

Virtual Device Documentation



Mobility. Agility. Security.

v1.0.0

Overview

VirtualDeviceLib is a C# library (DLL) used to interact with a Fiscal Device Management System (FDMS). This library provides several public methods to fetch device configurations, check device status, manage fiscal days, and process Fiscal Documents. This documentation offers basic examples of how to load the library in a C# application, call its methods, and handle possible error scenarios.

Referencing

To include VirtualDeviceLib in your project, reference the DLL in your Visual Studio project.

1. Right-click on the **References** section in your project.
2. Click on **Add Reference**.
3. Browse to the location (preferably your application path) of the VirtualDeviceLib DLL and add it.

Usage Example

1. Loading and Using the Library

```
using VirtualDeviceLib;
using Newtonsoft.Json.Linq;

namespace TestVirtualDeviceApp
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create an instance of the library
            var virtualDevice = new VirtualDevice();
            // Example: Fetch Device Configurations
            string configResult = virtualDevice.GetConfig();
            Console.WriteLine($"Device Config Result: {configResult}");
            // Example: Check Device Status
            string statusResult = virtualDevice.GetStatus();
            Console.WriteLine($"Device Status Result: {statusResult}");
        }
    }
}
```

2. Standard Responses

Error Response

The Error response is indicated by **Code 0** and a **Message** indicating the error message.

```
{  
  "Code": "0",  
  "Message": "Error Message"  
}
```

Success Response

The success is indicated by **Code 1**, a **Message** indicating the error message, and the **Data** section which might be Device information (for GetStatus), Device Status Data (for GetStatus), Z Report Data (for OpenFiscalDay and CloseFiscalDay), Receipt Data (for SubmitReceipt)

```
{  
  "Code": "1",  
  "Message": "Success Message",  
  "QRCode": "",  
  "FDMSInvoiceNo": "",  
  "FiscalDayNo": "",  
  "VerificationCode": "",  
  "VerificationLink": "https://fdmstest.zimra.co.zw",  
  "DeviceID": "18489",  
  "Data": { ... }  
}
```

3. Public Methods

GetConfig()

- Fetches the device configurations from the FDMS server.
- **Returns:** Success message with configuration details or an error message.

Example:

```
string config = virtualDevice.GetConfig();
```

GetStatus()

- Retrieves the current status of the device from the FDMS server.
- **Returns:** Success message with status details or an error message.

Example:

```
string status = virtualDevice.GetStatus();
```

OpenFiscalDay()

- Opens a new fiscal day on the FDMS server, provided the last fiscal day is closed.
- **Returns:** Success message with the Z Report data for the previous fiscal day or an error message

Example:

```
string openFiscalDayResult = virtualDevice.OpenFiscalDay();
```

CloseFiscalDay()

- Closes the current fiscal day on the FDMS server.
- **Returns:** Success message with the Z Report data for the current fiscal day or an error message.

Example:

```
string closeFiscalDayResult = virtualDevice.CloseFiscalDay();
```

```
SubmitReceipt(string receiptType, string receiptCurrency, string invoiceNo, string referenceNumber, decimal invoiceAmount, decimal invoiceTaxAmount, string receiptNotes, string receiptLines, bool receiptLinesTaxInclusive, string moneyTypeCode, string receiptPrintForm, string buyerRegisterName, string buyerTradeName, string vatNumber, string buyerTIN, string buyerPhoneNo, string buyerEmail, string buyerProvince, string buyerStreet, string buyerHouseNo, string buyerCity)
```

- Submit a Fiscal Invoice, Credit Note, or Debit Note to the FDMS server.
- **Returns:** Success message with the QR Code and Receipt Information as the Data value or an Error Message.

