

Paywire Receiver Integration, Testing Instruction and Transaction Examples 2022-04-06



Table of Contents

Paywire Receiver: Software Download	3
Paywire Receiver: Testing Steps.....	3
Test on the dev computer.....	3
Test from the LAN server	5
Test using remote desktop software or a different computer within the LAN	5
Paywire Receiver: Transaction Examples	6
Sale Request	6
Refund Request	7
Void Request	8
Batch Inquiry Request	8
Batch Close Request	9
Initialize Reader Connection / Get Reader Info	9

Paywire Receiver: Software Download

Paywire Receiver v3.1.4:

https://project.paywire.com/setup/PaywireReceiver/PaywireReceiver_3.1.4.zip

Sample .Net 3.1 Paywire Receiver Tester project (need Visual Studio 2019):

https://project.paywire.com/setup/ReceiverTestForm/PaywireReceiverClientSample_v20210303.zip

Please ensure the latest .Net runtime is installed on all computers that run this application.

Paywire Receiver: Testing Steps

Test on the dev computer

1. **Install** the Paywire Receiver: make sure it is allowed to run by all firewall settings.
2. **Run** Paywire Receiver and the icon will appear in the system tray near the Windows clock on the bottom right of the screen.
3. **Double click** the Paywire Receiver icon to enter the preference view.

Note: If you are testing within the LAN from a different computer within the LAN, please always use port 7777 (Fig. 1). If testing from the same computer, both 7777 and 7778 can be used, but 7778 requires wss:// and the self-signed license installed with the installer. If any changes are made, click on "Save", exit and restart the Paywire Receiver application.

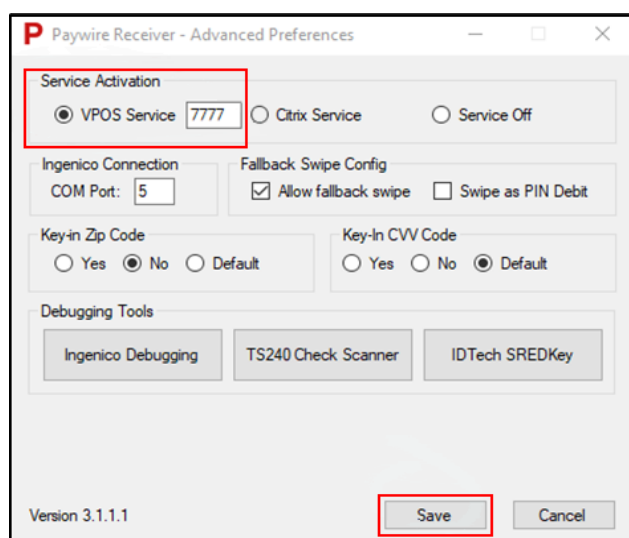


Figure 1

4. Open the sample project with Visual Studio 2019, and try to build and run the tester.
5. On the tester (Fig. 2), set the receiver endpoint to **ws://127.0.0.1:7777** (prefix for 7777 is **ws://**) and the **prefix for localhost testing 7778 is wss://**.
6. Click on "Test Connection" and there will be a request message generated.
7. Make sure the reader is connected and click on the "Send" button. The expected output should resemble Figure 2 below:

