Shift4 Payments Integration: API Integration Guide

Shift4 Payments’ API Integration Guide provides a framework for building and testing your integration with Shift4 Payments’ DOLLARS ON THE NET® application program interface (API). By reviewing the API Integration Guide, you will be able to:

- Select an API and a communication protocol: HTTPS POST or TCP/IP
- Install Shift4 Payments’ Universal Transaction Gateway® (UTG®)
- Understand the basic format for sending and receiving Shift4 Payments’ API messages
- Understand the Access Token process
- Retrieve an Access Token
- Send a test transaction to Shift4 Payments
- Use DOLLARS ON THE NET for certification testing
- Understand the extensive functionalities that you can add to your interface

Certifying with Shift4 Payments requires dedicated resources and strong attention to detail. While the API Integration Guide provides the information you need to understand Shift4 Payments’ DOLLARS ON THE NET API and how to create (or update) your interface, it is important that you add the functionalitites as agreed to in your approved Integration Plan.

You should also refer to the complete functions guides in MyPortal API Corner to ensure that you apply the correct format to all Shift4 Payments API messages.

For detailed information about Shift4 Payments field descriptions, test card numbers, error codes, API Options, and more, you can also refer to the Shift4 Payments Integration: Reference Guide in MyPortal API Corner.

As always, if you have any questions or need assistance during this process, you can email your assigned Shift4 Payments API Analyst at apisupport@shift4.com.
Selecting an API and Protocol

Before you begin building your interface, you will need to decide how your interface will communicate with Shift4 Payments’ DOLLARS ON THE NET API. We support integrations to DOLLARS ON THE NET via Shift4 Payments’ UTG or direct server-to-server POST.

Interfaces using the UTG will offer our joint customers advanced security, speed, and reliability when processing transactions. The UTG delivers our proprietary Derived Unique Key Per Transaction with Moving Target Encryption (DUKPT w/MTE) data transport, a PCI DSS compliant, strong data encryption methodology that repeatedly changes its algorithm, making it very difficult to decipher. The UTG also allows merchants to support EMV as well as secure offline transaction processing. Finally, the UTG creates comprehensive trace logs that Shift4 Payments can use to help you identify errors and troubleshoot issues as needed.

Requirement: Shift4 Payments’ UTG is required to support EMV and secure offline capability.

Some merchants, such as those who don’t use Windows, won’t be able to use the UTG. The good news is that you can still support these merchants by integrating to DOLLARS ON THE NET via direct server-to-server POST. It’s important to note that if you are developing a server-to-server interface, you must still install the UTG for testing and certification as described in this document. This is required because gathering trace files directly from the testing server is a difficult process.

When integrating to connect via direct server-to-server, as a final certification step, a round of testing will be performed using our certification direct server-to-server POST URL. The certification direct server-to-server POST URL is https://cfapi.shift4test.com/api/S4Tran_Action.cfm.

Requirement: The initial testing for server-to-server processing requires use of the UTG because gathering trace files directly from the testing server is a difficult process.

Shift4 Payments has implemented a load-balancing solution that selects the best available server for processing live server-to-server transactions (i.e., in production). After completing your certification, live transactions are posted using the following URL: https://dotn.shift4api.net/api/s4tran_action.cfm.
HTTPS POST Protocol

HTTPS POST can be used whether your interface uses the UTG or server-to-server POST. Although the APIs are different, the format of the API messages will be the same. The only difference is that interfaces using the UTG may have a few additional parameters to send that apply to UTG-controlled PIN pads.

The HTTPS POST protocol uses key-value pairs to specify the parameters for each API request and response. Many of our developer partners like to use the HTTPS POST protocol because the key names are self-defined, making it easier to read. For example, a simple sale in HTTPS POST might be displayed as:

```
STX=YES&VERBOSE=YES&CONTENTTYPE=XML&APIsignature=%24&APIFormat=O&FunctionRequestCode=1&D&RequestorReference=247717356378592&Invoice=354646111&CardNumber=432100000001119&Vendor=Happy%20Pet%20Store%3APOS_Simulator%3Av2021_07_12&CardType=CC&CardEntryMode=M&CardPresent=Y&ExpirationDate=12285&Clerk=22245&Date=082
```

TCP/IP Protocol

TCP/IP can only be used for integrating to Shift4 Payments using the UTG. The TCP/IP protocol uses data blocks to specify the parameters for each API request. Instead of naming the fields in the data block, the data is interpreted according to its placement in the data block. A simple sale in TCP/IP might be displayed as:

```
$01E247717356378592
$64648111432100000001119_000Happy_Pet_Store:POS_Simulator:v2021_07_12_001CM
Y12282224508242612313450000000000002500000000000000000000000000000000
65 Main Street
65000
0080003
50000000021261000_000 Halifax HTML Tags</p> 0221 333
```
Using Shift4 Payments’ UTG for Certification Testing

All software providers, including those supporting direct server-to-server POST environments, must install the UTG for testing and certification. Most software providers will perform a basic UTG installation. However, if UTG4Cloud is noted in your Integration Plan, you should follow the instructions for Performing a UTG4Cloud Installation (below) instead.

This section will guide you through the installation and configuration of the UTG for testing and certification. Please note that configuring the UTG for live transaction processing (i.e., production) requires different settings that do not apply to the test environment.

---

**Note:** Due to security concerns, computers with IP addresses from outside of the United States may be blocked by Shift4 Payments’ firewall. If you encounter issues with a blocked IP address, contact your Shift4 Payments API Analyst for assistance.

---

**Requirement:** If you support UTG4Cloud, please skip ahead to the Performing a UTG4Cloud Installation section of this document to install UTG4Cloud.

---

**Requirement:** US English must be selected as the language for Windows in order to load the UTG properly. Be sure to configure Windows before attempting to install the UTG.
Performing a Basic UTG Installation

Please complete the following steps to perform a basic UTG installation to allow for interface testing. Where applicable, please follow the directions for HTTP POST or TCP/IP.

WARNING! Review the Universal Transaction Gateway: PA-DSS Implementation Guide located in MyPortal Documentation Corner before installing the UTG.

1. Gather and save the following information from your assigned API Analyst:
   - Serial Number
   - Slot Number (One per UTG)

2. Run the Shift4 Payments Probe utility available at www.shift4.com/downloads/s4probe.zip, and then complete the following steps:
   a. In the Available Items list, select Connect to $$$ on the Net(tm) via Internet using Normal Route.
   b. In the Server field, enter “ns.shift4test.com”.
   c. Click Next.
   d. Click Test.
   e. The Results window will pop up indicating “SUCCESS”, “SOME ROUTES FAILED”, or “FAILURE”. If a failure is indicated, your network or firewall may need reconfiguration. Verify that ports 26880 and 26881 are open in your network or firewall and try running the Shift4 Payments Probe utility again. When the test result reads “SUCCESS”, click close to exit.

3. Install the UTG available at https://myportal.shift4.com/downloads/utg2setup.exe, and then complete the following steps:
   a. Follow the steps in the Installation Wizard, accepting the default location settings.
   b. Continue to follow the steps in the Installation Wizard.
   c. On the last screen of the Installation Wizard, verify that Run Universal Transaction Gateway® TuneUp now is selected and click Finish.

4. In UTG TuneUp, click the API Interfaces tab to add either an HTTP or TCP/IP interface.

Note: The UTG is capable of hosting multiple interfaces with different settings. When naming your interface according to the directions below, you may want to choose a name that identifies the specific interface and configuration.
• *(If applicable)* If you chose HTTP as your method, click **Add > HTTP**, and then complete the following steps:

  i. In the Task Description field, enter a name to identify your interface, for example: “HttpApi”.
  
  ii. From the Host Address drop-down menu, select the IP address of your machine.
  
  iii. In the Port field, enter the port number used by your interface for communication with the UTG; the default is 16448 ($4040 hex).
  
  iv. Click **OK**.

• *(If applicable)* If you chose TCP/IP as your method, click **Add > TCP/IP**, and then complete the following steps:

  i. In the Task Description field, enter a name to identify your interface, for example: “TcpHost”.
  
  ii. From the Host Address drop-down menu, select the IP address of your machine.
  
  iii. In the Port field, enter the port number used by your interface for communication with the UTG; the default is 17476 ($4444 hex).
  
  iv. Click **OK**.

5. Click the **Express** tab.

   a. *(For all installations – whether using TCP or HTTP interfaces)* Click the **TCP/IP** subtab and enter “ns.shift4test.com” in the Host field.

   b. Click the **Identity** sub-tab, and then complete the following steps:

      i. In the Serial field, enter the provided serial number.
      
      ii. In the Slot field, enter the provided slot number.
      
      iii. From the Host Address drop-down menu, select the IP address of your machine.
      
      iv. In the Location field, enter your company’s ZIP/postal code.

   c. Leave the Api Serial Number, Api Password, and Verify Api Password fields blank. *(These fields have been deprecated with the introduction of Shift4 Payments’ Access Token process, which is explained in the **Understanding the Access Token Process** section of this document.)*

6. Click the **Advanced** tab, and then complete the following steps:

   a. In the Environment section, select **Test**.

   b. In the Options section, click **Trace Configuration** and then configure the parameters for your trace files:

      i. In the Directory field, the default file path is C:\Shift4\UTG2\Trace. If you would like to store your trace files in an alternate location, enter the directory location here.

      ii. Select **Verbose** to provide the comprehensive transaction data in trace files that is required for completing certification test scripts.
iii. Click **OK**.

7. **(If applicable)** If you need to add devices, click the **Devices** tab, and then complete the following steps:
   a. In the Device section, click **Add**.
      i. In the Device Name field, enter a unique name that will identify this device.
      ii. From the Device Type drop-down menu, select the PIN pad device.
      iii. In the API Terminal ID field, enter a value of 1-32 alphanumeric characters to identify the PIN pad. This value should be unique as it is used by the interface to prompt the PIN pad.

---

**Requirement:** To prompt a UTG-controlled PIN pad in a request, the API Terminal ID configured in UTG TuneUp must be specified in the **TerminalID** field.

---

- **(If applicable)** If connecting the device directly to the machine running the UTG using a USB or serial port, complete the following steps:
  a. In the Connection section, select **RS-232**.
  b. From the Port drop-down menu, select the COM port to which the device is connected.
  c. From the Baud Rate drop-down menu, select the connection speed as indicated by the device manufacturer or as listed in the *Universal Transaction Gateway: Using EMV External Devices* guide located in MyPortal Documentation Corner.

- **(If applicable)** If connecting the device to a machine that is running a UTG Stub, complete the following steps:
  a. In the Connection section, select **UTG-STUB**.
  b. In the Address field, enter the static IP address of the machine running the UTG Stub.
  c. In the Port field, enter the port to which the UTG Stub is listening. (The default port is 17478.)

- **(If applicable)** If using a point-to-point encryption (P2PE) PIN pad device that will be connected directly to the network via an Ethernet cable, complete the following steps:
  a. In the Connection section, select **Direct TCP/IP**.

---

**Note:** If your device is shipped with a USB connection, you will need to virtualize the COM port using the driver supplied by the manufacturer.
b. In the Address field, enter the static IP address of the P2PE device.

c. In the Port field, enter the port to which the device is listening. (The default port is 12000.)

iv. Click OK in the PinPad window and click OK again in the Terminal List window.

8. Click Save in UTG TuneUp.

9. Close the UTG TuneUp window.

10. From the Start menu, select Programs > Shift4 Corporation > Universal Transaction Gateway > UTG (v2) Stand Alone to verify your connection between the UTG and DOLLARS ON THE NET.

   a. When the UTG has made a connection to DOLLARS ON THE NET, the words “Ready No Failures” should appear in the Global Status row and the row should turn green. (If there is a red error row on the screen, contact your assigned API Analyst.)

   b. Select Advanced from the menu bar and make sure Verbose is selected.

---

Tip: By default, the UTG automatically runs as a service when Windows starts. You may find it helpful to change the UTG startup method to manual so that you can launch the UTG Stand Alone as needed for testing.
Performing a UTG4Cloud Installation

This is a two-part installation with (1) a “Local” UTG that will be connected directly to a PIN pad device, and (2) a “Central” UTG that is server based – designed to be off premise.

Installing a Local UTG

Please complete the following steps to install a local UTG for testing (it is recommended that this UTG installation be on a machine that is in close physical proximity to the PIN pad device):

WARNING! Review the Universal Transaction Gateway: PA-DSS Implementation Guide located in MyPortal Documentation Corner before installing the UTG.

Requirement: US English must be selected as the language for Windows in order to load the UTG properly. Be sure to configure Windows before attempting to install the UTG.

1. Gather and save the following information from your API Analyst:
   - Serial Number
   - Slot Number (One per UTG)
2. Run the Shift4 Payments Probe utility available at www.shift4.com/downloads/s4probe.zip, and then complete the following steps:
   a. In the Available Items list, select Connect to $$$ on the Net(tm) via Internet using Normal Route.
   b. In the Server field, enter “ns.shift4test.com”.
   c. Click Next.
   d. Click Test.
   e. The Results window will pop up indicating “SUCCESS”, “SOME ROUTES FAILED”, or “FAILURE”. If a failure is indicated, your network or firewall may need reconfiguration. Verify that ports 26880 and 26881 are open in your network or firewall and try running the Shift4 Payments Probe utility again. When the test result reads “SUCCESS”, click close to exit.
3. Install the UTG available at https://MyPortal.Shift4.com/downloads/UTG2setup.exe, and then complete the following steps:
   a. Follow the steps in the Installation Wizard, accepting the default location settings.
   b. Continue to follow the steps in the Installation Wizard.
c. On the last screen of the Installation Wizard, verify that Run Universal Transaction Gateway® TuneUp now is selected and click Finish.

4. In UTG TuneUp, click the API Interfaces tab to add a UTG4Cloud interface and then complete the following steps:

   Note: The UTG is capable of hosting multiple interfaces with different settings. When naming your interface according to the directions below, you may want to choose a name that identifies the specific interface and configuration.

   a. Click Add > UTG4Cloud.

   b. In the Task Description field, enter a name to identify your interface, for example: “4Cloud”.

   c. From the Host Address drop-down menu, select the IP address of your machine.

   d. In the Port field, enter the port number used by your interface for communication with the “local” UTG; the default is 13107 ($3333 hex).

   e. From the Transport Format drop-down menu, select Http or Tcp.

   f. Click OK.

5. Click the Express tab.

   a. (For all installations – whether using TCP or HTTP interfaces) Click the TCP/IP subtab and enter “ns.shift4test.com” in the Host field.

   b. Click the Identity sub-tab, and then complete the following steps:

      i. In the Serial field, enter the provided serial number.

      ii. In the Slot field, enter the provided slot number.

      iii. From the Host Address drop-down menu, select the IP address of your machine.

      iv. In the Location field, enter your company’s ZIP/postal code.

      v. Leave the Api Serial Number, Api Password, and Verify Api Password fields blank. (These fields have been depreciated with the introduction of Shift4 Payments’ Access Token process, which is explained in the Understanding the Access Token Process section of this document.)
6. Click the Advanced tab, and then complete the following steps:
   a. In the Environment section, select Test.
   b. In the Options section, click Trace Configuration and then configure the parameters for your trace files:
      i. In the Directory field, the default file path is C:\Shift4\UTG2\Trace. If you would like to store your trace files in an alternate location, enter the directory location here.
      ii. Select Verbose to provide the comprehensive transaction data in trace files that is required for completing certification test scripts.
      iii. Click OK.

7. Click the Devices tab, and then complete the following steps:
   a. In the Device section, click Add.
      i. In the Device Name field, enter a unique name that will identify this device.
      ii. From the Device Type drop-down menu, select the PIN pad device.
      iii. In the API Terminal ID field, enter a value of 1-32 alphanumeric characters to identify the PIN pad. This value should be unique as it is used by the interface to prompt the PIN pad.

   **Requirement:** To prompt a UTG-controlled PIN pad in a request, the API Terminal ID configured in UTG TuneUp must be specified in the TerminalID field.

   • (If applicable) If connecting the device directly to the machine running the UTG using a USB or serial port, complete the following steps:
      a. In the Connection section, select RS-232.
      b. From the Port drop-down menu, select the COM port to which the device is connected.

   **Note:** If your device is shipped with a USB connection, you will need to virtualize the COM port using the driver supplied by the manufacturer.
• *(If applicable)* If connecting the device to a machine that is running a UTG Stub, complete the following steps:
  a. In the Connection section, select **UTG-STUB**.
  b. In the Address field, enter the static IP address of the machine running the UTG Stub.
  c. In the Port field, enter the port to which the UTG Stub is listening. (The default port is 17478.)

• *(If applicable)* If using a P2PE PIN pad device that will be connected directly to the network via an Ethernet cable, complete the following steps:
  a. In the Connection section, select **Direct TCP/IP**.
  b. In the Address field, enter the static IP address of the P2PE device.
  c. In the Port field, enter the port to which the device is listening. (The default port is 12000.)

  iv. Click **OK** in the PinPad window and click **OK** again in the Terminal List window.

8. Click **Save** in UTG TuneUp.

9. Close the UTG TuneUp window.

10. From the Start menu, select **Programs > Shift4 Corporation > Universal Transaction Gateway > UTG (v2) Stand Alone** to verify your connection between the UTG and DOLLARS ON THE NET.

  a. When the UTG has made a connection to DOLLARS ON THE NET, the words “Ready No Failures” should appear in the Global Status row and the row should turn green. (If there is a red error row on the screen, contact your assigned API Analyst.)

  b. Select **Advanced** from the menu bar and make sure **Verbose** is selected.

---

**Tip:** By default, the UTG automatically runs as a service when Windows starts. You may find it helpful to change the UTG startup method to manual so that you can launch the UTG Stand Alone as needed for testing.
Installing a Central UTG

To install a UTG that will act as your system’s Central UTG, complete the following steps on another workstation or server:

1. Run the Shift4 Payments Probe utility available at www.shift4.com/downloads/s4probe.zip, and then complete the following steps:
   a. In the Available Items list, select **Connect to $$$ on the Net(tm) via Internet using Normal Route**.
   b. In the Server field, enter “ns.shift4test.com”.
   c. Click **Next**.
   d. Click **Test**.
   e. The Results window will pop up indicating “SUCCESS”, “SOME ROUTES FAILED”, or “FAILURE”. If a failure is indicated, your network or firewall may need reconfiguration. Verify that ports 26880 and 26881 are open in your network or firewall and try running the Shift4 Payments Probe utility again. When the test result reads “SUCCESS”, click **close** to exit.

2. Install the UTG available at https://MyPortal.Shift4.com/downloads/UTG2setup.exe, and then complete the following steps:
   a. Follow the steps in the Installation Wizard, accepting the default location settings.
   b. Continue to follow the steps in the Installation Wizard.
   c. On the last screen of the Installation Wizard, verify that **Run Universal Transaction Gateway® TuneUp now** is selected and click **Finish**.

3. In UTG TuneUp, click the **API Interfaces** tab to add either an HTTP or TCP/IP interface.

---

**Note:** The UTG is capable of hosting multiple interfaces with different settings. When naming your interface according to the directions below, you may want to choose a name that identifies the specific interface and configuration.

- **(If applicable)** Click **Add > HTTP**, and then complete the following steps:
  i. In the Task Description field, enter a name to identify your interface, for example: “HttpApi”.
  ii. From the Host Address drop-down menu, select the IP address of your machine.
  iii. In the Port field, enter the port number used by your interface for communication with the UTG; the default is 16448 ($4040 hex).
  iv. Click **OK**.
• *(If applicable) Click Add > TCP/IP, and then complete the following steps:
  i. In the Task Description field, enter a name to identify your interface, for example: “TcpHost”.
  ii. From the Host Address drop-down menu, select the IP address of your machine.
  iii. In the Port field, enter the port number used by your interface for communication with the UTG; the default is 17476 ($4444 hex).
  iv. Click OK.

4. Click the Express tab.
   a. *(For all installations – whether using TCP or HTTP interfaces) Click the TCP/IP subtab and enter “ns.shift4test.com” in the Host field.
   b. Click the Identity sub-tab, and then complete the following steps:
      i. In the Serial field, enter the provided serial number.
      ii. In the Slot field, enter the provided slot number.
      iii. From the Host Address drop-down menu, select the IP address of your machine.
      iv. In the Location box, enter your company’s ZIP/postal code.
      v. Leave the Api Serial Number, Api Password, and Verify Api Password boxes blank. (These boxes have been deprecated with the introduction of Shift4 Payments’ Access Token process, which is explained in the Understanding the Access Token Process section of this document.)

5. Click the Advanced tab, and then complete the following steps:
   a. In the Environment section, select Test.
   b. In the Options section, click Trace Configuration and then configure the parameters for your trace files:
      i. In the Directory field, the default file path is C:\Shift4\UTG2\Trace. If you would like to store your trace files in an alternate location, enter the directory location here.
      ii. Select Verbose to provide the comprehensive transaction data in trace files that is required for completing certification test scripts.
      iii. Click OK.

6. Click Save in UTG TuneUp.

7. Close the UTG TuneUp window.

8. From the Start menu, select Programs > Shift4 Corporation > Universal Transaction Gateway > UTG (v2) Stand Alone to verify your connection between the UTG and DOLLARS ON THE NET.
a. When the UTG has made a connection to DOLLARS ON THE NET, the words “KeyPage SUCCESS” should appear in the TcpClient line of the Express Manager row. (If there is a red error row on the screen, contact your assigned API Analyst.)

b. Select Advanced from the menu bar and make sure Verbose is selected.

Tip: By default, the UTG automatically runs as a service when Windows starts. You may find it helpful to change the UTG startup method to manual so that you can launch the UTG Stand Alone as needed for testing.

Disabling the Shift4 UTG2 Service

When the UTG is installed, by default, a service is enabled which runs the UTG in the background. This service is intended for use in live production environments, but conflicts with the UTG Stand Alone application (which is used during certification testing). If the UTG being installed will be used for certification testing, the following steps should be completed in order to disable the service.

To disable the Shift4 UTG2 service, complete the following steps:

1. Click the Open Windows Explorer icon, and the Open Windows Explorer menu will expand.
2. In the Windows Start Menu, enter “run” in the Search field. (A file list will populate.)
3. Under Programs, select Run.
4. In the Run window, in the Open field, enter “services.msc” and click OK.

![Run window with services.msc entered]

5. In the Services window, double-click Shift4 UTG2. (You may need to scroll down to locate the service.)

![Services window with Shift4 UTG2 highlighted]

**Note:** After a UTG installation, this service displays “Automatic” in the Startup Type column.

6. In the Shift4 UTG2 Properties (Local Computer) window, select Disabled from the Startup type list, and then click OK.
7. In the Services window, the Shift4 UTG2 is now listed as “Disabled” in the Startup Type column. Close the Services window. (The background service is now disabled and should not interfere with running the UTG Stand Alone application.)
Formatting Shift4 Payments’ API Messages

Your interface will send API messages that allow the merchant to process payments through DOLLARS ON THE NET using a specific set of function request codes (FRCs). This section describes the basic template for formatting requests in HTTPS POST or TCP/IP. Shift4 Payments’ responses will match the formatting of these requests; however, your interface must be able to accept additional data in responses.

To allow Shift4 Payments to return all of the fields that are available in a particular response, the API Option ALLDATA must be included in every request. Your application must be designed to ignore any fields in responses that are not applicable to a given request.

ASCII

Shift4 Payments uses the 7-bit ASCII standard. When referring to the standard, keep in mind that Shift4 Payments does not support Unicode.

RequestorReference Field

Shift4 Payments requires a unique 12-character alphanumeric value to be sent each time a request is made. Therefore, a RequestorReference field is required. This value must be different from the invoice number and new for each request sent, including subsequent requests. Shift4 Payments returns the corresponding value in the correlated response – which should be used by your application to match up requests with responses. This value will also facilitate troubleshooting in production.

HTTPS POST

Shift4 Payments HTTPS POST messages require sending data in key-value pairs. Response data can be received in either key-value pairs or in XML format.

URL Encoding

Shift4 Payments requires the values (in key-value pairs) to be URL encoded using the RFC 3986 standard, including all reserved characters (e.g., '?', '&', '=', '+', '%', '/') and non-printable characters.

It’s important to note that when URL encoding, only the value of the key-value pair should be URL encoded. For example, if the key is equal to “X” and the value is equal to “75%”, then:

- “75%” should be URL encoded to “75%25”.
- ‘X’ should be joined with an equal sign (‘=’) to the URL-encoded value.

The final string for this example would be “X=75%25”. (A common mistake that some developers make is URL encoding the entire string, which would result in the string equaling “X%3D75%25” for the example shown above, which is incorrect and would cause a parsing error when evaluated by our API.)

Requirement: All values must be URL encoded correctly. Failure to do so will result in transaction errors.
**URL Decoding**

Shift4 Payments’ API responses will be URL encoded. Your interface must be able to URL decode all returned values. A ‘+’ character should be converted to a space. A ‘%’ character in a response indicates that the two characters that follow are a hexadecimal representation of an ASCII character. When your interface encounters “%xx” (where xx is a hexadecimal value), your application should replace that three-character string with the appropriate ASCII character. For example, the string “USD+%24100.00” would be decoded by your application into the string “USD $100.00”.

**HTTPS POST Request Format**

The table below illustrates the basic structure of an HTTPS POST request, followed by an in-depth description of the format:

<table>
<thead>
<tr>
<th>STX= YES</th>
<th>VERBOSE= YES</th>
<th>CONTENTTYPE=XML or TEXT</th>
<th>Key-Value Pairs Required for Specific Functions</th>
<th>ETX= YES</th>
</tr>
</thead>
</table>

- **STX= YES** – This key-value pair is used to indicate the start of every request.
- **VERBOSE= YES** – The VERBOSE value in a request must be “YES” for all applicable data to be returned in the response.
- **CONTENTTYPE=XML or TEXT** – The CONTENTTYPE key must be set to the value “XML” or “TEXT”. A value of “XML” will cause the response to be returned in an XML string. A value of “TEXT” will cause the response to be returned in plaintext.
- **Key-Value Pairs Required for Specific Functions** – The key-value pairs required in your request will vary depending on the function being executed. (A complete list of FRCs is available in the *Shift4 Payments Integration: HTTP POST Complete Functions Reference Guide* located in MyPortal API Corner.) If there is no data for a particular value in a request, then that key-value pair shouldn’t be sent.
- **ETX= YES** – This key-value pair indicates the end of every request.

**TCP/IP**

Shift4 Payments TCP/IP messages contain a start character, a Transaction Header Data Block, a series of additional data blocks depending on the desired function, an end character, and a longitudinal redundancy character.

**TCP/IP Request Format**

The table below illustrates the basic structure of a TCP/IP request, followed by an in-depth description of the format:

<table>
<thead>
<tr>
<th>Start Character (ASCII 02)</th>
<th>Transaction Header Data Block</th>
<th>Data Blocks Required for Specific Functions</th>
<th>End Character (ASCII 03)</th>
<th>Longitudinal Redundancy Character</th>
</tr>
</thead>
</table>

- **Start Character** — ASCII 02 begins every request.
- **Transaction Header Data Block** — The Transaction Header Data Block contains various request parameters, including the FRC.
• **Data Blocks Required for Specific Functions** — The data blocks included in your request will vary depending on the desired function. In each data block, currency fields use an implied decimal, so no decimal should be used. Fixed-length fields are right padded with spaces. The last field in each data block does not require padding if it is not a fixed-length field. If a field is not being used, it should be filled with spaces.

• **End Character** — ASCII 03 indicates the end of every request.

• **Longitudinal Redundancy Character** — The final character in the Shift4 Payments message is the longitudinal redundancy character. This character is no longer validated by Shift4 Payments, so any valid ASCII value may be sent in this field. A missing longitudinal redundancy character at the end of a message will cause the request to hang.

**Understanding the Access Token Process**

Shift4 Payments has set up a process for authenticating API requests using an Access Token. This section explains how an Access Token is retrieved and used (both in our testing and production environments).

---

**Note:** If you’ve already certified with Shift4 Payments and are not yet using an Access Token to authenticate your API requests, we will work with you to create a mutually agreed upon transition plan.

