FISCAL EDGE DESKTOP CLIENT SETUP GUIDE FOR SAGE EVOLUTION

JUNE 23, 2024

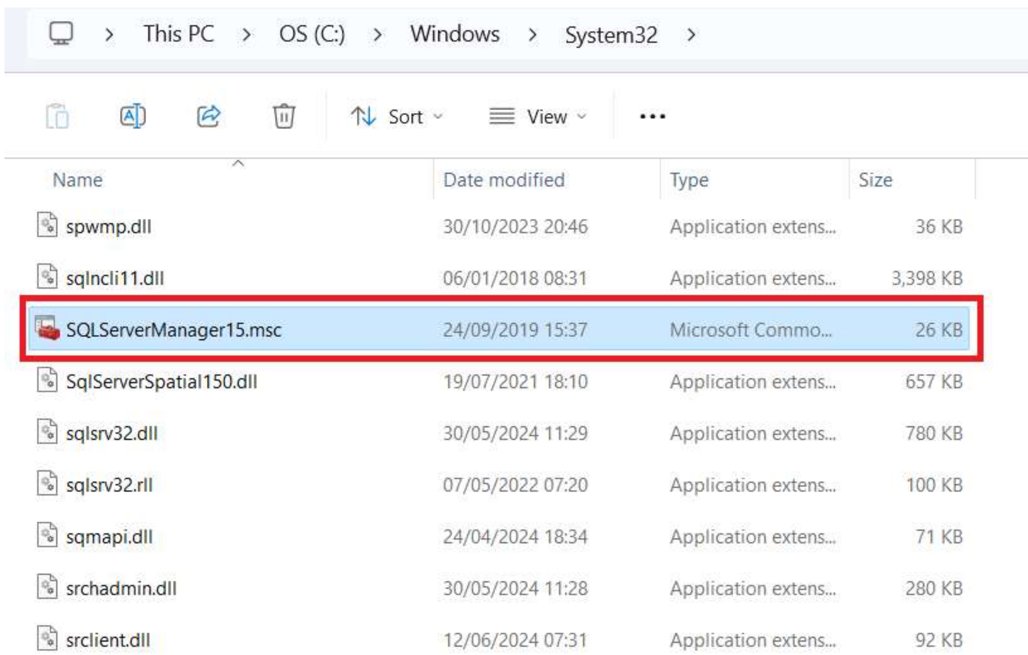
FISCAL EDGE SOLUTIONS LIMITED
Lusaka, Zambia

Contents

Prerequisite: Ensure TCP Access to SQL Server Database

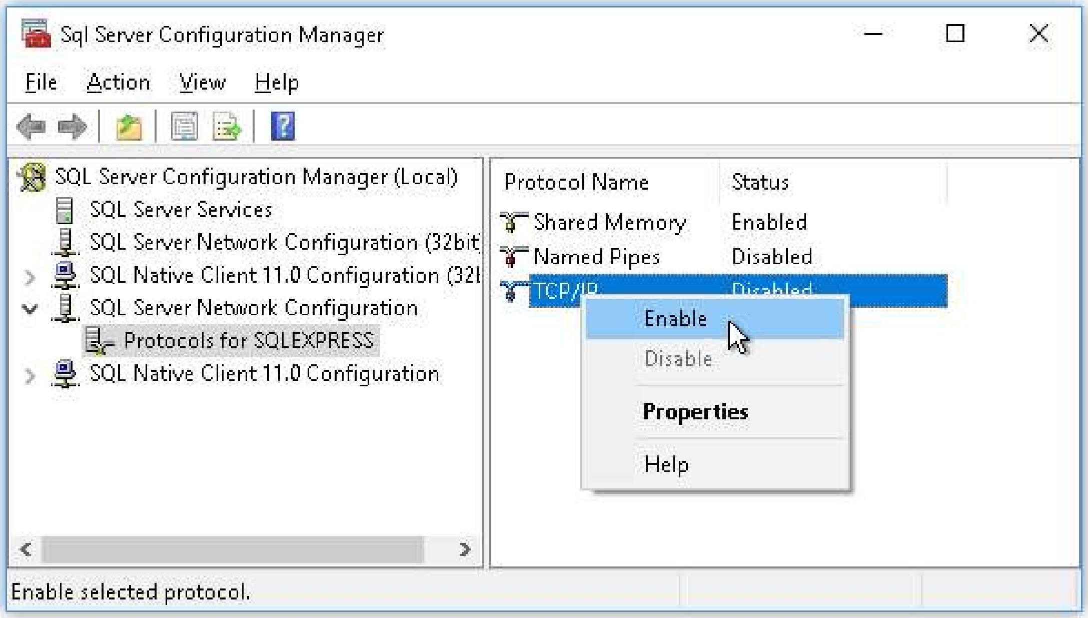
The Fiscal Edge Desktop Client application requires access to the SQL Server hosting the Sage database (or pervasive database for pastel partner). To enable TCP protocol for the database, follow these steps:

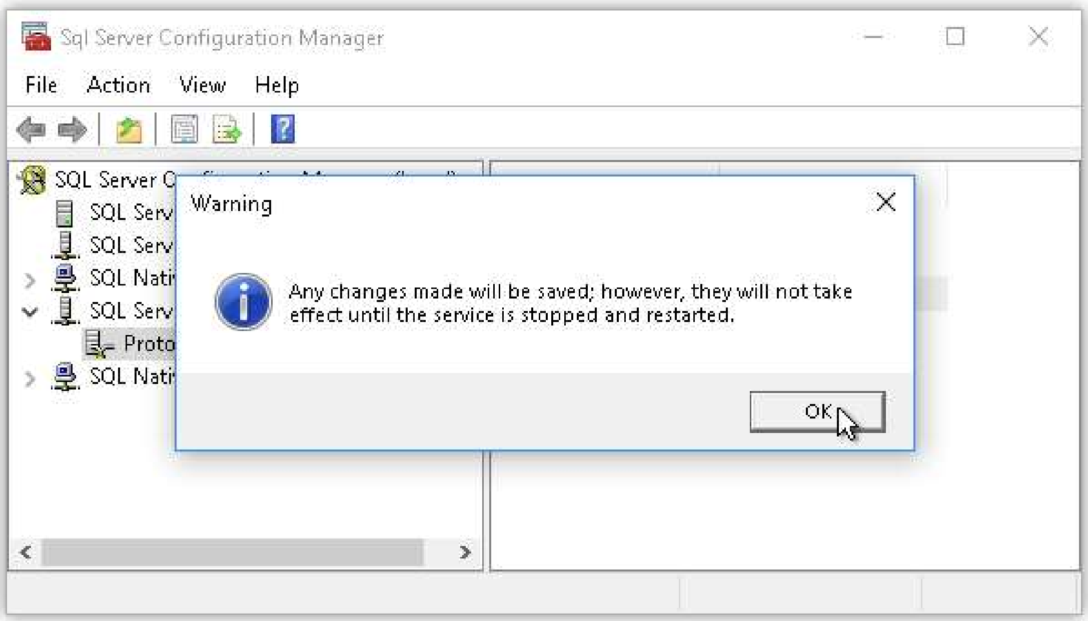
1. Open SQL Configuration Manager:
    - If you cannot locate SQL Configuration Manager in your Start menu, you can find it in the following directory: **C:\Windows\System32**.
    - Look for **SQLServerManager15.msc** and open it.



