Blackboard transact™ | Cashnet®

Web Services

Integration Guide

Version 3.3 | March 2018

Trademarks & Copyright

www.cashnet.com | info@cashnet.com

Disclaimers

This document contains confidential information and is intended for distribution to authorized external parties only.

The content of this document is provided for informational use only and is subject to change without notice. Blackboard Inc. assumes no liability or responsibility for any errors or inaccuracies that may appear in this document.

Contact Information

For information and support prior to the deployment process, contact your Cashnet Project Manager.

For post-deployment support, contact Cashnet Support at (800) 231-9182 or support@cashnet.com.

About This Document

Scope

This document provides an overview of the real-time integration process using web services. Included in this document are the data fields that can be passed into Cashnet and posted to the client ERP or other campus system, several examples of web service lookups and posts/responses, and a preview of the web service response schema.

Audience

This document is intended for Cashnet clients and business partners who wish to integrate external programs with Cashnet using web services. This document assumes that the audience is familiar with web services and their standard configurations.

Glossary

The table below defines terms relevant to this document's content.

Table 1: Document glossary.

TERM	DEFINITION
1098-T	Cashnet module that offers the option for students to view 1098-T tax statements in electronic format.
Cashnet-ERP Interface	Communication mechanisms between the Cashnet application and campus systems.
eBill	Cashnet module that automates bill presentment, displaying a comprehensive view of all account information including non-tuition charges.
eRefund	Cashnet module that provides the ability to issue electronic refunds to students through ACH direct deposit
Extract	Type of interface that is used to send information about activities that occur in Cashnet to other systems.
Import	Type of interface that is used to retrieve information from other systems and load that information into Cashnet.
Note Code	Cashnet identifier for an account restriction, hold, or derogatory record.
Payment Plan Software (IPP)	Cashnet module that offers multiple tools for institutions to develop flexible tuition payment terms.

References

This document refers to the following documents and other resources.

Table 2: Referenced documents.

TITLE	AUTHOR	URL / FILE LOCATION
Transaction Inquiry Service Quick Guide	Cashnet	Support Library > Modules and Services > Inquiry and Reporting
SAO Summary Service Quick Guide	Cashnet	Available upon request

Contents

1 Web Services Integration Overview

This section answers some of the main questions regarding the purpose of ERP integration and how it can be done using web services.

1.1 What is the Purpose of ERP Integration?

One of Cashnet's primary strengths is its ability to integrate in real-time as well as batch with ERP or other campus systems.

There are many facets to ERP integration, but the main purposes are to:

- **Retrieve student balances** when students use the Cashnet suite of payment products, so that all the information displayed in Cashnet is up-to-date.
- **Post payment information** to the campus system after a customer makes a payment in Cashnet.
- **Post status information** to the student system after students' statuses change in a Cashnet module such as 1098-T, eRefund, or Payment Plans.

1.2 What Are Interfaces?

The communication mechanisms between the Cashnet application and campus systems are known as **interfaces**.

1.2.1 Cashnet Interface Terminology

Cashnet interfaces are categorized by the direction in which data flows:

- **Imports** are used to retrieve information from other systems and load that information into Cashnet. Imports usually contain data about who Cashnet customers are, what their balances are, and whether there any restrictions on their accounts.
- **Extracts** are used to send information about activities that occur in Cashnet to other systems. Extracts usually contain information about payments made in Cashnet.

Interfaces are also categorized by the method in which the data is transferred:

- **Batch interfaces** exchange information through text files, which are transferred between Cashnet and the other system through the FTP server once per day (or in some cases less frequently).
- **Real-time interfaces** exchange information through instantaneous requests, which Cashnet makes to the other system whenever a specific action is performed, such as a customer logging in or completing a payment.

Because only real-time interfaces may use web services to exchange information, this document focuses specifically on real-time interfaces.

In many cases, your institution will also want to exchange batch files with Cashnet for certain information. For more details on that topic, ask your Cashnet representative about additional documentation.

1.3 What Is a Web Service?

A web service is an application programming interface (API) that allows systems to exchange information over the Internet at a network address. A web service is always on and can therefore be called by systems with access at any time.

1.4 How Do I Configure Web Services for Cashnet?

During a client's deployment of the Cashnet Payment Platform, the Cashnet Deployment Team works with the client's technical team to integrate the systems in real-time.