---

**Retrieving an Access Token**

Your interface will need to retrieve an Access Token via a token exchange per merchant account. This process requires a globally unique ID (Client GUID), which will be provided by Shift4 Payments, and an Auth Token, which will be provided by the merchant’s DOLLARS ON THE NET Account Administrator. After performing a token exchange, the Access Token will identify your interface and the merchant account in all subsequent API requests. This gives merchants more control over who is able to make API requests on their behalf.

**Client GUID**

Your API Analyst will provide you with a Client GUID when you begin a new integration project or recertify an existing interface. Your Client GUID is permanent and will be the same for testing and production. Therefore, it must be hard coded into your application to identify the interface version across all merchant accounts and must not be a configurable field. The Client GUID serves as a digital thumbprint that uniquely identifies your interface (and specific version) to Shift4 Payments.

**Auth Token**

An Auth Token gives your application permission to make API requests on behalf of a given merchant account. Your API Analyst will serve as the DOLLARS ON THE NET Account Administrator and provide a test Auth Token for your interface. The test Auth Token can be reused as needed only during certification.
In production, the merchant’s DOLLARS ON THE NET Account Administrator will provide an Auth Token for your interface. An Auth Token in production is valid only for a single token exchange for a specific merchant account within a limited timeframe (24 hours, 3 days, 7 days, or 30 days). Therefore, an Auth Token should never be stored by your application, and should be immediately discarded after an Access Token has been acquired.

If there is a token exchange failure in production or if an Access Token has been revoked, the merchant’s DOLLARS ON THE NET Account Administrator will need to provide a new Auth Token for your interface.

**Generating a New Access Token**

Interfaces must be built to account for the fact that a new Access Token may be required if there is a security incident or compliance concern. In an event such as this, the DOLLARS ON THE NET Account Administrator will provide a new Auth Token for your interface. After a new Access Token has been generated for the affected account, the Account Administrator can revoke the old one. This ensures the merchant can continue processing transactions.

**Performing a Token Exchange**

To begin building and testing your interface, you will need to perform a Token Exchange (FRC CE) request using the Client GUID and test Auth Token provided by your API Analyst.

---

**Requirement:** The Client GUID supplied by your API Analyst must be hard coded into your application because it will permanently identify that version of your interface across all merchant accounts.

---

This section will walk you through the process of completing an Access Token exchange in HTTPS POST or TCP/IP. For additional information about the general format for API messages, see the [Formatting Shift4 Payments API Messages](#) section of this document.

The Access Token you receive will be used to identify your interface and the merchant account to DOLLARS ON THE NET during development, testing, and certification. Once your certification is complete, you may continue to use the Access Token for additional interface testing as needed.

**HTTPS POST**

Please follow the directions below to connect to Shift4 Payments, send an FRC CE in HTTPS POST, and receive an Access Token for testing.

---

**Requirement:** All key-value pairs in the example below are prior to URL encoding. Your application must URL encode all values in the key-value pairs before concatenation.
1. From the Start menu, select **Programs > Shift4 Corporation > Universal Transaction Gateway > UTG (v2) Stand Alone**.

2. Build a simple request to exchange your Client GUID and Auth Token for an Access Token. The following key-value pairs will be required, and each key-value pair must be separated by an ampersand (‘&’):

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STX</td>
<td>YES</td>
</tr>
<tr>
<td>VERBOSE</td>
<td>YES</td>
</tr>
<tr>
<td>CONTENTTYPE</td>
<td>&lt;Either “XML” or “TEXT” to specify the return message format&gt;</td>
</tr>
<tr>
<td>APISignature</td>
<td>$</td>
</tr>
<tr>
<td>APIFormat</td>
<td>0</td>
</tr>
<tr>
<td>FunctionRequestCode</td>
<td>CE</td>
</tr>
<tr>
<td>RequestorReference</td>
<td>&lt;Unique 12-character alphanumeric descriptor to facilitate your interface matching up requests with responses&gt;</td>
</tr>
<tr>
<td>Vendor</td>
<td>&lt;Your Vendor* description to identify the software and version that was used&gt;</td>
</tr>
<tr>
<td>Date</td>
<td>[MMDDYY]</td>
</tr>
<tr>
<td>Time</td>
<td>[HHMMSS]¹</td>
</tr>
<tr>
<td>AuthToken</td>
<td>&lt;Your supplied AuthToken&gt;</td>
</tr>
<tr>
<td>ClientGUID</td>
<td>&lt;Your supplied ClientGUID&gt;</td>
</tr>
<tr>
<td>APIOptions</td>
<td>ALLDATA</td>
</tr>
<tr>
<td>ETX</td>
<td>YES</td>
</tr>
</tbody>
</table>

*For Vendor parameters, see the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner.

¹Time must be formatted in military format (e.g., 4:14:49 p.m. would be formatted “161449”) for the local time zone where the merchant makes the sale.

3. Post the string, including a proper HTTP header, to the IP address and port number that you configured in UTG TuneUp (default port: 16448).

4. Shift4 Payments will process the request and send a response with an Access Token.
   
   a. If an error is returned, evaluate the error code, verify the parameters that were sent, and retry.

5. Store the Access Token in your application for use in all future requests.

**TCP/IP**

Please follow the directions below to connect to Shift4 Payments, send an FRC CE in TCP/IP, and receive an Access Token for testing.
1. From the Start menu, select **Programs > Shift4 Corporation > Universal Transaction Gateway > UTG (v2) Stand Alone**.

2. Open a socket to the IP address and port number that you configured in UTG TuneUp (default port: 17476).

3. Build a simple request to exchange your Client GUID and Auth Token for an Access Token as outlined below:

   Start Character (ASCII 02)
   Transaction Header Data Block
   Vendor (Data Block 000)
   API Options (Data Block 023)
   Token Exchange (Data Block 095)
   End Character (ASCII 03)
   Longitudinal Redundancy Character

   The parameters that need to be sent in the corresponding data blocks are as follows:

   **Transaction Header Data Block**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>APISignature</td>
<td>$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>APIFormat</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>FunctionRequestCode</td>
<td>CE</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>RequestorReference</td>
<td>&lt;Unique 12-character alphanumeric descriptor to facilitate your interface matching up requests with responses&gt;</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>ErrorIndicator</td>
<td>&lt;One space&gt;</td>
</tr>
<tr>
<td>17</td>
<td>6</td>
<td>PrimaryErrorCode</td>
<td>&lt;Six spaces&gt;</td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td>SecondaryErrorCode</td>
<td>&lt;Three spaces&gt;</td>
</tr>
<tr>
<td>26</td>
<td>10</td>
<td>MerchantID</td>
<td>&lt;Ten zeros&gt; (Deprecated field.)</td>
</tr>
<tr>
<td>36</td>
<td>10</td>
<td>TranID</td>
<td>&lt;Ten spaces&gt;</td>
</tr>
<tr>
<td>46</td>
<td>10</td>
<td>Invoice</td>
<td>&lt;Ten spaces&gt;</td>
</tr>
</tbody>
</table>

   **000 Vendor**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Field Separator</td>
<td>ASCII 028</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Data Block Indicator</td>
<td>000</td>
</tr>
<tr>
<td>4</td>
<td>Up to 64</td>
<td>Vendor</td>
<td>&lt;Your Vendor description†&gt;</td>
</tr>
</tbody>
</table>

   †For Vendor parameters, see the **Shift4 Payments Integration: Reference Guide** located in MyPortal API Corner.

   **023 API Options Data Block**
<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Field Separator</td>
<td>ASCII 028</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Data Block Indicator</td>
<td>023</td>
</tr>
<tr>
<td>4</td>
<td>Up to 255</td>
<td>APIOptions</td>
<td>ALLDATA</td>
</tr>
</tbody>
</table>

**095 Token Exchange Data Block**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Field Separator</td>
<td>ASCII 028</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Data Block Indicator</td>
<td>095</td>
</tr>
<tr>
<td>4</td>
<td>51</td>
<td>AuthToken</td>
<td>&lt;Your supplied Auth Token&gt;</td>
</tr>
<tr>
<td>55</td>
<td>Up to 51</td>
<td>ClientGUID</td>
<td>&lt;Your supplied Client GUID&gt;</td>
</tr>
</tbody>
</table>

4. Send the request to the UTG.
5. Shift4 Payments will process the request and send a response with an Access Token.
   a. If an error is returned, evaluate the error code, verify the parameters that were sent, and retry.
6. Store the Access Token in your application for use in all future requests.

**Sending Your First Transaction Request**

This section describes how to perform a sample payment request as a “Hello, World!” message. After you’ve completed this test, you can continue building the other FRCs for your interface.

---

**Note:** The requests below are for testing purposes only. Additional parameters will be required later in the development process to test your interface for real-world transaction processing.

---

**HTTPS POST**

Please follow the directions below to connect to Shift4 Payments and send a test transaction.

---

**Requirement:** All key-value pairs in the example below are prior to URL encoding. Your application must URL encode all values in the key-value pairs before concatenation.
1. From the Start menu, select **Programs > Shift4 Corporation > Universal Transaction Gateway > UTG (v2) Stand Alone**.

2. Build a simple request using the following key-value pairs, with each key-value pair separated by an ampersand (`&`):

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STX</td>
<td>YES</td>
</tr>
<tr>
<td>VERBOSE</td>
<td>YES</td>
</tr>
<tr>
<td>CONTENTTYPE</td>
<td>&lt;Either “XML” or “TEXT” to specify the return message format&gt;</td>
</tr>
<tr>
<td>APISignature</td>
<td>$</td>
</tr>
<tr>
<td>APIFormat</td>
<td>0</td>
</tr>
<tr>
<td>FunctionRequestCode</td>
<td>1B</td>
</tr>
<tr>
<td>RequestorReference</td>
<td>&lt;Unique 12-character alphanumeric descriptor to facilitate your interface matching up requests with responses&gt;</td>
</tr>
<tr>
<td>Vendor</td>
<td>&lt;Your Vendor‡ description to identify the software and version that was used&gt;</td>
</tr>
<tr>
<td>CardEntryMode</td>
<td>&lt;Either ‘M’, ‘1’, or ‘2’ to indicate how the card data was entered (i.e., ‘M’ for manual card entry, ‘1’ for track 1 only or dual track, or ‘2’ for track 2 only)&gt;</td>
</tr>
<tr>
<td>CardPresent</td>
<td>&lt;Either ‘Y’ or ‘N’ to indicate whether the card was physically present&gt;</td>
</tr>
<tr>
<td>ExpirationDate</td>
<td>[MMYY]</td>
</tr>
<tr>
<td>Clerk</td>
<td>&lt;5-digit numeric identifier used to identify a clerk&gt;</td>
</tr>
<tr>
<td>Date</td>
<td>[MMDDYY]</td>
</tr>
<tr>
<td>Time</td>
<td>[HHMMSS]§</td>
</tr>
<tr>
<td>SaleFlag</td>
<td>$</td>
</tr>
<tr>
<td>PrimaryAmount#</td>
<td>100.00</td>
</tr>
<tr>
<td>CardNumber</td>
<td>&lt;Please select a test card number from the Acceptable Test Card Numbers section of the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner&gt;</td>
</tr>
<tr>
<td>APIOptions</td>
<td>ALLDATA</td>
</tr>
<tr>
<td>CustomerReference</td>
<td>&lt;Unique max 25-character alphanumeric descriptor&gt;</td>
</tr>
<tr>
<td>TaxIndicator</td>
<td>&lt;Either ‘Y’ or ‘N’ to indicate whether tax was charged&gt;</td>
</tr>
<tr>
<td>TaxAmount</td>
<td>&lt;Amount of tax charged&gt;</td>
</tr>
<tr>
<td>Key</td>
<td>Value/Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DestinationZipCode</td>
<td>&lt;Max 9-digit ZIP/postal code where merchandise was shipped or where goods/services were rendered&gt;</td>
</tr>
<tr>
<td>ProductDescriptor1</td>
<td>&lt;Max 40-character text description of items purchased or services sold (e.g., “Groceries”)&gt;</td>
</tr>
<tr>
<td>AccessToken</td>
<td>&lt;Your Access Token&gt;</td>
</tr>
<tr>
<td>ETX</td>
<td>YES</td>
</tr>
</tbody>
</table>

‡For Vendor parameters, see the *Shift4 Payments Integration: Reference Guide* located in MyPortal API Corner.
§Time must be formatted in military format (e.g., 4:14:49 p.m. would be formatted “161449”) for the local time zone where the merchant makes the sale.
ǁSaleFlag is only needed in an Auth (FRC 1B) or Sale (FRC 1D) request to specify whether the direction of the funds should credit the consumer or the merchant.
#For a complete list of trigger values, please see the *Shift4 Payments Test Server Logic* section of the *Shift4 Payments Integration: Reference Guide* located in MyPortal API Corner.

3. Post the string, including a proper HTTP header to the IP address, and port number that you configured in UTG TuneUp (default port: 16448).

4. Shift4 Payments will process the request and send a response.
   - If you received a response, check the UTG Task Explorer window. In the dark blue History area, you should see the details of the transaction you just completed.
   - If you did not receive a response:
     i. In the UTG Task Explorer window, verify the connection between the UTG and DOLLARS ON THE NET by ensuring that “KeyPage SUCCESS” is displayed in the TcpClient line of the Express Manager.
     ii. Verify the message is formatted correctly. Confirm that the IP address and port number are correctly configured in UTG TuneUp.
     iii. In the UTG Task Explorer window, watch the yellow Requests area while you send the request again. You should see information about your transaction appear in this area when it is processed by the UTG and disappear when a response has been sent.
     iv. Verify that a response has been received.

5. Log in to DOLLARS ON THE NET to view the transaction by completing the following steps:
   a. Open a browser and navigate to https://dotn.shift4test.com. Enter the login information that your Shift4 Payments API Analyst provided and click Log In.
   b. From the menu, select Transactions > Select Merchant.
   c. On the Select Merchant page, select the merchant for which you would like to view transactions and click Submit.
d. From the menu, select **Transactions > Current Transactions.**
TCP/IP

1. From the Start menu, select Programs > Shift4 Corporation > Universal Transaction Gateway > UTG (v2) Stand Alone.

2. Open a socket to the IP address and port number that you configured in UTG TuneUp (default port: 17476).

3. Build a simple request as outlined below:

<table>
<thead>
<tr>
<th>Start Character (ASCII 02)</th>
<th>Transaction Header Data Block</th>
<th>Vendor (Data Block 000)</th>
<th>Standard Transaction (Data Block 001)</th>
<th>Level 2 Data (Data Block 008)</th>
<th>Purchasing Card Data (Data Block 009)</th>
<th>API Options (Data Block 023)</th>
<th>Access Token (Data Block 094)</th>
<th>End Character (ASCII 03)</th>
<th>Longitudinal Redundancy Character</th>
</tr>
</thead>
</table>

The parameters that need to be sent in the corresponding data blocks are as follows:

**Transaction Header Data Block**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>APISignature</td>
<td>$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>APIFormat</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>FunctionRequestCode</td>
<td>1B</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>RequestorReference</td>
<td>&lt;Unique 12-character alphanumeric descriptor to facilitate your interface matching up requests with responses&gt;</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>ErrorIndicator</td>
<td>&lt;One space&gt;</td>
</tr>
<tr>
<td>17</td>
<td>6</td>
<td>PrimaryErrorCode</td>
<td>&lt;Six spaces&gt;</td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td>SecondaryErrorCode</td>
<td>&lt;Three spaces&gt;</td>
</tr>
<tr>
<td>26</td>
<td>10</td>
<td>MerchantID</td>
<td>&lt;Ten zeros&gt;</td>
</tr>
<tr>
<td>36</td>
<td>10</td>
<td>TranID</td>
<td>&lt;Ten spaces&gt;</td>
</tr>
<tr>
<td>46</td>
<td>10</td>
<td>Invoice</td>
<td>&lt;Ten spaces&gt;</td>
</tr>
<tr>
<td>56</td>
<td>Up to 32</td>
<td>CardNumber</td>
<td>&lt;Please select a test card number from the Acceptable Test Card Numbers section of the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner&gt;</td>
</tr>
</tbody>
</table>

**000 Vendor Description Data Block**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Field Separator</td>
<td>ASCII 028</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Data Block Indicator</td>
<td>000</td>
</tr>
</tbody>
</table>
### Offset Properties

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Up to 64</td>
<td>Vendor</td>
<td></td>
</tr>
</tbody>
</table>

||For the Vendor parameters, see the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner.

#### 001 Standard Transaction Data Block

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Field Separator</td>
<td>ASCII 028</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Data Block Indicator</td>
<td>001</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>CardType</td>
<td>&lt;Two spaces&gt;</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>CardEntryMode</td>
<td>&lt;Either ‘M’, ‘1’, or ‘2’ to indicate how the card data was entered (i.e., ‘M’ for manual card entry, ‘1’ for track 1 only or dual track, or ‘2’ for track 2 only)&gt;</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>CardPresent</td>
<td>&lt;Either ‘Y’ or ‘N’ to indicate whether the card was physically present&gt;</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>ExpirationDate</td>
<td>[MMYY]</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>Clerk</td>
<td>&lt;5-digit numeric identifier used to identify a clerk&gt;</td>
</tr>
<tr>
<td>17</td>
<td>6</td>
<td>Date</td>
<td>[MMDDYY]</td>
</tr>
<tr>
<td>23</td>
<td>6</td>
<td>Time##</td>
<td>[HHMMSS]</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>SaleFlag##</td>
<td>S</td>
</tr>
<tr>
<td>30</td>
<td>14</td>
<td>PrimaryAmount###</td>
<td>000000000010000</td>
</tr>
<tr>
<td>44</td>
<td>14</td>
<td>SecondaryAmount</td>
<td>00000000000000</td>
</tr>
<tr>
<td>58</td>
<td>1</td>
<td>Response</td>
<td>&lt;One space&gt;</td>
</tr>
<tr>
<td>59</td>
<td>6</td>
<td>Authorization</td>
<td>&lt;Six spaces&gt;</td>
</tr>
<tr>
<td>65</td>
<td>1</td>
<td>AVSResult</td>
<td>&lt;One space&gt;</td>
</tr>
<tr>
<td>66</td>
<td>1</td>
<td>AVSStreetVerified</td>
<td>&lt;One space&gt;</td>
</tr>
<tr>
<td>67</td>
<td>1</td>
<td>AVSZipVerified</td>
<td>&lt;One space&gt;</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>ValidAVS</td>
<td>&lt;One space&gt;</td>
</tr>
<tr>
<td>69</td>
<td>Up to 128</td>
<td>TrackInformation</td>
<td>&lt;One space&gt;</td>
</tr>
</tbody>
</table>

# Time must be formatted in military format (e.g., 4:14:49 p.m. would be formatted “161449”) for the local time zone where the merchant makes the sale.

## SaleFlag is only needed in an Auth (FRC 1B) or Sale (FRC 1D) request to specify whether the direction of the funds should credit the consumer or the merchant.
### For a complete list of trigger values, please see the *Shift4 Payments Test Server Logic* section of the *Shift4 Payments Integration: Reference Guide* located in MyPortal API Corner.

#### 008 Level 2 Card

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Field Separator</td>
<td>ASCII 028</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Data Block Indicator</td>
<td>008</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>CustomerReference</td>
<td>&lt;Unique max 25-character alphanumeric descriptor&gt;</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>TaxIndicator</td>
<td>&lt;Either ‘Y’ or ‘N’ to indicate whether tax was charged&gt;</td>
</tr>
<tr>
<td>30</td>
<td>14</td>
<td>TaxAmount</td>
<td>&lt;Amount of tax charged&gt;</td>
</tr>
<tr>
<td>44</td>
<td>Up to 9</td>
<td>DestinationZipCode</td>
<td>&lt;Max 9-digit ZIP/postal code where merchandise was shipped or where goods/services were rendered&gt;</td>
</tr>
</tbody>
</table>

#### 009 Purchasing Card

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Field Separator</td>
<td>ASCII 028</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Data Block Indicator</td>
<td>009</td>
</tr>
<tr>
<td>4</td>
<td>Up to 40</td>
<td>ProductDescriptor1</td>
<td>&lt;Max 40-character text description of items purchased or services sold (e.g., “Groceries”) &gt;</td>
</tr>
<tr>
<td>44</td>
<td>1</td>
<td>Caret</td>
<td>ASCII 094</td>
</tr>
<tr>
<td>45</td>
<td>Up to 40</td>
<td>ProductDescriptor2</td>
<td>&lt;40 spaces&gt;</td>
</tr>
<tr>
<td>85</td>
<td>1</td>
<td>Caret</td>
<td>ASCII 094</td>
</tr>
<tr>
<td>86</td>
<td>Up to 40</td>
<td>ProductDescriptor3</td>
<td>&lt;40 spaces&gt;</td>
</tr>
<tr>
<td>126</td>
<td>1</td>
<td>Caret</td>
<td>ASCII 094</td>
</tr>
<tr>
<td>127</td>
<td>Up to 40</td>
<td>ProductDescriptor4</td>
<td>&lt;One space&gt;</td>
</tr>
</tbody>
</table>

#### 023 API Options

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Field Separator</td>
<td>ASCII 028</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Data Block Indicator</td>
<td>023</td>
</tr>
<tr>
<td>4</td>
<td>Up to 255</td>
<td>APIOptions</td>
<td>ALLDATA</td>
</tr>
</tbody>
</table>
094 AccessToken Data Block

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Field Separator</td>
<td>ASCII 028</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Data Block Indicator</td>
<td>094</td>
</tr>
<tr>
<td>4</td>
<td>Up to 51</td>
<td>AccessToken</td>
<td>&lt;Your Access Token&gt;</td>
</tr>
</tbody>
</table>

4. Send the request to the UTG.

5. Shift4 Payments will process the request and send a response.
   a. If you received a response, check the UTG Task Explorer window. In the dark blue History area, you should see the details of the transaction you just completed.
   b. If you did not receive a response:
      i. In the UTG Task Explorer window, verify the connection between the UTG and DOLLARS ON THE NET by ensuring that “KeyPage SUCCESS” is displayed in the TcpClient line of the Express Manager.
      ii. Verify the message is formatted correctly. Confirm that the IP address and port number are correctly configured in UTG TuneUp.
      iii. In the UTG Task Explorer window, watch the yellow Requests area while you send the request again. You should see information about your transaction appear in this area when it is processed by the UTG and disappear when a response has been sent.
      iv. Verify that a response has been received.

6. Close the socket.

7. Log in to DOLLARS ON THE NET to view the transaction by completing the following steps:
   a. Open a web browser and navigate to https://dotn.shift4test.com. Enter the login information provided to you by your Shift4 Payments API Analyst and click Log In.
   b. From the menu, select Transactions > Select Merchant.
   c. On the Select Merchant page, select the merchant for which you would like to view transactions and click Submit.
   d. From the menu, select Transactions > Current Transactions.
Development Testing

As you develop your interface, you will use our test environment to simulate a connection with a processor. Using our test environment, your assigned API Analyst will help you troubleshoot the functionalities you’re building into your interface and verify that you can successfully process transactions through Shift4 Payments.

You already set up a connection to the test environment when you installed the UTG. This section explains what you need to know about the test environment to use it effectively.

Note: The test environment is not designed for performance testing or load testing.

Using Test Cards

As a simulated environment, the test environment recognizes a limited number of test card numbers. You can find a list of valid test card numbers in the Shift4 Payments Test Card Numbers section of the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner. We can also provide physical test cards if you want to test card-swipe capability. Just ask your assigned API Analyst for assistance.

We’re happy to say that you will not need to go through the painstaking trouble of testing EMV chip cards or contactless functionality. Shift4 Payments’ UTG is already certified for EMV and NFC/contactless with the processors and its device control capability manages these payment flows.

Understanding Trace Files

The UTG creates logs (trace files), which record all of the UTG’s activities from the moment you start it. Shift4 Payments uses these trace files during certification to verify that your interface is working as designed and to help you troubleshoot as needed.

Important: Your Trace Configuration was set up when you installed the UTG. To confirm the location of your trace files in UTG TuneUp, select the Advanced tab and click Trace Configuration. In the Trace Parameters window, the location of your trace files will be noted in the Directory box. (By default, log files are located in C:\Shift4\UTG2\Trace.) Verbose should also be selected. If no changes are required, click OK and close UTG TuneUp.
Collecting Trace Files

The UTG must be set to Verbose mode to write detailed information to the trace files. Before running test transactions, please ensure that the UTG is set to Verbose by completing the following steps:

1. From the Start menu, select Programs > Shift4 Corporation > Universal Transaction Gateway > UTG (v2) Stand Alone.
2. Select Advanced and ensure that there is a check mark next to Verbose.

Rolling Over Trace Files

The UTG is constantly logging its activities in trace files, so these files can become very large and difficult to read. To make trace files much easier to analyze and reduce Shift4 Payments response time, the following process should be used to roll over trace files before and after each test:

1. In the UTG Stand Alone, select Advanced > Rollover before beginning a test to start a new log file.
2. After running the test, select Advanced > Rollover again to complete the test log.
3. Go to the location where your trace files are held (as verified in the Important note above) and find the most recently modified TRC file (this should have a smaller file size than the rest of the files, but be greater than zero).
4. Rename the file, identifying the test you just completed (i.e., “test1.TRC”), and email the file to your assigned Shift4 Payments API Analyst.

---

Important: Every test should be isolated into its own trace file using the Rollover function before and after each test.
Using the DOLLARS ON THE NET Web Interface during Testing

While you design and develop your interface, we recommend using our DOLLARS ON THE NET web interface to review and troubleshoot your test transactions. This section explains how to use the applicable auditing features in DOLLARS ON THE NET during testing, including how to view and batch transactions as well as how to print and export reports. Please note that the instructions in this section are applicable to testing only and should not be used in production.

WARNING! The instructions provided in this section are for development and certification testing only and should not be used in production.

Logging In to Your DOLLARS ON THE NET Test Account

You can log in to your DOLLARS ON THE NET test account at https://dotn.shift4test.com using the Account Number, Username, and Password that your API Analyst provided. When you first log in, you will be prompted to create your own password.

Changing Administrative-Level Settings on Your Test Account

Because the DOLLARS ON THE NET test environment is a shared resource, you will not have access to Account Administrator functions. If you need to make any administrative-level changes to your account, such as creating IT’S YOUR CARD® (IYC) gift cards for testing, please email your assigned Shift4 Payments API Analyst.

Viewing Test Transactions

To view the test transactions that you process, complete the following steps:

Note: If there is only one merchant account assigned to your user profile for testing, you may skip ahead to step 3. In addition, the system only allows for the selection of one merchant at a time by default. To enable the selection of multiple merchants, complete the following steps:

1. From the menu, select User > Change Profile.
2. On the Change Profile page, in the Other Options area, select Multi-MID selection.
3. Click Apply.

1. From the menu, select Transactions > Select Merchant.
2. On the Select Merchant page, select the merchant(s) for which you would like to view transactions and click Submit.
3. Select **Transactions > Current Transactions**.

4. Select the desired viewing options and click **Apply** to display the corresponding test transactions.

---

**Note:** An entire row that is highlighted red indicates a problem transaction. If you see any problem transactions in a batch, you will need to void them before batching the transactions by following the instructions in the **Processing Batches** section of this document.

---

### Searching for Transactions

To search for a specific transaction, use the Search field located in the menu. You can enter the sale amount, clerk number (left padded with zeros if less than five digits), card number (first four or last four), customer name, or invoice number (full number or last four numbers) into the Search field and press the **Enter** key (or click and select **Quick Search**).

To search using additional criteria, click and select **Advanced Search** from the list.

### Editing Transactions in Groups

To edit a group of transactions go to **Transactions > Current Transactions in Grid**. While not recommended for use in production, this option is helpful for voiding problem transactions and authorizations prior to batching during testing. You can find the instructions for how to do this in the **Closing Batches** section of this document.

### Processing Batches

Reusing invoice numbers can cause problems. In production, a merchant may mistakenly undercharge a consumer and lose money. In testing, reusing invoice numbers can yield unexpected results. For these reasons, we require a unique 10-character invoice number to identify every transaction on a per-batch basis.

If your application reuses invoice numbers, or while in development your interface used an invoice number more than once, then you will need to close the batch before using a given invoice number again.