2. Expand the **SQL Server Network Configuration**:

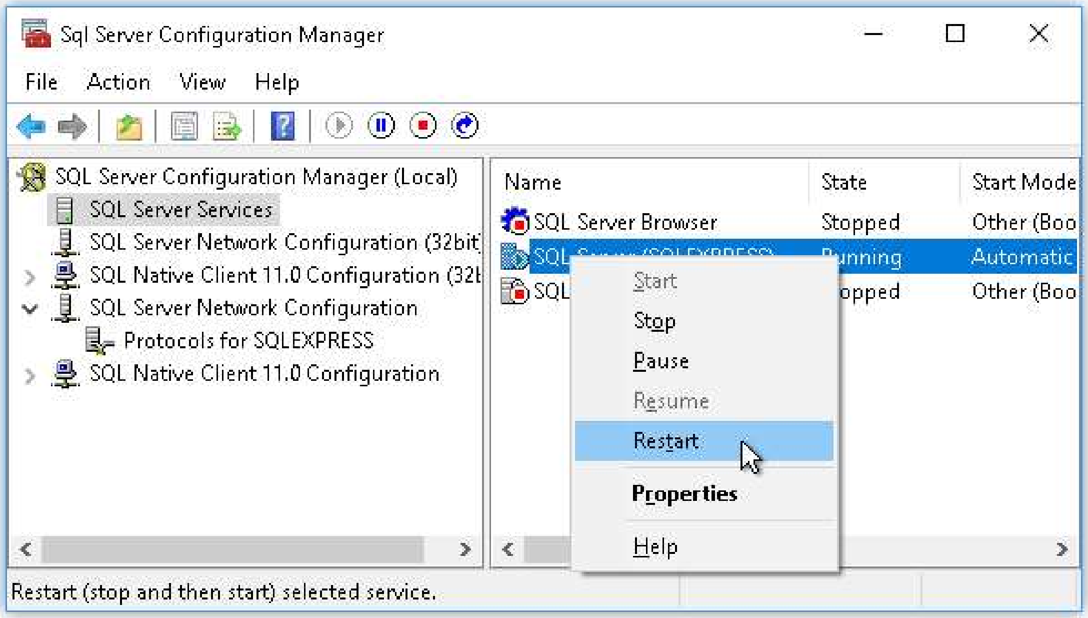
    - In the SQL Configuration Manager, expand the **SQL Server Network Configuration** heading.
    - Click on **Protocols for SQLEXPRESS** (or the name of your SQL Server instance). Note that if you have multiple instances of SQL Server, they will all appear here. Choose the server hosting your Sage database.
    - Ensure the **TCP/IP** protocol is enabled.





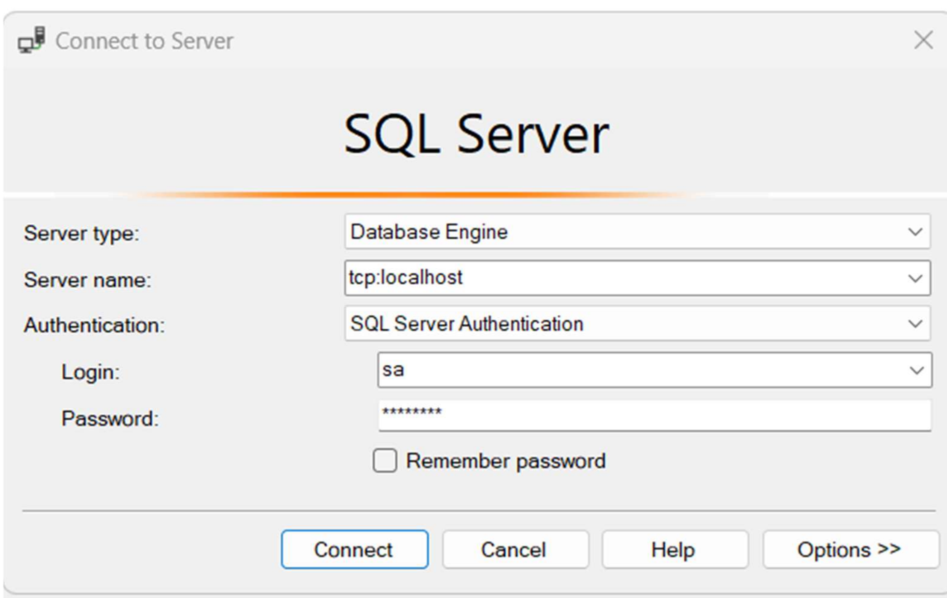
3. Click on **SQL Server Services**:

    - In the SQL Configuration Manager, click on **SQL Server Services**.
    - This will display a list of SQL Server instances in the right-hand pane. Verify that the **SQLEXPRESS** instance (or the instance running your Sage database) is **Running**.
    - If you are running multiple instances, they will all appear here. Right-click on the instance running your Sage database and select **Restart**.



4. **Verify TCP Connection to the Server**:

- Open **Microsoft SQL Server Management Studio** (SSMS).
- Attempt to connect to the server using the following format:
    - For the default server instance: tcp:localhost
    - For a named server instance: tcp:localhost\{YOUR_SERVER_NAME}
- Ensure that the connection is successful, confirming that TCP access is properly configured.



Execute SQL Scripts

Please download the script file from here

FiscalInfo Table

The FiscalInfo table serves to keep track of invoice fiscalization.

