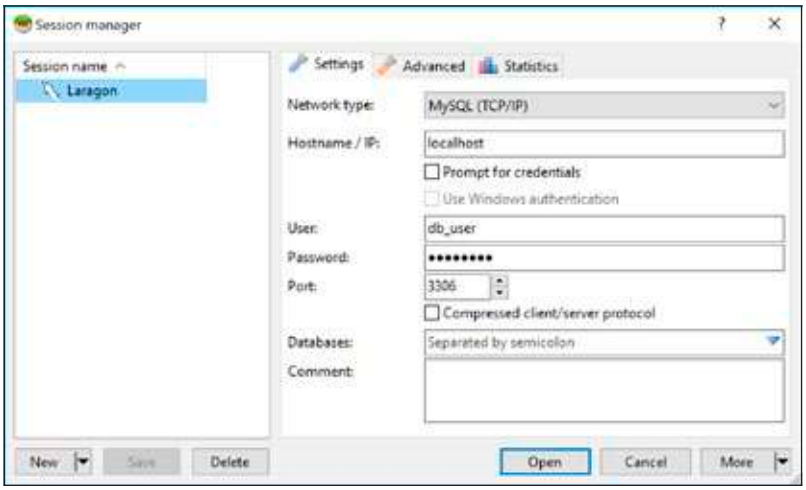
- A working .SQL backup file (generated from the backup procedure) is inside the directory C:\sbr-mysql\backup data.
- HeidiSQL was installed during the Laragon installation.
- The ADB SBR system is in maintenance mode. During this period, no user will be able to access the system.

STEP 1

Launch **Laragon** using the desktop shortcut or the Start Menu item.

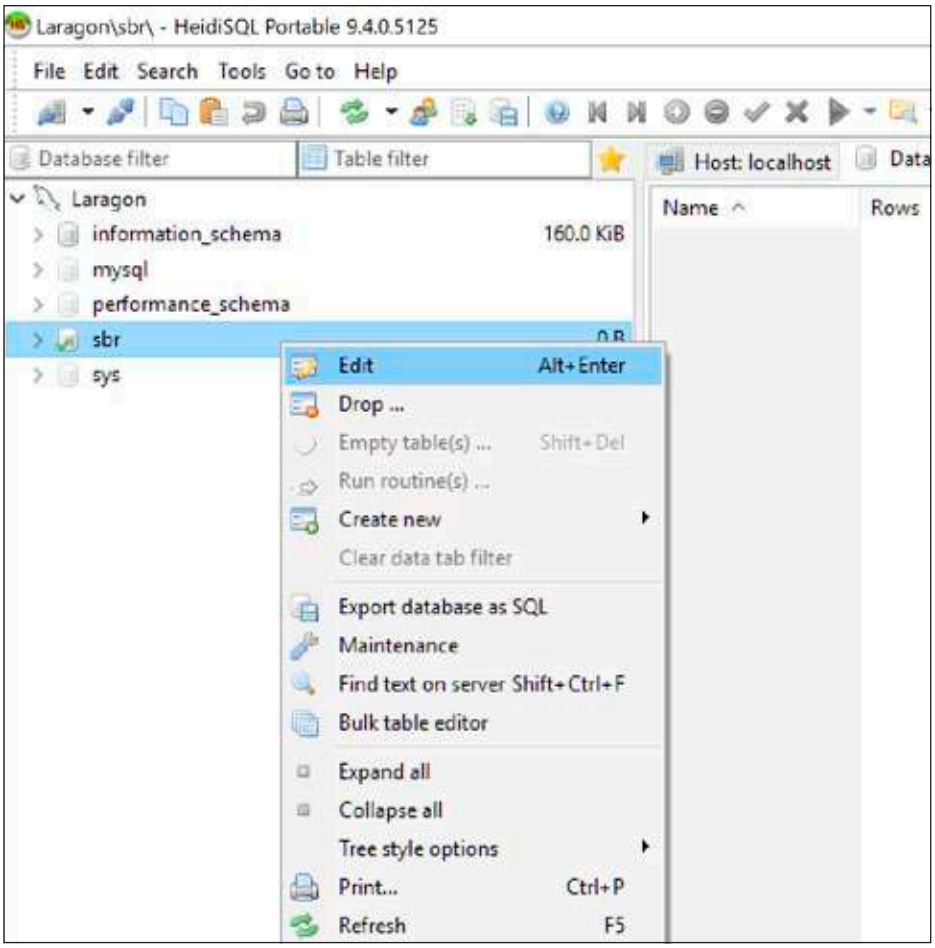
STEP 2

In the Session Manager, enter the **username** and **password** of the account defined in Step 6 of ***Section II.C.7. Set up the application database,*** and click Open to connect to the database.



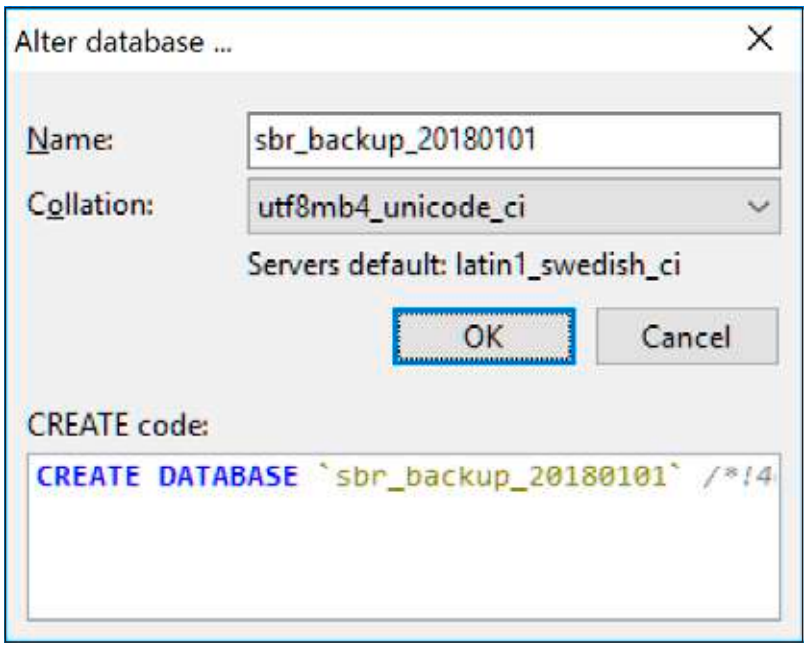
STEP 3

From the left pane, right click on the **SBR database** and click **Edit**.



STEP 4

Rename the database. **Renaming the database ensures that the backup restoration will not overwrite the current ADB SBR database.** It is recommended to specify a database name that is informative (e.g. inclusion of current date or description of a major system change).

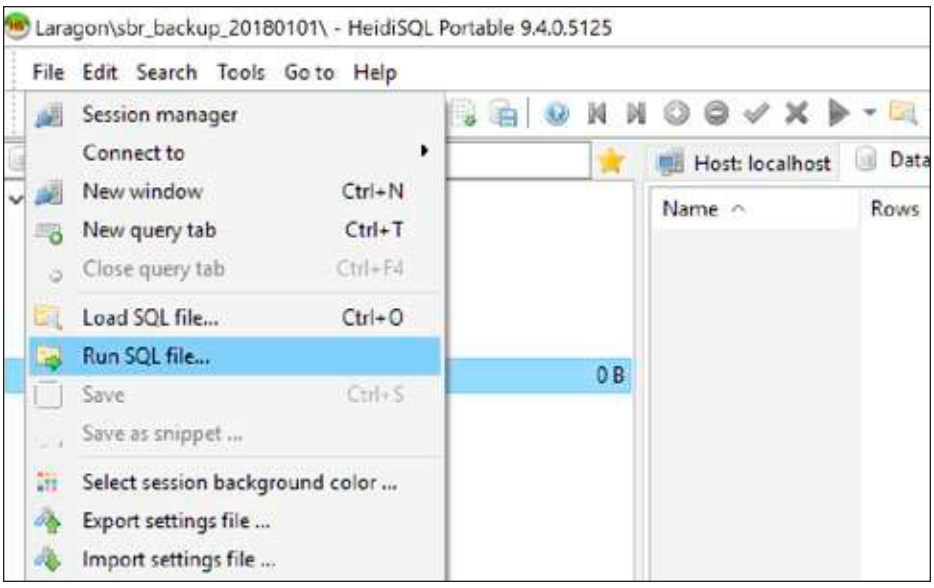


STEP 5

Click **OK**.

STEP 6

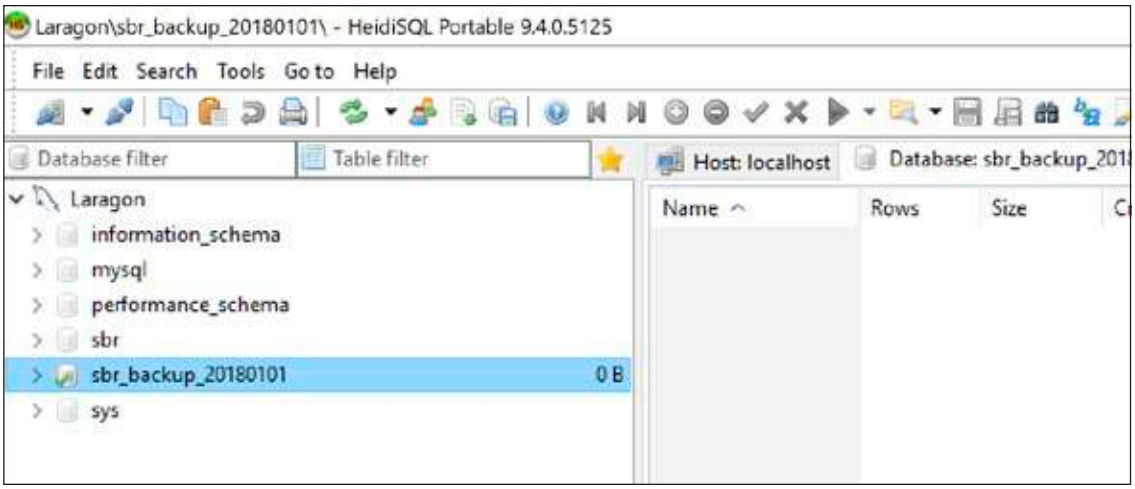
In the HeidiSQL window menu bar, select **File > RunSQL file.**



STEP 7

Navigate to the folder **C:\sbr-mysql\backup data** and select the **.SQL backup file to be restored.** Click **Open**. The restored version of the database will be created and the ADB SBR web application is automatically linked to the newly restored version.

The pre-restoration database renamed in Step 5 above is retained to allow for other database management or maintenance purposes deemed necessary.



USER GUIDE

III.A. Account Types and Access Restrictions

Each user is assigned a role or account type to determine the modules or features he or she can access. The type of account assigned ensures that only the features relevant to the user's work functions are visible and usable. The current version of the system has four account types—Administrator, Supervisor, Encoder, and Stakeholder.

	ADMINISTRATOR	SUPERVISOR	ENCODER	STAKEHOLDER
CAPACITY	Access and control over major components of the system	Access and control over major components of the system	Access to specific components of the system (data entry features)	Limited access to certain components of the system
ROLE	Responsible for system installation, configuration, and maintenance	Responsible for the data integrity (review and approval) of records entered into the system	Responsible for entering and updating records into the system	Only allowed to view pre-selected basic information of the establishment records

Disclaimer: All names, businesses, and data presented in the ADB SBR system screenshots or mentioned in the publication are all randomly generated and fictitious. Any resemblance to actual persons, or businesses is purely coincidental.