When developing your interface, it is common to have many transactions appear on your DOLLARS ON THE NET test account, so periodically batching the test transactions will help to clear out the invoices that posted to your account. This improves troubleshooting and makes it easier to view the test transactions you’ve recently processed.

The steps outlined below will guide you in cleaning up your test account by removing any problem transactions or authorizations prior to closing a batch.

### Voiding Problem Transactions

Problem transactions are items flagged by DOLLARS ON THE NET for auditors to review. There are many reasons for a transaction to be flagged as a problem, such as insufficient authorizations, declined transactions that have not been programmatically voided, or an expired card. Before a batch can be settled, a merchant needs to fix or exclude problem transactions.

For testing purposes, you should void problem transactions before closing the batch. This can easily be accomplished using the Current Transactions in Grid page.
To void problem transactions using the Current Transactions in Grid page, complete the following steps:

1. From the menu, select **Transactions > Current Transactions in Grid**.
2. In the viewing options area, select **Sales, Authorizations, Refunds, Problems**, and all of the card types. Ensure the Voids, Non-Problems, Duplicates Only, Modified Only, Referrals Only, and Errors Only options are cleared.
3. To view all of the problem transactions on one page, select **5000** from the Per Page list.
4. Click **Apply**.
5. In the auditing options list, select **Change Selected to Void**.
6. To select all of the problem transactions, click **All**.
7. Click **Submit**.
8. Repeat steps 5-7 as needed until all problem transactions have been voided.

### Changing Authorizations

In order to remove used invoice numbers that are associated with authorization only transactions, you must change the authorizations to have a sale/refund status. Once they have a sale/refund status, you can close them in a batch which will allow those invoice numbers to be used again.

To change authorizations to a sale/refund status, complete the following steps:

1. From the menu, select **Transactions > Current Transactions in Grid**.
2. In the viewing options area, select **Authorizations, Non-Problems**, and all of the card types. Ensure the Sales, Refunds, Voids, Problems, Duplicates Only, Modified Only, Referrals Only, and Errors Only options are cleared.
3. To view all of the transactions on one page, select **5000** from the Per Page list.
4. Click **Apply**.
5. In the auditing options list, select **Change Selected to Sale/Refund**.

6. To select all Auth Only transactions, click **All**.

7. Click **Submit**.

8. Repeat steps 5-7 as needed until all authorizations have been changed to a have a sale/refund status.

**Closing Batches**

After you have voided any problem transactions and changed authorizations to sales/refunds, complete the following steps to close the batch:

1. From the menu, select **Transactions > Current Transactions**.

2. On the Current Transactions page, click **Select Business Date**.

3. In the Business Date window, click on the current date and select **All transactions regardless of date** to include all past, present, and future transactions in the batch and click **Apply**.

4. Select **Sales, Refunds, Non-Problems**, all card types, and all card entry modes. Ensure the other options are cleared, and then click **Apply**.
5. At the bottom of the page, click **Close Batch**.

![Batch Close page](image)

6. The Batch Close page will be displayed and DOLLARS ON THE NET will begin to process the batch. Do not navigate away.

7. *(If applicable)* If Fraud Sentry® indicates any issues, such as duplicate transactions, simply enter your username and password at the bottom of the Batch Close User Confirmation page, then click **Close Batch**.

![Batch Close User Confirmation page](image)

**WARNING:**
Batch contains possible duplicates. While this trigger does not indicate a definite problem, you will want to double check the transactions below before continuing.

**Batch Summary**
- **Merchant Name:** DAN YC HOTEL
- **Business Date:** Wed Feb 22, 2017
- **Submit Date:** Wed Feb 22, 01:53 PM 2017

![Batch Summary](image)

**YOUR ATTENTION IS REQUIRED!**
One or more warnings were detected during the batch preparation and the process has been paused. The Fraud Sentry settings require your approval to continue. You must return to the bottom of this page to continue this batch submittal process...

![Approval required message](image)

8. When the batch has been successfully submitted, the Batch Close page will display a summary of the batch and the batch process.
Reporting

The reports available in DOLLARS ON THE NET should reconcile to the end-of-day reports produced by your interface.

Reports can be run by selecting Reports from the DOLLARS ON THE NET side menu, which is accessible from the Current Transactions page and the Archived Transactions page when a specific date is selected. The reports will be presented as PDF documents.

You can select from the following available reports:

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail</td>
<td>This report will contain all of the transactions listed on the Current Transactions page or for a specific date in Archived Transactions. The details will include information like the invoice number, card type, business date, masked card number, transaction amount, subtotals by card type, and grand total, which a merchant’s auditor will use to compare to their end of day report.</td>
</tr>
<tr>
<td>Totals Only</td>
<td>This report will only contain the subtotals and grand totals listed for the Current Transactions or selected Archived Transactions.</td>
</tr>
<tr>
<td>Detail w/ Auth Summary</td>
<td>This report will contain all of the transactions listed on the Current Transactions page or for a specific date in Archived Transactions, as well as the authorization information for each transaction.</td>
</tr>
<tr>
<td>Detail w/ Audit Trail</td>
<td>This report will contain the audit trail associated with each transaction listed on the Current Transactions page or Archived Transactions page when a specific date is selected. This report contains the raw information that was transmitted to Shift4 Payments’ data center for every request sent related to a transaction. The information is displayed in both ASCII and its hexadecimal representation (for clarity), and may be helpful when troubleshooting.</td>
</tr>
<tr>
<td>Full Detail</td>
<td>This report will contain all details associated with each transaction listed on the Current Transactions page or Archived Transactions page when a specific date is selected. This report combines the detail from the Detail w/ Audit Trail report, which contains the raw information that was transmitted to Shift4 Payments’ data center for every request sent for a transaction, as well as the authorization information.</td>
</tr>
</tbody>
</table>
Exporting Transaction Reports
Transaction reports are also available for export. From the DOLLARS ON THE NET side menu, select Export and then choose the desired report configuration.

Design Considerations
Now that you have performed a token exchange and sent a “Hello, World!” message, it’s time to develop your interface! We’ve provided the following sections to assist you in designing your interface. Please be sure to incorporate the features that apply to your Shift4 Payments Integration Plan.

• Understanding TrueToken
• Utilizing TokenStore
• Requirements for EMV Processing with Shift4 Payments
• Quick Chip EMV
• Supporting “Swipe Ahead” for Faster Transactions at the POS
• Receipt Printing Requirements
• Supporting Built-In Receipt Printing Capability on Devices
• Using UTG4Cloud in API Requests
• Understanding Invoices
• Basic Transaction Flow
• Timeouts and Communication Failures
• Handling Referral Responses
• One and Two Pass Verification
• Offline Processing
• End-of-Day Reporting
• Preventing Double Charges on Amended Transactions
• Partial Authorization
• Split-Tender Transactions
• NFC/Contactless Payments
• Using the Notes Field
• Level 2 Data and Purchasing Card Data
• Using the Surcharge Field
• Capturing Sales
• Gift Card Processing
• Processing Refunds
• Using the Lodging Fields
• Processing EBT Cards
• Understanding the MetaToken Process
• Prompt Confirmation Function
• Supporting Digital Signature Capture
• Using the Terms and Conditions Function
• Using the Process Forms Function
• Supporting the Cash Back Function
• Utilizing On-Demand Card Read
• Sending the Input Prompt Function
• Getting Device Information
• Using Promotional Codes for Private-Label Purchases
• Processing MO/TO from a Retail Location
• Requesting a Totals Report
• Developing an Automated Remote Export from DOLLARS ON THE NET

Warning: Review the PCI DSS and PA-DSS documentation when planning your integration design.

Understanding TrueToken

Current card association regulations specify that a merchant must encrypt all stored payment card numbers. Furthermore, these regulations do not allow the storing of card track information or card security codes (CSCs).

Using Shift4 Payments’ TrueTokenization® eliminates the need for both encryption key management and encryption algorithm maintenance in your interface. Tokenization is not encryption, but works together with our True P2PE® solution to relieve the merchant and the interface from the burden of storing, processing, and transmitting sensitive cardholder data (CHD), greatly reducing the scope of PCI DSS assessments and nearly eliminating the risk of a data breach. More information on TrueTokenization and True P2PE can be found on the Shift4 Payments website.

A TrueToken® is a unique, 16-character value created by DOLLARS ON THE NET to reference CHD. In and of itself, a TrueToken has no meaning and no decipherable relationship to CHD. The first four digits in a TrueToken are the last four digits of a payment card number followed by a random 12-character alphanumeric code.

Whenever the payment card number is used in a request, a TrueToken will be returned in the corresponding response in the UniqueID field. Your interface should be designed to store this TrueToken for future use. The latest TrueToken received should be used in any subsequent request that references the same card data.

WARNING! Sending both the payment card number and a TrueToken in a request will result in an error.

Short-Term Data Storage

Short-term data, including the full track data, CSC and PIN, or P2PE PIN block, is stored with the TrueToken until it is consumed – either when a transaction is processed or if it expires (whichever comes first). The length of time during which short-term data is held is configurable to 24, 48, 72, or a maximum duration of 96 hours for pre-authorization to remain PCI compliant.

TrueToken Expiration

A TrueToken will remain valid for 3 days, 7 days, 14 days, 1 month, 3 months, 6 months, 9 months, 12 months, 18 months, 24 months, or until the card’s expiration date. If a payment card expires prior to the TrueToken’s expiration date, then the TrueToken will expire regardless of the expiration date configured in DOLLARS ON THE NET.
Tip: If you want to determine if a TrueToken’s expiration date exceeds the card’s expiration date before storing a TrueToken, you can support use of the RETURNEXPDATE API Option.

Short-term data storage and the expiration of TrueTokens are configured within the merchant’s DOLLARS ON THE NET account’s Token Store Settings. More information about TokenStore is in the Utilizing TokenStore section of this document.

Utilizing TokenStore

By default, each merchant account has a TokenStore, a secure repository for all of their tokens in Shift4 Payments’ PCI-compliant data centers. Shift4 Payments’ TokenStore allows merchants to process refunds, card-on-file payments, incremental authorizations, and more without relying on actual CHD.

Sharing Tokens between Merchant Accounts

Many enterprise and multi-location merchants want to share tokens across different revenue centers and properties. Shift4 Payments offers two ways to do this: TokenShare and Global TokenStore. There are a few main differences between how TokenShare and Global TokenStore work:

- TokenShare allows merchants to share tokens among accounts. Merchants can use the tokens stored in their own account as well as the tokens stored in other merchants’ accounts when permission has been granted to do so.
- A Global TokenStore is an account that acts as a centralized repository for multiple merchants’ tokens. Its sole purpose is to act as a shared TokenStore for permitted merchant accounts.

Whether you choose to support TokenShare, Global TokenStore, or both depends on your interface requirements and the merchant’s preference. Examples of how your interface can support them are explained in detail below.

How TokenShare Works

To support TokenShare, your interface must store the (correlated) Shift4 Payments serial number of the merchant account where a given token was created. When a merchant needs to use a token that resides in another merchant’s TokenStore, your application will send the token and populate the serial number of the account where the token is located in the TokenSerialNumber field in a request. This will cause DOLLARS ON THE NET to locate that token for use.

An example of this process is given below.

Note: The example provided below is a simplified scenario intended to help illustrate the TokenShare process. Please note that other implementations may work differently depending on the transaction(s) being performed and the merchant’s account setup.

Also, TrueTokens are made up of 16 alphanumeric characters, but letters are used in this example to represent the tokens. The serial numbers used are also simplified for the purpose of this example only.
1. Stores 1 and 2 each have tokens in their respective TokenStore accounts in Shift4 Payments’ data centers.

![Diagram of TokenStores in Stores 1 and 2]

2. A customer comes into Store 1 (serial 123) and uses a credit card ending with 1119 to purchase a pair of shoes. When the transaction is approved, Token C is generated by DOLLARS ON THE NET and stored in Store 1’s TokenStore.

![Diagram of transaction at Store 1]

3. The customer later returns the pair of shoes at Store 2 (serial 124). Store 2 doesn’t have Token C in their TokenStore; therefore, the interface must provide Token C and specify “123” in the TokenSerialNumber field in the return request. This allows DOLLARS ON THE NET to locate Token C in Store 1’s TokenStore to process the return.

![Diagram of return request at Store 2]
4. When Store 2 completes the refund transaction, Token Z is created by DOLLARS ON THE NET and added to Store 2’s TokenStore.

**Important:** If a merchant is processing a transaction with a payment card or using a token from their TokenStore, the transaction will be processed as usual and no TokenSerialNumber needs to be referenced.

---

**How Global TokenStore Works**

To support Global TokenStore, your interface will need to keep track of the Global TokenStore’s serial number for use in certain requests. When a merchant needs to use a token that resides in the Global TokenStore, your interface will send the token and populate the Global TokenStore’s serial number in the TokenSerialNumber field in a request. This will cause DOLLARS ON THE NET to locate the token in the Global TokenStore account for that merchant’s use.

An example of this process is given below.

---

**Note:** The example provided below is a simplified scenario intended to help illustrate the Global TokenStore process. Please note that other implementations may work differently depending on the transaction(s) being performed and the merchant’s account setup.

Also, TrueTokens are made up of 16 alphanumeric characters, but letters are used in this example to represent the tokens. The serial numbers used are also simplified for the purpose of this example only.
1. Store 1 (serial 123) and Store 2 (serial 124) each have tokens in their respective TokenStore accounts in Shift4 Payments’ data centers. The Global TokenStore holds tokens that Store 1 and Store 2 have deposited into it.

2. A customer comes into Store 1 (serial 123) and uses a credit card ending with 1119 to purchase a pair of shoes. When the transaction is approved, Token C is generated by DOLLARS ON THE NET and stored in Store 1’s TokenStore.
3. To make this card available for use by other parts of the business, the interface must deposit Token C, which is stored in Store 1’s TokenStore, into the Global TokenStore account. This is accomplished by sending a TokenStore Duplicate (FRC E2) request along with serial ‘123’ in the TokenSerialNumber field (as that is the current location of the token) using the Access Token for the Global TokenStore account (as that is the account where Token C will be deposited). This causes DOLLARS ON THE NET to create a new token, Token G, in the Global TokenStore.

4. The customer later returns the pair of shoes at Store 2 (serial 124). Store 2 doesn’t have the correct token to process the refund because it is not stored in their TokenStore. Therefore, the interface must provide Token G and specify “789” in the TokenSerialNumber field in the return request. This allows DOLLARS ON THE NET to locate Token G in the Global TokenStore to process the return.
5. When Store 2 completes the refund transaction, Token Z is created by DOLLARS ON THE NET and added to Store 2’s TokenStore.

![Diagram showing the process of creating Token Z](image)

**Important:** If a merchant is processing a transaction with a payment card or using a token from their TokenStore, the transaction will be processed as usual and no TokenSerialNumber needs to be referenced.

---

**TokenStore Functions**

Shift4 Payments has two functions associated with the TokenStore: TokenStore Add (FRC E0) and TokenStore Duplicate (FRC E2).

**TokenStore Add (FRC E0)**

FRC E0 is used to add CHD to a local or Global TokenStore and receive a TrueToken. Common uses of FRC E0 are card-on-file processing or collecting CHD when an order or reservation is placed but not billed right away. FRC E0 is sent when a purchase is not immediately processed and CHD needs to be tokenized for later use. When using FRC E0, short-term data will not be consumed during the tokenization process; instead, the short-term data will remain with the TrueToken for the amount of time configured by the merchant in their Token Store Settings in DOLLARS ON THE NET.

**TokenStore Duplicate (FRC E2)**

FRC E2 is used to generate a new TrueToken using an existing TrueToken. When the FRC E2 request is sent, a new TrueToken will be returned to be used in place of the original token. The FRC E2 request will store short-term data if sent by the interface until it is used or for the period configured in the merchant’s DOLLARS ON THE NET account. When a merchant has a TrueToken that is going to expire and wants to replace it, FRC E2 can also be used to generate a new TrueToken. The FRC E2 request can also be used to deposit a TrueToken into a Global TokenStore.

**Use Case 1: Local TokenStore Add without Card Validation**

**Actors:** Interface, Shift4 Payments

**Preconditions:** CHD has been captured by a merchant and is ready for the interface to send.
Main Flow: This flow illustrates receiving a TrueToken for a card.

1. The interface sends the TokenStore Add request (FRC E0) using the merchant’s Access Token to store CHD locally.
2. Shift4 Payments returns a response with a TrueToken in the UniqueID field.
3. The interface vendor stores the TrueToken in its database instead of CHD for use in subsequent transactions.

Optional Flow: This flow illustrates validating a card before storing a TrueToken.

4. A Verify Card request (FRC 2F) is sent using the resultant TrueToken.
5. A response with no errors is received.
6. The interface interrogates the Response, ValidAVS, and CVV2Valid fields (where applicable), and stores the TrueToken if valid.

Post Conditions: A new TrueToken has been created and stored in the merchant’s TokenStore.

Use Case 2: Global TokenStore Duplicate Using an Existing TrueToken

Actors: Interface, Shift4 Payments

Preconditions: A merchant has a TrueToken that needs to be duplicated in the Global TokenStore after a purchase.

Main Flow: This flow illustrates receiving a TrueToken for a card without checking its validity.

4. The interface sends the TokenStore Duplicate request (FRC E2) using the Global TokenStore’s Access Token.
5. Shift4 Payments returns a response with a new TrueToken in the UniqueID field.
6. The interface vendor stores the new TrueToken in its database (instead of the old TrueToken) for use in subsequent transactions.

Post Conditions: The card data has been added to the Global TokenStore.
Requirements for EMV Processing with Shift4 Payments

EMV has been a complete paradigm shift in the U.S. payments industry. The good news is that Shift4 Payments is handling almost all of the changes to development so that you can quickly and easily support it.

Certifying for EMV with Shift4 Payments requires UTG-controlled P2PE PIN pads that are configured specifically for the Shift4 Payments’ test environment. This requires a Shift4 Payments P2PE test key and, if you plan to certify for debit, any U.S. debit key.

**Important:** The Shift4 Payments P2PE test key is different from the production key and these two types of keys cannot be interchanged.

The PIN pad(s) that you use for testing must be loaded with the required Shift4 Payments test key by the device manufacturer or reseller before certification can begin. You can find key injection facilities (KIFs) that can inject Shift4 Payments test keys for EMV by going to shift4.com/dotn/integration/third-party-devices.cfm?EMV and selecting a device.

**Requirement:** In EMV processing, your application must populate the API Terminal ID that you configured in UTG TuneUp in the TerminalID field to prompt the PIN pad.

Additionally, your application must support the Enhanced Receipts API Option and print both a merchant copy and consumer copy of the receipt for approvals (sale or auth), declines, referrals, errors, refunds, and voids. The Tip line must not be printed on the refund receipt, and the receipt text cannot be parsed or manipulated in any way.

For more information about EMV receipt requirements, please refer to the Receipt Printing Requirements section of this document.

**Note:** Certifying for EMV with Shift4 Payments does not require the use of chip cards because the UTG, which is already certified for EMV with the processors, controls the PIN pad and handles the chip card functionality.
Quick Chip EMV

In typical EMV processing, the consumer’s card must remain inserted into the PIN pad from the moment a transaction is totaled until an approval, decline, or referral response is returned from the card issuer. As a result, consumers have complained that transaction times are much slower with EMV chip cards than they are with magnetic stripe cards.

However, consumers can experience EMV transaction times that are comparable to those processed with magnetic stripe cards with quick chip EMV. Shift4 Payments provides support for quick chip EMV with all of the major card brands, including Visa Quick Chip for EMV, Mastercard M/Chip Fast, Amex Quick Chip, and Discover Quick Chip.

Quick chip EMV allows the consumer to remove their EMV chip card before the transaction is sent online to the card issuer for authorization. This reduces the amount of time the card is required to remain in a terminal, giving consumers a perceived speed increase because they can put their card away earlier than they can with typical EMV processing.

If you are certified with Shift4 Payments for EMV, you don’t need to make any changes to your interface to support this feature because quick chip functionality is completely handled by the UTG.

If you’d like to learn more about how you can enhance the EMV checkout experience with faster transaction times, see the Supporting “Swipe Ahead” for Faster Transactions at the POS section of this document.

Tip: Quick chip simply gives consumers a perceived speed increase as they are able to remove their card while the typical EMV authorization process flow occurs. However, interface designers can certify to support Swipe Ahead, which provides an actual speed increase to EMV transaction times. For more information, see the Supporting “Swipe Ahead” for Faster Transactions at the POS section of this document.
Supporting “Swipe Ahead” for Faster Transactions at the POS

“Swipe Ahead” functionality allows swipe, insert, contactless, or manual payment card entry at the Point of Sale (POS) before the merchant has finished ringing up a transaction.

You can support Swipe Ahead functionality by sending a Swipe Ahead request (FRC 96). This will prompt the consumer to enter their card information by displaying the Insert/Swipe (or Tap, if applicable) screen when the clerk begins scanning items.

**Requirement:** A UTG-controlled PIN pad is required to support FRC 96.

For FRC 96 requests, the Global Timer should be set to 10 minutes to allow the consumer time to enter their payment card data, if desired, prior to the transaction being totaled. If the consumer’s payment data is entered, the “Please wait for cashier screen” will display while the clerk continues to scan items.

Depending on your system or the merchant’s business practices, when the transaction has been totaled, you will send an Online Auth (FRC 1B) or Online Sale (FRC 1D) request to send the transaction online for processing.

**Sending Line Item(s) Requests with Swipe Ahead**

After sending an FRC 96 request, you can send display line item(s) requests (i.e., FRC 92, FRC 95, etc.) to display details about the item(s) being purchased to the consumer. After the consumer’s card data has been entered and the FRC 96 response is returned, the line item(s) requests will be processed in the order received. Therefore, you may choose to wait until the FRC 96 response is received to send line item(s) requests or send line item(s) requests immediately following the FRC 96 request.

**Tip:** If you send line item(s) requests before the FRC 96 response has been received, you should set a timer that is greater than the time currently remaining on the FRC 96 request. This allows time for the UTG to receive and respond to line item(s) requests after the FRC 96 response has been received.

**Clearing the Swipe Ahead Data Cache**

If an FRC 96 request has been sent and a consumer’s payment card data has been entered, a PIN Pad Reset request (FRC 97) can be sent to reset the PIN pad to idle if it hasn’t yet been sent online for processing. FRC 97 may be helpful if there is a need to change the consumer’s payment method, such as if the consumer decides that they want to pay using a different payment card or cash.

In addition to FRC 97, the Swipe Ahead data cache will also be cleared if another FRC 96 request is sent or if the transaction is processed.
**Use Case 3: Sending a “Swipe Ahead” Request before Sale or Authorization**

**Actors:** Interface, Shift4 Payments

**Preconditions:** A merchant is processing a sale.

**Main Flow:** This flow illustrates sending a Swipe Ahead (FRC 96) request prior to a sale or authorization request, allowing the consumer to swipe, insert, or tap their card prior to a transaction being totaled.

1. The clerk has begun scanning items for purchase.
2. The interface sends an FRC 96 request to the UTG, causing the PIN pad to prompt the consumer to insert, swipe, or tap their payment card. If applicable, the payment card entry is followed by a prompt for PIN.
3. The UTG begins processing the transaction and securely holds the data in cache. The transaction is not sent online for processing at this stage. If the card was inserted, the PIN pad prompts the consumer to remove their card.
4. The PIN pad displays a “Please wait for the cashier” message on the PIN pad.
5. The UTG sends the FRC 96 response to the interface.
6. The interface sends an FRC 1B or FRC 1D request with the final transaction amount for processing.
7. The PIN pad prompts the consumer to accept the total amount via the “Amount OK” screen.
8. The PIN pad displays “Authorizing, please wait” while the FRC 1B or FRC 1D request is sent for processing.
9. A response indicating approval is returned and the PIN pad prompts the consumer for signature, if applicable.

**Post Conditions:** An approved transaction awaits settlement at the end of the day.

---

**Tip:** The “Amount OK” screen can be skipped if you send the BYPASSAMOUNTOK API Option or if the “Bypass Amount OK” option was selected in UTG TuneUp when configuring the PIN pad.
Receipt Printing Requirements

Shift4 Payments’ receipt printing requirements are based on the processors’ specifications for both format and content and apply to all certification types. Shift4 Payments helps you to meet these requirements by providing information and determining which items need to be printed in the receipt text.

As part of your certification with Shift4 Payments, we’ll review samples of your receipts to ensure that your interface meets all receipt printing requirements.

**Requirement:** The receipt text returned in the response from Shift4 Payments is generated by the processor and is not controlled by Shift4 Payments; therefore, the interface must be able to handle variable text returned by the processor.

**WARNING!** Receipt text should never be parsed or manipulated in any way.

Presenting Receipts

Your interface will be required to properly handle generating receipts for both the consumer and merchant (consumer only for card-not-present transactions). The presentation method (e.g., digital or on paper) is based on the processor’s requirements, merchant’s policies, and/or consumer’s preference. Detailed information about the requirements for consumer receipts and merchant receipts is provided below.

**Consumer Receipts**

How and when in the payment flow a receipt is presented to the consumer varies based on the merchant’s industry and processing flow. Below are some examples of specific requirements:

- Lodging merchants are required to present receipts to consumers at two points: when the billing terms are printed on the folio at check in and when the final invoice is printed on the folio at check out.
- For card-not-present transactions, such as e-commerce and MO/TO (mail order/telephone order), an invoice that contains the customer-receipt text should be provided to the consumer. Depending on the merchant’s operational practices, this invoice may be provided via email, a statement on an online portal, or printed and shipped with purchased goods.
- Refund receipts should never include a tip line.

**Merchant Receipts**

Your interface must provide copies of receipts for the merchant’s record and use in chargeback defense. Merchant-receipt copies can be generated and stored electronically when a signature is digitally captured. If a signature is not digitally captured for a card-present transaction, a paper record of the consumer’s signature is required.
**Enhanced Receipts**

The processors require merchants processing EMV payments to use enhanced receipts. Enhanced receipts must be printed for approvals (sales or authorizations), declines, referrals, errors, refunds, and voids of previously approved transactions. To support enhanced receipts, your interface must be able to receive separate variable receipt text for both the consumer and merchant.

When you use the ENHANCEDRECEIPTS API Option, the receipt text provided by Shift4 Payments will be returned in two separate fields: CustomerReceiptText and MerchantReceiptText. The printed receipts will display the items sold as well as the receipt text returned by Shift4 Payments.

- Below is an enhanced receipt sample, illustrating the combination of the receipt text Shift4 Payments returns and the information that your interface will provide. The text that your interface is responsible for is at the top, and the shaded information is an example of the receipt text that Shift4 Payments provides.

- The consumer receipt text (shaded left) is returned in the CustomerReceiptText field and the merchant receipt text (shaded right) is returned in the MerchantReceiptText field.

---

<table>
<thead>
<tr>
<th>ABC Retail</th>
<th>ABC Retail</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 Crossing Road</td>
<td>14 Crossing Road</td>
</tr>
<tr>
<td>Las Vegas, NV 89444</td>
<td>Las Vegas, NV 89444</td>
</tr>
<tr>
<td>(702) 555-5555</td>
<td>(702) 555-5555</td>
</tr>
</tbody>
</table>

| BL Notebook x5 | $ 25.00 |
| Chalkboard     | $ 50.00 |
| BLK Messenger Bag | $ 20.00 |

Subtotal $ 95.00

| Total $100.00 |

---

**CREDIT CARD SALE**

02/15/2016 09:47:21

CARD TYPE VISA
ENTRY METHOD KEYED
CARD # XXXXXXXXXXXX1119
INVOICE 0807735372
CLERK 1
APPROVED OK6672

AMOUNT USD $100.00
TOTAL USD $100.00

I AGREE TO PAY ABOVE TOTAL AMOUNT ACCORDING TO CARD ISSUER AGREEMENT IF CREDIT VOUCHER

---

CARDHOLDER COPY

---

MERCHANT COPY

---

**External Use NDA**

© 2018 Shift4 Payments, LLC. All rights reserved.

Version 2.34

Page 56 of 137
Requirement: The receipt text that Shift4 Payments returns must be presented on the same receipt as the transaction details.