ZraInvoice View

The ZraInvoice view is specifically designed to extract invoices for the purpose of fiscalization. Ensuring the integrity of the data presented by this view is of utmost importance. It is essential to note that Sage stores both invoices and purchase orders within the same table. Therefore, meticulous care must be taken to filter out purchase orders, ensuring that only invoice data is extracted.

Required UDFs

Object	UDF Name	Type	Values	Force	Default
Sales Invoice & Sales Order	DestinationCountryCode	Lookup	ZM;ZA;	No	ZM
Sales Invoice & Sales Order	ZRALOCALPURCHASEORDER	String		No	

Key Fields to Note:

1. **DestinationCountryCode:** This field indicates the destination country code for a sold product in case of an export. For instance, "ZW" represents Zimbabwe. To implement this, you will need to create a User-Defined Field (UDF) where users can input this information. Would be best to enquire from the client the countries where they normally export to and make sure include those countries in the lookup options.
2. **CurrencyType:** Since different clients may have varied default currency configurations, it's crucial to pay attention to this field and ensure alignment with each client's specific configuration.
3. **LocalPurchaseOrder:** For invoices containing the tax code C2, it is necessary to provide an additional field. This field should be stored in a User-Defined Field (UDF) that can be created either on the invoice or the customer profile. The script must be adjusted accordingly to extract and utilize this field.
4. **BranchId:** This field is typically set to 000 for companies with a single branch. However, for companies with multiple branches and multiple databases, each database must be registered with the ZRA (Zambia Revenue Authority) and a corresponding BranchId must be obtained and used in the script.

ZraInvoiceItem View

The application depends on his view to extract corresponding line items for invoices.

Key Fields to Note:

1. **ItemClassificationCode:** This field refers to standard UN codes that must be provided for each item in the inventory. It should be added to the inventory through a User-Defined Field (UDF) or by utilizing any existing unused fields on the item creation page. The script must be adjusted to use this field. Note that this is a mandatory field, and a null value will lead to a validation error.
2. **PackagingUnitCode:** These are standard codes that need to be provided for each item in the inventory. This field should be added using a UDF or any existing unused fields on the item creation page. The script must be updated accordingly to use this value.
3. **QuantityUnitCode:** These are standard codes that need to be provided for each item in the inventory. This field should be added using a UDF or any existing unused fields on the item creation page. The script must be updated accordingly to use this value.
4. **TaxLabel:** Great care should be taken to ensure that the tax labels match the configuration on Sage. Any mismatch here would lead to invoices being fiscalized with the wrong tax code, which could have far-reaching implications for the business.

ZraStockMaster View

This view is used to extract stock items for sync with the ZRA system.

Required UDFs

Object	UDF Name	Type	Values	Force	Default
Inventory	UNSPSC	String		Yes	
Inventory	PackagingUnitCode	Lookup	See docs	Yes	
Inventory	QuantityUnitCode	Lookup	See docs	Yes	
Inventory	OriginCountryCode	Lookup	See docs	Yes	

Key Fields to Note:

1. **UNSPSC**: Taxpayers must assign a UNSPSC code from the ZRA published list to each item in their inventory.
2. **OriginCountryCode**: This field indicates the ISO code for the country where the item is generally sourced. If the item is sourced from multiple countries, any one of those countries' codes can be assigned.
3. **PackagingUnitCode**: The value for this field must be selected from the ZRA published list.
4. **QuantityUnitCode**: The value for this field must be selected from the ZRA published list.

PurchaseInfo Table

This table is used to keep track of purchases invoices that has been transmitted to the ZRA.

ZraPurchase View

The Zrapurchase view is specifically designed to extract purchase invoices. Ensuring the integrity of the data presented by this view is of utmost importance.

Required UDFs

Object	UDF Name	Type	Values	Force	Default
GRV & Purchase Order	Origin	Lookup	local;foreign	Yes	

Key Fields to Note:

1. **Origin:** This field is used to indicate whether a purchase was local or foreign.
2. **CurrencyType:** Since different clients may have varied default currency configurations, it's crucial to pay attention to this field and ensure alignment with each client's specific configuration.
3. **BranchId:** This field is typically set to 000 for companies with a single branch. However, for companies with multiple branches and multiple databases, each database must be registered with the ZRA (Zambia Revenue Authority) and a corresponding BranchId must be obtained and used in the script.