Example:

```
var myReceiptLines = new List<receiptLine>();

myReceiptLines.Add(new receiptLine()
{
    receiptLineType = "Sale", //Optional
    receiptLineNo = 1, //Optional
    receiptLineHSCode = "", //Optional
    receiptLineName = "Test Item 1",
    receiptLinePrice = 100.00M,
    receiptLineQuantity = 1,
    receiptLineTotal = 100.00M,
    taxCode = "B", //NonVatable item [B] Vatable Item [A]
    taxPercent = 0M, //NonVatable item [0] Vatable Item [15]
});
myReceiptLines.Add(new receiptLine()
{
    receiptLineType = "Sale", //Optional
    receiptLineNo = 3, //Optional
    receiptLineHSCode = "", //Optional
    receiptLineName = "Test Item 2",
    receiptLinePrice = 100.00M,
    receiptLineQuantity = 1,
    receiptLineTotal = 100.00M,
    taxCode = "A", //NonVatable item [B] Vatable Item [A]
    taxPercent = 15.00M, //NonVatable item [0] Vatable Item [15]
});
string receiptLines = JsonConvert.SerializeObject(myReceiptLines);
```

```
var submitReceipt = virtualDevice.SubmitReceipt("FiscalInvoice",
"USD", "INV001", "REF000", 110.00M, 13.04M, "Test Invoice", receiptLines,
true, "CASH", "InvoiceA4", "Axis Solutions Pvt Ltd", "Axis Solutions",
"123456789", "1234567890", "0770000000", "developers@axissol.com",
"Harare", "Bargate Road, Vainona", "60", "Harare");
```



```
//ReceiptLines Class
public class receiptLine
{
    public string receiptLineName { get; set; }
    public int receiptLineNo { get; set; }
    public decimal receiptLineQuantity { get; set; }
    public string receiptLineType { get; set; }
    public decimal receiptLineTotal { get; set; }
    public int taxID { get; set; }
    public string receiptLineHSCode { get; set; }
    public decimal receiptLinePrice { get; set; }
    public string taxCode { get; set; }
    public decimal taxPercent { get; set; }
}
```

GetLicense()

- Gets the device license status.
- **Returns:** Success message with the license details or an error message

Example:

```
string getLicenseResult = virtualDevice.GetLicense();
```

4. Error Guide

1. GetConfig()

Common Error Messages:

- **Error:** "Failed to get Device Configurations"
 - **Description:** Could not fetch the device configurations from the server.
 - **Possible Fixes:**
 - Ensure the device is connected to the FDMS server.
 - Check network connectivity between the device and server.
 - Verify that the correct device ID is being used.

2. GetStatus()

Common Error Messages:

- **Error:** "Failed to get Device Status"
 - **Description:** Could not fetch the current device status from the FDMS server.
 - **Possible Fixes:**
 - Ensure the device is properly connected to the network.
 - Check if the FDMS server is reachable.
 - Verify if the device configuration is correct.

3. OpenFiscalDay()

Common Error Messages:

- **Error:** "Failed to open a new Fiscal Day"
 - **Description:** A fiscal day is already open, and the system is trying to open another one.
 - **Possible Fixes:**
 - Ensure the previous fiscal day is closed before attempting to open a new one.
 - Check the status of the device to verify if the day is open or closed.
- **Error:** "Failed to get Fiscal Day Status"
 - **Description:** Unable to retrieve the current fiscal day status from the server.
 - **Possible Fixes:**
 - Ensure the server is reachable and the device is connected.
 - Review the network configuration and try fetching the fiscal day status again.

- **Error:** "Fiscal Day No: {number} is already open"
 - **Description:** The system detected that a fiscal day is already open, preventing a new day from being opened.
 - **Possible Fixes:**
 - Close the currently open fiscal day before opening a new one.

4. **CloseFiscalDay()**

Common Error Messages:

- **Error:** "Failed to close Fiscal Day"
 - **Description:** The current fiscal day could not be closed due to some issues.
 - **Possible Fixes:**
 - Ensure there are no pending transactions or processes that need to be completed.
 - Check if the device is properly connected to the FDMS server.

-
- **Error:** "Failed to get the Current Fiscal Day Status"
 - **Description:** Unable to retrieve the current fiscal day status.
 - **Possible Fixes:**
 - Ensure the FDMS server is reachable and the device is online.

-
- **Error:** "Fiscal Day No: {number} is already closed"
 - **Description:** The system detected that the current fiscal day is already closed, and thus, no further action is required.
 - **Possible Fixes:**
 - Ensure that no duplicate actions are being performed, and confirm the fiscal day status before closing.

5. **SubmitReceipt(arg1, ..., argn)**

Common Error Messages:

Error: "Device License Error - Device Not Licensed"

Description: The fiscalisation process failed because the device is not licensed.

Possible Fixes:

- Ensure that the device has a valid license installed.
- Verify the license payment status with the license provider.

Error: "Device License Error - License Status: {status}"

Description: The fiscalisation process failed due to an issue with the license status.

Possible Fixes:

- Check the license status for further details.
 - Contact support if the license status is unclear or incorrect.
-

Error: "The current Fiscal Day has exceeded the maximum hours limit"

Description: Fiscalisation failed because the current fiscal day is beyond the maximum allowable time.