Additional API Options, including PRINTTIPLINE, can also be included with the ENHANCEDRECEIPTS API Option as needed to change the configuration of a receipt. For more information on API Options, please refer to the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner.

Formatting Enhanced Receipts

The receipt text that Shift4 Payments returns will be fully justified, with word spacing adjusted so that the text is flush with both margins.

The column width of the receipt text must be specified by the interface in transaction requests using the ReceiptTextColumns field. Acceptable column width values are between 000 and 048 to allow the receipt text to wrap to fit your receipt. To produce readable receipts in production, do not set the column value too small.

The font used for receipt text needs to be monospaced and non-proportional. For HTML presentation, the spaces can be converted to non-breaking spaces to maintain proper text alignment.

General Receipts vs. Enhanced Receipts

The processors only require that merchants processing EMV use enhanced receipts. However, we strongly recommend that all integrations support enhanced receipts for a few reasons.

First of all, your interface will be required to provide less data for enhanced receipts because more data is provided by Shift4 Payments. For example, the table below outlines the information your interface is required to provide on a general receipt compared with an enhanced receipt:

<table>
<thead>
<tr>
<th>General Receipt Items Provided by Your Interface</th>
<th>Enhanced Receipt Items Provided by Your Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Merchant Name, Address, and Phone Number</td>
<td>• Merchant Name, Address, and Phone Number</td>
</tr>
<tr>
<td>• Date and Time of Transaction</td>
<td>• Line-Item Details (Including Amounts)</td>
</tr>
<tr>
<td>• Invoice Number</td>
<td>• Tax Information</td>
</tr>
<tr>
<td>• Line-Item Details (Including Amounts)</td>
<td>• Subtotal and Total</td>
</tr>
<tr>
<td>• Tax Information</td>
<td>• A List of Restrictions for Refunds Related to the Transaction</td>
</tr>
<tr>
<td>• Subtotal and Total</td>
<td>• Receipt Text</td>
</tr>
<tr>
<td>• Last 4 Digits of Card Number</td>
<td>• (If applicable) URL on the Receipt (E-Commerce Only)</td>
</tr>
<tr>
<td>• Fully Spelled-Out Card Type (Except JCB)</td>
<td></td>
</tr>
<tr>
<td>• Transaction Amount</td>
<td></td>
</tr>
<tr>
<td>• Authorization Code</td>
<td></td>
</tr>
<tr>
<td>• A List of Restrictions for Refunds Related to the Transaction</td>
<td></td>
</tr>
<tr>
<td>• Receipt Text</td>
<td></td>
</tr>
<tr>
<td>• (If applicable) URL on the Receipt (E-Commerce Only)</td>
<td></td>
</tr>
</tbody>
</table>
Supporting Built-In Receipt Printing Capability on Devices

If you support devices with built-in receipt printers, you can print receipts by sending a Print Receipt request (FRC F1) along with the text to be printed in the ReceiptText field following a transaction that requires a printed receipt. (For information regarding which transactions require printed receipts, please see the Receipt Printing Requirements section of this document.) This FRC also includes the ability to print scannable bar codes on receipts.

The receipt text will automatically wrap when using a device's built-in receipt printer. However, the following tags may be included in the ReceiptText field to modify how the receipt text will be displayed:

<table>
<thead>
<tr>
<th>Start Tag</th>
<th>End Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;$Cut&gt;</td>
<td></td>
<td>Advances the receipt a few lines so that it can be torn off.</td>
</tr>
<tr>
<td>&lt;$Bold&gt;</td>
<td>&lt;$Bold$&gt;</td>
<td>Bolds the text between two tabs.</td>
</tr>
<tr>
<td>&lt;$Reverse&gt;</td>
<td>&lt;$Reverse$&gt;</td>
<td>Prints white text on a black background.</td>
</tr>
<tr>
<td>&lt;$DoubleChar&gt;</td>
<td>&lt;$DoubleChar$&gt;</td>
<td>Prints double-wide characters.</td>
</tr>
<tr>
<td>&lt;$BigChar&gt;</td>
<td>&lt;$BigChar$&gt;</td>
<td>Prints text in a very large font.</td>
</tr>
</tbody>
</table>

As an example of using these tags to change how the text may be displayed, if you want to print “Groceries” in bold, it would be sent as:

“<$Bold>Groceries<$Bold$>”

Printing Bar Codes

To print scannable bar codes on a receipt, send the desired bar code tags from the list below with the receipt text in the ReceiptText field.

<table>
<thead>
<tr>
<th>Start Tag</th>
<th>End Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;$BarCode&gt;</td>
<td>&lt;$BarCode$&gt;</td>
<td>Prints a bar code from the data between the tags.</td>
</tr>
<tr>
<td>&lt;$BarCode128&gt;</td>
<td></td>
<td>Switches to Code 128 format for bar code printing.</td>
</tr>
<tr>
<td>&lt;$BarCodeEAN8&gt;</td>
<td></td>
<td>Switches to EAN-8 format (7 digits) for bar code printing.</td>
</tr>
<tr>
<td>&lt;$BarCodeEAN13&gt;</td>
<td></td>
<td>Switches to EAN-13 format (12 digits) for bar code printing.</td>
</tr>
<tr>
<td>&lt;$BarCodeHeightNNN&gt;</td>
<td></td>
<td>Configures the height of the bar code in pixels where NNN is the 3-digit value for the pixel width.</td>
</tr>
<tr>
<td>&lt;$BarCodeWidthNNN&gt;</td>
<td></td>
<td>Configures the width of the bar code in pixels where NNN is the 3-digit value for the pixel width.</td>
</tr>
</tbody>
</table>
### Using UTG4Cloud in API Requests

UTG-controlled devices are required for merchants to perform certain functions, including EMV processing, as well as functions like Terms and Conditions requests (FRC CF), Prompt Confirmation requests (FRC 82), and Process Forms requests (FRC 86). Merchants that have client-server-based payment systems where a central or cloud-based application is physically located in a different geographical location than the merchant will likely have difficulty implementing UTG-controlled devices (due to latency over the internet). Shift4 Payments’ UTG4Cloud gives merchants using cloud-based systems a solution for this issue so that they may use UTG-controlled devices.

Using UTG4Cloud with a cloud-based POS or property management system (PMS) requires additional configuration. A minimum of two UTGs are required for a merchant to use UTG4Cloud: a Local UTG and a Central UTG:

- The Local UTG is on the merchant’s property in the same physical location as the PIN pad device(s) and is used solely to control the device(s).
- The Central UTG is typically installed in the merchant’s data center alongside the POS/PMS and processes API requests from the POS/PMS.

Details for configuring UTG4Cloud in production are located in the *Universal Transaction Gateway Quick Installation Guide* located in MyPortal Documentation Corner.

---

**Note:** Some devices apply the bar code settings that were last used. Therefore, we recommend sending the format, including the height, width, and layout configuration (i.e., horizontal or vertical), and alignment for each bar code in the ReceiptText field to ensure that it is printed as desired.

<table>
<thead>
<tr>
<th>Start Tag</th>
<th>End Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;$BarCodeHorizontal&gt;</td>
<td></td>
<td>Switches the bar code to a horizontal layout.</td>
</tr>
<tr>
<td>&lt;$BarCodeVertical&gt;</td>
<td></td>
<td>Switches the bar code to a vertical layout.</td>
</tr>
<tr>
<td>&lt;$BarCodeAlignCenter&gt;</td>
<td></td>
<td>Prints the bar code in the middle of the receipt.</td>
</tr>
<tr>
<td>&lt;$BarCodeAlignLeft&gt;</td>
<td></td>
<td>Prints the bar code on the left side of the receipt.</td>
</tr>
<tr>
<td>&lt;$BarCodeAlignRight&gt;</td>
<td></td>
<td>Prints the bar code on the right side of the receipt.</td>
</tr>
</tbody>
</table>
Diagram: API Requests Using a Cloud-Based POS/PMS and a UTG-Controlled Device

The diagram below is used to illustrate the communication flow for API requests with UTG4Cloud and a UTG-controlled device. In the examples below, “device credentials” refers to the information that identifies a specific Local UTG to DOLLARS ON THE NET (e.g., DeviceService, DeviceGuid, and DeviceExtensions).

Step 1: Acquiring Device Credentials (Green Arrows)

a. An API request is initiated at a property that is using a web browser to access a cloud-based POS/PMS using a UTG-controlled device.

The web browser at the property sends a request for device credentials to the property’s Local UTG (i.e., the UTG4Cloud API Interface).

The property’s Local UTG establishes a connection to DOLLARS ON THE NET.

DOLLARS ON THE NET generates and returns the device credentials to the property’s Local UTG.

The property’s Local UTG sends the device credentials to the property’s web browser.

The property’s web browser sends the device credentials and API Terminal ID to the cloud-based POS/PMS.

Step 2: Processing a Request

The cloud-based POS/PMS sends the API request, including the device credentials and API Terminal ID, to the Central UTG.

The Central UTG sends the API request to DOLLARS ON THE NET.

DOLLARS ON THE NET sends the API request to the property’s Local UTG.

The property’s Local UTG processes the request with the specified UTG-controlled device.

The property’s Local UTG returns the response to DOLLARS ON THE NET.

DOLLARS ON THE NET forwards the response to the Central UTG.

The Central UTG forwards the response to the cloud-based POS/PMS.

The cloud-based POS/PMS forwards the response to the property’s web browser.
**Prompting the UTG-Controlled PIN Pad**

To request device credentials as in step 2 above, an empty request is sent to the Local UTG using the following information, depending on the protocol you’re using:

- **HTTP Interfaces:** Post an empty string to the IP address and port number that you configured in UTG TuneUp for the UTG4Cloud API Interface (default port: 13107).

- **For TCP/IP Interfaces:** Send an empty request including ASCII 02 followed by the Transaction Header Data Block, ASCII 03, and the Longitudinal Redundancy Character to the IP address and port number that you configured in UTG TuneUp for the UTG4Cloud API Interface (default port: 13107):

<table>
<thead>
<tr>
<th>Start Character (ASCII 02)</th>
<th>Transaction Header Data Block</th>
<th>End Character (ASCII 03)</th>
<th>Longitudinal Redundancy Character</th>
</tr>
</thead>
</table>

As displayed in the diagram above, DOLLARS ON THE NET will respond with the device credentials. Your Cloud based POS/PMS interface must be coded to send these device credentials along with the API Terminal ID in a request to the Central UTG when you need to prompt a UTG-controlled PIN pad.

**HTTP Device Credentials Request Examples**

Two HTTP examples using UTG4Cloud are provided below. The first example uses AJAX with CORS, which is the preferred method as it is more secure. The second example uses AJAX with JSONP. The browser will execute an empty AJAX POST to the Local UTG’s UTG4Cloud Interface. The Local UTG will return the device credentials. The AJAX call will gather these values and post them to the POS/PMS server for use in the authorization request.
CORS:

```javascript
var activateDevice = function() {
    $('##dialog-cardPresent').html('<p>Processing on the secure entry device.</p>);
    $.ajax({
        type: "POST",
        dataType: "json",
        crossDomain: true,
        url: "#application.settings.cp_utgURL#",
        timeout: 5000,
        success: function(data, textStatus, jqXHR) {
            $('##form-checkout input[name='DeviceService']').val(data.DeviceService);
            $('##form-checkout input[name='DeviceGuid']').val(data.DeviceGuid);
            $('##form-checkout input[name='DeviceExtensions']').val(data.DeviceExtensions);
            $('##form-checkout').submit();
        },
        error: function(jqXHR, textStatus, errorThrown){
            console.log("Device Activation Failed: "+textStatus);
            $('##dialog-cardPresent').append('<h3 class="red">Device Activation Failed - Please check UTG.</h3>');
        }
    });
};
```

JSONP:

```javascript
var activateDevice = function() {
    $('##dialog-cardPresent').html('<p>Processing on the secure entry device.</p>);
    $.ajax({
        type: "GET",
        dataType: "jsonp",
        url: "#application.settings.cp_utgURL#",
        timeout: 5000,
        success: function(data, textStatus, jqXHR) {
            $('##form-checkout input[name='DeviceService']').val(data.DeviceService);
            $('##form-checkout input[name='DeviceGuid']').val(data.DeviceGuid);
            $('##form-checkout input[name='DeviceExtensions']').val(data.DeviceExtensions);
            $('##form-checkout').submit();
        },
        error: function(jqXHR, textStatus, errorThrown){
            console.log("Device Activation Failed: "+textStatus);
            $('##dialog-cardPresent').append('<h3 class="red">Device Activation Failed - Please check UTG.</h3>');
        }
    });
};
```
Diagram: API Requests Using a Cloud-Based POS/PMS without a UTG-Controlled Device

The diagram below is used to illustrate the communication flow for API requests with UTG4Cloud without a UTG-controlled device.

1. An API request that doesn’t require a UTG-controlled device is initiated at a property that is using a web browser to access a cloud-based POS/PMS.
2. The cloud-based POS/PMS sends the API request to the Central UTG.
3. The Central UTG sends the API request to DOLLARS ON THE NET.
4. DOLLARS ON THE NET processes the request.
5. DOLLARS ON THE NET sends the response to the Central UTG.
6. The Central UTG forwards the response to the cloud-based POS/PMS.
7. The cloud-based POS/PMS forwards the response to the property’s web browser.

Understanding Invoices

It is crucial that an interface handle invoice numbers correctly because Shift4 Payments uses the invoice number as a unique identifier for transactions. The combination of a card number and invoice number create a unique key for identifying transactions on a per-batch basis in DOLLARS ON THE NET. Therefore, invoice numbers must be unique per individual batch and for each transaction flow.

The invoice number is used along with a TrueToken for any subsequent requests relating to the initial request, such as upgrading an authorization to a sale, incremental authorizations, or a void. For subsequent requests, the invoice number from the initial request is used to link to the transaction.

Refunds and credits are not considered subsequent requests, so these transactions require a new invoice number. If the same invoice number is used for a refund as was used for the sale, the consumer could end up with a credit instead of being charged for the sale. For example, if invoice 1225151234 is used for a $100 sale and later that same day, a refund is processed for $25 using the same invoice number, the customer will receive a $25 net credit instead of being charged a net of $75 because the invoice was overwritten before the batch was processed by the merchant.
Important: Shift4 Payments strongly recommends your interface be designed to never reuse an invoice number once a payment flow is complete within an individual batch.

**Incremental Authorizations and Rollbacks (Lodging Only)**

Incremental authorizations are commonly used in lodging where the final amount for the guest’s bill is unknown. While an initial authorization will be made and an invoice number will be assigned to the guest’s bill (i.e., the folio), additional transactions may also be needed, such as if the guest extends their stay or purchases incidentals. These additional transactions, or incremental authorizations, are added to the same invoice number.

At Shift4 Payments, incremental authorizations share the same invoice number and are individually identified by a TranID, which is returned in the TranID field. When the guest checks out, the sale is captured for the final amount, including the initial authorization and all approved incremental authorizations.

Sometimes, incremental authorizations are declined, requiring the interface to void the incremental authorization and return the invoice to a prior state. This is called a transaction rollback. To perform a rollback, the interface will send a Void request (FRC 08) along with the correlated TranID and the associated invoice number. This causes DOLLARS ON THE NET to void the declined authorization and roll the transaction back to the last valid authorization on file.

When a rollback is used to void a non-approved transaction, the void will not be passed to the issuer or processor.
Use Case 4: Processing an Incremental Authorization (Lodging Only)

Actors: Interface, Shift4 Payments, Clerk

Preconditions: The merchant has an approved initial transaction for a guest’s folio.

Main Flow 1: This flow illustrates an incremental authorization request used to increase the amount on an existing authorization.

1. The clerk initiates an authorization transaction through the interface.
2. The interface sends an Online Auth request (FRC 1B) and the TrueToken from the initial transaction to the UTG with an amount greater than the existing authorization.
3. The UTG sends the TrueToken and transaction request to DOLLARS ON THE NET for processing.
4. DOLLARS ON THE NET returns a resultant TrueToken and approval result to the UTG.

Alternate Flow 1: This flow illustrates a referral response. (Main Flow; Step 4.)

1. The clerk sends a Void request (FRC 08) with the TranID that was returned with the decline response to roll back the transaction.
2. The interface sends a Get Voice Center Information request (FRC 22).
3. A response is returned with the data provided to Shift4 Payments in the VoiceMerchantAccount and VoicePhoneNumber fields.
4. The clerk calls the phone number provided and follows the instructions.
5. The Voice Authorization Center approves the transaction and a voice authorization code is provided.
6. The interface sends an Offline Auth request (FRC 05) to add the incremental authorization to the invoice in DOLLARS ON THE NET.

Alternate Flow 2: This flow illustrates the card issuer declining the incremental authorization. (Main Flow; Step 4.)

Follow the steps in Use Case 5: Rolling Back a Transaction.

Post Conditions The authorization transaction is complete and the interface stores the resultant TrueToken to use for a subsequent transaction (e.g., incremental authorization, sale, refund, or void).
Use Case 5: Rolling Back a Transaction

Actors: Interface, Shift4 Payments

Preconditions: A merchant has an approved initial transaction and is voiding a subsequent transaction only.

Main Flow: This flow illustrates rolling back an invoice to a prior state.

1. The merchant attempts an incremental authorization.
2. A decline response is received.
3. The interface sends a Void request (FRC 08) with the TranID that was returned with the decline response.
4. The specified transaction is voided. A response is received with the ErrorIndicator equal to ‘N’.
5. The merchant must obtain a different form of payment, if applicable.

Exception Flow: This flow illustrates an error response.

1. A response is received with the ErrorIndicator equal to ‘Y’.
2. The interface determines the cause of the error condition and responds appropriately.

Post Conditions The transaction is rolled back to the previous state.
Basic Transaction Flow

For Online Auth (FRC 1B) and Online Sale (FRC 1D) transactions, the diagram below outlines a basic transaction flow. For other requests, the ErrorIndicator field can be checked to see if the transaction was processed successfully. The complete definitions for the available Response codes are available in the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner.
Timeouts and Communication Failures

Communication failures can occur at various points in the transaction process and cause the transaction to timeout. This section outlines the amount of time that your interface should allow for transaction responses, the communication error codes your interface should receive, and guidelines for handling communication failures.

Global Timer

Interfaces will need to implement a Global Timer to monitor the time between a request and a response. The Global Timer must be configurable because the values may need to be changed later. Based on the devices supported, the Global Timer must never allow less time than the device-timer setting in UTG TuneUp.

- For transactions with card entry using a UTG-controlled PIN pad, initially set the Global Timer to 120 seconds.
  - For Swipe Ahead requests (FRC 96) only, a timer should be set to 10 minutes.
- For transactions with card entry other than a UTG-controlled PIN pad, initially set the Global Timer to 65 seconds.

When a transaction response exceeds the time limit configured in the Global Timer, the interface will need to treat the timeout as a communication error.

Requirement: An Unknown Status Report that displays transactions where the approval status is unknown must be made available for an auditor’s review. This can either be included in the end-of-day report or as a separate Unknown Status Report. Demonstrating that unresolved communication failures are included in one of these reports is a certification requirement.

Communication Error Codes

The following is a list of communication error codes:

<table>
<thead>
<tr>
<th>Communication Error Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
</tr>
<tr>
<td>9957</td>
</tr>
</tbody>
</table>

If one of these error codes is received, refer to the Basic Transaction Flow and follow the diagram to check the status of the invoice.
When a Communication Failure Occurs

When a communication failure occurs, it is important to determine the status of the transaction in progress to determine the point at which the error occurred. For example, a communication failure does not mean that the card issuer did not receive or approve the transaction; if the failure occurred before the card issuer responded, the merchant could have a valid approval.

To determine the status of the transaction, a Get Invoice Information request (FRC 07) must be sent by the interface. The FRC 07 request will need to include the invoice number of the transaction in the Invoice field and the TrueToken (if available) in the Unique ID field. If the TrueToken is not available, include the invoice number in the Invoice field and the last four digits of the payment card number in the CardNumber field. Follow the Basic Transaction Flow section to handle the transaction correctly. If the FRC 07 request is unsuccessful, the interface can resend the request once before logging the transaction on the unknown status report.

When a communication failure occurs, the interface should complete the following steps:

1. Wait for 1 to 3 seconds (in production environment) or 3 to 5 seconds (in test environment) and send an FRC 07 request to check the invoice status.
2. Handle the FRC 07 response (if received) by logging the transaction in the Unknown Status Report for an auditor’s review if any of the following are true:
   a. The response code is ‘e’ and the error code in the Authorization field indicates a communication error.
   b. The transaction status still cannot be determined from the FRC 07 response.
   c. The Response field is blank.
3. If the FRC 07 fails, one retry can be attempted before logging the transaction in the Unknown Status Report.

WARNING! If a transaction fails, do not send a Void request (FRC 08). Your interface must log the error condition for the merchant’s review.

Handling an ‘e’ Response Value Returned in an FRC 07 Response

When an ‘e’ response is returned on an FRC 07, an error exists in the transaction being looked up. Check the Authorization field to determine which error code has been returned. Based on the code returned in the Authorization field, the transaction should be either logged or voided. Refer to the Basic Transaction Flow section to review the Authorization field codes and determine the appropriate action.

If the FRC 07 returns ‘Y’ in the Error Indicator field, this does not mean that there was an error with the transaction. Instead, it indicates that there is an error with the FRC 07 itself. Refer to the Basic Transaction Flow section to determine the type of error being returned for the request and how to handle it.
**When a Gift Card Balance Loading Communication Failure Occurs:**

When a communication failure occurs while loading a balance on a gift card, send a Balance Inquiry request (FRC 61) to check the balance.

- If the Global Timer expires before receiving a response or if the interface receives a communication error from the UTG, log the transaction for an auditor’s review.
- If you receive the FRC 61 response and the correct balance (prior balance plus requested load amount) is not reflected, log the transaction for an auditor’s review and adjustment.

---

**WARNING!** Never send a Void request (FRC 08) for communication errors related to gift cards. Your interface must log the error condition for the merchant’s review.

---

**Logging Communication Failures for Auditor Review**

When unresolved exceptions, such as timeouts and communication errors, occur during the processing flow, a transaction must be logged for an auditor’s review. Logging the transaction allows the merchant’s auditor to review the batch of transactions and fix potential issues, such as double charges.

**Handling Referral Responses**

A response code of ‘R’ indicates the card issuer is unable to approve a transaction without processing a voice referral. Voice referrals are commonly caused when an issuer has concerns over fraudulent activity. Voice referrals can also happen when a merchant is processing in offline mode.

The way your interface responds to a voice referral is based on the merchant’s business practices and industry. The interface will either need to void the transaction or process the transaction using an authorization code (provided by the card-issuing bank over the phone). An e-commerce interface will automatically void referral responses because there is no clerk to call the card issuer. Interfaces for other industries may need to add the ability to follow Shift4 Payments’ voice referral flow (detailed below). A best practice in interface design is to make referral handling configurable by dollar amount and merchant preference.

**Authorization Codes**

A successful voice referral will result in a card issuer providing a voice authorization code that the clerk will enter into the interface. Your interface will then pass the authorization code to DOLLARS ON THE NET using the Offline Auth request (FRC 05) or Offline Sale request (FRC 06).
Invalid Authorization Codes

Historically, merchants have experienced issues with clerks entering invalid authorization codes. This can occur when a clerk mistypes the code given by the voice authorization number or makes up a code to avoid having to call the card issuer. Invalid authorization codes can cause merchants to lose revenue when a purchase cannot be authorized by the card issuer or a chargeback occurs. In addition to lost revenue, merchants can also be fined by the processor when an invalid authorization code is entered.

Valid authorization codes are six alphanumeric characters (no special characters); your interface should verify that the code entered meets this criteria. Shift4 Payments also requires a warning message be displayed on the screen where the clerk enters the authorization code. This message will need to warn the clerk that the authorization code entered must match the one provided by the issuer exactly.

Use Case 6: Handling Voice Referrals

Actors: Clerk, Consumer, Interface, Shift4 Payments, Voice Authorization Center

Preconditions: A transaction request has received a referral response.

The merchant is not e-commerce, where referral requests are automatically voided.

Main Flow: This flow illustrates successfully obtaining a voice authorization.

1. The clerk and consumer choose to continue the transaction after receiving an ‘R’ response.
2. The interface sends a Get Voice Center Information request (FRC 22).
3. A response is returned with the data provided to Shift4 Payments in the VoiceMerchantAccount and VoicePhoneNumber fields.
4. The interface displays the information from the response for the clerk’s use.
5. The clerk calls the phone number provided and follows the instructions.
6. The Voice Authorization Center approves the transaction and a voice authorization code is provided.
7. The interface displays a field in which the clerk will enter the voice authorization code.
8. The clerk enters and submits the voice authorization code.
9. The interface will send an Offline Auth request (FRC 05) or Offline Sale request (FRC 06).

Alternate Flow 1: This flow illustrates the merchant or consumer cancelling the transaction prior to attempting to get a voice authorization code. (Main Flow; Step 1.)

Follow steps in Use Case 13: Voiding an Invoice.

Alternate Flow 2: This flow illustrates the card issuer declining the transaction. (Main Flow; Step 6.)

Follow steps in Use Case 13: Voiding an Invoice.

Exception Flow 1: This flow illustrates a failure attempting to get the voice referral data.

3. The FRC 22 request returns an error code.
4. The clerk refers to the merchant’s backup voice referral procedures.
5. Return to Main Flow; Step 5.

Post Conditions: The voice authorization code is recorded for approved transaction.
AVS/CSC Verification

Address Verification System (AVS) and Card Security Code (CSC) verification are two separate verification systems that are often used together when processing card-not-present transactions and manually entered cards in card-present scenarios.

AVS and CSC data may be collected through prompts on a PIN pad, entered into the interface by the clerk, or entered into an e-commerce web page by the consumer. The AVS and CSC data is collected and sent with the initial authorization or sale request, but it is not sent in any subsequent requests for a given invoice. If AVS or CSC verification fails on an initial request, Shift4 will not reattempt verification if new data is provided in a subsequent request. Therefore, to retry AVS and CSC verification with new data, the initial transaction must be voided and a new transaction must be started.

Importantly, AVS and CSC verification are completed separately from transaction authorization. This means that even though a bank may approve a transaction, the AVS or CSC verification may fail. A merchant can also have their account set to return a declined response whenever AVS or CSC is invalid.

AVS Validation

Processors use AVS data, including the cardholder name, street number and/or postal code, to validate the identity of the consumer against the information on file with the card issuer. AVS validation helps a merchant achieve better interchange and/or discount rates.

One and Two Pass Verification

Understanding One and Two Pass Verification

There are two methods to verify the cardholder’s AVS information:

- One Pass Verification
- Two Pass Verification

One Pass Verification

With One Pass Verification, Shift4 goes out to the processor and requests the full amount at the same time the AVS/CSC is being verified. If the AVS or CSC is invalid, then the full amount is put on authorization hold. A void (FRC 08) must be performed to release the authorization hold on the customer’s card. Depending on the amount requested, the cardholder’s Open to Buy balance can be significantly affected. The balance on the customer card will be reduced by the amount being held as an authorization. As a result, the customer may not be able to make purchases elsewhere until the authorization hold is voided.

If the authorization hold is for a large enough amount, the cardholder may be declined for insufficient funds.
Two Pass Verification

With Two Pass Verification Shift4 first goes out to the processor to verify the AVS and CSC using a $0 or $1 authorization. If the AVS and CSC are valid then Shift4 goes out to the processor to authorize the full amount requested. If the AVS or CSC is invalid then an ‘f’ code is returned in the Response field, and the transaction will need to be automatically voided.

The major advantage of Two Pass Verification is evident when a Two Pass verification fails, the full amount requested is not subject to an authorization hold. Despite financial institutions supporting real-time reversals, it may still take several hours or a day for the authorization hold on the funds to be removed. Since the authorization is for $1 or $0, the potential impact to the customer’s Open to Buy balance is minimized.