To enable Cashnet to perform real-time lookup and posting via web services, Cashnet **Clients** are responsible for:

1. Creating the web service mechanism that Cashnet will consume.
2. Providing the school's Web Services Description Language (WSDL) file containing the web service scheme to the Cashnet Deployment Team.

Note: The Cashnet environment supports both the SOAP and REST protocols for web services.

2 Real-Time Lookup Data

This section lists the information that Cashnet's real-time lookup may request from the client's campus system.

The lookup request from Cashnet will be HTTPS Post or Get to the client's application. The request will contain the web service data, as shown in the following examples.

Note: For the complete lookup schema, see Appendix A on page 25.

2.1 Student Demographic Information

The table below lists the student demographic information that may be requested by Cashnet from the ERP system.

Note: In the tables throughout this document, the tags identify how the fields are coded in the Cashnet system.

Note: The max length values are only shown for the text data types.

Table 3: Student Demographic Information

PIECE OF INFORMATION	CASHNET TAGS	REQUIRED, OPTIONAL, OR CONDITIONAL?	MAX LENGTH	REMARKS
Student ID	cust_code	Required	50	
Last Name	last_name	Optional	30	
First Name	first_name	Optional	20	
Customer Group	cust_group	Optional	1	
PIN/Password	cust_pin	Conditional—required if SSO is not used	12	
Address Line 1	address_1	Optional	50	
Address Line 2	address_2	Optional	50	
Address Line 3	address_3	Optional	50	
City	city	Optional	30	
State	state	Optional	2	
Zip	zip	Optional	10	
Country	country	Optional	50	
Area code	area_code	Optional	3	
Phone Number	phone_number	Optional	10	

PIECE OF INFORMATION	CASHNET TAGS	REQUIRED, OPTIONAL, OR CONDITIONAL?	MAX LENGTH	REMARKS
Email Address	email_id	Optional	70	
Date of Birth	dob	Optional		mm/dd/yyyy format
Student Attribute 1	custattr_1	Optional	50	Specific to student
Student Attribute 2	custattr_2	Optional	50	Specific to student
Student Attribute 3	custattr_3	Optional	50	Specific to student
Student Attribute 4	custattr_4	Optional	50	Specific to student
Alternate ID – 1	alternateid_1	Optional	50	
Alternate ID Type – 1	alternateid_1_type	Optional	20	
Alternate ID – 1 Email	alternateid_1_email	Optional	70	
Alternate ID – 1 Active	alternateid_1_activeflag	Optional	1	Y/N
Alternate ID – 2	alternateid_2	Optional	50	
Alternate ID Type – 2	alternateid_2_type	Optional	20	
Alternate ID – 3	alternateid_3	Optional	50	
Alternate ID Type – 3	alternateid_3_type	Optional	20	
Alternate ID – 4	alternateid_4	Optional	50	
Alternate ID Type – 4	alternateid_4_type	Optional	20	

2.2 Student Balances

The table below lists the student balance information that may be requested by Cashnet from the campus system.

Note: The balance information that is retrieved will vary depending on how the client chooses to display balances in Cashnet. Balances can be displayed as a summary, by term, or in some other manner supported by the campus system. For additional details, ask your Deployment Team.

Table 4: Student Balances Information

PIECE OF INFORMATION	CASHNET TAGS	REQUIRED, OPTIONAL, OR CONDITIONAL?
Item Code	item_code	Required

PIECE OF INFORMATION	CASHNET TAGS	REQUIRED, OPTIONAL, OR CONDITIONAL?
Bill Description	bill_desc	Optional
Bill Number	bill_no	Optional
Balance Due	balance_due	Required
Minimum Due	minimum_due	Optional
Other Data	balotherdata	Optional

2.3 Installment Plan Balances

The table below lists the installment plan balance information that may be requested by Cashnet from the campus system.

Note: The balance information that is retrieved will vary depending on how the client chooses to display balances in Cashnet. Balances can be displayed as a summary, by term, or in some other manner supported by the campus system. For additional details, ask your Deployment Team.