III.B. Getting Started

III.B.1. Launch the Application

> **Note**
>
> Before launching the application, make sure the computer is connected to the internet through a secure and reliable network.

STEP 1

Launch the internet browser.

STEP 2

In the browser address bar, input the official ADB SBR web address (or URL) and press **Enter** from the keyboard.

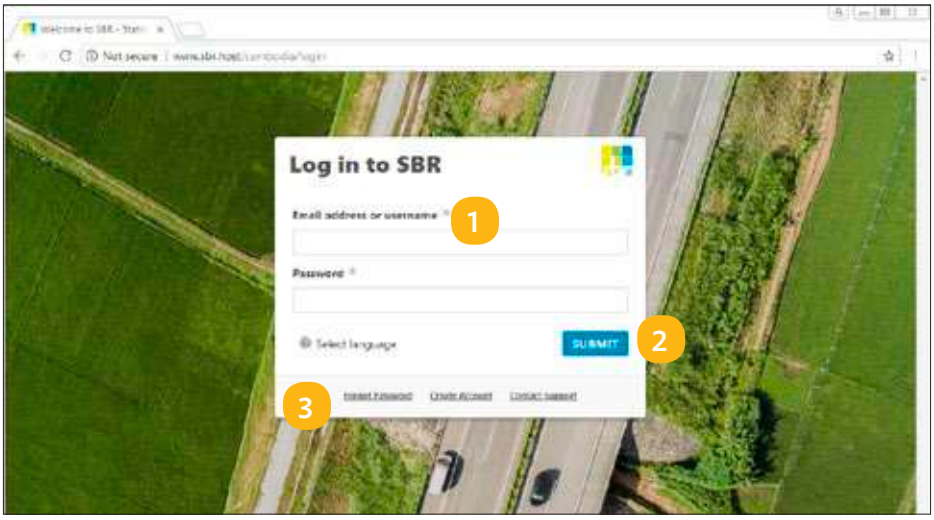
Recommended internet browsers:

Google Chrome
version 61 or later

Mozilla Firefox
version 55 or later

Safari
version 10.1 or later

Get to know the ADB SBR login page



1 **Login fields**
Section III.B.3. Log in

2 **SUBMIT button**

3 **Help links**
Forgot password (*Section III.C.3. Reset password*)
Create account (*Section III.B.2. Create an account*)
Contact support

III.B.2. Create an Account

STEP 1

From the login page, click on the **Create account link.**

STEP 2

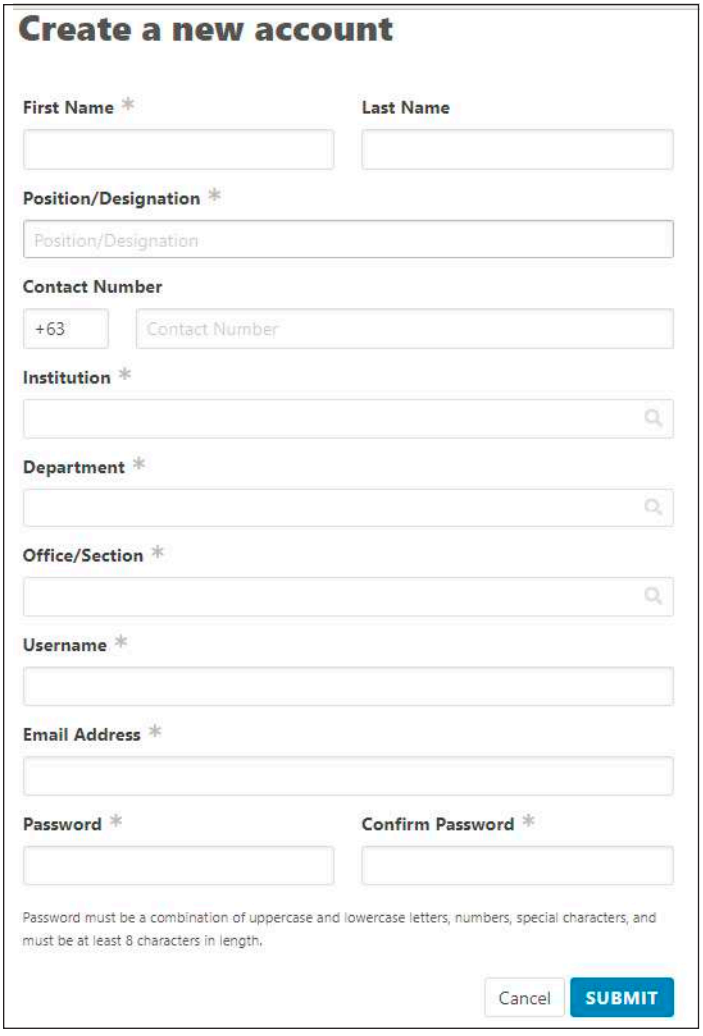
Completely **fill-in** the form with the requested user information and click **Submit**.

STEP 3

Check the **verification request** sent to the email provided and click Verify. This will forward the account creation request to the system supervisor for approval.

STEP 4

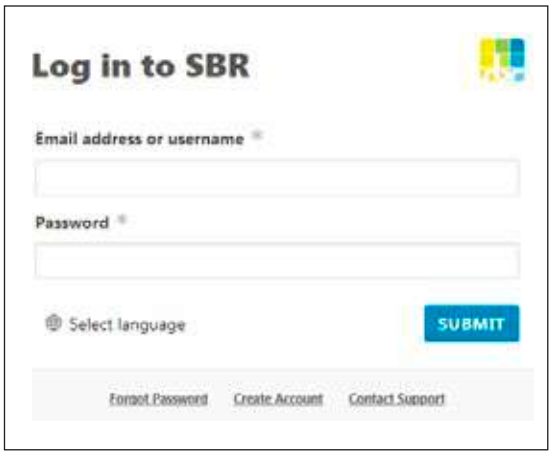
An **account approval confirmation** will be sent to the email provided. Once the request has been approved, log in using the email address/username and password provided.



*The form will not be submitted if a **required field (*)** is left blank or contains errors.*

III.B.3. Log In

Enter the registered email address or username and password and click **Submit**.



Can't log in?

Invalid email address / username

If the user has not registered for an account, see ***Section III.B.2. Create an account.***

If the user has already created an account, check if the login information is correct and that it matches the information entered during account creation.

If the user wants to use another email address or username, see ***Section III.C.1. Update profile.*** The user must be logged in to access this feature.

Invalid password

Five failed attempts will lock the account, and the user will not be able to log in even with the correct password. To recover a locked account, click **Forgot Password** or **Contact Support** from the login page.

III.B.4. Explore the Home Page

1 **MAIN MODULES**
Manage records inside the system depending on role
Note that available modules will vary per account type

2 **DASHBOARD**
Summary of key action items
Note that available information will vary per account type

3 **CLASSIFICATIONS**
View geographic, economic activity, and product classifications

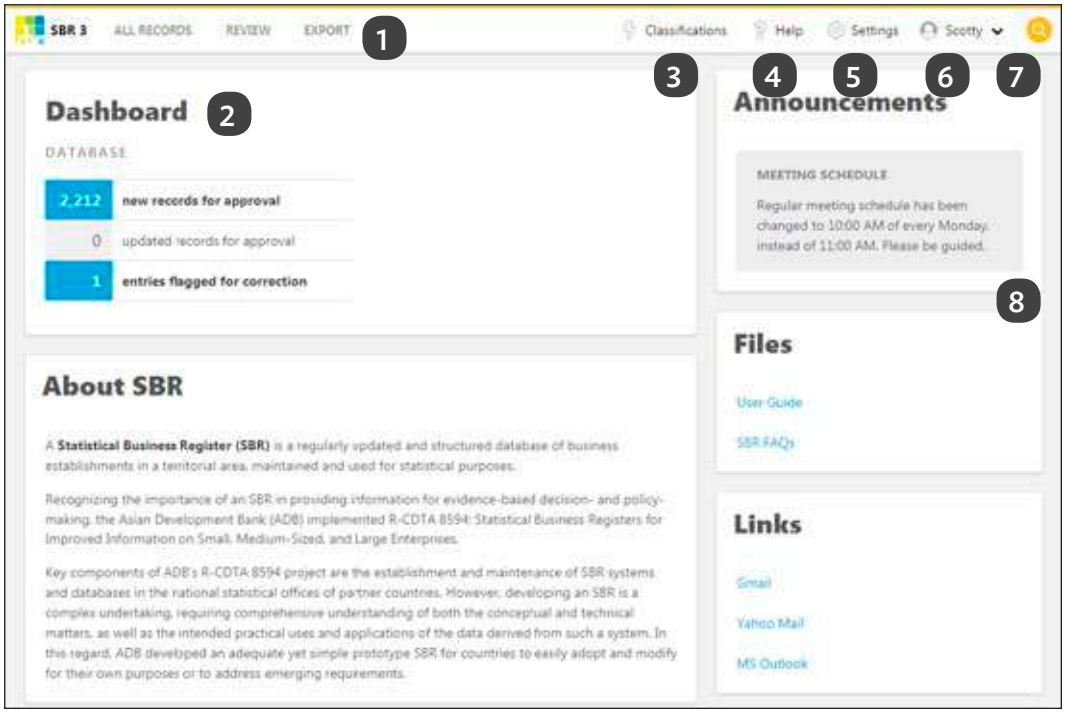
4 **HELP**
View specific instructions or tips for the current page
Note that content varies per page

5 **SETTINGS**
Manage users, data sources, classifications, home page content, and backups (limited to Administrator and Supervisor account types)

6 **USER ACCOUNT**
Update your profile, change your password, select your preferred language, and log out

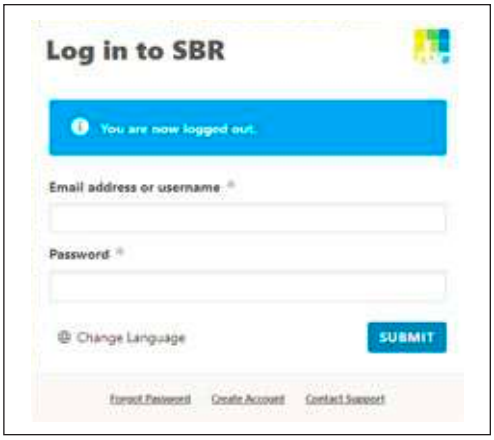
7 **SEARCH**
Find a record by entering its business name or ID number

8 **ANNOUNCEMENTS, FILES, LINKS**
Read announcements, download reference files, and view external links set by the Supervisor or Administrator



III.B.5. Log Out

From the top bar, hover over your account username, and select **Log out.**



III.C. User Account

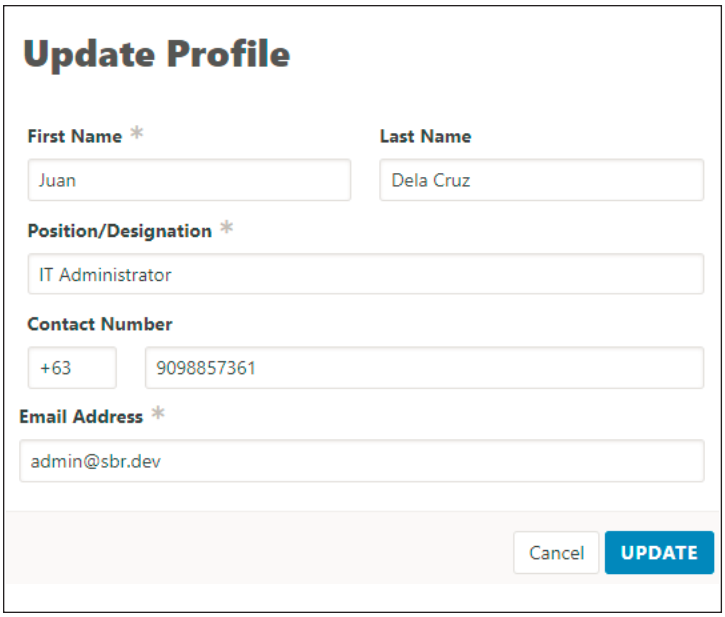
III.C.1. Update Profile

STEP 1

From the top bar, hover over the **Account username**, and click **Update profile**.