How One and Two Pass Verification Works at the Merchant Level

When a merchant account is created, Shift4 Payments staff will set a dollar threshold that controls whether One Pass or Two Pass is utilized. Transactions for an amount at or below the threshold will use One Pass Verification, while transactions for an amount higher than the threshold will use Two Pass. For example, if the threshold is set at $100.00, then transactions for an amount at or below $100.00 will use One Pass Verification, while transactions for an amount higher than $100.00 will use Two Pass Verification. As another example, if the threshold is set at $0, then all transactions will use Two Pass. If the threshold is set to $999,999,999,999.99, then all transactions will use One Pass. The merchant should let Shift4 Payments staff know what value best suits their business practices.

Offline Processing

If a merchant connects with Shift4 Payments via direct server-to-server POST and they lose their internet connection, they will be unable to process transactions. However, the UTG has a few capabilities that make it simple for your interface to support offline transaction processing when a merchant’s internet connection goes down.

Offline mode mimics online transaction processing by locally tokenizing CHD and returning an ‘R’ (referral) response when a card is swiped, tapped, or manually entered. While offline, the merchant can use their backup voice referral procedures to obtain a referral code for all of the transactions that they process (since an FRC 22 Get Voice Center Information request won’t work offline). For added security, the UTG will hold the tokenized CHD in an encrypted file until connectivity is reestablished.

Shift4 Payments’ Secure Offline Stand-In® (SOS) feature provides advanced offline processing by allowing a merchant to set a threshold (a maximum dollar value) below which they are comfortable issuing automatic “stand-in” authorizations without requesting voice authorizations. All transactions above this amount will prompt for voice authorization.

WARNING! Not all transactions processed using SOS may be approved once the UTG comes back online. Also, transactions processed using SOS will not qualify as EMV transactions in a card-present environment. However, the benefits of being able to accept transactions while offline may be worth the risk for some merchant environments.

The UTG creates a special authorization code range for offline transactions. All Stand-In authorization codes are formatted as “44**44”, where “**” is a wildcard. Once the merchant’s internet connectivity is restored, the UTG will transmit the transaction data and acquire authorizations (if required) and normal processing can continue.
During the certification process, we will run a few tests to make sure that you can support offline processing. You can learn how to configure offline mode and SOS by following the steps in the Configuring Offline Mode section of the Universal Transaction Gateway: Quick Installation Guide located in MyPortal Documentation Corner. Your API analyst will tell you how to set your UTG Stand Alone to test in offline mode when you’re ready to do so.

**End-of-Day Reporting**

Shift4 Payments requires that your interface provide an end-of-day report to enable a merchant’s auditor to compare the day’s transactions between your interface and DOLLARS ON THE NET prior to settlement. Pre-settlement auditing is a crucial step that allows a merchant to manage their business and monitor transactions for issues like employee fraud, double charges, and other potential problems prior to settlement.

When a transaction is logged for an auditor’s review, the end-of-day report will display those logged transactions. The reports must include the following items:

- Grand total for all card types
- Subtotal by card type
- The ability to display individual transaction details and group those transactions by card type
- The Shift4 Payments invoice number
- The last 4 digits of each card number

---

**Requirement**: An Unknown Status Report that displays transactions where the approval status is unknown must be made available for an auditor’s review. This can either be included in the end-of-day report or as a separate Unknown Status Report. Demonstrating that unresolved communication failures are included in one of these reports is a certification requirement.

Below is an example of a report that a merchant can generate in DOLLARS ON THE NET that can be used to reconcile transactions against your application’s end of day report.

<table>
<thead>
<tr>
<th>Invoice</th>
<th>Clerk</th>
<th>Type</th>
<th>Business Date</th>
<th>Card Number</th>
<th>Amount</th>
<th>Tran</th>
<th>Entry</th>
<th>Customer Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0708500051</td>
<td>00001</td>
<td>VS</td>
<td>Mon 10/03/2016</td>
<td>432100000001119</td>
<td>$100.00</td>
<td>S</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>0708500053</td>
<td>00001</td>
<td>VS</td>
<td>Mon 10/03/2016</td>
<td>486100000002257</td>
<td>$75.43</td>
<td>S</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>0708500054</td>
<td>00001</td>
<td>MC</td>
<td>Mon 10/03/2016</td>
<td>543200000003332</td>
<td>$254.01</td>
<td>S</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>0708500055</td>
<td>00001</td>
<td>MC</td>
<td>Mon 10/03/2016</td>
<td>525000000001111</td>
<td>$25.54</td>
<td>S</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>0708500056</td>
<td>00001</td>
<td>MC</td>
<td>Mon 10/03/2016</td>
<td>222100000000009</td>
<td>$50.00</td>
<td>S</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>0708500057</td>
<td>00001</td>
<td>AX</td>
<td>Mon 10/03/2016</td>
<td>375400000002221</td>
<td>$100.00</td>
<td>S</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>0708500058</td>
<td>00001</td>
<td>NS</td>
<td>Mon 10/03/2016</td>
<td>601100000004444</td>
<td>$100.00</td>
<td>S</td>
<td>M</td>
<td></td>
</tr>
</tbody>
</table>

**DAN IYC HOTEL (76000)**

<table>
<thead>
<tr>
<th>Type</th>
<th>Sales</th>
<th>Refunds</th>
<th>Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>AX</td>
<td>$100.00</td>
<td>0</td>
<td>$100.00</td>
</tr>
<tr>
<td>MC</td>
<td>$328.55</td>
<td>0</td>
<td>$328.55</td>
</tr>
<tr>
<td>NS</td>
<td>$100.00</td>
<td>0</td>
<td>$100.00</td>
</tr>
<tr>
<td>VS</td>
<td>$175.43</td>
<td>0</td>
<td>$175.43</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Count</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>7</td>
</tr>
</tbody>
</table>
Preventing Double Charges on Amended Transactions

There are certain circumstances that require a sales transaction to be amended by the merchant. When an Online Sale (FRC 1D) request is submitted, DOLLARS ON THE NET searches the current batch for the invoice number specified. If the specified invoice is not found, a new transaction with the specified invoice number is created. For Food and Beverage (F&B) merchants, this may create potential double charge scenarios when a transaction has been marked as settled or captured, and a tip adjustment is made for the transaction after batch closing.

For example, if an F&B authorization contains a charge for $10 and another for a $2 tip, the consumer should expect to be charged $12. Because there are no existing invoices to assign the charges, duplicate invoices may be created for both during settlement. Another $12 is charged ($10 plus $2 tip), which is approved for the whole amount of $12. Instead of being charged $12, the consumer is charged $22 (a $10 initial charge plus the $12 tip adjust request equals $22).

The INVMUSTEXIST API Option prevents this situation from occurring by triggering an error when no invoice is found instead of automatically creating a new invoice. Your interface should include this API Option for sales only F&B transactions.

---

**Note:** This API option is for an Online Sale (FRC 1D) transaction only.

---

Partial Authorization

The card brands require certain industries to support partial authorization, or the ability to approve a transaction for less than the amount requested. Instead of a decline, the card issuer returns an authorization for the available funds. The merchant will then request an additional form of payment for the remaining balance.

While partial authorization is required in some industries, it is optional in others, such as e-commerce and lodging. This is based on the merchant category code (MCC), which is assigned by the merchant’s processor. Some merchants choose to support partial authorization when it’s not required to prevent consumer dissatisfaction and potential lost revenue.

To enable merchants to accept partial authorization, your interface must include the ALLOWPARTIALAUTH API Option in authorization and sale requests. When doing so, your interface will need to interrogate the amount in the PrimaryAmount field in the transaction response (comparing it to the amount requested) to determine if a transaction is partially approved. Your interface must be able to capture the remaining balance with a new invoice number and as a new transaction.
Use Case 7: Partial Authorization

Actors: Interface, Clerk, Consumer

Preconditions: A merchant supports partial authorization.

Main Flow: This flow illustrates a partial authorization response, resulting in a split-tender transaction using another payment card.

1. An Online Sale request (FRC 1D) is sent for the full transaction amount. The request must include the API Options ALLDATA, ALLOWPARTIALAUTH and ENHANCEDRECEIPTS.
2. Response Code ‘A’ is received with no error codes and both merchant and consumer receipts are generated. The amount displayed in the PrimaryAmount field is for less than the full transaction amount.
3. The clerk informs the consumer that the transaction was partially approved and requests another form of payment for the remaining amount.
4. The consumer provides another payment card.
5. An Online Sale request (FRC 1D) is sent for the remaining transaction amount using a new invoice number. The request must include the API Options ALLDATA, ALLOWPARTIALAUTH and ENHANCEDRECEIPTS.
6. Response Code ‘A’ is received with no error codes, and the PrimaryAmount field is equal to the remaining transaction amount.
7. Receipts are generated.

Alternate Flow 1: This flow illustrates a partial authorization resulting in a split-tender transaction using cash.

4. The consumer provides cash.
5. The clerk follows the merchant’s procedures for processing cash transactions.
6. Both merchant and consumer receipts are generated.

Exception Flow 1: This flow illustrates a partial authorization resulting in a void.

4. The consumer cannot provide another form of payment.
5. Follow steps in Use Case 13: Voiding an Invoice.

Exception Flow 2: This flow illustrates a partial authorization when services that have already been rendered and another form of payment cannot be obtained.

4. The clerk has captured a partial payment for services, but the consumer cannot provide another form of payment.
5. The clerk follows the merchant’s procedures for notifying the auditor of the loss.

Post Conditions: An approved transaction awaits settlement at the end of the day.
Split-Tender Transactions

Split-tender transactions are when a payment is split between multiple tenders at the request of the consumer(s). For example, after a group of five friends eat dinner together, they may prefer to split the bill. Split tender allows merchants to accommodate these requests by processing multiple transactions to make up the full amount of the bill.

To process a split-tender transaction, each payment should be treated as an individual transaction request with a unique invoice number.

NFC/Contactless Payments

Shift4 Payments allows merchants to accept contactless payments, such as Apple Pay®, Samsung Pay®, and Android Pay®. The ability to accept contactless payments is configurable by the merchant in UTG TuneUp. A UTG-controlled device equipped with NFC/contactless technology is required.

To support contactless payments, your interface will need to be able to accept a CardEntryMode of ‘R’ in a response.

Shift4 Payments only supports the tap-and-pay functionality of mobile wallet applications and does not currently support in-app purchases.

---

Note: Shift4 Payments’ demo environment doesn’t support testing NFC/contactless transactions. Because our UTG is already certified with the processors to support NFC/contactless payments and its device control capability manages these payment flows, testing NFC/contactless functionality is unnecessary.

---

Using the Notes Field

Shift4 Payments has a Notes field that’s displayed within an individual transaction record in DOLLARS ON THE NET. This is a free-form field that supports the use of HTML tags. The Notes field is primarily used for the merchant’s reference and is not passed on to the processor.

The merchant’s auditor can use the Notes field to review data about a transaction, such as to determine the cause of a problem transaction or for chargeback defense. Depending on your interface design and/or the merchant’s desired use of the Notes field, it can contain data such as the workstation used, items purchased, clerk notes, order number, or receipt details.

In a payment flow where multiple requests are sent, the Notes field will be overwritten with the value sent in the last request. If you plan to support the use of the Notes field, Shift4 Payments will test to confirm that your interface can properly pass notes using HTML formatting.
Level 2 Data and Purchasing Card Data

Level 2 data and purchasing card data provide enhanced transaction information as compared to the basic (i.e., level 1) data included in the Standard Transaction Data Block. Level 2 data includes the CustomerReference, TaxIndicator, TaxAmount, and DestinationZIPCode for a given transaction. Purchasing card data includes a text description of the items or services purchased.

By providing level 2 data and purchasing card data to the processor, merchants can improve their discount and interchange rates. Because this information is available to auditors in DOLLARS ON THE NET, it can also be very useful for chargeback defense.

Historically, the payment card industry has used enhanced transaction data specifically for corporate and government purchasing cards. However, it is now common for card issuers to provide enhanced transaction data for all payment cards to help prevent cardholders from requesting unnecessary chargebacks.

**Requirement:** To help merchants improve their chargeback defense and achieve the best discount and interchange rates, Shift4 Payments requires level 2 data and purchasing card data for all certifications.

Using Level 2 Data and Purchasing Card Data in an API Call

Level 2 data and purchasing card data are commonly used in a sale request. The table below describes the fields and the values to include in them.

To format the API call, visit the FRC 1D – Online Sale section of the Shift4 Payments Integration: HTTP POST Complete Functions Guide or the 008 Level 2 and 009 Purchasing Card Data sections in the Shift4 Payments Integration: TCP/IP Complete Functions Guide located in MyPortal API Corner. For detailed definitions, please refer to the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner.

<table>
<thead>
<tr>
<th>Field</th>
<th>Bytes</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomerReference</td>
<td>25</td>
<td>A unique identifier of the consumer or transaction. In lodging transactions, this may be unique transaction details, such as a reservation code or third-party booking source.</td>
</tr>
<tr>
<td>TaxIndicator</td>
<td>1</td>
<td>‘Y’ or ‘N’ to indicate whether sales tax was included in the transaction amount.</td>
</tr>
<tr>
<td>TaxAmount</td>
<td>14</td>
<td>The amount of sales tax included in the transaction. The taxes charged are included in the value specified in the PrimaryAmount field. If no tax was charged, ‘0’ should be sent because some processors do not use the TaxIndicator field. The tax amount is used by businesses to track tax expenses (for accounting purposes). Identifying the tax amount also helps consumers understand the total amount they are billed.</td>
</tr>
</tbody>
</table>
### Using the Surcharge Field

Sometimes merchants charge an extra fee when a consumer wants to use a credit or debit card to make a purchase. This helps merchants recoup some of the costs for processing credit and debit card transactions.

When a merchant adds a surcharge to a transaction, the processors require that they use the Surcharge field to identify the surcharge amount to the processors. This helps the merchant avoid experiencing unnecessary chargebacks from consumers who may not remember that an extra fee was included in their final transaction amount.

To use the Surcharge field, your interface should populate the Surcharge field and increase the amount in the PrimaryAmount field accordingly (i.e., Surcharge+PrimaryAmount). For example, if a transaction at an antique store is $100 and there’s a $5 surcharge for the transaction, the POS application will send $5 in the Surcharge field and $105 in the PrimaryAmount field.

### Capturing Sales

There are two common flows used to capture sales: “Authorization and Capture” or “Straight Capture.” The capture method used depends on the merchant’s business model and/or your interface’s design.

#### Authorization and Capture

An Authorization and Capture flow is used when the final amount of the transaction is unknown at the time of authorization or when a merchant needs to guarantee funds before completing a transaction. Below are some common examples of Authorization and Capture scenarios:

- A guest checks in to a hotel and the final amount of the bill is unknown because the length of stay may change and incidental charges may occur. When the guest checks out, the sale is captured for the final amount.
- A consumer makes an e-commerce purchase for an item that has yet to be shipped. Due to industry regulations, the merchant will authorize the transaction amount to guarantee the funds for the purchase and then capture the sale when it’s prepared for shipment.
- A consumer makes a purchase at an F&B establishment and the tip is unknown, so the merchant will authorize the check before requesting a tip and capturing the sale.

There are two basic steps to an Authorization and Capture flow prior to settlement:
• An Online Auth request (FRC 1B) is sent to DOLLARS ON THE NET to request a hold be placed on the cardholder’s funds.
• When appropriate, an Online Sale request (FRC 1D) captures the transaction. This can happen one of two ways: Trickle Flow or Batch Flow:
  a. In a Trickle Flow, each sale is captured by the interface on an individual basis after authorization, such as when an item is shipped, a rental car is returned, or a hotel guest checks out.
  b. In a Batch Flow, the sales are captured in groups. This is usually done at the end of the business day or at shift changes. This is sometimes performed by a centralized auditing account or corporate location instead of the interface.

Note: Not every card type supports an Authorization and Capture flow. For example, debit authorizations are not supported for an Authorization and Capture flow because the funds are immediately captured.

Straight Capture

A Straight Capture flow is used when a merchant knows the final amount of the sale at the time of the request. Below are some common examples of Straight Capture scenarios:

• A consumer makes a purchase at a retail location and leaves the store with their merchandise (e.g., a shoe store).
• A consumer makes an e-commerce purchase, such as a software download, that is immediately fulfilled.
• A consumer makes a purchase at an F&B establishment where a tip is known in advance, the PIN pad will prompt for a tip, or no tip is expected (e.g., fast food).

In a Straight Capture flow, an FRC 1D immediately captures the transaction and gets approval for it. At the time of approval, the transaction is ready for settlement.

Settlement

Settlement is the last step to complete before a sale is finalized with the banks. Settlement is often completed by an auditor in DOLLARS ON THE NET. When a merchant’s batch is closed, all of the captured transactions completed for the date(s) selected are sent to the processor. This triggers the movement of funds from the cardholders’ banks to the merchant’s bank account as well as the calculation of processing fees.

Use Cases for Capturing Sales

The following use cases demonstrate capturing sales. For incremental authorizations, please see the Incremental Authorizations and Rollbacks (Lodging Only) section in this document.
Note: The use cases provided below are examples and do not replace the flows outlined in your Shift4 Payments Integration Plan.

Also, by including API Options in a request, many of the flows illustrated below will be altered; for example, the ability to manually enter a card can be forced or restricted.

API Options are described throughout this document and in the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner.

WARNING! To improve card data security, Shift4 Payments strongly recommends that all merchants use a P2PE device for all card-present transactions.

Use Case 8: Authorization
Actors: Consumer, Merchant
Preconditions: A merchant is ready to send an Online Auth request (FRC 1B) for a transaction that will later be captured as an Online Sale request (FRC 1D).
Main Flow: This flow illustrates an approval response.
   1. The consumer’s card is captured or a previously stored TrueToken is retrieved.
   2. An FRC 1B is sent.
   3. The request is sent to the processor.
   4. A response indicating approval is returned.
Alternate Flow 1: This flow illustrates a decline response. (Main Flow; Step 4.) Follow the steps in Use Case 13: Voiding an Invoice.
Alternate Flow 2: This flow illustrates a referral response. (Main Flow; Step 4.) Follow the steps in Use Case 6: Handling Voice Referrals.
Post Conditions: An approved authorization awaits settlement at the end of the day.

Use Case 9: Authorization with a UTG-Controlled PIN Pad
Actors: Consumer, Merchant
Preconditions: A merchant is ready to process an Online Auth request (FRC 1B) for a payment using a UTG-controlled PIN pad. The transaction will later be captured as an Online Sale request (FRC 1D).
Main Flow: This flow illustrates an approval response.
   1. An FRC 1B is sent.
2. The PIN pad prompts the consumer to insert, swipe, tap, or key the card and walks the consumer through the steps to complete the transaction.

3. The request is sent to the processor.

4. A response indicating approval is returned.

Alternate Flow 1: This flow illustrates a decline response. (Main Flow; Step 4.)
Follow the steps in Use Case 13: Voiding an Invoice.

Alternate Flow 2: This flow illustrates a referral response. (Main Flow; Step 4.)
Follow the steps in Use Case 6: Handling Voice Referrals.

Post Conditions: An approved authorization awaits settlement at the end of the day.

Use Case 10: Capturing a Previous Authorization
Actors: Consumer, Merchant
Preconditions: A merchant is ready to capture a sale for a previously authorized amount.
Main Flow: This flow illustrates an approval response.

1. An FRC 1D is sent with the invoice number for the previously authorized amount. The previously stored TrueToken is specified in the UniqueID field.

2. The invoice is updated in DOLLARS ON THE NET.

3. A response indicating an approval is returned.

Post Conditions: An approved transaction awaits settlement.

Use Case 11: Straight Capture
Actors: Consumer, Merchant
Preconditions: A merchant is ready to process an Online Sale request (FRC 1D) for a retail purchase, straight F&B purchase, or an e-commerce transaction that will be immediately fulfilled.
Main Flow: This flow illustrates an approval response.

1. The consumer’s card is swiped or keyed, or a previously stored TrueToken is retrieved.

2. An FRC 1D request is sent.

3. A response indicating approval is returned.

Alternate Flow 1: This flow illustrates a decline response. (Main Flow; Step 3.)
Follow the steps in Use Case 13: Voiding an Invoice.

Alternate Flow 2: This flow illustrates a referral response. (Main Flow; Step 3.)
Follow the steps in Use Case 6: Handling Voice Referrals.

Post Conditions: An approved transaction awaits settlement.
Use Case 12: Straight Capture with a UTG-Controlled PIN Pad
Actors: Consumer, Merchant
Preconditions: A merchant is ready to capture a straight sale.
Main Flow: This flow illustrates an approval response.

1. An FRC 1D is sent.
2. The PIN pad prompts the consumer to insert, swipe, tap, or key the card and walks the consumer through the steps to complete the transaction.
3. The request is sent to the processor.
4. A response indicating an approval is returned.

Alternate Flow 1: This flow illustrates a decline response. (Main Flow; Step 5.)
Follow the steps in Use Case 13: Voiding an Invoice.

Alternate Flow 2: This flow illustrates a referral response. (Main Flow; Step 5.)
Follow the steps in Use Case 6: Handling Voice Referrals.

Post Conditions: An approved transaction awaits settlement.

Use Case 13: Voiding an Invoice
Actors: Interface, Shift4 Payments
Preconditions: A merchant has an invoice that needs to be voided.
Main Flow: This flow illustrates voiding an entire invoice.

1. The interface sends a Void request (FRC 08).
2. The entire invoice is voided in DOLLARS ON THE NET.
3. A response is received with the ErrorIndicator equal to ‘N’.

Exception Flow: This flow illustrates an error response. (Main Flow; Step 3.)
Follow the Basic Transaction Flow to determine the proper resolution.

Post Conditions: The requested transaction has been successfully voided.
Gift Card Processing

Shift4 Payments has made building gift card functionality into your interface simple. The FRCs and API Options used for IYC, Shift4 Payments’ processor-neutral gift card program, are listed below. These functions and options are the same as for third-party gift card solutions.

Note: Administrative access to DOLLARS ON THE NET is required to create test gift card numbers. Your assigned API Analyst will provide you with a set of IYC gift card numbers specifically for testing.

Purchasing Gift Cards

Purchases of gift cards are handled by collecting the funds from the consumer and then using an Activate/Reload request (FRC 24) to enable the card (and/or add funds).

Redeeming Gift Cards

Authorizations and sales work the same for gift cards as they do for other card types. These functions are explained in detail in the Capturing Sales section of this document. There are also a few functions specific to processing gift cards, which are described below.

Issuing Refunds

The process flow for making a refund to a gift card varies based on the source of the funds being returned. If a refund is being issued to the same IYC gift card that was used for a purchase, it will be processed using an Online Sale request (FRC 1D) with the SaleFlag field set to ‘C’ for credit. If a refund is being issued to a different IYC gift card than the one that was used for the original purchase, it will be processed using an FRC 24. The correct use of FRC 1D and FRC 24 in gift card returns helps ensure that reporting, accounting, and escheatment requirements are met. To learn about these rules and regulations, read Shift4 Payments’ How to Get the Most Out of a Gift Card Program: Gift Card Best Practices white paper.

Supporting Split Tender and Partial Authorization

Merchants are required to support partial authorization when processing gift cards so that if a consumer cannot pay for the total purchase amount with a gift card, they will be able to pay for the remaining balance with another form of payment. This requires your interface to support both split-tender functionality and the API Option ALLOWPARTIALAUTH. For more information, visit the Partial Authorization and Split-Tender Transactions sections of this document.

Providing a Gift Card Balance

The IYCBalance and IYCAvailableBalance fields both indicate the funds available on a gift card, and will be returned in response to an Inquiry (FRC 61), Online Auth (FRC 1B), or Online Sale (FRC 1D) request. The IYCBalance field shows the total balance on the card including pending transactions. The IYCAvailableBalance field will show the available balance of a card less the amount of pending transactions.
**IYC API Options**

The API Options GCCASHOUT, IYCRECHARGE, IYCACTIVEONLY, and IYYDEACTIVEONLY are used with IYC functions along with the commonly required API Option ALLDATA. These API Options are defined in the *Shift4 Payments Integration: Reference Guide* located in MyPortal API Corner. The API Option ENHANCEDRECEIPTS is not used with IYC functions.

**IYC Gift Card States**

There are three states for IYC gift cards:

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Indicates a card that has been activated. The card may or may not have a balance.</td>
</tr>
<tr>
<td>Inactive</td>
<td>Indicates a card that has not yet been activated by a merchant. Balances can be preloaded to an inactive card, but Shift4 Payments recommends against this for security reasons.</td>
</tr>
<tr>
<td>Deactivated</td>
<td>Indicates a card that was intentionally deactivated by a merchant. If supported by the processor, any remaining funds will be left on the card after deactivation.</td>
</tr>
</tbody>
</table>

**IYC FRCs**

There are four FRCs used specifically for IYC process flows:

<table>
<thead>
<tr>
<th>FRC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRC 24</td>
<td>FRC 24 requests that a new gift card be activated for use, that a new gift card be loaded with funds, or that an active gift card be loaded with additional funds.</td>
</tr>
<tr>
<td>FRC 25</td>
<td>FRC 25 requests that an active gift card be deactivated so that it can’t be used to process transactions. The API Option GCCASHOUT must be sent in the request to return any remaining funds to the consumer. If GCCASHOUT is not sent, a balance will remain on the gift card (if doing so is supported by the processor).</td>
</tr>
<tr>
<td>FRC 26</td>
<td>FRC 26 requests that a previously deactivated gift card be reactivated. If there was a balance remaining on the gift card at the time of deactivation, the balance will again become available for use. Funds cannot be added to a gift card during a Reactivate request.</td>
</tr>
<tr>
<td>FRC 61</td>
<td>FRC 61 requests that a gift card’s balance, masked card number, expiration date, and discount percentage be returned. The IYCAvailableBalance and IYCBalance fields can be used to determine the amount of funds on the card. Merchants may be required to pay a fee for an inquiry request depending on their processor agreement.</td>
</tr>
</tbody>
</table>
**Gift Card Processing Use Cases**

The use cases described below apply to only IYC and third-party gift cards. They do not apply to other card types, including private-label gift cards.

**Use Case 14: Gift Card Balance Inquiry**

**Actors:** Interface, Shift4 Payments

**Preconditions:** A merchant has swiped or manually keyed a gift card and needs to determine the balance of the card.

**Main Flow:** This flow illustrates a successful balance inquiry.

1. An Inquiry request (FRC 61) is sent to determine the balance of a gift card.
2. A response is received with no errors.
3. The IYCBalance and IYCAvailableBalance fields are examined to determine the card’s balance.
4. Receipts are generated, if applicable.

**Exception Flow:** This flow illustrates a balance inquiry error response.

2. An error response is received.
3. Determine the cause of the error and follow merchant procedures.

**Post Conditions:** The balance of the gift card is successfully retrieved.
Use Case 15: Loading or Recharging a Gift Card

Actors: Merchant, Interface

Preconditions: A merchant has collected payment for a gift card and is ready to load the payment onto an active gift card. Payment can be collected as a purchase using a payment card, cash, check, return, or other tender.

Main Flow 1: This flow illustrates successfully loading a balance to a new gift card.

1. An Activate/Reload request (FRC 24) is sent with the IYCBalance field specifying the amount to load on the card. The request must include the API Options ALLDATA and IYCDEACTIVEONLY.
2. A response is received with no errors.
3. The correct balance is indicated in the IYCBalance field.
4. Receipts are generated.

Main Flow 2: This flow illustrates successfully adding to the balance of an active gift card.

1. An FRC 24 is sent with the IYCBalance field specifying the amount to add to the card. The request must include the API Options ALLDATA, IYCACTIVEONLY, and IYCRECHARGE.
2. A response is received with no errors.
3. The correct balance is indicated in the IYCBalance field.
4. Receipts are generated.

Exception Flow 1: This flow illustrates an error response or incorrect balance. (Main Flow 1 or 2; Step 3.)

Log the transaction for an auditor’s review and follow merchant procedures.

Post Condition: The consumer has an active gift card with the correct balance and receipt.

Use Case 16: Cashing Out a Gift Card

Actors: Merchant, Interface

Preconditions: A merchant has swiped or manually entered a gift card that needs to be cashed out without deactivating it. The interface has already successfully completed the steps in Use Case 14: Gift Card Balance Inquiry and recorded the card’s balance.

Main Flow: This flow illustrates successfully cashing out a gift card without deactivating it.