Possible Fixes:

- Close the current fiscal day and open a new one to continue transactions.
 - Check the system clock and ensure it is correctly synchronized.
-

Error: "Invoice Number already exists in your fiscal records"

Description: The provided invoice number is already recorded in the system.

Possible Fixes:

- Use a unique invoice number for each transaction.
 - Review existing fiscal records for the invoice number in question.
-

Error: "Fiscal Day is currently closed"

Description: The fiscal day has been closed, preventing further transactions until a new day is opened.

Possible Fixes:

- Open a new fiscal day to process further transactions.
 - Confirm the status of the fiscal day before attempting operations.
-

Error: "Invalid ReceiptPrintForm {receiptPrintForm}"

Description: The specified receipt print form is invalid.

Possible Fixes:

- Verify the receipt print form against the allowed formats.
- Update the print form configuration if necessary.

Error: "Buyer TIN {buyerTIN} is not a valid TIN"

Description: The Tax Identification Number (TIN) provided for the buyer is invalid.

Possible Fixes:

- Check the TIN for accuracy and compliance with regulatory standards.
 - Validate the TIN through official channels.
-

Error: "FDMS VAT Amount: {taxAmount} Is not equal to the invoice Tax Amount: {invoiceTaxAmount}"

Description: The VAT amount from the FDMS does not match the expected invoice tax amount.

Possible Fixes:

- Ensure that the correct VAT rates are applied to the invoice.
 - Review the calculation of the invoice tax amount for accuracy.
-

Error: "Invalid Receipt lines provided {receiptLines}"

Description: The receipt lines provided do not meet the necessary criteria for processing.

Possible Fixes:

- Check that all required fields for receipt lines are correctly filled out.
 - Ensure receipt lines correspond to valid products/services.
-

Error: "Amount {invoiceAmount} Exceeds the original Invoice Amount"

Description: The specified amount exceeds the original invoice amount recorded in the system.

Possible Fixes:

- Ensure that the transaction amount is within the original invoice limit.
 - Review the invoice and transaction amounts for discrepancies.
-

Error: "Currency {receiptCurrency} does not match original invoice currency"

Description: The currency used in the current transaction does not match that of the original invoice.

Possible Fixes:

- Confirm that the correct currency is used in the transaction.
- Update the currency settings in the system if necessary.

Error: "Invoice Number {invoiceNo} does not exist in your fiscal records"

Description: The provided invoice number cannot be found in the fiscal records.

Possible Fixes:

- Double-check the invoice number for accuracy.
 - Review fiscal records to ensure the invoice was recorded correctly.
-

Error: "Error saving transaction to Queue {Message}"

Description: The system encountered an error while trying to save the transaction to the queue.

Possible Fixes:

- Review system logs for more detailed error information.
 - Ensure the transaction data is valid and complete.
-

Error: "Invalid Receipt Currency. Valid Currency options: {allowedCurrencies}"

Description: The receipt currency provided is invalid or not among the allowed currencies.

Possible Fixes:

- Confirm that the receipt currency is among the supported options.
 - Update the currency configuration if necessary.
-

Error: "Invalid Invoice Amount - Invoice Amount: {invoiceAmount} must be equal to sum of all Line Item Amounts: {totalReceiptLineTotal}"

Description: The invoice amount must equal the total of all line item amounts provided.

Possible Fixes:

- Check the calculations for line item amounts to ensure accuracy.
 - Adjust the invoice amount to match the sum of line items.
-

Error: "Invalid Invoice Amount - Invoice Amount: {invoiceAmount} must be equal to sum of all Line Item Amounts plus sum of all Line Item VAT Amounts: {amount}"

Description: The invoice amount must equal the sum of line item amounts plus the VAT amounts.

Possible Fixes:

- Review the line item and tax calculations for accuracy.
 - Adjust the invoice total to ensure it matches the expected sums.
-

Error: "Invalid receiptTotal - receiptTotal must be greater than 0 for FiscalInvoice and DebitNote."

Description: The total amount for the receipt must be positive for fiscal invoices and debit notes.

Possible Fixes:

- Review the receipt total and ensure it is correctly calculated.
- Verify the type of transaction being processed.

1. GetLicense()

Common Error Messages:

- **Error: "License Status: Device Not Licensed"**
 - **Description:** Could not fetch the device license.
 - **Possible Fixes:**
 - Ensure the device license is paid and activated.
 - Open fiscal day to load license if it has been recently paid or activated.