STEP 2

Enter the **updated information** in their respective fields, and click Update.



Changing the account type?

The account type is assigned by the system supervisor and **cannot be changed by the user through the Update profile page**. To request for a change in account type, see ***Section III.G.2. Contact support***.

Changing the email address?

Check the verification request sent to the email provided and click Verify. This will save the new email address in the system.

III.C.2. Change Password

STEP 1

From the top bar, hover over your **account** username, and click **Change password.**

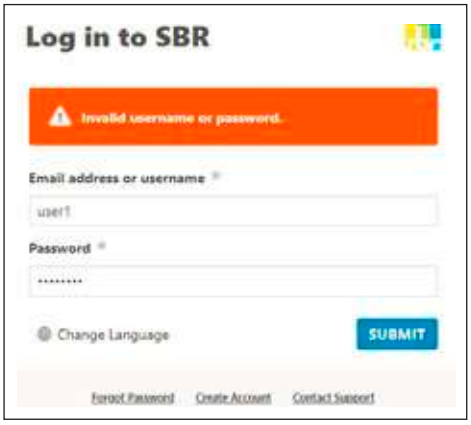
STEP 2

Enter your current password and new password. Confirm your new password and click **Update**.

III.C.3. Reset Password

STEP 1

From the login page, click on the ***Forgot password*** **link.**



STEP 2

Enter the **e-mail address** associated with your account.

STEP 3

Check your e-mail and click on the ***Reset password*** **link.**

STEP 4

Enter your **new password** and click Submit.

III.C.4. Select the Display Language

STEP 1

From the top bar, hover over your **account username**, and click **Select language**.

STEP 2

Click on the preferred language to be used.

Still in English?

Some words and phrases may not have a direct translation in the selected language; in such cases, the system will refer to the original English expression.

Pre-login

The select language feature is also accessible on the login page of the ADB SBR.

III.D. Records

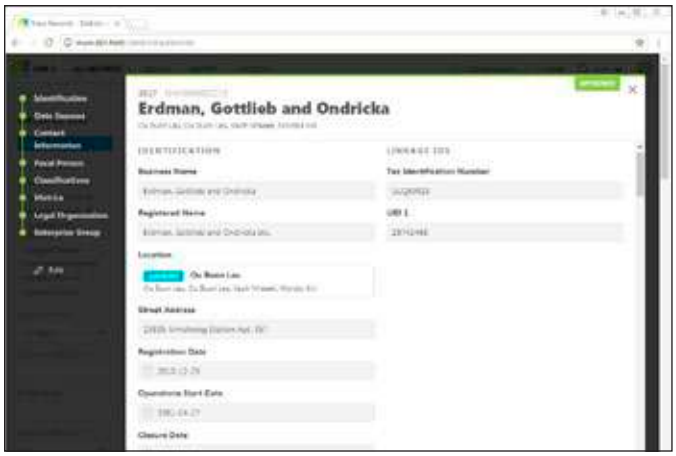
III.D.1. View Record Information

STEP 1

From the top bar, click **All Records**.

STEP 2

Select the **specific record** to be viewed, and click on the **view icon** or the **business name.**



III.D.2. Create a New Record

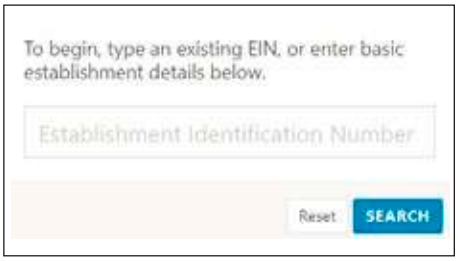
> **Note**
>
> This function is limited to the **Encoder** account type.

STEP 1

From the top bar, click **Create**.

STEP 2

If the EIN is known, enter it in the field and click Search.

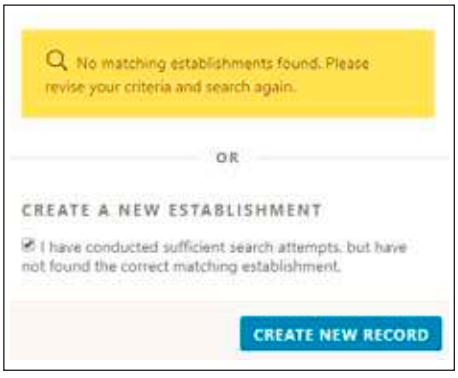


Otherwise, enter the **other requested information** in their respective fields, and click Search.

STEP 3

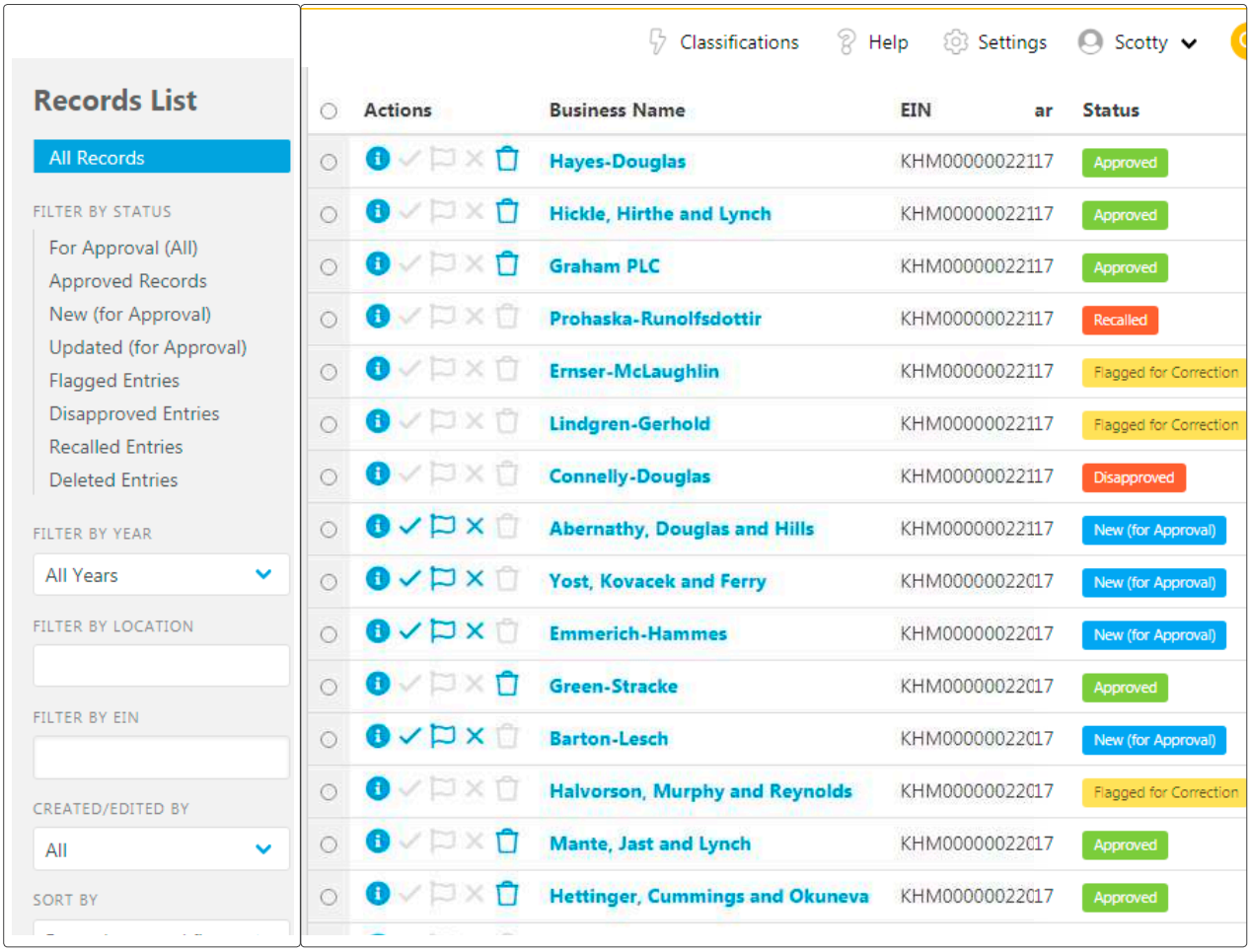
If **no matching records** are found, mark the checkbox, and click **Create New Record**.

Do the same, if there are **returned matches that do not pertain to the same establishment.**



Search

Enter the EIN, business name, or other unique ID to find a specific record. This feature may be used from any page in the system, and not just the Records pages.

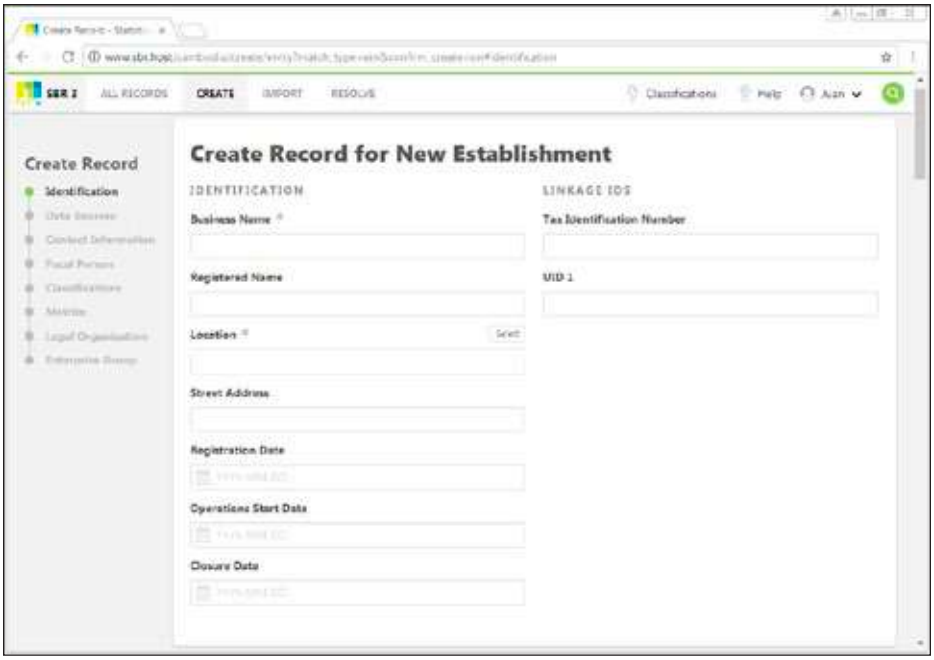


Filter and Sort

Use the different filters available to narrow the list of records to be displayed, or change the sequence of the records displayed by selecting from the options.