5. An Online Sale request (FRC 1D) is sent for a gift card’s recorded balance. The request must include the API Options ALLDATA and ENHANCEDRECEIPTS.
6. A response is received with no error codes.
7. The balance from the FRC 1D request is returned to the consumer in cash.
8. Receipts are generated.

Exception Flow: This flow illustrates a deactivation error response.

2. An error response is received.
3. Log the transaction for an auditor’s review and follow merchant procedures.
Post Conditions: The balance of the gift card has been returned to the consumer in cash.
Use Case 17: Deactivating a Gift Card
Actors: Interface, Merchant
Preconditions: A merchant has swiped or manually entered a gift card that needs to be deactivated. The merchant or interface has already successfully completed the steps in Use Case 14: Gift Card Balance Inquiry and recorded the card’s balance.
Main Flow: This flow illustrates deactivating a gift card with cashback.

1. A Deactivate request (FRC 25) is sent. The request must include the API Options ALLDATA, GCCASHOUT, and IYCACTIVEONLY.
2. A response is received with no error codes. With the value “D E A C T I V A T E D” in the ReceiptText field, indicating the card has been deactivated.
3. The balance is returned to the consumer in cash.
4. Receipts are generated.

Alternate Flow: This flow illustrates deactivating a gift card and not cashing it out. (This may happen in rare instances, such as when a gift card has been lost or stolen.)

1. An FRC 25 request is sent. The request must include the API Options ALLDATA and IYCACTIVEONLY.
2. A response is received with no error codes and the value “D E A C T I V A T E D” in the ReceiptText field, indicating the card is deactivated.
3. Receipts are generated.

Exception Flow: This flow illustrates receiving an error response to a deactivation request. (Main Flow; Step 2, or Alternate Flow; Step 2.)

1. An error response is received.
2. Log the transaction for an auditor’s review and follow merchant procedures.

Post Conditions: The card has been deactivated and the reason recorded.

Use Case 18: Reactivating a Gift Card
Actors: Merchant, Interface
Preconditions: A merchant has swiped or manually entered a gift card that needs to be reactivated.
Main Flow: This flow illustrates successfully reactivating a gift card.

1. A Reactivate request (FRC 26) is sent with the API Options ALDDATA and IYCDEACTIVEONLY.
2. A response is received with no error codes, indicating the card is reactivated.
3. Receipts are generated, if applicable.

Exception Flow: This flow illustrates receiving an error response to a reactivation request.

1. An error response is received.
2. Log the transaction for an auditor’s review and follow merchant procedures.

Post Conditions: The gift card has been reactivated.
Use Case 19: Processing Refunds to a Gift Card

Actors
Consumer, Merchant, Interface

Preconditions:
A merchant is processing a refund to a gift card.

Main Flow:
This flow illustrates issuing a refund to a new gift card.

1. The consumer initiates a return to a merchant.
2. The merchant is not issuing the refund to the payment card used for the purchase.
3. A gift card is swiped or manually keyed and an Activate/Reload request (FRC 24) is sent with the amount to be loaded to the gift card in the IYCBalance field. The request must include the API Options ALLDATA and IYCDEACTIVEONLY.
4. A response is received with no errors.
5. The correct balance is indicated in the IYCBalance field.

Alternate Flow:
This flow illustrates issuing a refund to the gift card that was used for the purchase. (Main Flow; Step 2.)

2. The merchant is issuing the refund to the gift card used for the purchase.
3. An Online Sale request (FRC 1D) is sent with the amount to be refunded in the PrimaryAmount field and the SaleFlag set to ‘C’ using a new invoice number. The gift card that was used for the original purchase is used or the TrueToken from the sale is retrieved. The request must include the API Options ALLDATA and ENHANCEDRECEIPTS. The Notes field must include a reference to the original sale.
4. A response is received with no errors.
5. The correct balance is indicated in the IYCAvailableBalance field.

Exception Flow 1:
This flow illustrates receiving a refund error response. (Main Flow; Step 4, Alternate Flow; Step 4.)

4. An error response is received.
5. Log the transaction for an auditor’s review and follow merchant procedures.

Exception Flow 2:
This flow illustrates a gift card refund response with an incorrect balance. (Main Flow; Step 5, Alternate Flow; Step 5.)

Follow the steps in Use Case 15: Loading or Recharging a Gift Card.

Post Conditions:
An approved transaction awaits settlement.
Processing Refunds

There are several options for processing refunds, and there are several factors to consider when choosing how to process refunds, including:

- The fees that the merchant will incur
- The consumer’s available funds
- The card type used for the sale

Use the information below to determine which option for processing refunds best fits your current business model.

**Refunds for Debit and Gift Cards**

Sales for debit and gift cards are processed in real time, so when a purchase is completed, the amount of the transaction is immediately deducted from the consumer’s account. Because of this, issuers do not always support issuing refunds to these card types. The best practice is to refund these transactions in cash, as a gift card, or store credit.

**Refund Flow Chart for Credit Cards and Private Label Cards**
Using the Lodging Fields

Shift4 Payments requires interfaces supporting lodging and hospitality merchants to provide specific data input fields. Sending this information can help merchants receive better processing rates. Also, since this data is typically provided on consumers’ bank statements (along with the information from the level 2 card data and purchasing card fields), it helps to lessen the likelihood that the charges will be disputed.

For more information about the level 2 card data and purchasing card fields, see the Level 2 Data and Purchasing Card Data section in this document.

What Are the Lodging Fields?

The lodging fields contain information that pertains to what a consumer is being billed for and why, including stay dates, late charges, and incidental charge details. The fields are defined below.
## Hotel Check-In Field (Data Block 013)

The hotel check-in field is required for authorization requests.

<table>
<thead>
<tr>
<th>Field</th>
<th>Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HotelEstimatedDays</td>
<td>2</td>
<td>The estimated number of days of the guest’s stay. This value is included in the initial authorization request as well as subsequent authorization requests (where the value is likely to change).</td>
</tr>
</tbody>
</table>

## Hotel Check-Out Fields (Data Block 003)

The following hotel check-out fields are required for lodging sale requests, not authorizations.

<table>
<thead>
<tr>
<th>Field</th>
<th>Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrimaryChargeType</td>
<td>1</td>
<td>Indicates the transaction type using one of the following values: '1' – Lodging '2' – Restaurant '3' – Gift Shop</td>
</tr>
<tr>
<td>SpecialCode</td>
<td>1</td>
<td>Provides additional detail about lodging transactions, including: '1' – No Special Code '2' – Assured Reservation/No Show '3' – Advance Deposit '4' – Delayed Charge (Requires HotelAdditionalCharges Field to be Specified) '5' – Express Check-Out Service '6' – Assured Reservation/Normal Stay</td>
</tr>
<tr>
<td>HotelAdditionalCharges</td>
<td>6 Max</td>
<td>Provides additional detail about hotel charges. Up to six values can be specified in one request to indicate different types of charges, including: '0' – N/A '2' – Restaurant '3' – Gift Shop '4' – Minibar '5' – Telephone '6' – Other '7' – Laundry</td>
</tr>
<tr>
<td>ArrivalDate</td>
<td>6</td>
<td>Indicates the arrival date of a guest’s hotel stay in MMDDYY format. The date specified cannot be a future date. The arrival date needs to be one day before the sale date for hotel transactions that are a straight sale, such as advanced deposit, no-show charge, and late charges. Providing the date as specified helps merchants qualify for the best processing rate.</td>
</tr>
<tr>
<td>DepartureDate</td>
<td>6</td>
<td>Indicates the departure date of guest’s hotel stay in MMDDYY format. The date specified cannot be a future date. The departure date needs to be the day of the sale for hotel transactions that</td>
</tr>
</tbody>
</table>
Field | Bytes | Description
--- | --- | ---
are a straight sale, such as advanced deposit, no-show charge, and late charges. Providing the date as specified helps merchants qualify for the best processing rate.

**Examples of How to Use the Lodging Fields**

Some examples to help you determine how to use the lodging fields are given below.

---

**Note:** The following examples do not cover all possible lodging transactions.

**Hotel Check-In – FRC 1B/05**

A guest is checking in to a room on 3/31/17 and checking out 4/4/17.

<table>
<thead>
<tr>
<th>Hotel Estimated Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
</tr>
</tbody>
</table>

**Hotel Advanced Deposit – FRC 1D/06**

Today is 4/3/17, and a guest has booked a room for 4/30/17 to 5/4/17.

<table>
<thead>
<tr>
<th>Primary Charge Type</th>
<th>Special Code</th>
<th>Hotel Additional Charges</th>
<th>Arrival Date</th>
<th>Departure Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
<td>4/2/17</td>
<td>4/3/17</td>
</tr>
</tbody>
</table>

**Hotel Stay with No Incidental – FRC 1D/06**

A guest checked in to a room on 3/31/17. Today is 4/4/17, and the guest is checking out. The SpecialCode field will be set to ‘5’ if express check-out was used. If express check-out wasn’t used, the field should be set to ‘6’ for a stay with a reservation or ‘1’ for a stay with no reservation.

If additional hotel charges are included in the bill, specify them in the HotelAdditionalCharges field.

<table>
<thead>
<tr>
<th>Primary Charge Type</th>
<th>Special Code</th>
<th>Hotel Additional Charges</th>
<th>Arrival Date</th>
<th>Departure Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,5, or 6</td>
<td>0</td>
<td>3/31/16</td>
<td>4/4/17</td>
</tr>
</tbody>
</table>

**Hotel No Show Charge – FRC 1D/06**

Today is 4/1/17, and a guest is being billed for a missed reservation that was booked for 3/31/17. Current industry regulation stipulates that the hotel may only collect one night’s room rate plus taxes and fees for a missed reservation.
<table>
<thead>
<tr>
<th>Primary Charge Type</th>
<th>Special Code</th>
<th>Hotel Additional Charges</th>
<th>Arrival Date</th>
<th>Departure Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
<td>3/31/17</td>
<td>4/1/17</td>
</tr>
</tbody>
</table>
Hotel Check-Out Delayed Charge – FRC 1D/06

Today is 4/5/17, and a guest is being billed for minibar charges after they checked out on 4/4/17.

<table>
<thead>
<tr>
<th>Primary Charge Type</th>
<th>Special Code</th>
<th>Hotel Additional Charges</th>
<th>Arrival Date</th>
<th>Departure Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3/31/17</td>
<td>4/4/17</td>
</tr>
</tbody>
</table>

Processing EBT Cards

To process EBT (Electronic Benefits Transfer) cards, your interface will need to send one of three API Options with the transaction depending on the purpose: EBTFOOD, EBTCASH, or EBTWITHDRAW.

A complete list of supported banks and processors can be found at shift4.com/dotn/integration/banks-processors.cfm. Using this list, you can determine which processors and banks support EBT.

For additional information about these API Options, see the API Options section in the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner.

Use Case 20: EBT Straight Sale for Groceries
Actors: Consumer, Merchant
Preconditions: A merchant is ready to process an Online Sale Request (FRC 1D) for EBT items.
Main Flow: This flow illustrates an approval response.
   1. The consumer’s EBT card is swiped for EBT items.
   2. The transaction is sent as an FRC 1D request, including the EBTFOOD API option.
   3. A response indicating an approval is returned.
Alternate Flow 1: This flow illustrates a decline response. (Main Flow; Step 3.)
   Follow the steps in Use Case 13: Voiding an Invoice.
Post Conditions: An approved transaction awaits settlement at the end of the day.

Use Case 21: EBT Straight Sale for Non-Grocery Items with Cash Back
Actors: Consumer, Merchant
Preconditions: A merchant is ready to process an Online Sale request (FRC 1D) for EBT items.
Main Flow: This flow illustrates an approved EBT transaction with cash back.
   1. The consumer’s EBT card is swiped.
   2. The transaction is sent as an FRC 1D request, including the EBTCASH API option.
   3. A response indicating an approval is returned.
Alternate Flow 1: This flow illustrates a decline response. (Main Flow; Step 3.)
   Follow the steps in Use Case 13: Voiding an Invoice.
Use Case 22: EBT Straight Sale with Cash Withdraw

Actors: Consumer, Merchant

Preconditions: A merchant is ready to process an FRC 1D request for EBT items.

Main Flow: This flow illustrates a successful EBT cash withdrawal transaction.

1. The consumer’s EBT card is swiped.
2. The transaction is sent as an FRC 1D request, including the EBTWITHDRAW API option.
3. A response indicating an approval is returned.

Alternate Flow 1: This flow illustrates a decline response. (Main Flow; Step 3.) Follow the steps in Use Case 13: Voiding an Invoice.

Post Conditions: An approved transaction awaits settlement at the end of the day.
Understanding the MetaToken Process

The MetaToken™ provides a way to track card usage across multiple transactions and multiple revenue centers to support a merchant’s marketing campaigns and other analytic operations. The MetaToken feature can also be set up to track card usage across multiple Shift4 Payments accounts.

The MetaToken returns the same value every time a transaction is processed, but is designed so that it can’t be used to process transactions. The MetaToken is a 16-digit faux card number, and the first digits of the MetaToken are outside of the standard bank identification number (BIN) ranges (i.e., the first 6 digits of a card number used to identify the card issuer), so it is intentionally unable to pass Luhn mod 10.

MetaTokens can be received in two formats based on the merchant’s preference. If the MetaTokenType is specified as “IL”, the last four digits of the MetaToken will be the same as the last four digits of the payment card; if the MetaTokenType is specified as “F6”, the last six digits of the MetaToken will match that of the first six digits of the payment card.

MetaTokens are unique across a specified set of accounts except in rare instances, such as when the UTG is offline or a merchant is experiencing communication issues. In these cases, the UTG may return a stand-in MetaToken that is made up of 16 zeros. You may want to consider adopting a process that replaces these stand-in MetaTokens with valid MetaTokens after the merchant’s communications are restored by following the steps in Use Case 24: Obtaining a MetaToken for an Existing Transaction.

Use Case 23: Obtaining a MetaToken for Each Authorization or Sale Transaction

Ares: Clerk, Consumer, Shift4 Payments

Preconditions: To track consumer usage, a resort wants a MetaToken for each authorization or sale across all revenue centers. The consumer’s card has been captured or a TrueToken is being used.

Main Flow: This flow illustrates a MetaToken being returned.

1. The transaction is sent as an Online Auth request (FRC 1B) or Online Sale request (FRC 1D), including the RETURNMETATOKEN API Option and the MetaTokenType field.
2. If MetaToken is enabled for the account and a MetaToken exists for the card, the existing MetaToken is returned. If a MetaToken does not exist for the card, one will be generated and returned.

Exception Flow 1: This flow illustrates a MetaToken not being returned due to a configuration issue.

2. If MetaToken is not enabled for the account, the transaction will not return a MetaToken. Contact Shift4 Payments.

Exception Flow 2: This flow illustrates MetaToken functionality when a communication failure has occurred.

2. If the UTG is offline, the UTG will return a MetaToken with all zeros. (See Use Case 24: Obtaining a MetaToken for an Existing Transaction to learn how to replace stand-in tokens with a valid MetaToken.)

Post Conditions: A MetaToken is returned in a transaction response.
Use Case 24: Obtaining a MetaToken for an Existing Transaction

Actors: Interface, Shift4 Payments

Preconditions: A resort wants to obtain a MetaToken for an existing transaction.

Main Flow: This flow illustrates generating a MetaToken from an existing transaction.

1. The interface sends a Get MetaToken request (FRC CD), including the MetaTokenType field and one of the following: a card number, P2PE data, or TrueToken.
2. If MetaToken is enabled for the account and a MetaToken corresponds to the details provided, it will be returned. If a MetaToken does not correspond, one will be generated and returned.

Exception Flow 1: This flow illustrates a MetaToken not being returned due to a configuration issue.

2. If MetaToken is not enabled for the account, an error will be returned.

Exception Flow 2: This flow illustrates Get MetaToken functionality during a communication failure.

2. If the merchant is offline, it will return the error, “Function not supported in offline mode.”

Post Conditions: A MetaToken is returned in a Get MetaToken response.

Prompt Confirmation Function

This feature enables a merchant to prompt a consumer to enter or confirm their email address, legal text, or other text with a UTG-controlled PIN pad using the Prompt Confirmation function (FRC 82). The Prompt Confirmation function is a pass-through function, meaning the values provided in the request and response will not be recorded by DOLLARS ON THE NET or passed on to the processor. The function can be used before, after, or independent of a transaction request.

Use Case 25: Prompt Confirmation for Email on PIN Pad

Actors: Consumer, Clerk

Preconditions: A consumer signs up to receive email notifications from a merchant.

Main Flow: This flow illustrates the consumer responding to a Prompt Confirmation request on a UTG-controlled PIN pad.

1. The clerk enters the consumer’s email address into the interface.
2. The interface sends a Prompt Confirmation request (FRC 82), including a question, such as, “Is this your correct email address?” in the PromptConfirmQuestion field and the known email address in the PromptConfirmValue field.
3. The PIN pad prompts the consumer to verify their email address.
4. The consumer’s response is sent to the interface in the PromptConfirmResult field.

Post Conditions: The consumer’s response is passed through to the interface.
Use Case 26: Prompt Confirmation for Legal Text on PIN Pad Device

Actors: Consumer, Clerk

Preconditions: A merchant requires legal consent from a consumer.

Main Flow: This flow illustrates the consumer responding to a Prompt Confirmation request on a UTG-controlled PIN pad.

1. The interface sends a Prompt Confirmation request (FRC 82), including a question, such as, “Do you agree to the following terms?” in the PromptConfirmQuestion field and the known legal text in the PromptConfirmValue field.
2. The PIN pad prompts the consumer for their consent.
3. The consumer response is sent to the interface in the PromptConfirmResult field.

Post Conditions: The consumer response is passed through to the interface.

Supporting Digital Signature Capture

For card-present transactions on a PIN pad with digital signature capability, merchants can collect digital signatures for verification instead of paper signatures. Some merchants prefer digital signature capture because it alleviates the burden of storing and collecting signatures on paper and allows the signature to be conveniently stored within the transaction record in DOLLARS ON THE NET.

---

**Note:** When you send a sale or authorization request using a UTG-controlled PIN pad device with digital signature capability, the UTG automatically prompts the PIN pad for the consumer’s signature upon authorization and uploads the signature to DOLLARS ON THE NET. Therefore, implementing digital signature capture will be the easiest for merchants using UTG-controlled PIN pads.

---

**Digital Signature Capture Function Requests**

Common FRCs used for digital signature capture include the following:

<table>
<thead>
<tr>
<th>FRC</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>47 – Request Signature</td>
<td>Allows the merchant to obtain a signature without processing a payment card, such as when a hotel guest is checking in for their stay and accepting an estimate of charges.</td>
</tr>
<tr>
<td>20 – Signature Upload</td>
<td>Uploads the captured signature to an existing invoice in DOLLARS ON THE NET.</td>
</tr>
</tbody>
</table>

If a signature is required for additional non-transaction functions, such as Terms and Conditions (FRC CF) or Process Forms (FRC 86), this can be accomplished by sending FRC 47 followed by FRC 20.
Digital Signature Capture API Options

There are four API Options specific to signature capture:

<table>
<thead>
<tr>
<th>API Option</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYPASSSIGCAP</td>
<td>This option is used to suppress requesting a digital signature from a consumer for a given transaction. Instead, a signature line will be included in the receipt text (if applicable).</td>
</tr>
<tr>
<td>NOSIGNATURE</td>
<td>This option is used to suppress requesting a signature from a consumer for a given transaction. Additionally, sending this option will ensure that a signature line will not be included in the receipt text (if applicable).</td>
</tr>
<tr>
<td>RETURNSIGNATURE</td>
<td>This option is used to return the signature captured on a PIN pad to the interface. The returned signature can then be used by the interface to display to the clerk or print on a receipt.</td>
</tr>
<tr>
<td>PLCCSIGNATURE</td>
<td>This option is used with Request Signature (FRC 47). When performing an FRC 47, if PLCCSIGNATURE is included, the screen on the PIN pad will display “I have received and agree with the Terms and Conditions” instead of “Please sign and tap OK with pen”.</td>
</tr>
</tbody>
</table>

**WARNING!** Using the NOSIGNATURE API Option to skip collecting signatures on low-dollar-amount transactions can increase the merchant’s chargeback risk.

Plotting Digital Signatures Using the Shift4 Payments 8-Byte Format

The following rules apply to all digital signature captures:

- The stylus is off the device at the beginning of the digital signature capture, therefore a “pen up” character (‘%’), is recorded.
- As the consumer signs, the digital signature captured by the PIN pad or signature-only device is translated and parsed into the Shift4 Payments 8-byte format based on coordinates along the x and y axes. The location where the consumer initially touches the stylus to the signature capture device is the first 8-byte (“x,y”) coordinate recorded.
- Line segments are automatically created and assigned an 8-byte value as the consumer moves the stylus across the device.
- Every time the consumer lifts the stylus, a “pen up” character (‘%’) is recorded.
- When the consumer lifts the stylus and accepts the signature, recording ceases and the signature capture record is finalized.

Example Signature Capture – “HELLO”

To further explain how digital signature capture functions, we’ve provided an example of a signature using the printed word “HELLO.” Because printed block letters are based on straight lines rather than curves, far less data will be required than there would be to capture an actual consumer signature; however, the process is identical.
In its simplest form, the lines of the printed letter “H” are parallel to the x and y axes on the signature capture grid. Therefore, the coordinates and “pen up” characters for the printed letter “H” might be communicated using the 8-byte format as follows:

Note: The stylus is off the device at the beginning of the digital signature capture, therefore a “pen up” character (‘%’), is recorded.

Important: To compress data, the coordinates represent a continuous line until a “pen up” character (‘%’) is recorded.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Simplified Digital Signature Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>%00220018000220062%0052001800520062%0022004200520042%</td>
</tr>
</tbody>
</table>

The following table breaks down the printed letter “H” into each 8-byte segment of coordinates based on the consumer’s action:

<table>
<thead>
<tr>
<th>Capture</th>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>The stylus is up before consumer begins to write; the screen is blank and a “pen up” character is recorded</td>
<td></td>
</tr>
<tr>
<td>00220018</td>
<td>Consumer touches the stylus to the device at coordinate (0022,0018)</td>
<td></td>
</tr>
<tr>
<td>00220062</td>
<td>Consumer moves stylus to coordinate (0022,0062)</td>
<td></td>
</tr>
<tr>
<td>%</td>
<td>Consumer lifts stylus</td>
<td></td>
</tr>
<tr>
<td>00520018</td>
<td>Consumer touches stylus to device at coordinate (0052,0018)</td>
<td></td>
</tr>
<tr>
<td>00520062</td>
<td>Consumer moves stylus to coordinate (0052,0062)</td>
<td></td>
</tr>
<tr>
<td>%</td>
<td>Consumer lifts stylus</td>
<td></td>
</tr>
<tr>
<td>00220042</td>
<td>Consumer touches stylus to device at coordinate (0022,0042)</td>
<td></td>
</tr>
<tr>
<td>00520042</td>
<td>Consumer moves stylus to coordinate (0052,0042); letter “H” is complete</td>
<td></td>
</tr>
<tr>
<td>%</td>
<td>Consumer lifts stylus</td>
<td></td>
</tr>
</tbody>
</table>

The process outlined in the previous table repeats for each remaining letter (E, L, L, and O). After the last letter (O), the consumer touches a button on the device to either accept or reject the signature. After acceptance, the signature capture for the entire word “HELLO” might look like this:
**Simplified Digital Signature Capture**

<table>
<thead>
<tr>
<th>Letter</th>
<th>Simplified Digital Signature Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>%0022001800220062%0052001800520062%0022004200520042%</td>
</tr>
<tr>
<td>E</td>
<td>00920062006200620062001800920018%0062004200880042%</td>
</tr>
<tr>
<td>L</td>
<td>010200180102006201320062%</td>
</tr>
<tr>
<td>L</td>
<td>014200180142006201720062%</td>
</tr>
<tr>
<td>O</td>
<td>01860018020801802120022021200580208006201860062018200580182002201860018%</td>
</tr>
</tbody>
</table>

**“HELLO” Plotted on Grid**

A grid-based interpretation of the example signature using the 8-byte format to communicate the coordinates of the signature is provided below. Notice that the (0,0) coordinate is in the upper left-hand corner of the screen. Four of the coordinates from the raw trace are also provided as points of reference.

---

**Example Digital Signature Capture – DOLLARS ON THE NET Transaction Record**

When the consumer accepts their signature on a UTG-controlled device, Shift4 Payments sends the captured signature to DOLLARS ON THE NET, where it is attached to the transaction record as displayed in the example below:
Complete Letter “H” Raw Trace

The previous examples depict highly simplified data. However, the raw trace for a consumer’s digital signature is typically very large. This difference in size primarily occurs because there are significantly more coordinates required to capture a “live” signature. For example, a consumer’s “H” might look something like this:

The raw trace that was captured for this letter “H” is shown below. The trace consists of pairs of lines depicting 8-byte coordinates, each containing the following information:

- The signature
- “00000E3C” to denote that this line in the raw trace file contains signature capture data
- The day, date, and time stamp for the signature
- 8-byte x or y plot for the coordinate

The first line of each pair captures the location of the coordinate on the x axis. The second line captures the location of the coordinate on the y axis. For example, as indicated in red in the raw trace below, the first stroke of the signature is at (0739,0135):
Note: The stylus is off the device at the beginning of the digital signature capture, therefore a “pen up” character ("%"), is recorded.
Sending Digital Signature Capture Data Using a PNG File

Instead of sending a signature in the Shift4 Payments 8-byte format, you can send a signature in PNG format. To do this, use the Signature Upload request (FRC 20) after authorization and include the following parameters:

For HTTPS POST:

Add the fields PhotoType=[1 byte] and PhotoData=[up to 4 megabytes].

---

**Requirement:** The PhotoType field indicates the type of photo, and it must send ‘P’ for a PNG file.

---

For TCP/IP:

Use Data Block 088 Photo.

---

**Requirement:** Only photo data or Shift4 Payments 8-byte format data may be sent in an FRC 20 request; **NOT** both.

---

Using the Terms and Conditions Function

A Terms and Conditions request (FRC CF) displays a terms and conditions form on a UTG-controlled PIN pad. This will allow a consumer to either accept or decline the merchant’s terms and conditions displayed on the PIN pad device. This function is a pass-through function, so the data is not stored or processed by the UTG. The function can be used before, after, or independent of a transaction request.

To support the Terms and Conditions function, your interface will send FRC CF and the TermsAnd Conditions field containing the text to display (up to 4,096 characters). This will cause the UTG to activate the PIN pad and display the terms and conditions form. The “Accept” option will be disabled until the consumer selects “I acknowledge”.

© 2018 Shift4 Payments, LLC. All rights reserved.
External Use NDA

Version 2.34
Page 106 of 137
Use Case 27: Processing Terms and Conditions with a UTG-Controlled PIN Pad

Actors: Consumer, Clerk

Preconditions: A consumer must accept a merchant’s terms and conditions before a transaction is processed, such as to sign up and pay for a gym membership.

Main Flow: This flow illustrates the consumer being presented terms and conditions on a UTG-controlled PIN pad.

1. The clerk initiates an action requiring a consumer’s confirmation.
2. The interface sends a Terms and Conditions request (FRC CF) to the UTG.
3. The PIN pad displays the merchant’s terms and conditions on the PIN pad screen.
4. The consumer taps “I acknowledge” on the PIN pad screen.
5. The consumer taps “Accept” on the PIN pad screen.
6. The PIN pad returns the value ‘Y’ to the UTG in the TermsandConditionsResult field.
7. The interface proceeds to complete the transaction flow.

Alternate Flow 1: This flow illustrates the consumer declining a merchant’s terms and conditions on a UTG-controlled PIN pad.

4. The consumer declines the terms and conditions by tapping “Decline” on the PIN pad screen.
5. The PIN pad returns the value ‘N’ to the UTG in the TermsandConditionsResult field.
6. The transaction is canceled by the interface.

Exception Flow 1: This flow illustrates an error condition occurring during the transaction flow.