ZraPurchaseItem View

This is used to extract purchase invoice line items.

Where to find Fiscal Details

The fiscal detail is written back to the **InvNum** table in the below columns:

Signature = cDPOrdServiceTaskNo

InternalData = cDSOrdServiceTaskNo

ZRA Invoice Number = cDSMExtOrderNum

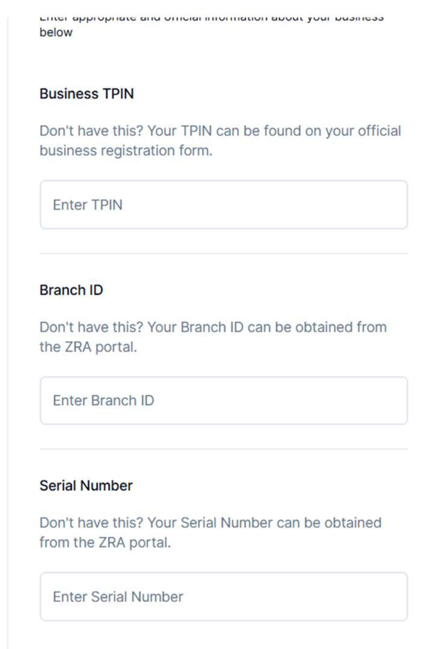
Trust your eyes

Before creating any of the views mentioned above, it is highly recommended to first execute the SELECT part of the query and carefully examine the accuracy of the output results. Here's why:

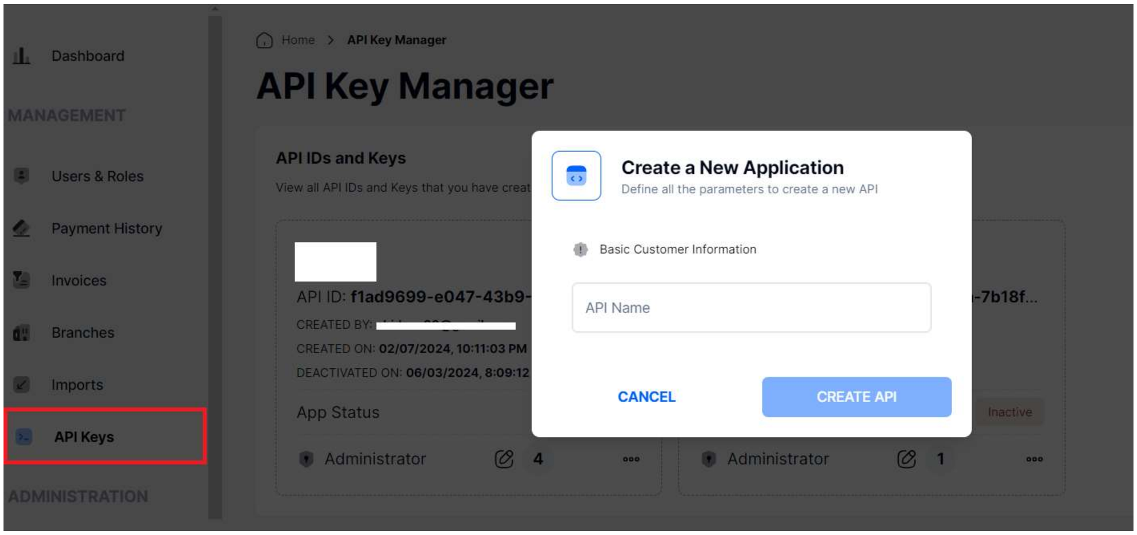
1. **Accuracy Verification**: Ensure that the data retrieved is correct and meets the expected criteria. This step helps in identifying any discrepancies or errors in the query logic.
2. **Critical Dependence**: The application relies on these views for fiscalization. Any inaccuracies in the data presented by these views will be carried forward and could lead to fiscalization issues.
3. **Prevent Potential Issues**: Inaccurate data used for fiscalization can have significant negative implications. By verifying the output before creating the views, you minimize the risk of such issues.