STEP 4

Enter all available establishment information in their respective fields, and click **Finish**.



Once submitted, the record will be included in the list for approval of the supervisor. To cancel the submission, see ***Section III.D.4. Recall a submitted record.***

Preliminary matching

To avoid creating duplicate entries, the system will initially scan the database for possible matches.

Matching establishment found?

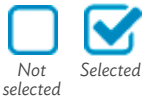
If a returned matching record pertains to the exact same establishment, click **Select**, and proceed to **Step 4**. Since the record is already linked to other existing records in the system, the form will be pre-filled with available information from previous records of the establishment. This speeds up the record creation process.

INPUT FIELD TYPES

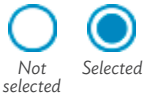
Text field
Most basic and common type; allows the user to enter plain text information

Search-assisted text field
Allows the user to input a partial value and select from the suggestions

Drop-down list
Allows the user to select one from a list of pre-defined options



Check box
Allows the user to select one or more from a list of pre-defined options



Radio buttons
Allows the user to select one from a list of pre-defined options

Date picker
Displays a monthly calendar that allows the user to select a specific date

Add/expand field
Allows the user to add a field of the same type or expand the current field to include disaggregations

SPECIAL INDICATORS

Required
The form will not be submitted if a required field is left blank or contains errors

Automatically generated input
Grayed out fields do not allow manual input from the user because the value is system-generated

III.D.3. Edit an Existing Record

> **Note**
>
> This function is limited to the **Encoder** account type.

STEP 1

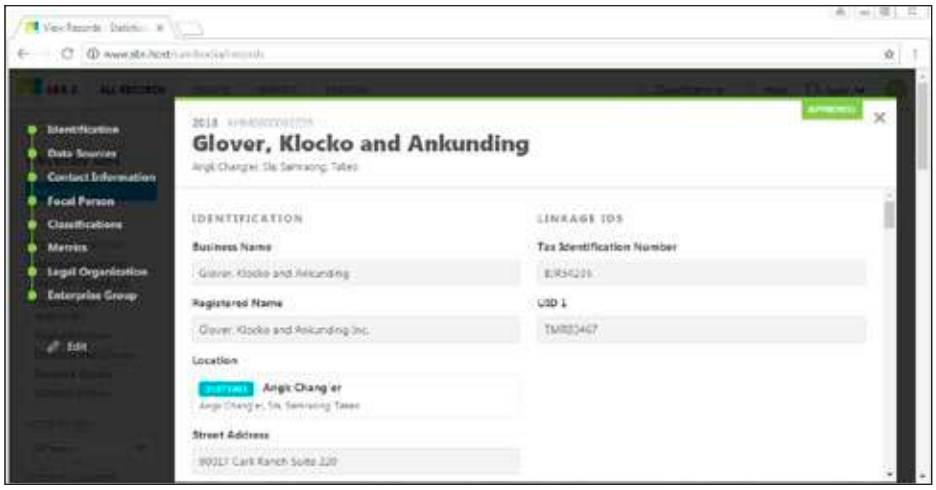
From the top bar, click **All Records.**

STEP 2

Select the **specific record** to be edited, and click on the **view icon.**

STEP 3

From the left pane, click **Edit**.



STEP 4

Enter the establishment information in their respective fields, and click Finish.

Once submitted, the record will already be included in the list for approval of the supervisor. To cancel the submission, see ***Section III.D.4. Recall a submitted record.***

III.D.4. Recall a Submitted Record

> **Note**
>
> This function is limited to the **Encoder** account type.

STEP 1

From the top bar, click **All Records.**

STEP 2

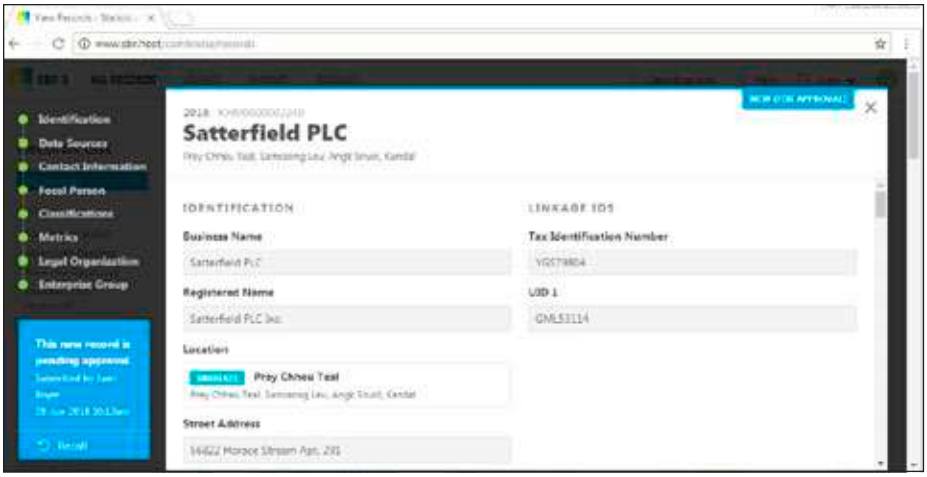
Select the **specific record** to be recalled, and click on the **view icon.**

STEP 3

From the left pane, click **Recall**.

STEP 4

Enter the reason for recalling the record, and click **Submit**.



Once recalled, the record will be removed from the list for approval of the supervisor.

III.D.5. Review a Submitted Record

> **Note**
>
> This function is limited to the **Supervisor** account type.

STEP 1

From the top bar, click **Review** to display all records submitted for approval.

STEP 2

Select the **specific record** to be reviewed, and click on the **view icon** to open the detailed information of the record.

STEP 3

Based on the reviewed information, **approve, flag, or disapprove** the record using the buttons on the left pane.

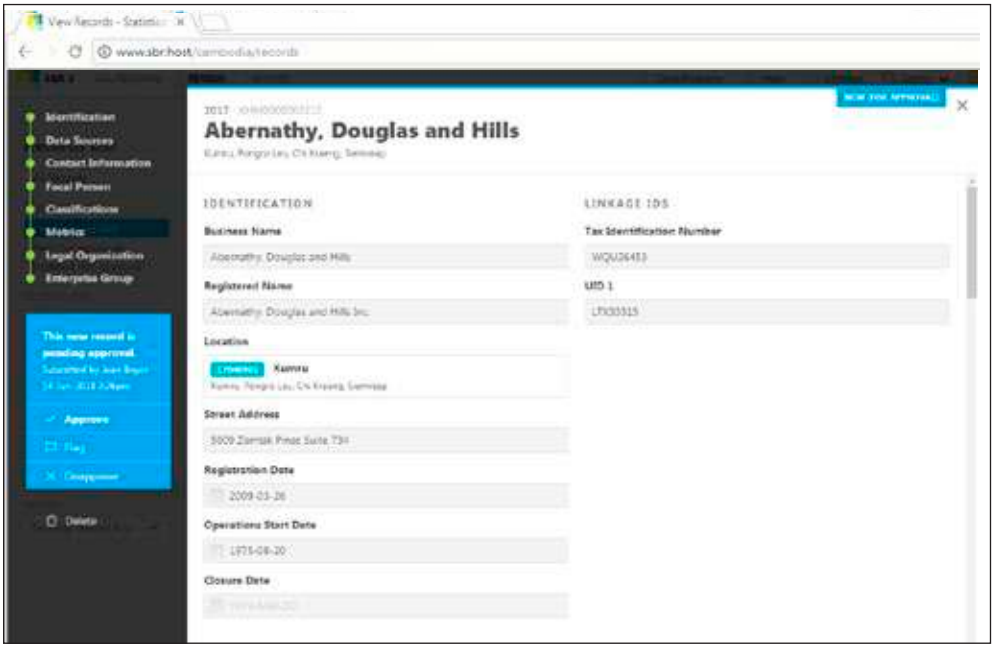
Actions	Business Name	EIN	Location	Year	Status
	Abernathy, Douglas and Hills	KHM0000002210	Kumru	2017	New (for Approval)
	Yost, Kovacek and Ferry	KHM0000002209	Chey Chumneah	2017	New (for Approval)
	Emmerich-Hammes	KHM0000002208	Souriya	2017	New (for Approval)

Action Shortcuts

Entries can be approved, flagged, or disapproved by clicking on the respective icons, without having to open the entry for a detailed view.

Actions for Multiple Records

Click on the record selector(s) ○ and select the desired action from the upper-right portion of the screen.



Approve

Accepts the record information as the establishment information for the year indicated (until further updates); information will be visible to stakeholder account types and will form part of exported information to create reports, statistics, etc.

Disapprove

Rejects the record information as the establishment information for the year indicated.

Flag for correction

Withholds approval or disapproval of the record to a later time, while providing instructions to the assigned encoder to revise the record for approval.

III.D.6. Delete an Approved Record

> **Note**
>
> This function is limited to the **Supervisor** account type.

STEP 1

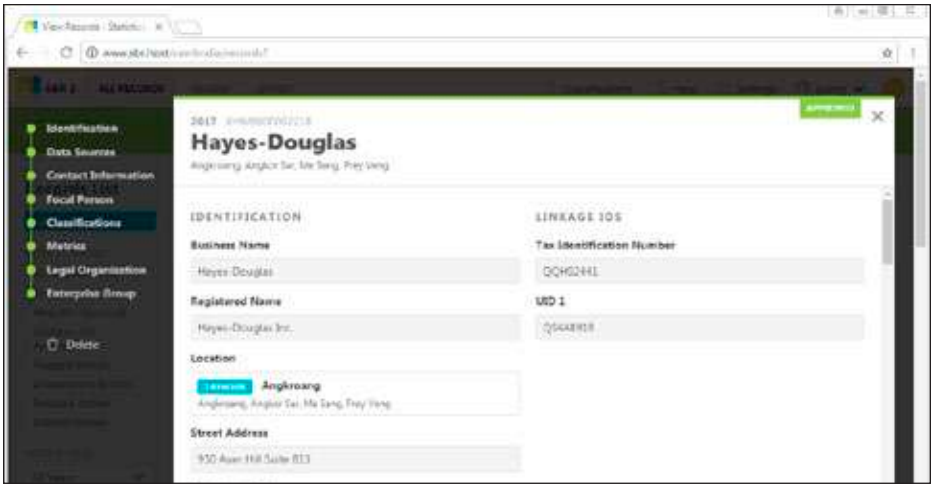
From the top bar, click **All Records.**

STEP 2

Select the **specific record** to be deleted, and click on the **view icon.**

STEP 3

From the left pane, click **Delete**.



STEP 4

Enter the reason for deleting the record, and click Submit.

Once deleted, the record will be removed from the ADB SBR database and the information will not form part of exported datasets for reports, statistics, etc.

III.E. Import and Export

III.E.1. View Imports List

> **Note**
>
> This function is limited to the **Encoder** account type.

STEP 1

From the top bar, click **Import**.