6. The transaction receives an error or timeout response.
7. Follow the merchant’s procedures.

Post Conditions: The interface has recorded the consumer’s acceptance of the terms and conditions, and it proceeds to complete the normal transaction flow.
Using the Process Forms Function

The Process Forms function (FRC 86) allows a merchant to display custom forms and text on a UTG-controlled PIN pad. For example, a merchant would display this form to prompt a consumer to acknowledge data stored by the merchant, such as a birth date, email address, or loyalty program number. Additionally, a merchant could use this function to display order or reservation information for the consumer’s approval before processing payment.

This function is a pass-through function, so the data is not stored or processed by the UTG. This function can be used before, after, or independent of a transaction request.

To support the Process Forms request, your interface will send FRC 86, including the form name and up to ten custom messages to display. Each custom message will have a numeric value (1-10) that will map the message to a position defined on the form. This will activate the PIN pad and prompt the consumer with the merchant’s custom form(s) and data. The consumer will then select an option based on the merchant-defined response buttons. The UTG will then return the button ID that was pressed to the interface in the FormResponse field.

Use Case 28: Using the Process Forms Function to Record a Consumer Response

Actors: Consumer, Clerk

Preconditions: A merchant has custom forms designed and loaded onto a UTG-controlled PIN pad.

Main Flow: This flow illustrates presenting a consumer a custom form to complete using a UTG-controlled PIN pad.

1. The interface sends a Process Forms request (FRC 86) to the UTG.
2. The PIN pad displays the custom form for the consumer to complete.
3. The consumer completes the form using the on-screen button(s).
4. The PIN pad sends the button ID to the UTG.
5. The response is sent to the interface with the button ID in the FormResponse field.

Exception Flow: This flow illustrates an error condition occurring during the transaction flow.

5. The request receives an error or timeout response.
6. The UTG will return the error or timeout message to the interface.

Post Conditions: The interface receives the button ID pressed by the consumer.
Supporting the Cash Back Function

Shift4 Payments allows merchants using UTG-controlled PIN pads to accept cash back requests from consumers using debit cards. To enable this capability, the PIN pad must be configured for cash back via UTG TuneUp. For information, see the Configuring Devices section of the Universal Transaction Gateway: Quick Installation Guide located in MyPortal Documentation Corner.

**Requirement:** To enable cash back, the merchant must use a UTG-controlled PIN pad and configure it for cash back functionality.

To prompt a consumer for cash back, your interface will send the ALLOWCASHBACK API Option in an Online Auth (FRC 1B) or Online Sale (FRC 1D) request. (Cash back is only supported by debit cards.)

In the response, the UTG will populate the Cashback field with the cash amount that the consumer requested and adjust the PrimaryAmount accordingly. If the ENHANCEDRECEIPTS API Option is used, the UTG will update the receipt text as well.

**Use Case 29: Cash Back Transaction Flow**

**Actors:** Interface, Clerk, Consumer

**Preconditions:** A merchant has configured their PIN pad to allow for cash back.

**Main Flow:** This flow illustrates an approved cash back transaction.

1. The interface sends an FRC 1B or FRC 1D for $100 and the ALLOWCASHBACK API Option.
2. The consumer is prompted to approve the purchase amount and swipe their card.
3. If debit is selected, the consumer is prompted for cash back.
4. The consumer requests cash back.
5. The consumer is prompted with the cash back form.
6. The consumer selects a cash back amount of $20.
7. The consumer is prompted to approve the total amount of $120.
8. The consumer is prompted to enter their PIN.
9. The request is sent for approval.
10. A response indicating approval is received.
11. The interface prompts the clerk to give the consumer the approved cash back amount and receipts are generated.

**Alternate Flow 1:** This flow illustrates a decline response. (Main Flow; Step 10.)

Follow the steps in Use Case 13: Voiding an Invoice.

**Post Conditions:** An approved transaction awaits settlement at the end of the day.
Utilizing On-Demand Card Read

The On-Demand Card Read function (FRC DA) prompts a P2PE-enabled UTG-controlled PIN pad to request a pass-through card swipe, causing the output of the swipe to be returned directly to the interface without any action or validation by Shift4 Payments or the processor.

Based on a card's BIN range, an FRC DA response may be returned in one of the following formats outlined in the sections below.

Payment Cards

Card data that falls within the card brands' BIN ranges will be returned to the interface encrypted in the P2PEBlock field and a P2PEDeviceType will be returned. If entered, the consumer's CVV2Code will be embedded in the P2PE block. If the card was manually keyed, the CardNumber and ExpirationDate will also be embedded in the P2PE block.

For payment cards that are processed by Shift4 Payments but fall outside of the card brand BIN ranges (such as private-label and gift cards), the CardNumber, ExpirationDate and TrackInformation will be returned embedded in the P2PEBlock field and a P2PEDeviceType will be returned. If entered, the CVV2Code field will be returned in the P2PE block as well.

If you plan to store the P2PE block for future use, you will also need to store the P2PEDeviceType parameter that was returned. When using the P2PE block in future API requests, you will need to specify the correlated P2PEDeviceType parameter to reference that data.

Specialty Payment Cards Added to an Exclusion File

Specialty payment cards, such as private-label and gift cards that the merchant doesn’t process through Shift4 Payments, may be excluded from Shift4 Payments' encryption when using FRC DA. This is accomplished by loading a manufacturer-signed configuration file that specifies the BIN ranges onto an Ingenico RBA or Verifone MX PIN pad. Card swipes that fall within the BIN ranges noted in the exclusion file will be returned unencrypted including the CardNumber, ExpirationDate and TrackInformation. If entered, the CVV2Code field will be returned unencrypted as well. As a fail-safe measure, the UTG has logic that returns an error if the PIN pad attempts to pass data that falls within the card brands’ BIN ranges.

The signed configuration files for Ingenico and Verifone devices can be loaded on the PIN pad using the instructions in the Universal Transaction Gateway: Using EMV External Devices guide located in MyPortal Documentation Corner.

Non-Payment Cards

If a non-payment card (e.g., employee identification or consumer loyalty card) is swiped, the response will return the track data to the interface unencrypted in the TrackInformation field. Care should be taken to avoid having the numbers of a non-payment card fall within a card brand’s BIN range, as this data will be blocked by Shif4’s UTG.

API Options for On-Demand Card Read

The API Options RETURNEXPDATE, DISABLEMCE, USEMCE, and DISABLECONTACTLESS can be used with FRC DA. For detailed descriptions, see the API Options section of the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner.
Use Case 30: On-Demand Card Read Function

Actors: Interface, Clerk, Consumer

Preconditions: A merchant is ready to request a pass-through card read.

Main Flow: This flow illustrates a card swipe returning encrypted CHD.
1. An On-Demand Card Read (FRC DA) request is sent, activating the PIN pad.
2. The consumer swipes their card.
3. The PIN pad identifies the card type as a standard payment card.
4. The PIN pad returns the P2PE block to the UTG.
5. The UTG returns the P2PE block and P2PEDeviceType parameter to the interface to store for use in future API requests.

Alternate Flow 1: This flow illustrates a card swipe returning unencrypted CHD.
3. The PIN pad identifies the card type as a specialty payment card in the consumer’s exclusion file.
4. The PIN pad returns the unencrypted CHD to the UTG.
5. The UTG checks the BIN range of the unencrypted CHD and determines that it is outside of the payment card ranges processed by Shift4 Payments.
7. The unencrypted CHD is passed from the UTG to the interface.

Alternate Flow 2: This flow illustrates a card swipe returning data from a non-payment-card swipe.
3. The PIN pad identifies the card type as a non-payment card.
4. The swipe data is passed from the PIN pad to the UTG.
5. The UTG checks the BIN range of the unencrypted CHD and determines that it is outside of the standard payment card range.
6. The unencrypted CHD is passed from the UTG to the interface.

Exception Flow: This flow illustrates a card swipe returning an error when disallowed unencrypted CHD is detected.
5. The UTG checks the BIN range of the unencrypted CHD and determines that it is not allowed.
6. The UTG discards the data returned by the PIN pad.
7. The UTG returns a “NOT P2PE data” error (code 9076).
8. The interface informs the merchant of the error condition and to contact Shift4 Payments immediately.

Post Conditions: The interface receives the swipe data to use as needed.
Sending the Input Prompt Function

An Input Prompt request (FRC DB) prompts a UTG-controlled PIN pad to collect a specified value from a consumer. Your interface will specify the value from the list below in the DeviceInputIndex field. Each request will only collect one specified value; when multiple values need to be collected, separate requests must be sent. The PIN pad forms will control the length and format of the consumer’s input.

<table>
<thead>
<tr>
<th>DeviceInputIndex</th>
<th>Value Name</th>
<th>Return Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>CVV2</td>
<td>Numeric</td>
</tr>
<tr>
<td>002</td>
<td>Street Number</td>
<td>Numeric</td>
</tr>
<tr>
<td>003</td>
<td>ZIP Code</td>
<td>Numeric</td>
</tr>
<tr>
<td>004</td>
<td>Social Security Number (SSN)</td>
<td>Numeric, no formatting</td>
</tr>
<tr>
<td>005</td>
<td>Last 4 of SSN</td>
<td>Numeric</td>
</tr>
<tr>
<td>006</td>
<td>Date of Birth</td>
<td>MM/DD/YYYY</td>
</tr>
<tr>
<td>007</td>
<td>Annual Income</td>
<td>Numeric, no commas or decimals</td>
</tr>
<tr>
<td>008</td>
<td>Home Phone Number</td>
<td>Numeric, no formatting</td>
</tr>
<tr>
<td>009</td>
<td>Business Phone Number</td>
<td>Numeric, no formatting</td>
</tr>
<tr>
<td>010</td>
<td>Email Address (Requires Touchscreen)</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>011</td>
<td>Driver’s ID (Requires Touchscreen)</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>012</td>
<td>Tip</td>
<td>Numeric</td>
</tr>
</tbody>
</table>

The data collected will be returned in the DeviceInputResponse field.

This is a pass-through function and the collected values will not be stored in DOLLARS ON THE NET or passed to the processor. Shift4 Payments will not validate the consumer’s input.

Use Case 31: Prompting a Consumer for Input

Actors: Consumer, Clerk, Interface

Preconditions: A merchant is ready to collect consumer information using a UTG-controlled PIN pad.

Main Flow: This flow illustrates the consumer responding to an Input Prompt request.

1. The interface sends an Input Prompt request (FRC DB) to collect the desired value on a UTG-controlled PIN pad.
2. The consumer inputs the desired value on the PIN pad.
3. The UTG sends the consumer response to the interface in the PromptConfirmResult field.

Post Conditions: The consumer response is passed through to the interface.
Getting Device Information

Shift4 Payments created the Get Device Info request (FRC F2) to allow merchants to check the status of a specific device they’re using. This request will prompt the UTG to return useful device information in the Notes field in key-value pairs separated by a CRLF (carriage return; line-feed) pair.

To support this functionality, your interface will need to send an FRC F2 request with the API Terminal ID in the TerminalID field.

The following information may be returned:

- Number of card swipes, bad track reads, signatures, and reboots
- Terminal type
- Operating system version
- Application version
- Date of manufacture
- Information about the types of encryption keys that are injected

Additional information about the PIN pad may be returned, and the amount and type of information returned will depend on the PIN pad type.

If the PIN pad does not support FRC F2, an error will be returned.

Using Promotional Codes for Private-Label Purchases

Promotional codes are used with private-label credit card purchases to offer special financing terms, such as a no-interest period or a low promotional interest rate. At this time, Shift4 Payments only supports the use of promotional codes with Synchrony Private Label cards on a UTG-controlled PIN pad.

Promotional Codes in a Transaction Request

When a purchase request is sent with a promotional code, the UTG-controlled PIN pad will prompt the consumer to acknowledge the promotional disclosures.

Requirement: The API Option ENHANCEDRECEIPTS is required when processing promotional code transactions because the promotional terms returned in the receipt text fields must be provided to the consumer.

To include a promotional code in a transaction request, these fields must be added: EnhancedDataId and EnhancedDataValues. (These fields are in Data Block 068 Enhanced Authorization.) The EnhancedDataId field will be sent with a value of “PL” to indicate a private-label card. The EnhancedDataValues field is used to send key value pairs separated by the pipe operator ‘|’. For promotional codes, the first key value pair indicates a Promotional Code (05), which is a six digit number. The second key value pair indicates the Promo Type (F0), which is a two digit numeric code.

For example if the promotional code is 013116 and Promo Type 01, these values would be sent:

EnhancedDataId=PL
Promotional code values for testing are located in the Shift4 Payments Integration: Reference Guide located in MyPortal API Corner. The values used in production will be provided by the processor.

Promotional Codes and Refund Requests
When a refund is made to a private-label card and a promotional code was used for the sale, the interface will need to send the same codes used in the initial transaction in the refund request.

Use Case 32: Capturing a Promotional Code Sale
Actors: Consumer, Clerk
Preconditions: A merchant is ready to send an Online Sale request (FRC 1D) with a promotional code.
Main Flow: This flow illustrates an approval response with the consumer accepting appropriate promotional disclosures.

6. The transaction is sent as an FRC 1D request including a promotional code.
7. The consumer is prompted to capture (insert, swipe, tap, or key) the card on the PIN pad.
8. The UTG will guide the consumer through the appropriate steps, including appropriate promotional disclosures, on the PIN pad.
9. The UTG processes the FRC 1D request.
10. A response indicating an approval is returned.

Alternate Flow 1: This flow illustrates the consumer declining the promotional disclosures. (Main Flow; Step 3.)
Follow the steps in Use Case 13: Voiding an Invoice.

Alternate Flow 2: This flow illustrates a decline response. (Main Flow; Step 5.)
Follow the steps in Use Case 13: Voiding an Invoice.

Alternate Flow 3: This flow illustrates a referral response. (Main Flow; Step 5.)
Follow the steps in Use Case 6: Handling Voice Referrals.

Post Conditions: An approved transaction awaits settlement at the end of the day.
Processing MO/TO from a Retail Location

In certain circumstances, a retail merchant may want process a MO/TO (mail order/telephone order) transaction in store. For example, if a consumer wants to purchase a product and the desired size is not available, the merchant may want to process the transaction using their MO/TO system and send the correct item directly to the consumer.

Shift4 Payments has created processing flows that allow merchants to process these types of transactions.

Requirement: A UTG-controlled PIN pad is required to support this function.

Sending the Correct Parameters

Requirement: If a consumer is purchasing goods that are available in store as well as goods that are available through the merchant’s MO/TO system, two separate transactions must be processed with unique invoice numbers.

To process a MO/TO transaction in store, your interface should use either a TokenStore Add (FRC E0) request along with the API Option DISCARDTRACKINFO or TokenStore Duplicate (FRC E2). An FRC E0 request should be sent if the CHD is new to your system (not currently tokenized), whereas an FRC E2 should be sent if the CHD has already been tokenized, such as if an in-store retail transaction has already taken place.

When an FRC E0 is sent with the API Option DISCARDTRACKINFO, the card number and expiration date will be used to process the transaction and all remaining track data will be discarded.

The fields in Data Block 112 AVS/CVV Prompt (CVV2Prompt, StreetNumberPrompt, and PostalCodePrompt) should also be sent in the request as needed to prompt the PIN pad to collect AVS and/or CSC data. For example, if the merchant needs CSC data, but not the street number or postal code, your interface will send the following in an HTTPS request: CSC Prompt='Y', StreetNumberPrompt='N', PostalCodePrompt='N'. For a TCP/IP request, the data block would equal ‘YNN’ for this request. If AVS data has already been collected by the POS, it can be included in the request by populating the proper fields rather than collecting it using the PIN pad.

Note: The CHD, AVS, and CSC data is not validated upon collection unless an Online Sale request (FRC 1D) is sent or a Verify Card request (FRC 2F) is sent when storing the token.
### Use Case 33: Processing MO/TO from a Retail Location Using a Payment Card

**Actors:** Clerk, Consumer  

**Preconditions:** A consumer is making a purchase at a retail location that is being fulfilled by the merchant’s MO/TO facility. The merchant has no previous retail purchase or card on file. The merchant is using a UTG-controlled PIN pad.  

**Main Flow:** This flow illustrates processing a MO/TO transaction using a retail POS by creating a new TrueToken.  

1. A TokenStore Add (FRC E0) request is sent including ‘Y’ in the AVS/CVV Prompt fields and the API Option DISCARDTRACKINFO. The request is sent using the merchant’s Retail Access Token.  
2. The consumer is prompted to swipe the card using the PIN pad.  
3. The consumer is prompted to enter the requested AVS and CVV data on the PIN pad.  
4. The UTG processes the request.  
5. A response with no errors is received and a TrueToken is returned in the UniqueID field.  

**Optional Flow:** This flow illustrates a request to verify the card and AVS/CVV data.  

6. A Verify Card request (FRC 2F) is sent using the resultant TrueToken from the FRC E0 request.  
7. A response with no errors is received with the Response field equal to ‘A’, ValidAVS equal to ‘Y’, and ValidCVV equal to ‘Y’.  

**Post Conditions:** The card is tokenized and the TrueToken can be passed to the MO/TO interface. When needed, the MO/TO interface can use the TrueToken in sales using TokenShare.

### Use Case 34: Processing MO/TO from a Retail Location Using a Stored TrueToken

**Actors:** Clerk, Consumer  

**Preconditions:** A consumer is making a purchase at a retail location that is being fulfilled by the merchant’s MO/TO facility after completing a retail purchase or using a TrueToken on file. The merchant is using a UTG-controlled PIN pad.  

**Main Flow:** This flow illustrates processing a MO/TO transaction using an existing TrueToken.  

8. A TokenStore Duplicate (FRC E2) request is sent including the TrueToken from the previous Retail sale and ‘Y’ in the AVS/CVV Prompt fields. The request is sent using the merchant’s Retail Access Token.  
9. The UTG receives the request and activates the PIN pad.  
10. The consumer is prompted to enter the requested AVS and CVV data on the PIN pad.  
11. The UTG processes the request.  
12. A response with no errors is received and a new TrueToken is returned in the UniqueID field.  

**Optional Flow:** This flow illustrates a request to verify the card and AVS/CVV data.  

13. A Verify Card (FRC 2F) request is sent using the resultant TrueToken from the FRC E2 request.  
14. A response with no errors is received with the Response field equal to ‘A’, ValidAVS equal to ‘Y’, and ValidCVV equal to ‘Y’.  

**Post Conditions:** The card is tokenized and the TrueToken can be passed to the MO/TO interface. When needed, the MO/TO interface can use the TrueToken in sales using TokenShare.
Requesting a Totals Report

The Totals Report is a comma-separated list of records returned in the Notes field when a Totals Report request (FRC 5F) has been sent. Any time FRC 5F is sent, two types of records are returned: a Total Record, which indicates the total amount of sales and returns by card type, and a Final Record, which indicates the gross sales and returns by card type.

This function does not supersede the standard auditing and reporting tools that are included with Shift4 Payments’ products.

Important: Responses to Totals Report requests (FRC 5F) will be URL-encoded. Therefore, your interface must decode the data for parsing.

To interpret the FRC 5F response, please refer to the field names and descriptions in the following tables. Each record is terminated by a CRLF (carriage return; line-feed) pair.

**Total Record “T” – Totals by Card Type**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Max Bytes</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record Type</td>
<td>1</td>
<td>String</td>
<td>“T”</td>
</tr>
<tr>
<td>Card Type</td>
<td>2</td>
<td>String</td>
<td>Card Type Abbreviation (AX, DB, DC, MC, JC, NS, PL, VS)</td>
</tr>
<tr>
<td>Merchant ID</td>
<td>10</td>
<td>String</td>
<td>Merchant ID being reported</td>
</tr>
<tr>
<td>Terminal ID</td>
<td>24</td>
<td>String</td>
<td>Terminal Description being reported</td>
</tr>
<tr>
<td>Clerk ID</td>
<td>5</td>
<td>Numeric</td>
<td>Clerk ID being reported</td>
</tr>
<tr>
<td>Total Sales Count</td>
<td>6</td>
<td>Numeric</td>
<td>Zero-padded sales count for this terminal or clerk and card type</td>
</tr>
<tr>
<td>Total Returns Count</td>
<td>6</td>
<td>Numeric</td>
<td>Zero-padded credit count for this terminal or clerk and card type</td>
</tr>
<tr>
<td>Total Sale Amount</td>
<td>14</td>
<td>Currency</td>
<td>Sale total for this terminal or clerk and card type</td>
</tr>
<tr>
<td>Total Return Amount</td>
<td>14</td>
<td>Currency</td>
<td>Credit total for this terminal or clerk and card type</td>
</tr>
<tr>
<td>Total Cashback Amount</td>
<td>14</td>
<td>Currency</td>
<td>Cashback total for this terminal or clerk and card type</td>
</tr>
</tbody>
</table>
## Final Record “F” – Totals Including All Card Types

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Max Bytes</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record Type</td>
<td>1</td>
<td>String</td>
<td>“F”</td>
</tr>
<tr>
<td>Gross Sales Count</td>
<td>6</td>
<td>Numeric</td>
<td>Zero-padded sales count for this terminal and card type</td>
</tr>
<tr>
<td>Gross Returns Count</td>
<td>6</td>
<td>Numeric</td>
<td>Zero-padded credit count for this terminal and card type</td>
</tr>
<tr>
<td>Gross Sale Amount</td>
<td>14</td>
<td>Currency</td>
<td>Gross Sale total for this terminal and card type</td>
</tr>
<tr>
<td>Gross Return Amount</td>
<td>14</td>
<td>Currency</td>
<td>Gross Credit total for this terminal and card type</td>
</tr>
<tr>
<td>Gross Cashback Amount</td>
<td>14</td>
<td>Currency</td>
<td>Gross Cashback total for this terminal and card type</td>
</tr>
</tbody>
</table>

## Developing an Automated Remote Export from DOLLARS ON THE NET

Some merchants prefer to remotely generate batch settlement reports from DOLLARS ON THE NET without logging in to their account. To generate these reports, developers must develop the parameters for two API calls: a BatchSummary request (FRC BatchSummary) and a BatchDetail request (FRC BatchDetail). These API calls must be sent via direct server-to-server communication with DOLLARS ON THE NET.

- **BatchSummary** – Sending FRC BatchSummary returns a general batch summary, which includes archive information according to the data export definition configured by a DOLLARS ON THE NET administrator.
- **BatchDetail** – Sending FRC BatchDetail returns a detailed batch settlement report according to the data export definition configured by a DOLLARS ON THE NET administrator.

---

**Requirement**: The API requests for Automated Remote Export must be created using HTTP POST and sent to Shift4 Payments using direct server-to-server POST communication with DOLLARS ON THE NET.

---

## Creating and Testing API Requests for Automated Remote Export

In production, a DOLLARS ON THE NET administrator must create an API user and configure the required data export definitions for the Batch Summary and Batch Detail reports within DOLLARS ON THE NET before the API calls for Automated Remote Export can be built.

A DOLLARS ON THE NET administrator must provide the developer with the following parameters:

- The username for the API user
- The password for the API user
- The appropriate Export ID for each desired export

As previously mentioned in this document, your assigned API Analyst will act as your DOLLARS ON THE NET administrator during testing; therefore, they will provide you with these parameters for testing.
Testing will be conducted using the direct server-to-server POST URL for Shift4 Payments’ test environment: https://cfapi.shift4test.com/api/S4Tran_Action.cfm.

Tip: Shift4 Payments has also created a tool specifically for testing Automated Remote Export. This tool is available at https://dotn.shift4test.com/api/s4tran_form.cfm.

Creating and Testing a BatchSummary Request

Please follow the steps below to create and test a BatchSummary request.

1. Build a simple request for a Batch Summary report using the following data:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>REQUIRED/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>STX</td>
<td>YES</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>&lt;The DOLLARS ON THE NET account number&gt;</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>UserName</td>
<td>&lt;The username for the API user&gt;</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>Password</td>
<td>&lt;The password for the API user&gt;</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>FunctionRequestCode</td>
<td>BatchSummary</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>Notes</td>
<td>&lt;The appropriate Export ID&gt;</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>MerchantID</td>
<td>&lt;Comma-separated list of merchant IDs&gt;**</td>
<td>Optional</td>
</tr>
<tr>
<td>ReportStartDate</td>
<td>&lt;The beginning business date to return summary information in MM/DD/YYYY format.&gt;††</td>
<td>Optional</td>
</tr>
<tr>
<td>ReportEndDate</td>
<td>&lt;The ending business date to return summary information in MM/DD/YYYY format.&gt;</td>
<td>Optional</td>
</tr>
<tr>
<td>ETX</td>
<td>YES</td>
<td>REQUIRED</td>
</tr>
</tbody>
</table>

**If the MerchantID field is not sent in the request, the export will return batch summary information for all of the merchants that the API user has access to.

††If the ReportStartDate field is not sent in the request, the default value is 30 days prior to the request.

2. Post the string, including a proper HTTP header, to Shift4 Payments’ test environment using the following URL: https://cfapi.shift4test.com/api/S4Tran_Action.cfm.

3. Shift4 Payments will process the request and send a response.
   • If you did not receive a response, verify the message is formatted correctly, includes all of the required parameters defined in the table above, and that the string was posted to the proper URL.
Creating and Testing a BatchDetail Request

Please follow the steps below to create and test a BatchDetail request.

1. Build a simple request for a Batch Detail report using the following data:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>REQUIRED/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>STX</td>
<td>YES</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>&lt;The DOLLARS ON THE NET account number&gt;</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>UserName</td>
<td>&lt;The username for the API user&gt;</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>Password</td>
<td>&lt;The password for the API user&gt;</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>FunctionRequestCode</td>
<td>BatchDetail</td>
<td>Required</td>
</tr>
<tr>
<td>Notes</td>
<td>&lt;The appropriate Export ID&gt;</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>TranID</td>
<td>&lt;The Batch ID or a comma-separated list of Batch IDs you are requesting detail for&gt;</td>
<td>Optional</td>
</tr>
<tr>
<td>MerchantID</td>
<td>&lt;Comma-separated list of merchant IDs&gt;</td>
<td>Optional</td>
</tr>
<tr>
<td>ETX</td>
<td>YES</td>
<td>REQUIRED</td>
</tr>
</tbody>
</table>

If the MerchantID field is not sent in the request, the export will return batch summary information for all of the merchants that the API user has access to.

If the TranID field is not sent in the request, the export will return information on all of the unsettled transactions.

2. Post the string, including a proper HTTP header, to Shift4 Payments' test environment using the following URL: https://cfapi.shift4test.com/api/S4Tran_Action.cfm.

3. Shift4 Payments will process the request and send a response.
   - If you did not receive a response, verify the message is formatted correctly, includes all of the required parameters defined in the table above, and that the string was posted to the proper URL.

Performing the Automated Remote Export Functions in Production

After achieving successful results when testing Automated Remote Export, you can request batch settlement reports from production without logging in to DOLLARS ON THE NET via the following URL: https://dotn.shift4api.net/api/s4tran_action.cfm.