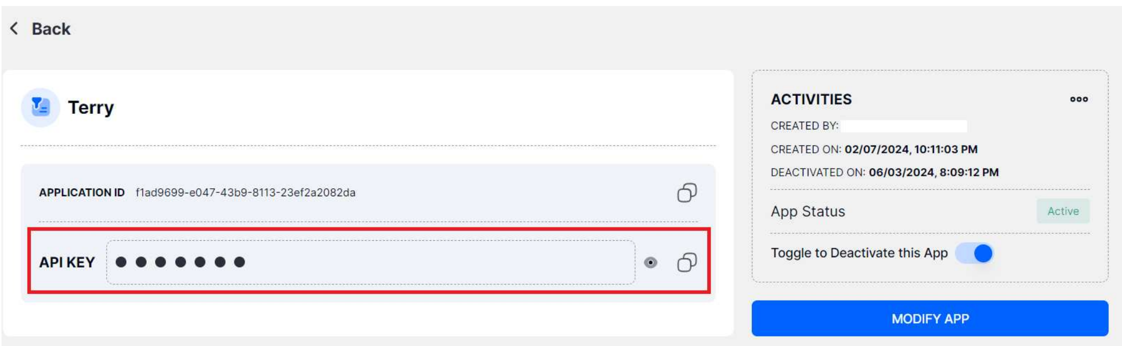
Register on Fiscal Edge Platform

This step requires that the business is already registered on the ZRA's Smart Invoicing Portal and has obtained the device serial number and branch code. These details are necessary to complete the process.



Once the registration process is completed, click on the API Keys tab and create an API Key


Dashboard
MANAGEMENT
Users & Roles
Payment History
Invoices
Branches
Imports
API Keys
ADMINISTRATION
Home > API Key Manager
API Key Manager
API IDs and Keys
View all API IDs and Keys that you have creat
API ID: f1ad9699-e047-43b9-
CREATED ON: 02/07/2024, 10:11:03 PM
DEACTIVATED ON: 06/03/2024, 8:09:12
App Status
Administrator
4
-7b18f...
Inactive
Administrator
1
Create a New Application
Define all the parameters to create a new API
Basic Customer Information
API Name
CANCEL
CREATE API


Back
Terry
APPLICATION ID f1ad9699-e047-43b9-8113-23ef2a2082da
API KEY
ACTIVITIES
CREATED BY:
CREATED ON: 02/07/2024, 10:11:03 PM
DEACTIVATED ON: 06/03/2024, 8:09:12 PM
App Status
Active
Toggle to Deactivate this App
MODIFY APP

Install Fiscal Edge Desktop Client

The Fiscal Edge Desktop Client application is a 32-bit .NET-based software. To function properly, it requires both the .NET 8 runtime and ASP.NET Core runtime. These runtimes are usually bundled with the application. While most Windows computers come with preinstalled .NET runtimes, we recommend installing the versions bundled with the application to avoid compatibility issues.

Installation Steps:

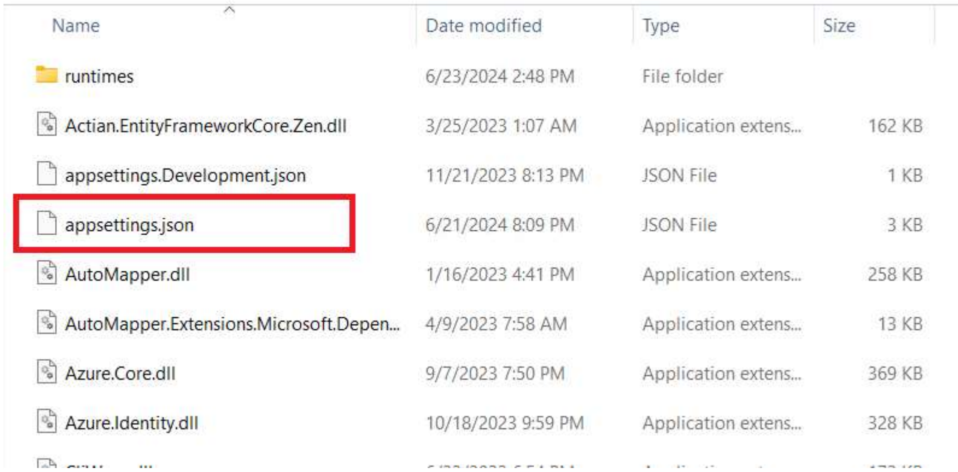
1. **Install the .NET Runtimes:**
   - Ensure the .NET 8 runtime and ASP.NET Core runtime are installed. Use the versions provided with the application package to prevent compatibility problems.
2. **Install the Fiscal Edge Client:**
   - Proceed with the installation of the Fiscal Edge Client application following the on-screen instructions.
3. **Update the Configuration File:**
   - Upon successful installation, navigate to the installation directory to update the configuration file.
   - If you selected the default location during installation, the file can be found at:

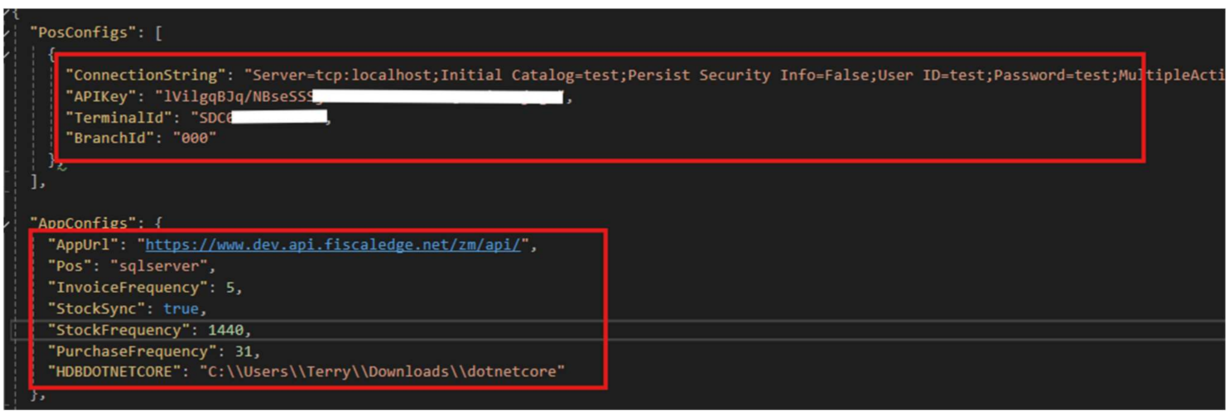
     C:\Program Files (x86)\Fiscal Edge Solutions\Fiscal Edge Client

   - If a custom location was chosen, refer to that directory for the configuration file.

Update Configuration File

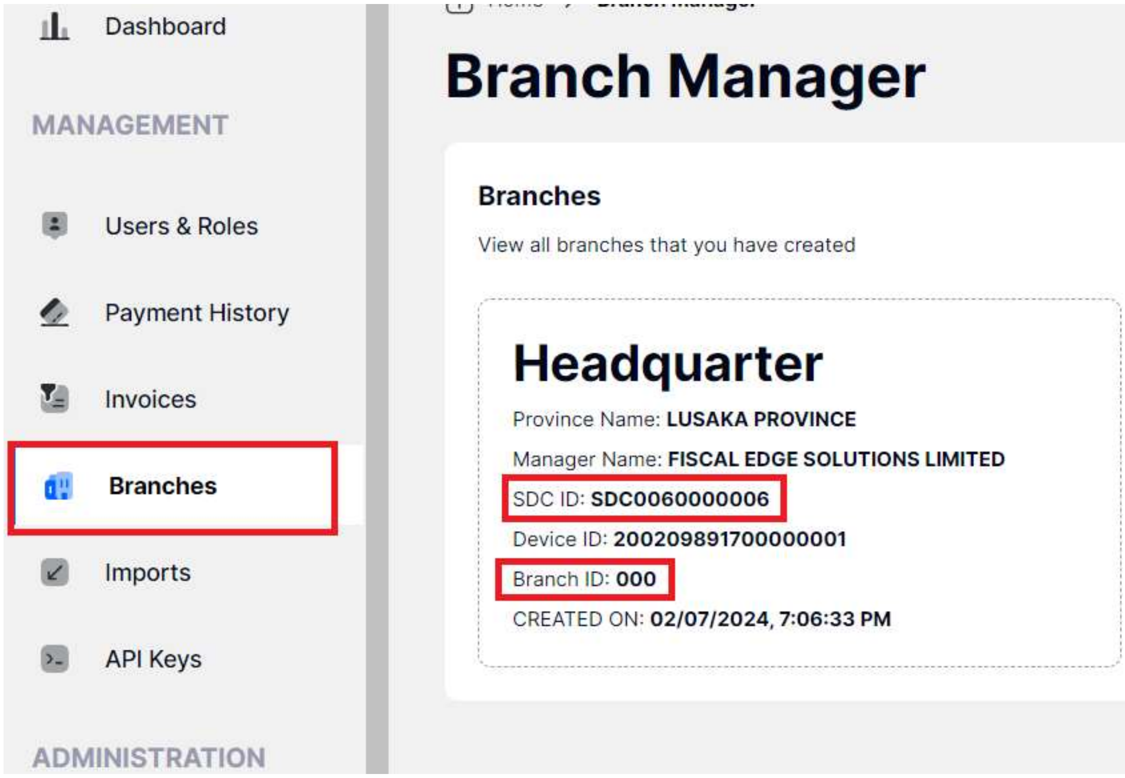
Open the *appsettings.json* file with your preferred text editor.