Filter and Sort

Use the different filters available to narrow the list of imports to be displayed, or change the sequence of the files displayed by selecting from the options.

Action buttons

View the import results (for Done status), progress (for Processing / Queued status), or the draft import configuration (for Pending Action status), and **download** the file.

Import new

Start the import process for a **new file.**

III.E.2. Import Records from File

Note

This function is limited to the **Encoder** account type.

Before starting the import, check if:

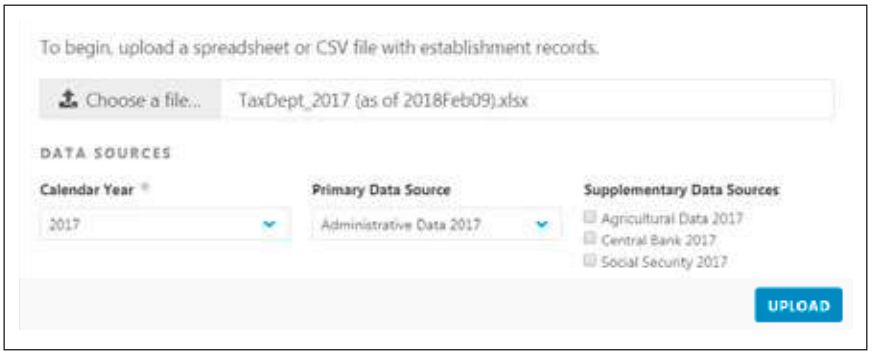
- The target file is in **XLSX**, **CSV**, or **ODS** format.
- The corresponding **data sources** for the records on file are defined. (If not, coordinate with an Administrator or Supervisor.)
- **One row corresponds to one record, and one cell corresponds to one variable.**
- **The required variables** (Business Name and Location Code) are not empty for each record. (This can be changed during system setup.)
- One row contains the **column headings or variable names**. (Not required but will be significantly helpful during the upload process.)

STEP 1

From the top bar, click **Import**.

STEP 2

From the upper right portion of the imports list, click **New**.



STEP 3

Select the **file** to be imported.

STEP 4

Select the **calendar year and data sources** to tag the imported records to, and click **Upload**.

STEP 5

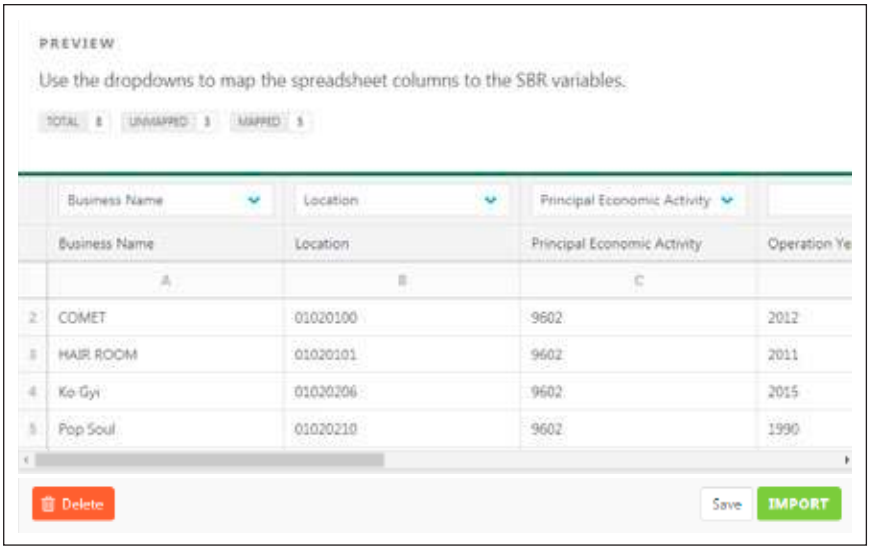
Configure the import by selecting the **Sheet name, row number of the headings, and start and end rows of records** to be imported.

STEP 6

Click **Next** to proceed, or **Delete** to cancel the import.

STEP 7

Using the preview of the spreadsheet, **select or map each of the spreadsheet columns to its corresponding SBR variable.** Unmapped columns will not be included in the import.



STEP 8

Click **Import** to proceed, **Save** to continue the process at a later time, or **Delete** to cancel the upload. The import process will continue even if you leave the import page or log out.

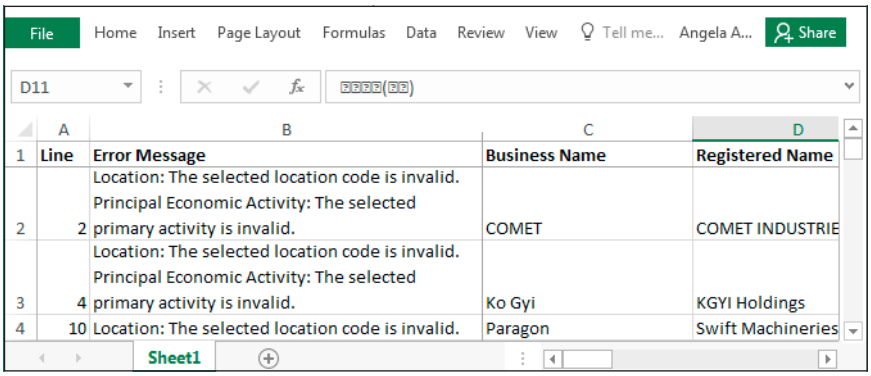
What to do with rejected records

STEP 9

From the import results, **download** the list of rejected records.

STEP 10

Using a spreadsheet processor, **validate and revise the record information** based on the provided **error message**. **Save** the revised file.



	A	B	C	D
1	Line	Error Message	Business Name	Registered Name
2	2	Location: The selected location code is invalid. Principal Economic Activity: The selected primary activity is invalid.	COMET	COMET INDUSTRIE
3	4	Location: The selected location code is invalid. Principal Economic Activity: The selected primary activity is invalid.	Ko Gyi	KGYI Holdings
4	10	Location: The selected location code is invalid.	Paragon	Swift Machineries

STEP 11

Repeat the import procedure (steps 1 to 8 above) for the revised file.

III.E.3. Resolve Record Matches from Import

Note

This function is limited to the **Encoder** account type.

Match to submit

The imported records are categorized based on the **number of potential establishment matches** they currently have in the system (**zero, one, or more than one**). Each record from each category must be matched to **complete the submission for approval.**

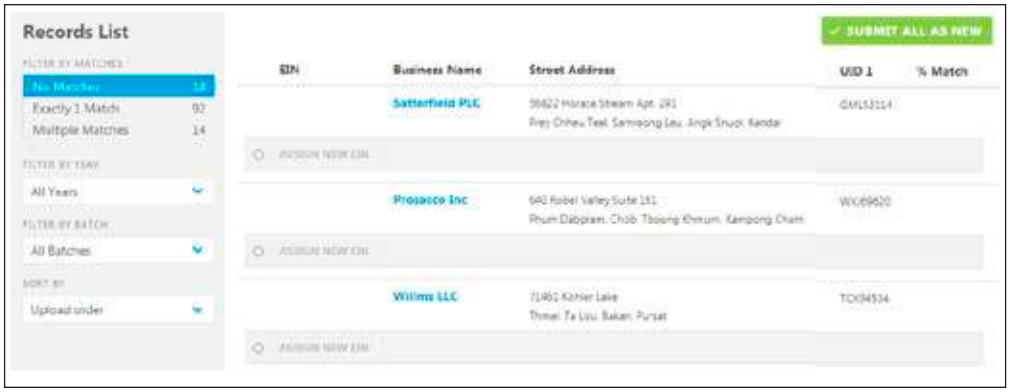
A **matching algorithm** is used to rate the **level of similarity** (see ***% Match*** column) between the Identification variables of the newly imported record as against the existing approved establishment records in the system. This is done for each imported record.

STEP 1

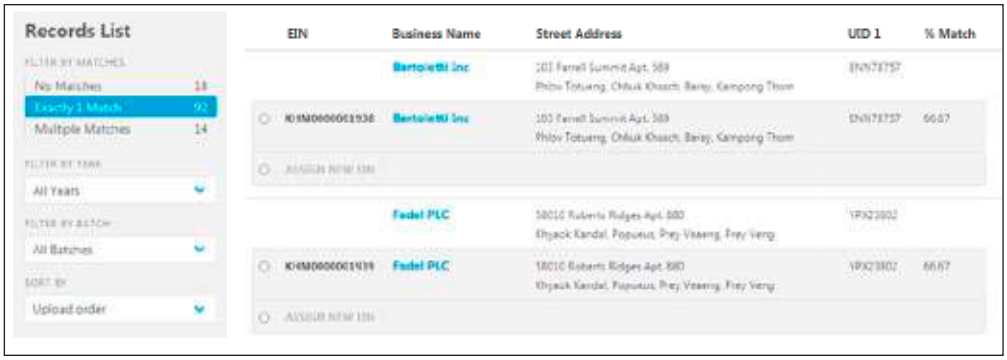
From the top bar, click **Resolve**.

STEP 2

For records with **no matching establishments**, mark the **Assign New EIN** selector ○ for the record/s and then click **Apply** from the upper-right portion of the screen. This s**ubmits the individually selected records as new establishments with new EINs.**



Alternatively, click **Submit All As New** without individually marking each selector. This automatically **submits ALL records as new establishments with new EINs.** As such, this feature must only be used after precautionary review and validation on the records have been conducted.



STEP 3*

For records with **matching establishments,** review the matching establishment/s.

Mark the selector ○ for the **correct matching establishment.** If none of the returned matches are correct, mark the **Assign New EIN** selector ○ for the record/s.

**This step may be accomplished for multiple records before proceeding to Step 4.*

STEP 4

After selecting an option for the record, click **Apply** from the upper right portion of the screen.

If **Assign New EIN** is selected, this **submits the record as a new establishment with a new EIN.**

If a matching establishment is selected, this **submits the record as a new or updated record for the linked establishment.**

III.E.4. Export List of Establishments

> **Note**
>
> This function is limited to the **Supervisor** account type.

STEP 1

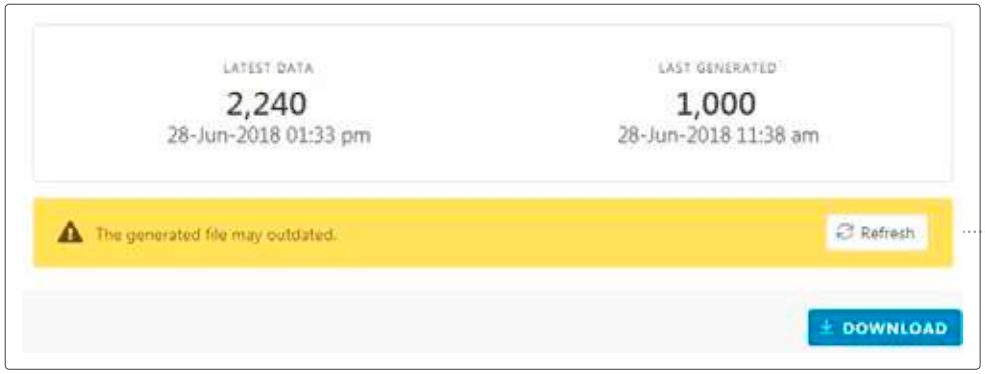
From the top bar, click **Export.**

STEP 2

From the side bar, select **All Establishments.**

STEP 3

Click **Download**. This saves an XLSX file of the **complete list of establishments and their latest approved Identification Variables.**



Update the data for export

The system will prompt for the need to refresh the export when the date / time in ***Latest Data*** is more recent than the date / time in ***Last Generated***.