# API Integration Guide Change Log

## 10/16/15 Change Log API Integration Guide

<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughout the Doc Unique ID was updated to reflect UniqueID being the field name and TrueToken being the value.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Minor Revision to UTG Instructions.</td>
</tr>
<tr>
<td>7</td>
<td>Moved how to format HTTP Post message format info with minor fixes to before Access Token</td>
</tr>
<tr>
<td>8 - 10</td>
<td>Access Token Process – Added and updated information.</td>
</tr>
<tr>
<td>12-13</td>
<td>Generating Trace Files  added</td>
</tr>
<tr>
<td>15</td>
<td>Enhanced the Testing section.</td>
</tr>
<tr>
<td>16</td>
<td>Swipe Ahead information was removed since it is deprecated by EMV</td>
</tr>
<tr>
<td>18</td>
<td>Updated the TrueToken section</td>
</tr>
<tr>
<td>18</td>
<td>Fixed references to merchant configuration</td>
</tr>
<tr>
<td>19</td>
<td>Fixed information on E0 short term data storage</td>
</tr>
<tr>
<td>20</td>
<td>Added blank response to the flow chart. Added 9020 and 9023 communication errors</td>
</tr>
<tr>
<td>40</td>
<td>Updated information on using GCRETURN API Option</td>
</tr>
<tr>
<td>41</td>
<td>Corrected information on  voids and authorizations</td>
</tr>
<tr>
<td>45</td>
<td>Fixed use case reference</td>
</tr>
<tr>
<td>45</td>
<td>Fixed MetaToken instructions</td>
</tr>
<tr>
<td>46</td>
<td>Added information on production vs test encryption keys</td>
</tr>
<tr>
<td>46</td>
<td>Fixed TerminalID definition</td>
</tr>
<tr>
<td>47</td>
<td>Clarified receipt requirements</td>
</tr>
</tbody>
</table>
11/5/15 Change Log API Integration Guide

<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Throughout replaced customer with consumer as appropriate</td>
</tr>
<tr>
<td>4-6</td>
<td>Updated UTG install instructions</td>
</tr>
<tr>
<td>17-20</td>
<td>Added DOLLARS ON THE NET for Integrators</td>
</tr>
<tr>
<td>23-36</td>
<td>Revamped TokenStore information</td>
</tr>
<tr>
<td>30</td>
<td>Updated Basic Transaction Flow</td>
</tr>
<tr>
<td>32</td>
<td>Updated AVS and CVV Verification article</td>
</tr>
<tr>
<td>53-57</td>
<td>Added a new section on Signature Capture</td>
</tr>
<tr>
<td>58</td>
<td>Added Terms and Conditions FRC</td>
</tr>
<tr>
<td>58</td>
<td>Added Process Forms FRC</td>
</tr>
</tbody>
</table>
### 01/18/16 Change Log API Integration Guide

<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Updated the document introduction</td>
</tr>
<tr>
<td>4-7</td>
<td>Added diagrams and explanation of processing functions</td>
</tr>
<tr>
<td>8-9</td>
<td>Explained communication protocols</td>
</tr>
<tr>
<td>11-13</td>
<td>Defined the UTG and combined setup for HTTP and TCP/IP Integrations</td>
</tr>
<tr>
<td>14-15</td>
<td>Combined Shift4 API Message Formats</td>
</tr>
<tr>
<td>20-21</td>
<td>Combined HTTP Post and TCP/IP Access Token Exchange Information</td>
</tr>
<tr>
<td>26-27</td>
<td>Moved Trace File information</td>
</tr>
<tr>
<td>28</td>
<td>Fixed DOLLARS ON THE NET for Integrators title</td>
</tr>
<tr>
<td>32</td>
<td>Updated Design Considerations index</td>
</tr>
<tr>
<td>34</td>
<td>Defined Invoices and described rollbacks</td>
</tr>
<tr>
<td>40</td>
<td>Reformatted response codes to a table</td>
</tr>
<tr>
<td>41</td>
<td>Updated information on Communication Failures</td>
</tr>
<tr>
<td>59-60</td>
<td>Added information on handling Voice Referrals</td>
</tr>
<tr>
<td>65</td>
<td>Added information on Offline Processing</td>
</tr>
<tr>
<td>66</td>
<td>Added End of Day Reporting from Reference Guide</td>
</tr>
<tr>
<td>67</td>
<td>Added Notes field and Contactless Payments information</td>
</tr>
<tr>
<td>84</td>
<td>Updated Refund information</td>
</tr>
<tr>
<td>90-92</td>
<td>Enhanced receipt printing information</td>
</tr>
<tr>
<td>103</td>
<td>Added information for Cashback and Surcharge new functionality</td>
</tr>
</tbody>
</table>
### 3/22/16 Change Log API Integration Guide

<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Throughout document replaced HTTP with HTTPS</td>
</tr>
<tr>
<td>8</td>
<td>Clarified HTTP and TCP instructions</td>
</tr>
<tr>
<td>9</td>
<td>Clarified Direct Post testing instructions</td>
</tr>
<tr>
<td>19</td>
<td>Add instruction not to include Shift4 in the vendor field.</td>
</tr>
<tr>
<td>20</td>
<td>Added detail for processing TCP transactions</td>
</tr>
<tr>
<td>21</td>
<td>Corrected AuthToken field</td>
</tr>
<tr>
<td>33</td>
<td>Added Short Term data definition</td>
</tr>
<tr>
<td>37</td>
<td>Fixed FRC Name</td>
</tr>
<tr>
<td>38</td>
<td>Added definition for Initial and Subsequent transactions</td>
</tr>
<tr>
<td>42</td>
<td>Added detail for using last 4 of a credit card in place of a token</td>
</tr>
<tr>
<td>46</td>
<td>Added detail on AVS/CVV logic</td>
</tr>
<tr>
<td>54</td>
<td>Added Enhanced Transaction Data article</td>
</tr>
<tr>
<td>74</td>
<td>Added Lodging and Hospitality Field detail</td>
</tr>
<tr>
<td>80</td>
<td>Added detail for using the ReceiptTextColumns field</td>
</tr>
<tr>
<td>94</td>
<td>Added detail for configuring the UTG to allow Cash Back</td>
</tr>
</tbody>
</table>

### 5/4/16 Change Log API Integration Guide

<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reformatted, numbered, and enhanced Use Cases.</td>
</tr>
<tr>
<td>55</td>
<td>Moved AVS Response Logic to Basic Transaction Flow section and added error handling details</td>
</tr>
<tr>
<td>56</td>
<td>Added explanation of logging transactions for an auditors review</td>
</tr>
<tr>
<td>67</td>
<td>Adjusted Partial Authorization verbiage</td>
</tr>
<tr>
<td>83</td>
<td>Revised It’s Your Card content</td>
</tr>
<tr>
<td>Page</td>
<td>Change</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>105</td>
<td>Fixed incorrect MetaToken API Option</td>
</tr>
<tr>
<td>109</td>
<td>Enhanced Prompt Confirmation Function details</td>
</tr>
<tr>
<td>121</td>
<td>Added On Demand Card Read Function</td>
</tr>
<tr>
<td>123</td>
<td>Added Input Prompt Function</td>
</tr>
<tr>
<td>124</td>
<td>Added Promotional Codes for Private Label Cards</td>
</tr>
<tr>
<td>125</td>
<td>Added MO/TO transactions from a Retail Location</td>
</tr>
<tr>
<td>128</td>
<td>Added Multi-TokenStore Add</td>
</tr>
</tbody>
</table>

### 3/6/17 Change Log API Integration Guide

<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Added new copyright date and ninth patent number.</td>
</tr>
<tr>
<td>3</td>
<td>Rewrote introduction.</td>
</tr>
<tr>
<td>4</td>
<td>Removed separate section about Shift4 payment functions as it is discussed in detail in other sections of the <em>API Integration Guide</em>.</td>
</tr>
<tr>
<td>4-5</td>
<td>Updated heading and rearranged content in this section.</td>
</tr>
<tr>
<td></td>
<td>Added explanation about the different APIs and protocols we support as well as information about the benefits of using the UTG.</td>
</tr>
<tr>
<td></td>
<td>Added new Requirement callout to emphasize that Shift4’s UTG is required to support EMV and secure offline capability.</td>
</tr>
<tr>
<td></td>
<td>Added information about our support for Direct Server-to-Server POST to focus on the fact that we support these environments (and how).</td>
</tr>
<tr>
<td></td>
<td>Added a new Requirement callout to emphasize that the initial testing for server-to-server processing requires the use of the UTG.</td>
</tr>
<tr>
<td></td>
<td>Added simple explanations describing the difference between HTTPS POST and TCP/IP.</td>
</tr>
<tr>
<td></td>
<td>Added screenshots illustrating the difference between HTTPS POST and TCP/IP.</td>
</tr>
<tr>
<td>Page</td>
<td>Change</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
</tr>
</tbody>
</table>
| 6-16  | • Updated heading and added an introduction explaining how installing the UTG fits into the Shift4 integration process.  
• Added Note callout to inform developers about the potential for IP addresses outside the U.S. to be blocked, and who to contact for assistance. 
• Added an Important callout directing cloud-based systems developers to UTG4Cloud installation instructions.  
• Updated text for UTG installations.  
• Changed the Warning about closing the unique code window to a Note callout.  
• Updated the Note callout about the API Interfaces tab.  
• Added information about the use of the API Terminal ID field and as well as a Note callout about what it does.  
• Added a Note callout about the need to virtualize the COM port if using a device with a serial connection.  
• Updated tip about UTG service vs. standalone.  
• Added UTG4Cloud UTG installation instructions. |
| 17-19 | • Updated heading, section formatting, and added simplified sub-headings.  
• Added information about ASCII.  
• Added information about the RequestorReference field.  
• Expanded explanation about HTTPS POST, including information about URL Encoding and URL Decoding. |
| 20    | • Updated heading, sub-headings, and reorganized section content.  
• Updated information about the Client GUID.  
• Updated information about the Auth Token.  
• Updated information about generating a new Access Token. |
| 21-24 | • Updated heading, introduction, and sub-headings.  
• Added a reminder about where to go for the formatting information.  
• Added a Requirement callout about the need to hard-code the Client GUID into the application and why.  
• Replaced the Note callout about the Vendor parameters with a notation about where to find the Vendor parameters.  
• Replaced the Note callout about the Vendor parameters with a notation about where to find the Vendor parameters. |
<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
</table>
| 25-29 | • Updated heading.  
        • Changed Important callout to a Note and updated wording.  
        • Minor updates to the formatting of the HTTPS POST request.  
        • Added a Requirement callout to remind developers using HTTPS POST that their application must URL encode all values prior to concatenation.  
        • Added a notation about where to find the Vendor parameters.  
        • Added information about how to verify the transaction details were displayed and what to do if they did not receive a response.  
        • Minor updates to the TCP/IP request.  
        • Added a notation about where to find the Vendor parameters. |
| 30-31 | • Updated heading and introduction.  
        • Updated “certification” environment to “test” environment in the Note callout.  
        • Arranged test card information into its own section.  
        • Removed separate warning that only test card numbers should be used.  
        • Added a Note callout to emphasize that the test environment is not designed for performance or load testing.  
        • Added *Understanding Trace Files* section.  
        • Added an Important callout to provide essential information about where trace files are saved and how to verify they’re being saved in the correct way.  
        • Updated instructions for collecting trace to “Collecting Trace Files” rather than “Creating Trace Files”.  
        • Added a screenshot to walk developers through the process of collecting trace files.  
        • Added instructions about *Rolling Over Trace Files*.  
        • Added an Important callout to remind developers that every test should be isolated into its own trace file. |
<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
</table>
| 32-37 | • Updated the section heading and general section information to ensure it focused on the use of DOLLARS ON THE NET in testing, not production.  
• Made minor text changes to the Warning callout emphasizing that the instructions in this section are provided for development and certification only and should not be used in production.  
• Rearranged the section, updated sub-headings, and included more sub-headings to improve navigation and readability.  
• Added a Note callout to skip to step 3 if there is one merchant account assigned to the test account profile (instead of multiple accounts that they need to select from).  
• Updated and reformatted the instructions.  
• Updated the wording in the Searching for Transactions section.  
• Added information about Editing Transactions in Groups.  
• Changed Batch Processing to Processing Batches and Problem Transactions to Voiding Problem Transactions.  
• Added a specific part about Voiding Authorizations.  
• Updated/added instructions that provided more detail about test environment usage of DOLLARS ON THE NET.  
• Updated the wording in another Warning callout about the instructions being only for development and certification testing only.  
• Updated the instructions for Closing Batches, including step-by-step screenshots.  
• Moved the Reporting and Exporting Transaction Reports sub-sections to the end of the section. |
| 38    | • Updated the introduction to the Design Considerations section.  
• Added a reminder to ensure that developers review the sections that pertain to their Shift4 Integration Plan. |
| 39    | • Updated the Understanding TrueTokens heading and made minor updates to the introduction.  
• Updated the description of a TrueToken.  
• Added a Tip callout about how to determine whether a TrueToken’s expiration date exceeds the card’s expiration date prior to storage. |
<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
</table>
| 40-46 | • Updated heading.  
• Added a short definition/description of what Shift4’s TokenStore is.  
• Added simple definitions about TokenShare and Global TokenStore are.  
• Added description about how TokenShare works.  
• Added an illustrated step-by-step example.  
• Added description about how Global TokenStore works.  
• Added an illustrated step-by-step example.  
• Updated the descriptions of TokenStore Functions, TokenStore Add, and TokenStore Duplicate.  
• Updated Use Case 1: Local TokenStore Add without Card Validation. Added an optional flow where the card can be validated before storing a TrueToken.  
• Minor updates to Use Case 2: Global TokenStore Duplicate Using an Existing TrueToken. |
| 47    | • Updated section heading and introduction.  
• Clarified that a UTG-controlled P2PE PIN pad configured specifically for Shift4’s test environment is required.  
• Added an Important callout specifying that Shift4’s P2PE test key is different from the production key and information about getting a Shift4 test key.  
• Added a Requirement callout about the use of API Terminal ID.  
• Updated information about the required use of the Enhanced Receipts API Option. Directed developers to read in detail in the Receipt Printing Requirements section.  
• Added a Note callout explaining that Shift4 does not require the use of chip cards during testing. |
<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
</table>
| 48-50 | • Updated heading and section text to be more explicit about Shift4’s receipt printing processes.  
  • Added a Requirement callout to clarify that receipt text is generated by the processor and that developers must ensure their interface is able to handle variable text as returned by the processor.  
  • Added a Warning callout warning developers that receipt text should never be parsed or manipulated in any way.  
  • Removed callout briefly mentioning which requests require receipts since this is covered in the *Presenting Receipts* subsection.  
  • Expanded explanatory information about *Receipt Text Columns*, *Presenting Receipts*, and the differences between *General Receipts* and *Enhanced Receipts*.  
  • Created a table to outline the differences and to emphasize why we recommend supporting enhanced receipts.  
  • Outlined requirements for supporting the Enhanced Receipts API Option.  
  • Replaced sample receipt image.  
  • Added a Requirement callout that the receipt text Shift4 returns must be presented on the same receipt as the transaction details.  
  • Pointed developers to the *Shift4 Integration: Reference Guide* for more information about additional API Options related to receipt configuration. |
| 51-55 | • Updated heading and introduction.  
  • Minor updates to the UTG4Gloud diagram and related information.  
  • Added instructions about how to prompt a UTG-controlled PIN pad.  
  • Made minor updates to the *HTTP Device Credentials Request Examples*. |
| 56-58 | • Updated section text, sub-headings, organization, and associated use cases. |
| 59 | • Updated *Basic Transaction Flow* diagram. |
| 60-61 | • Updated heading and section text.  
  • Added Requirement callout about using an Unknown Status Report.  
  • Created table for *Communication Error Codes*. |
| 62-63 | • Updated text in the *Handling Referral Responses* section and *Use Case 5: Handling Voice Referrals*. |
| 64-66 | • Updated text in the *Understanding AVS and CSC Verification* section.  
  • Updated formatting of the *AVS Processing Flow* diagram. |
<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
</table>
| 67    | • Updated heading.  
      | • Updated information about the requirements for offline processing and how it works.  
      | • Added a Warning callout about the risks of processing offline (and specifying that the benefits of being able to accept transactions while offline may be worth the risks in certain merchant environments).  
      | • Added explanation that we’ll run a few test transactions to ensure that offline processing can be supported.  |
| 68    | • Updated heading and made minor text updates.  
      | • Added a Requirement callout about the Unknown Status Report.  
      | • Created new sample end-of-day report. |
| 69-70 | • Updated heading and text for clarity and accuracy.  
      | • Added a use case for Partial Authorization. |
| 71    | • Updated heading.  
      | • Added an example to further explain the use of split tender. |
| 72    | • Updated heading and made minor text updates.  
      | • Added a Note callout to specify that testing NFC/contactless functionality is unnecessary because our UTG is already certified to support it. |
| 73    | • Updated heading and made minor text updates. |
| 74-75 | • Updated heading and text throughout the section.  
      | • Added information about Using Level 2 Data and Purchasing Card Data in an API Call.  
<pre><code>  | • Updated the table text and added the number of bytes in each field. |
</code></pre>
<p>| 76    | • Updated heading and added two more paragraphs of explanation. |</p>
<table>
<thead>
<tr>
<th>Page Range</th>
<th>Change</th>
</tr>
</thead>
</table>
| 77-81      | • Updated heading.  
• Updated information about authorization and capture, straight capture, and settlement.  
• Added a Note callout explaining that not every card type supports authorization and capture flows (e.g., debit authorizations).  
• Added a Note callout explaining that the use cases in this section are an example and do not replace the flows outlined in a Shift4 Integration Plan.  
• Added a Warning callout recommending the use of a P2PE device for card-present transactions.  
• Updated and rearranged all of the use cases for capturing sales. |
| 82-87      | • Updated heading and rearranged information in subsections.  
• Added a Note callout explaining that administrative access to DOLLARS ON THE NET is required to test gift card numbers.  
• Moved IYC API Options above IYC Gift Card States.  
• Updated the gift card processing use cases. |
| 88         | • Updated heading. |
| 89-91      | • Updated heading and introduction.  
• Updated information in the lodging fields.  
• Added a Note callout to emphasize that the examples given do not cover all possible lodging transactions. |
| 92-93      | • Updated heading.  
• Added information about what EBT is and the required API Options associated with processing EBT. Minor text updates.  
• Minor updates to the use cases. |
| 94-95      | • Updated heading.  
• Removed information about token migration.  
• Added more detailed information about the MetaToken process.  
• Minor text updates to the MetaToken use cases. |
| 96         | • Updated heading.  
• Minor updates to headings and introductions to the use cases. |
<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
</table>
| 97-102 | • Updated heading, subheadings, and introduction.  
       | • Added a Note callout about digital signature capture and the UTG.  
       | • Made a table about digital signature capture function requests.  
       | • Added more information about how to interpret the instructions for an example signature capture.  
       | • Updated text in Note callout for clarity.  
       | • Added Important callout about how to interpret the example.  
       | • Updated instances of “pen” to “stylus”.  
       | • Updated the 8-byte coordinates given for “O”.  
       | • Updated text introducing the example of a digital signature capture in a DOLLARS ON THE NET transaction record.  
       | • Updated text description of a complete letter “H” in raw trace.  
       | • Updated text for sending digital signature capture data in PNG format.  
       | • Changed Note callout about the PhotoType field to a Requirement since it describes a requirement. |
| 103   | • Updated heading, section text, and use case. |
| 104   | • Updated heading and made minor text updates. |
| 105   | • Updated heading and section text.  
       | • Added a Requirement callout about the requirements for supporting cash back.  
       | • Added a new cash back flow based on tests with a PIN pad. |
| 106-107 | • Updated heading, section text, and use case. |
| 108   | • Updated heading and made minor section text updates. |
| 109-110 | • Updated heading.  
<pre><code>   | • Minor text updates to the explanation and use case. |
</code></pre>
<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>111-113</td>
<td>• Updated heading.</td>
</tr>
<tr>
<td></td>
<td>• Added a Requirement callout that a UTG-controlled PIN pad is required to support this function.</td>
</tr>
<tr>
<td></td>
<td>• Added a subsection about sending the correct parameters.</td>
</tr>
<tr>
<td></td>
<td>• Added a Requirement callout to underscore that a separate transaction needs to be processed for goods that are available in store.</td>
</tr>
<tr>
<td></td>
<td>• Updated information for accuracy and clarity.</td>
</tr>
<tr>
<td></td>
<td>• Updated use case information.</td>
</tr>
<tr>
<td></td>
<td>• Removed Multi-Token Store Add</td>
</tr>
</tbody>
</table>

### 4/7/17 Change Log API Integration Guide

<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Formatting</strong></td>
</tr>
<tr>
<td></td>
<td>o Reformatted all of the tables for consistency across API documents.</td>
</tr>
<tr>
<td></td>
<td><strong>Mechanics</strong></td>
</tr>
<tr>
<td></td>
<td>o Ensured all instances of Authorization with Referral Code and Sale with Referral code had “with” lowercase for consistency across all API documents (and with CMoS).</td>
</tr>
<tr>
<td></td>
<td>o Updated use case numbers.</td>
</tr>
<tr>
<td></td>
<td>o Added trademark to MetaToken.</td>
</tr>
<tr>
<td>25-31</td>
<td><strong>Sending Your First Transaction Request</strong></td>
</tr>
<tr>
<td></td>
<td>o Updated the token exchange API call info to incorporate Level 2 Data and Purchasing Card Data.</td>
</tr>
<tr>
<td>37-38</td>
<td><strong>Using the DOLLARS ON THE NET Web Interface during Testing</strong></td>
</tr>
<tr>
<td></td>
<td>o Updated screenshots of DOLLARS ON THE NET to ensure that the new Referrals Only option is included in the navigation.</td>
</tr>
<tr>
<td>40</td>
<td><strong>Design Considerations</strong></td>
</tr>
<tr>
<td></td>
<td>o Added new sections to the list of design considerations</td>
</tr>
<tr>
<td>50</td>
<td><strong>Quick Chip EMV</strong></td>
</tr>
<tr>
<td></td>
<td>o Added section.</td>
</tr>
<tr>
<td>Page</td>
<td>Change</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
</tr>
</tbody>
</table>
| 51-52 | - Supporting “Swipe Ahead” for Faster Transactions at the POS  
| |   - Added section and use case. |
| 56-57 | - Supporting Built-In Receipt Printing Capability on Devices  
| |   - Added section. |
| 60, 62 | - Using UTG4Cloud in API Requests  
| |   - Minor text updates. |
| 67, 68 | - Timeouts and Communication Failures  
| |   - Added information about when to set the Global Timer in FRC 96 requests.  
| |   - Updated explanation of what to do When a Communication Failure Occurs. |
| 81-82 | - Level 2 Data and Purchasing Card Data  
| |   - Minor text updates to the table to match the table in the API Integration: Reference Guide. |
| 114  | - Sending the Input Prompt Function  
| |   - Minor text updates to the DeviceInputIndex table to match the table in the API Integration: Reference Guide. |
| 115  | - Getting Device Information  
| |   - Added section. |
| 122-123 | - Requesting a Totals Report  
| |   - Updated text. |
| 124-126 | - Developing an Automated Remote Export from DOLLARS ON THE NET  
| |   - Added section. |

### 5/10/17 Change Log API Integration Guide

<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
</table>
|       | - Formatting  
| |   - Minor updates to table formatting for improved readability in PDF format.  
| | - Mechanics  
| |   - Updated all instances of FRC 1B from Authorization to Online Auth. |
Updated all instances of FRC 1D from Sale to Online Sale.
- Updated all instances of FRC 05 from Authorization with Referral Code to Offline Auth.
- Updated all instances of FRC 06 from Sale with Referral Code to Offline Sale.
- Minor updates to the sentence structure of the FRCs for gift cards to better match updates made in the other API documents.

**Supporting Digital Signature Capture**
- Based on testing, updated the rules applying to all digital signature captures to indicate that the “pen up” (%) character will be recorded at the beginning of the SignatureBlock field.
- Updated data in tables to ensure that it accounted for the “pen up” (%) character showing up at the beginning of the SignatureBlock field in testing.
- Updated two note callouts to indicate the “pen up” (%) is recorded at the beginning of the SignatureBlock field.

### 06/15/18 -- Change Log API Integration Guide

<table>
<thead>
<tr>
<th>Page</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Formatting</strong></td>
<td>o New Shift4 Payments template applied to API Integration Guide.</td>
</tr>
<tr>
<td>6-10</td>
<td><strong>Performing a Basic UTG Installation</strong></td>
</tr>
<tr>
<td></td>
<td>o Replaced “enter your’ with “the provided”.</td>
</tr>
<tr>
<td></td>
<td>o Moved note in Step 5 to sub-step A.</td>
</tr>
<tr>
<td></td>
<td>o Added note for Step 8</td>
</tr>
<tr>
<td>13-14</td>
<td><strong>Performing a UTG Cloud Installation</strong></td>
</tr>
<tr>
<td></td>
<td>o Moved note in Step 5 to sub-step A.</td>
</tr>
<tr>
<td></td>
<td>o Replaced “enter you” with ‘the provided”.</td>
</tr>
<tr>
<td>16-17</td>
<td><strong>Installing a Central UTG</strong></td>
</tr>
<tr>
<td></td>
<td>o Replaced “enter your” with “the provided”.</td>
</tr>
<tr>
<td></td>
<td>o Moved note in Step 1 to sub-step A.</td>
</tr>
<tr>
<td>18-20</td>
<td><strong>Disabling the UTG2 Service</strong></td>
</tr>
<tr>
<td></td>
<td>o New section</td>
</tr>
<tr>
<td>23</td>
<td><strong>Understanding the Access Token Process</strong></td>
</tr>
<tr>
<td></td>
<td>o Updates to the sentence structure of the Client GUID to emphasize importance of GUID.</td>
</tr>
<tr>
<td></td>
<td><strong>Performing a Token Exchange</strong></td>
</tr>
<tr>
<td></td>
<td>o Removed Card Number row from Transaction Header Data Block.</td>
</tr>
<tr>
<td>24-25</td>
<td><strong>Performing a Token Exchange</strong></td>
</tr>
<tr>
<td></td>
<td>o Added Functional Request Code for first instance of acronym (FRC).</td>
</tr>
<tr>
<td></td>
<td>o Added footnote for formatting time.</td>
</tr>
<tr>
<td>29</td>
<td><strong>Sending Your First Transaction Request</strong></td>
</tr>
<tr>
<td>Page</td>
<td>Change</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>32</td>
<td>• <strong>001 Transaction Block</strong>&lt;br&gt;  o Added footnotes for formatting Time and SaleFlag use.</td>
</tr>
<tr>
<td>37</td>
<td>• <strong>Changing Administrative Level Setting on Your Test Account</strong>&lt;br&gt;  o Added Its Your Card (IYC) for first instance of acronym.</td>
</tr>
<tr>
<td>43</td>
<td>• <strong>Design Considerations</strong>&lt;br&gt;  o Removed ‘s’ from title to read: “Understanding True Token”&lt;br&gt;  o Added new entry: “Preventing Double Charges on Amended Transactions”.</td>
</tr>
<tr>
<td>44</td>
<td>• <strong>Understanding True Tokens</strong>&lt;br&gt;  o Removed ‘s’ from title to read: “Understanding True Token”</td>
</tr>
<tr>
<td>54</td>
<td>• <strong>Supporting “Swipe Ahead” for Faster Transactions at the POS</strong>&lt;br&gt;  o Added Point of Sale (POS) as first instance of acronym.</td>
</tr>
<tr>
<td>68</td>
<td>• <strong>Basic Transaction Flow</strong>&lt;br&gt;  o Edited and replaced Basic Transaction Flowchart</td>
</tr>
<tr>
<td>70</td>
<td>• <strong>When a Communication Failure Occurs</strong>&lt;br&gt;  o Updated sentence structure for clarity.&lt;br&gt;  o Added reference to Basic Transaction Flow.</td>
</tr>
<tr>
<td>73</td>
<td>• <strong>Understanding AVS and CSC Verification</strong>&lt;br&gt;  o Deleted original section</td>
</tr>
<tr>
<td>74</td>
<td>• <strong>One and Two Pass Verification</strong>&lt;br&gt;  o Added new section</td>
</tr>
<tr>
<td>75</td>
<td>• <strong>AVS Processing Flow</strong>&lt;br&gt;  o Deleted flowchart; added AVS/CVV process in Basic Transaction Flowchart.</td>
</tr>
<tr>
<td>76</td>
<td>• <strong>Offline Processing</strong>&lt;br&gt;  o Added special auth code range format for offline transactions.</td>
</tr>
<tr>
<td>77</td>
<td>• <strong>End of Day Reporting</strong>&lt;br&gt;  o Added description/explanation of the report screen capture.</td>
</tr>
<tr>
<td>78</td>
<td>• <strong>Preventing Double Charges on Amended Transactions</strong>&lt;br&gt;  o New section added</td>
</tr>
<tr>
<td>91</td>
<td>• <strong>Hotel Check Out Fields (Data Block 003)</strong>&lt;br&gt;  o Added ArrivalDate field and definition.</td>
</tr>
<tr>
<td>92</td>
<td>• <strong>Hotel Advanced Deposit</strong>&lt;br&gt;  o Updated Arrival Date and Departure Date information. The date range was inaccurate.</td>
</tr>
<tr>
<td>99</td>
<td>• <strong>Digital Signature Capture API Options</strong>&lt;br&gt;  o Added PLCCSIGNATURE API Option</td>
</tr>
</tbody>
</table>