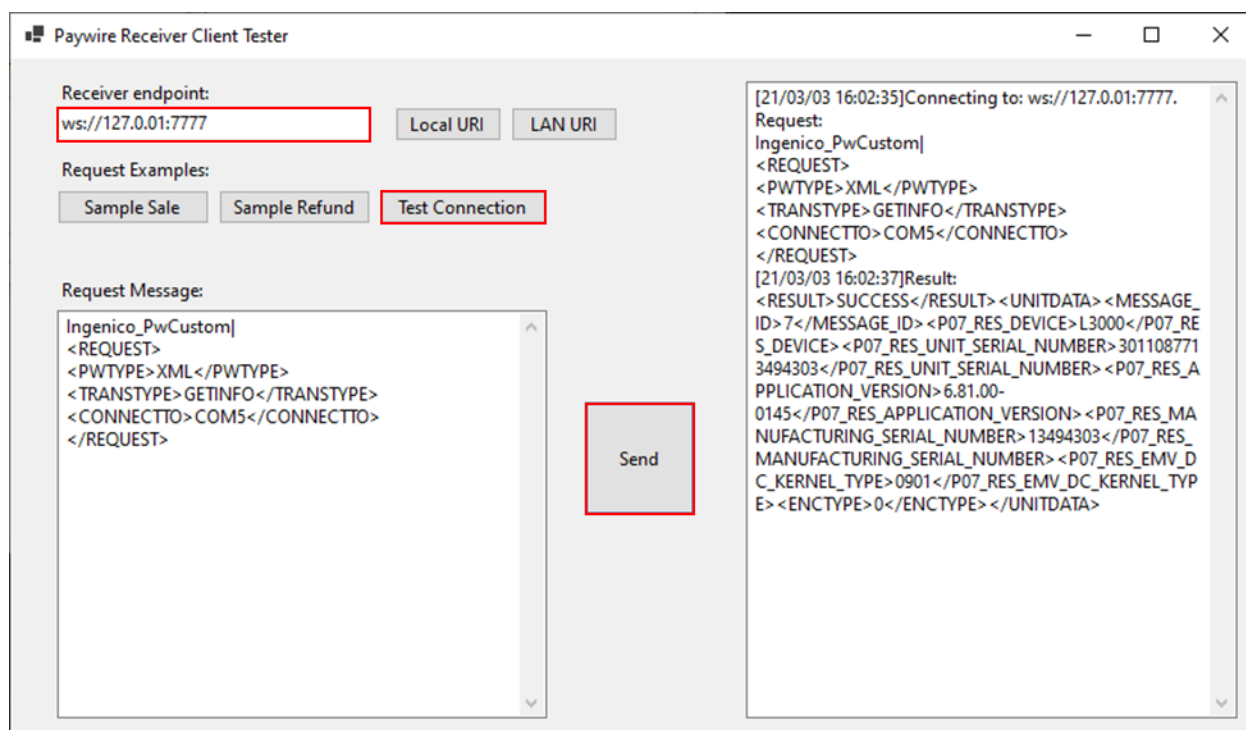


Figure 2

Test from the LAN server

1. On the user's computer (client computer), install and run the Paywire Receiver. Configure the service port to 7777 as shown in the previous step.
2. Copy the test application output from
\\PaywireReceiverClientTester\bin\Debug\netcoreapp3.1\ to the LAN server, make sure .Net runtime is installed on the server in order to run this application.
3. Run the testing from the server, find the private IP of the client computer, then in the server side tester, update the Receiver endpoint accordingly, for example:
ws://192.168.0.5:7777 (prefix for 7777 is ws:// not wss://).
4. Use the same method to test the connection between the server and the client computer.

Test using remote desktop software or a different computer within the LAN

1. Publish the tester using the remote desktop system to the client computer.
2. Run the tester from the remote desktop system on the client computer. Fill out the client IP and port based on the previous testing, for example: **ws://192.168.0.5:7777 (Prefix for 7777 is ws:// not wss://).**
3. Make sure Paywire Receiver is running on the client computer and the reader is connected, then test the connection.

Paywire Receiver: Transaction Examples

Version 20210401

Sale Request

```
processOnDemand|<REQUEST>
  <PWTYPE>XML</PWTYPE>
  <TRANSTYPE>SALE</TRANSTYPE>
  <CONNECTTO>COM5</CONNECTTO>
  <PWCLIENTID></PWCLIENTID>
  <PWKEY></PWKEY>
  <PWUSER></PWUSER>
  <PWPASS></PWPASS>
  <AMOUNT>1.00</AMOUNT>
  <URL>dbtranz</URL>
  <POSTTIMEOUT>120</POSTTIMEOUT>
  <ISDEBIT></ISDEBIT>
  <ISEMVFALLBACK></ISEMVFALLBACK>
  <ENTRYTIMEOUT>300</ENTRYTIMEOUT>
  <ALLOWZIP>0</ALLOWZIP>
  <ALLOWCVV>0</ALLOWCVV>
  <INVOICE></INVOICE>
  <HASCONSUMERFEE></HASCONSUMERFEE>
  <ADDRESS1></ADDRESS1>
  <ADDRESS2></ADDRESS2>
  <CITY></CITY>
  <COUNTRY></COUNTRY>
  <NAME></NAME>
  <ZIP></ZIP>
  <STATE></STATE>
  <EMAIL></EMAIL>
  <PHONE></PHONE>
  <DESCRIPTION></DESCRIPTION>
</REQUEST>
```