III.E.5. Export all Establishment Records

> **Note**
>
> This function is limited to the **Supervisor** account type.

STEP 1

From the top bar, click **Export.**

STEP 2

From the side bar, select **All Records.**

STEP 3

Click the **download** icon. This saves an XLSX file of the **complete list of establishments** and all their **approved records.**

Update the data for export

Refresh the export when the date / time in ***Latest Data*** is more recent than the date / time in ***Last Generated***. In this scenario, the status column will also indicate that the current exported data is **outdated**.

III.E.6. Export Customized Records List

> **Note**
>
> This function is limited to the **Supervisor** account type.

STEP 1

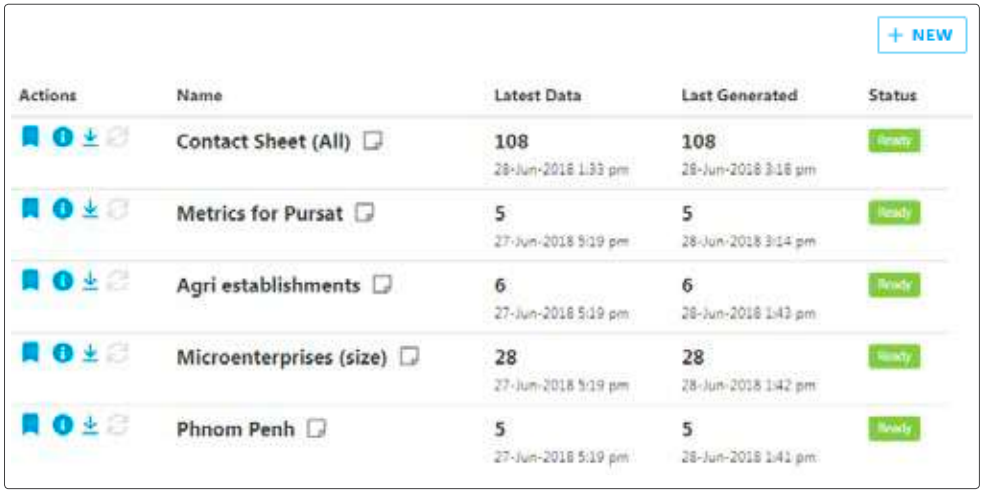
From the top bar, click **Export.**

STEP 2

From the side bar, select **Custom Exports.**

STEP 3

Click the **download** icon. This saves an XLSX file of the **customized list of establishments** in the system and their information on **selected variables.**



New custom export

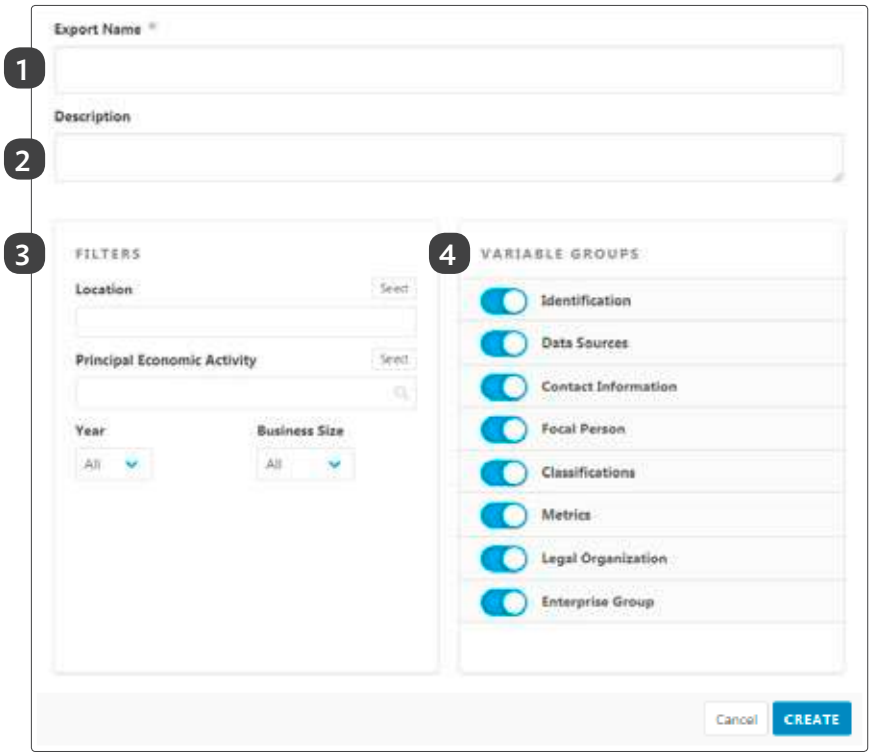
Bookmarks

Only a fixed number of custom exports can be stored by the system. When this limit is reached, the least recent custom export is automatically deleted to allow a new one. For exports that are commonly or periodically needed, bookmark them to save them from the auto-replace process.

Update the data for export

Refresh the export when the date / time in ***Latest Data*** is more recent than the date / time in ***Last Generated***. In this scenario, the status column will also indicate that the current exported data is **outdated**.

Create a new custom export



1 **Name**
Specify an easily identifiable name

2 **Description**
Include important notes and remarks about the custom export, for future reference in exporting the same data or editing the export parameters

3 **Filters**
Export a subset of the entire list by configuring the filters by location, principal economic activity, year, and / or business size

4 **Variables**
Select which variable groups are to be included in the export

III.F. Settings

III.F.1. Manage System Users

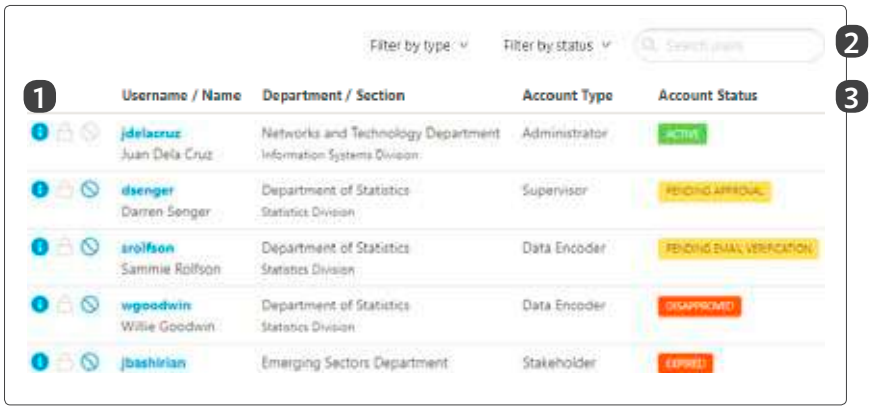
> **Note**
>
> This function is limited to the **Administrator** account type.

STEP 1

From the top bar, click **Settings.**

STEP 2

From the sidebar, select **User Accounts** (or **Pending Requests**).

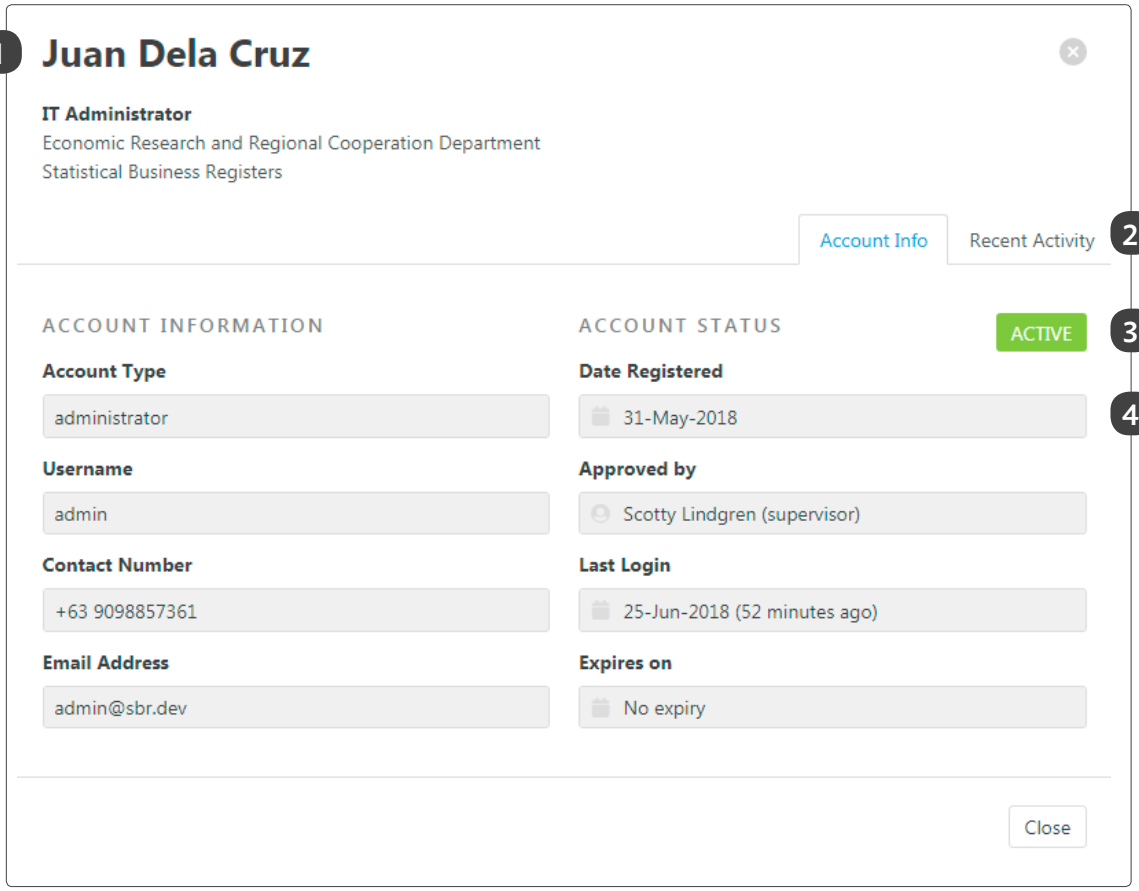


1 **Action icons**
View complete information, unlock, activate / deactivate accounts

2 **Search and filters**
Narrow down the list of users based on the account type, status, or entered name / username

3 **Basic information**
View key information such as username, name, department / section, account type, and status

Detailed view of a user account



1 **User identification**
Key information such as name, account type, department, and section

2 **Tabs**
Switch between viewing the account information, recent activity, and submissions

3 **Status**
Depending on the selected tab, different information will be displayed

4 **Information fields**
Depending on the selected tab, different information will be displayed

III.F.2. Manage Data Sources

Note

This function is limited to the **Administrator** and **Supervisor** account type.

Important

A data source must be created before any establishment record may be tagged under it (see ***Sections III.D.2. Create a new record*** and ***III.D.3. Edit an existing record***).

The system will only allow a single primary source per year.