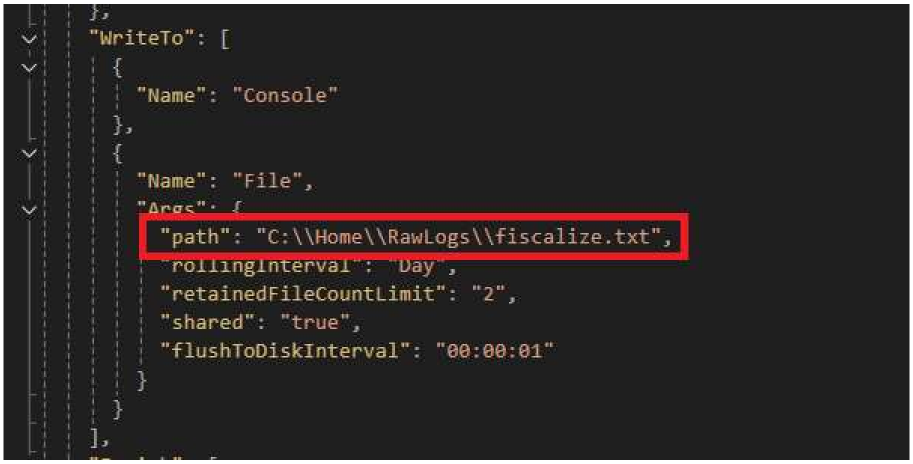


Configuration Parameters:

- **APIKey:** Use the API Key created in the API Key step above.
- **TerminalId:** Obtain this from the Fiscal Edge platform under the Branch tab.
- **BranchId:** Obtain this from the Fiscal Edge platform under the Branch tab.



- **ConnectionString:**
  - This refers to the connection string for the SQL Server database.
  - If you are unfamiliar with SQL Server connection strings, please refer to ConnectionStrings.com.
- **AppUrl:**
  - For testing, use: https://www.dev.api.fiscaledge.net/zm/api/
  - For production, use: https://www.api.fiscaledge.net/zm/api/
- **Pos:**
  - Use pastel for Pastel Partner.
  - Use sqlserver for other Sage products.
- **InvoiceFrequency:**
  - This refers to the time interval in seconds for the fiscalization service to run.
  - Recommended values are 3, 4, or 5 seconds.
- **StockFrequency:**
  - This refers to the time interval in minutes for the stock service to run.
  - Leave this at 1440 minutes.
- **PurchaseFrequency:**
  - This refers to the time interval in minutes for the purchase service to run.
  - Leave this at 31 minutes.
- **Path:**This specifies the logging location for the application.



The Log is your friend

Save the Configuration File:

- After updating the appsettings.json file with the necessary parameters, save the file.

Stop the Service:

- Locate the Fiscal Edge service in the Windows Services list.
- Stop the service.

Clear Previous Logs:

- Navigate to the log location specified in the configuration file.
- Delete the existing log file to ensure you have a clean log to work with.

Start the Service:

- Restart the Fiscal Edge service.

Check the New Log:

- After the service starts, a new log file will be created.
- Review this log for any errors. If the application is correctly configured, there should be no errors in the log.

Troubleshoot Errors:

- If you encounter an error, do not panic. Carefully review the error message; it usually contains details about what might have gone wrong.
- Common issues include misconfigured connection strings or missing fields in the SQL views.
- Once you identify the cause of the error, correct it and restart the service.

Alternative Approach:

- If errors persist, you can stop the service again, delete the log file, and restart the service. This ensures you are reviewing a newly generated log that reflects changes made after corrections.