Refund Request

```
processOnDemand|<REQUEST>
  <PWTYPE>XML</PWTYPE>
  <TRANSTYPE>CREDIT</TRANSTYPE>
  <CONNECTTO>COM5</CONNECTTO>
  <PWCLIENTID></PWCLIENTID>
  <PWKEY></PWKEY>
  <PWUSER></PWUSER>
  <PWPASS></PWPASS>
  <AMOUNT>1.00</AMOUNT>
  <URL>dbtranz</URL>
  <POSTTIMEOUT>120</POSTTIMEOUT>
  <ISDEBIT></ISDEBIT>
  <ISEMVFALLBACK></ISEMVFALLBACK>
  <ENTRYTIMEOUT>300</ENTRYTIMEOUT>
  <ALLOWZIP>0</ALLOWZIP>
  <ALLOWCVV>0</ALLOWCVV>
  <INVOICE></INVOICE>
  <ADDRESS1></ADDRESS1>
  <ADDRESS2></ADDRESS2>
  <CITY></CITY>
  <COUNTRY></COUNTRY>
  <NAME></NAME>
  <ZIP></ZIP>
  <STATE></STATE>
  <EMAIL></EMAIL>
  <PHONE></PHONE>
  <DESCRIPTION></DESCRIPTION>
</REQUEST>
```

Void Request

```
processOnDemand|<REQUEST>
  <PWTYPE>XML</PWTYPE>
  <TRANSTYPE>CREDIT</TRANSTYPE>
  <CONNECTTO>COM5</CONNECTTO>
  <PWCLIENTID></PWCLIENTID>
  <PWKEY></PWKEY>
  <PWUSER></PWUSER>
  <PWPASS></PWPASS>
  <AMOUNT>1.00</AMOUNT>
  <UID>Paywire_Transaction_UID_To_Void</UID>
  <URL>dbtranz</URL>
  <POSTTIMEOUT>120</POSTTIMEOUT>
  <ENTRYTIMEOUT>300</ENTRYTIMEOUT>
  <DESCRIPTION></DESCRIPTION>
</REQUEST>
```

Batch Inquiry Request

```
processOnDemand|<REQUEST>
  <PWTYPE>XML</PWTYPE>
  <TRANSTYPE>BATCHINQUIRY</TRANSTYPE>
  <CONNECTTO>COM5</CONNECTTO>
  <PWCLIENTID></PWCLIENTID>
  <PWKEY></PWKEY>
  <PWUSER></PWUSER>
  <PWPASS></PWPASS>
  <URL>dbtranz</URL>
  <POSTTIMEOUT>120</POSTTIMEOUT>
</REQUEST>
```

Batch Close Request

```
processOnDemand|<REQUEST>  
  <PWTYPE>XML</PWTYPE>  
  <TRANSTYPE>CLOSE</TRANSTYPE>  
  <CONNECTTO>COM5</CONNECTTO>  
  <PWCLIENTID></PWCLIENTID>  
  <PWKEY></PWKEY>  
  <PWUSER></PWUSER>  
  <PWPASS></PWPASS>  
  <URL>dbtranz</URL>  
  <POSTTIMEOUT>120</POSTTIMEOUT>  
  <DESCRIPTION></DESCRIPTION>  
</REQUEST>
```

Initialize Reader Connection / Get Reader Info

```
Ingenico_PwCustom|<REQUEST>  
  <PWTYPE>XML</PWTYPE>  
  <TRANSTYPE>GETINFO</TRANSTYPE>  
  <CONNECTTO>COM5</CONNECTTO>  
</REQUEST>
```