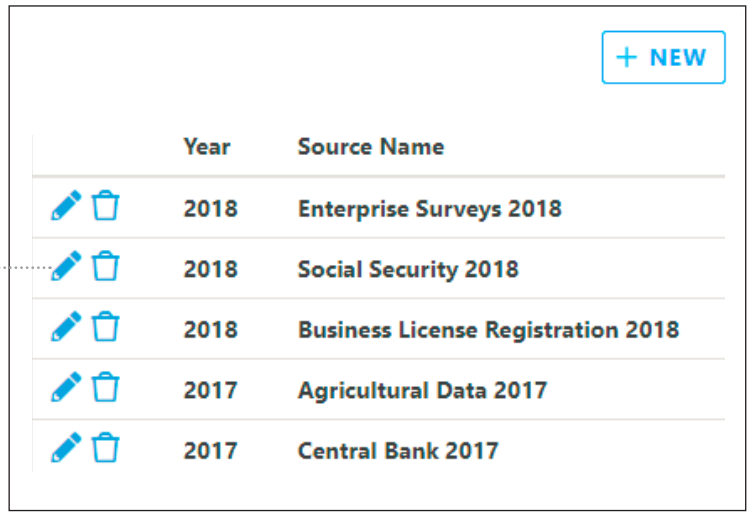
Importing multiple records from a file (see ***Section III.E.2. Import records from file***) will tag all the records to the selected primary and supplementary data sources.

STEP 1

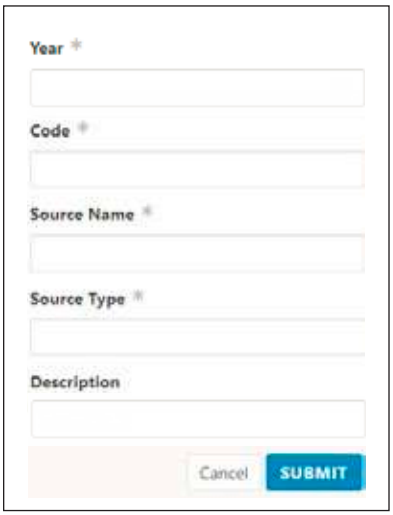
From the top bar, click **Settings**.

STEP 2

From the sidebar, select **Primary** or **Supplementary** (under **Data Sources**).



Action icons
Edit or delete data sources



ADD NEW

Enter the **year, code, name, type, and description** of the new data source

III.F.3. Manage Classifications

> **Note**
>
> This function is limited to the **Administrator** and **Supervisor** account type.

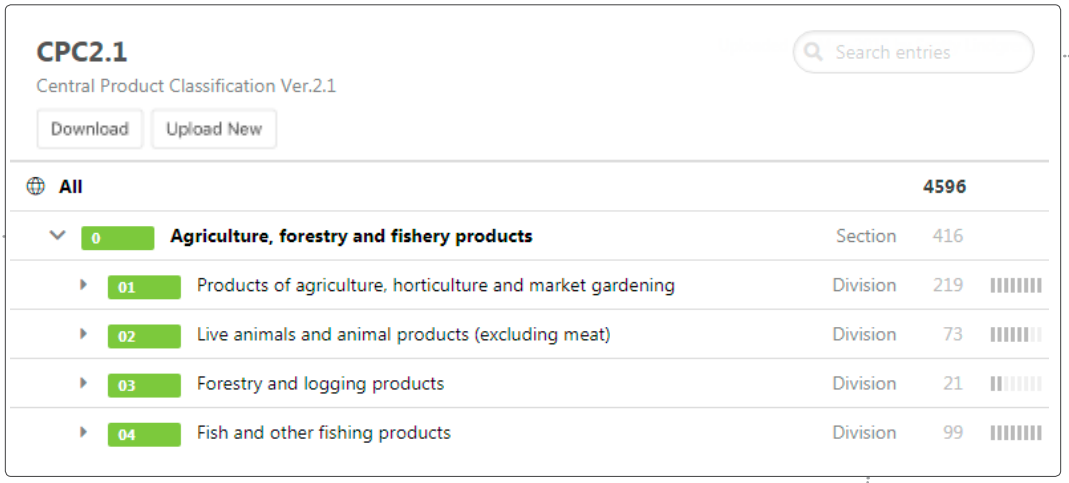
STEP 1

From the top bar, click **Settings.**

STEP 2

From the sidebar, select **Geographic, Economic Activity,** or **Product** (under **Classifications**).

Search
Look for specific entries by typing in the exact code or a partial description



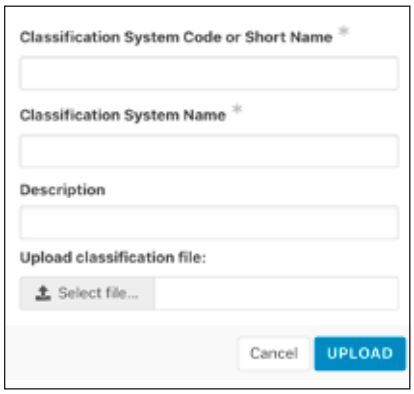
Expand and collapse
Click on an entry to show its subclassifications; click on a higher-level entry to collapse any displayed sub-classifications

Hierarchical levels
Specifies whether the entry is a section (highest level), division, group, class, or subclass (lowest level)

Update the classifications

STEP 1

Click Download to get a Microsoft Excel version of the codes and descriptions.



STEP 2

Open the file using **Microsoft Excel** and **revise the codes and descriptions that need updating.** Make no other changes to the structure and formatting to ensure compatibility for the succeeding steps. **Save the updated file.**

STEP 3

Back in the classifications page, click **Upload New**.

STEP 4

Enter the **code / short name** and **complete name** of the new classification, and **select the file to be uploaded.**

STEP 5

Click **Upload**.

Compatibility rules for the file:

- Must be in XLSX format
- Data must begin on row 2
- Columns A-H: The first eight columns must contain higher-level classification codes, and the code of the current item
- Column I: Description of current item name in local (primary) language / character set
- Column J: Description of current item name in secondary language / character set
- Column K (optional): Longitude in decimal format
- Column L (optional): Latitude in decimal format
- Column M (optional): Remarks

III.F.4. Manage Announcements

> **Note**
>
> This function is limited to the **Administrator** and **Supervisor** account type.

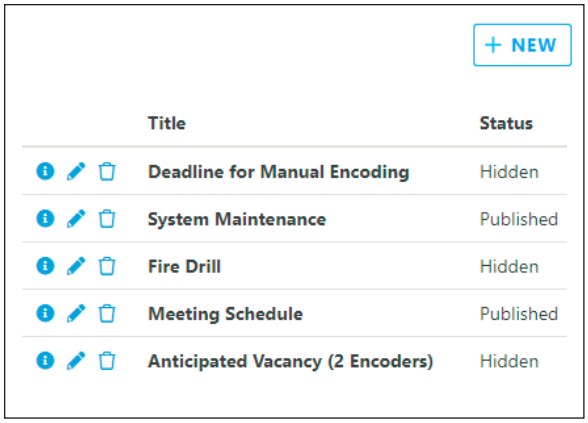
STEP 1

From the top bar, click **Settings**.

STEP 2

From the sidebar, select **Announcements**.

Action icons
View, edit, or delete



ADD NEW

Enter the **title, content, and status** (published or hidden) of the new announcement

Setting the status to **Published** allows all users logged into the system to view the announcement from the home page.

III.F.5. Manage Files

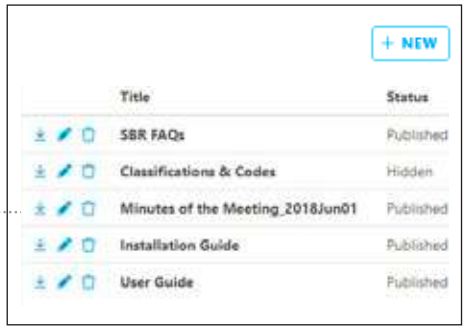
> **Note**
>
> This function is limited to the **Administrator** and **Supervisor** account type.

STEP 1

From the top bar, click **Settings**.

STEP 2

From the sidebar, select **Files**.

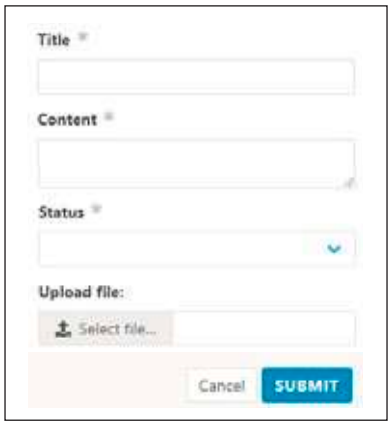


Action icons
Download, edit, or delete

ADD NEW

Enter the **title, content, and status** (published or hidden) of the new file, and **select the file** to be added

Setting the status to **Published** allows all users logged into the system to download the file from the home page



III.F.6. Manage Links

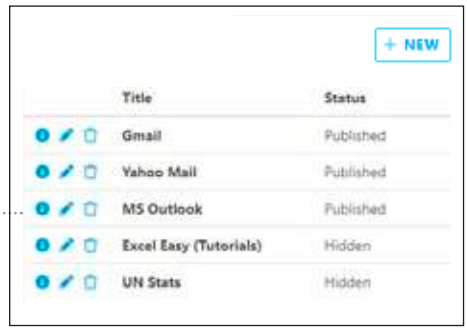
> **Note**
>
> This function is limited to the **Administrator** and **Supervisor** account type.

STEP 1

From the top bar, click **Settings**.

STEP 2

From the sidebar, select **Links**.



Action icons
View, edit, or delete

ADD NEW

Enter the **title, content (URL), and status** (published or hidden) of the new link

Setting the status to **Published** allows all users logged into the system to view and click the link from the home page

III.F.7. View Audit Log

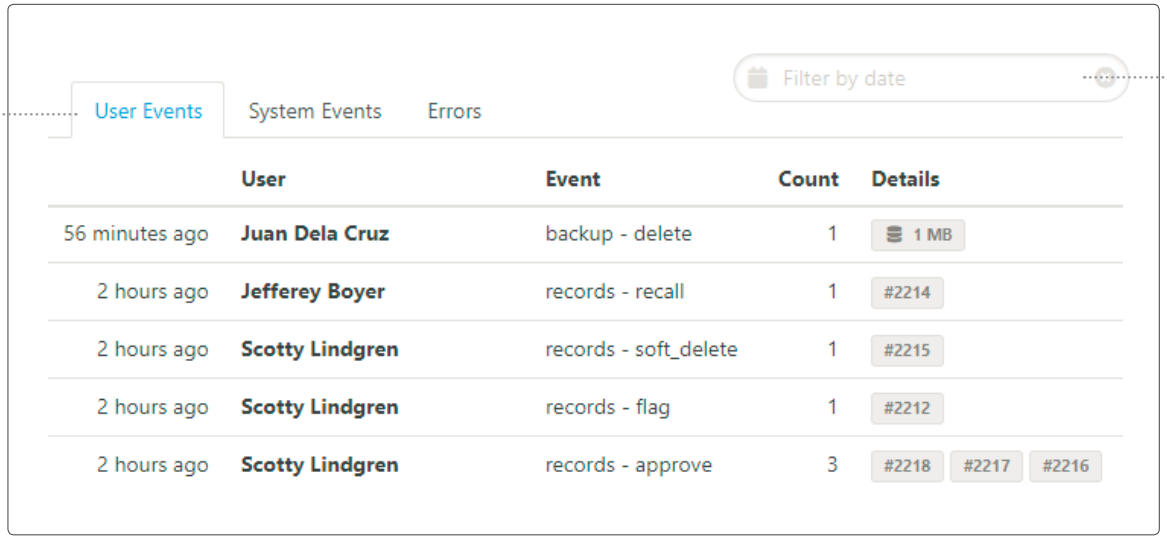
> **Note**
>
> This function is limited to the **Administrator** and **Supervisor** account type.

STEP 1

From the top bar, click **Settings.**

STEP 2

From the sidebar, select **Audit Log.**



Switch tabs
View activities categorized as User Events, System Events, and Errors

Filter by date
Select date to narrow down activity list

III.F.8. Create a Backup of the Database

> **Note**
>
> This function is limited to the **Administrator** and **Supervisor** account type.

STEP 1

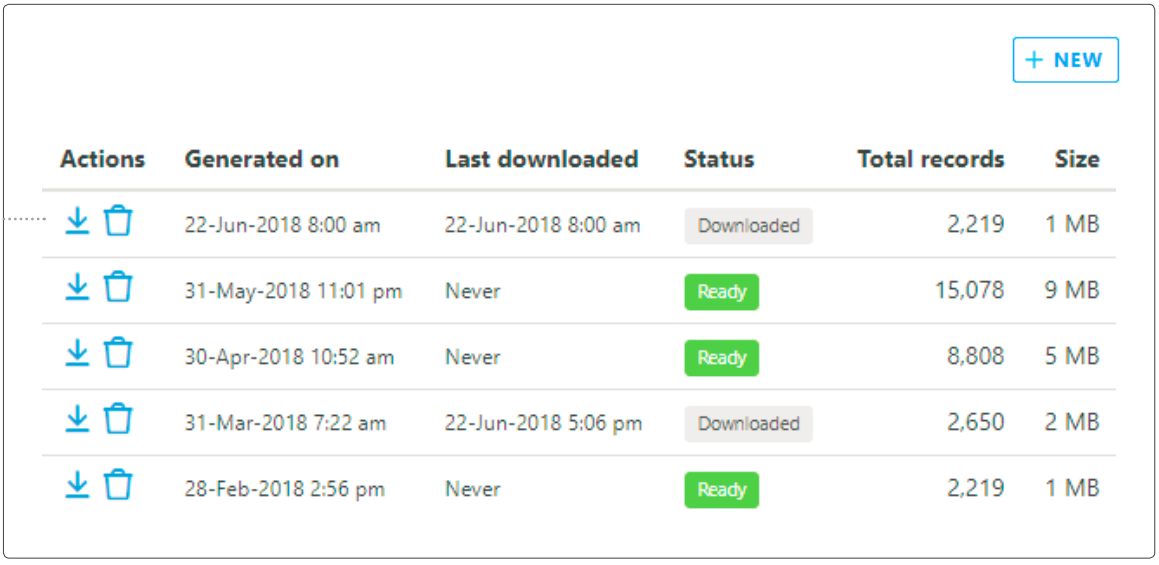
From the top bar, click **Settings.**

STEP 2

From the sidebar, select **Database Backup.**

STEP 3

At the top right portion of the screen, click **New.** Once completed, the backup will be included in the list of recent backups that will be available for download.

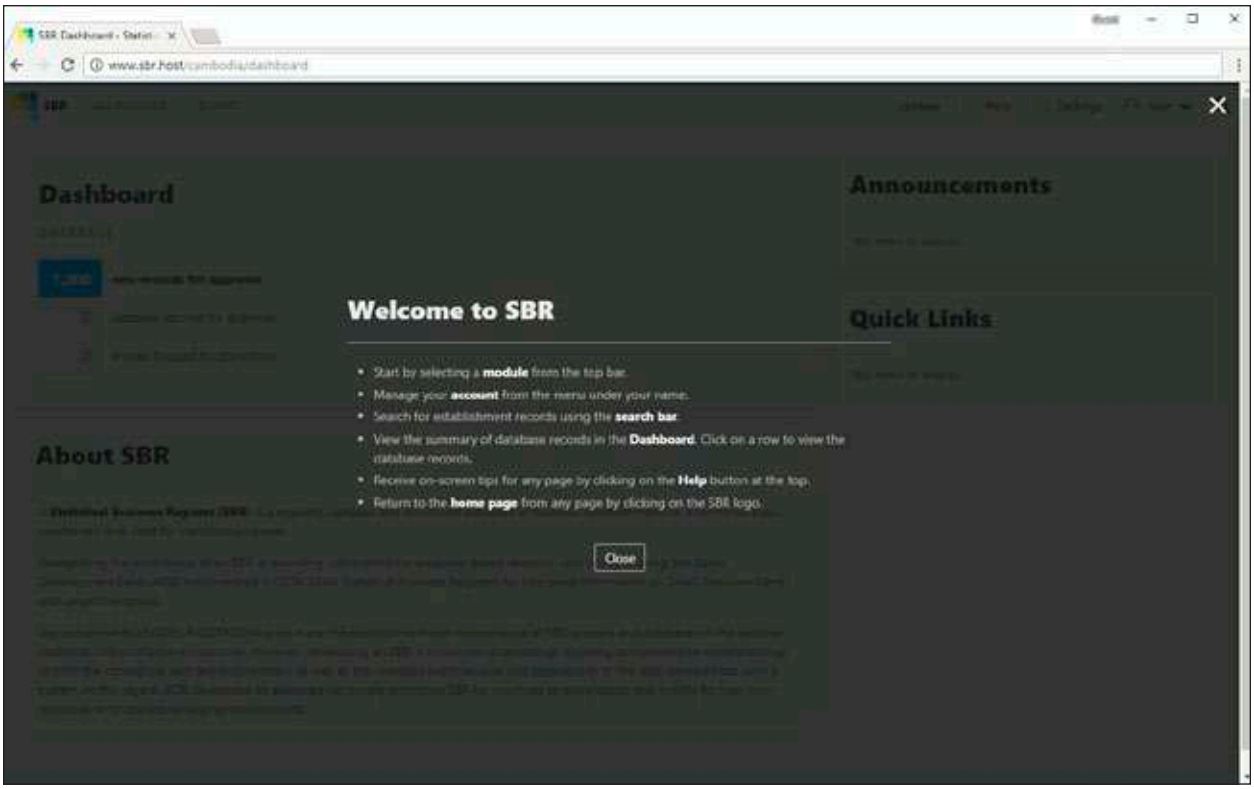


Previous backups
Download or delete previously created backups of the database

III.G. Support

III.G.1. View Help Guides

From the top bar, select **Help** to open the help guide. The help text varies per page.



III.G.2. Contact Support

STEP 1

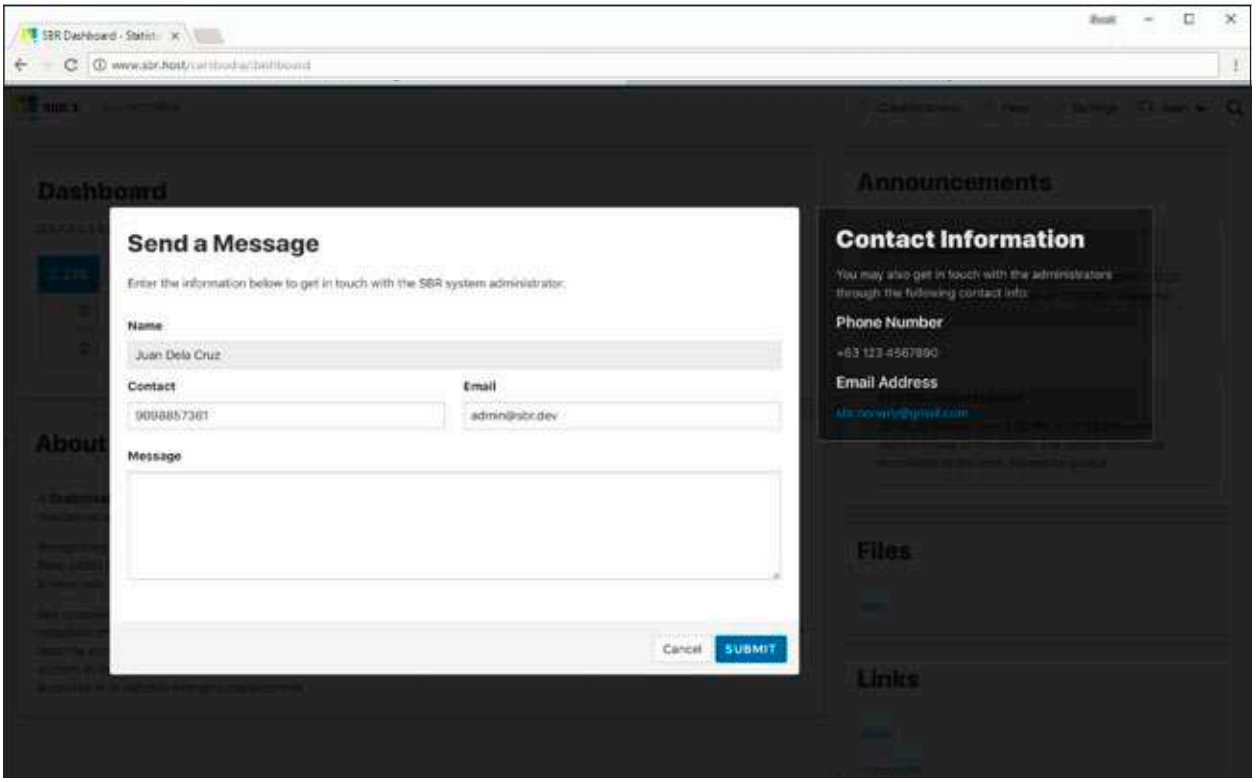
From any page in the system, scroll to the bottom of the screen and click **Contact support**.

STEP 2

Enter the details of the message to be sent to the system administrator. Click **Submit**.

STEP 3

Alternatively, the user may contact the system administrator through the information provided on the right portion of the screen.


Dashboard
Announcements
Send a Message
Enter the information below to get in touch with the SBR system administrator.
Name
Juan Dela Cruz
Contact
9998857361
Email
admin@sbr.dev
Message
Cancel
SUBMIT
Contact Information
You may also get in touch with the administrators through the following contact info:
Phone Number
+63 123 4567890
Email Address
About
Files
Links

User Guide for ADB Statistical Business Register

This user guide provides step-by-step instructions on how to set up, use, and maintain the statistical business register of the Asian Development Bank (ADB). It also outlines the key technical considerations that guided ADB's development of the software. This publication serves as a reference for countries looking to establish statistical business registers of their own using the software solution developed as part of this initiative. Statistical business registers are structured databases that provide information on business establishments and their activities, enabling a country's national statistical system to produce economic statistics and indicators. This helps governments design and implement economic policies that are data-driven.

About the Asian Development Bank

ADB is committed to achieving a prosperous, inclusive, resilient, and sustainable Asia and the Pacific, while sustaining its efforts to eradicate extreme poverty. Established in 1966, it is owned by 67 members—48 from the region. Its main instruments for helping its developing member countries are policy dialogue, loans, equity investments, guarantees, grants, and technical assistance.

ADB

ASIAN DEVELOPMENT BANK
6 ADB Avenue, Mandaluyong City
1550 Metro Manila, Philippines
www.adb.org