Table 5: Student Balances Information

PIECE OF INFORMATION	CASHNET TAGS	REQUIRED, OPTIONAL, OR CONDITIONAL?
Term Code	term_code	Required
Status	status	Optional
Grade Level	gradelevel	Optional
IPP Source Amount Code	ippsourceamountcode	Optional
Amount	Amount	Required

2.4 Note Codes

The table below lists the student note code information that may be requested by Cashnet from the campus system.

Note: Note codes in Cashnet are commonly referred to as holds, derogatory codes, or service restrictions in other systems.

Table 6: Student Balances Information

PIECE OF INFORMATION	CASHNET TAGS	REQUIRED, OPTIONAL, OR CONDITIONAL?	MAX LENGTH
Note Code	note_code	Optional	10
Expiration Date	note_exp_date	Optional	

2.5 Dynamic Bill Data

The table below lists the student recent activity information that may be requested by Cashnet from the campus system. This information is used to populate Cashnet's Dynamic Bill data in the ePayment site (aka Student Account Online), and is sent in an XML string within the real-time lookup.

Note: The Dynamic Bill data must be in an XML schema agreed upon with the Cashnet Deployment Team.

Table 7: Student Recent Activity Information

PIECE OF INFORMATION	CASHNET TAGS	REQUIRED, OPTIONAL, OR CONDITIONAL?	MAX LENGTH	REMARKS
Recent Activities	cust_other_data	Optional	2,147,483,647	Must be in XML format

2.6 Example Lookup Request

Below is an example lookup request.

Note: For simplicity, the examples in this document only show the XML relevant to the request. The web services wrapper and SOAP envelope have been removed.

```
<Cashnet>
   <studentid>P987654321</studentid>
   <command>lookup</command>
   <lookuptype>summary</lookuptype>  <!-- The value for 'lookuptype' may vary depending
on your Cashnet configurations. Confirm the value with your Cashnet Deployment Team. -->
<Cashnet>
```

2.7 Example Lookup Responses

Below are example XML responses returned to the Cashnet application.

Example Successful Lookup Response:

```
<?xml version="1.0" encoding="utf-8"?> <CustomerLookupResultDS
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <LookupResult>
   <result>0</result>
   <respmessage>Successful lookup</respmessage>
   <cust_code>20204</cust_code>
   <last_name>Mohr</last_name>
   <first_name>Robert</first_name>
   <address_1>401 Valley St</address_1>
   <city>Nevada City</city>
   <state>CA</state>
   <zip>95959-3013</zip>
   <country>USA</country>
   <area_code>198</area_code>
   <phone_number>325-1144</phone_number>
   <email_id>email@myuniversity.net</email_id>
   <custotherdata><![CDATA[<?xml version="1.0" encoding="utf-8" ?><activities><activity>
                <activity_description>Previous Balance</activity_description>
                <amount>3000.00</amount>
          </activity>
          <activity>
                <activity_date>02/01/2012</activity_date>
                <activity_description>Tuition</activity_description>
                <amount>3100.00</amount>
          </activity>
          <activity>
                <activity_date>02/15/2012</activity_date>
                <activity_description>Payment</activity_description>
                <amount>-437.28</amount>
          </activity>
          <activity>
                <activity_date>02/21/2012</activity_date>
                <activity_description>Current Balance Due</activity_description>
                <amount>5662.72</amount>
          </activity></activities>]]>
   </custotherdata>
   <dob>2/21/1998</dob>
  </LookupResult>
  <items>
   <item_code>BALANCE</item_code>
   <bill_desc>Balance</bill_desc>
   <bill_no>Spring 2012</bill_no>
   <balance_due>5662.72</balance_due>
```

```
    <minimum_due>5662.72</minimum_due>
    <balotherdata></balotherdata>
  </items>
  <IPPSourceAmount>
    <termcode>Spring 2012</termcode>
    <ippsourceamountcode>IPPBALANCE</ippsourceamountcode>
    <amount>5662.72</amount>
  </IPPSourceAmount>
  <notes>
    <multi_note_code>A_SPECIAL</multi_note_code>
    <multi_note_exp_date>06/30/2016</multi_note_exp_date>
  </notes>
  <notes>
    <multi_note_code>H_STUACCTS</multi_note_code>
  </notes>
</CustomerLookupResultDS>
```

Example Failed Lookup Response:

```
<?xml version="1.0" encoding="utf-8"?> <CustomerLookupResultDS
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<LookupResult>
    <result>100</result>
    <respmessage>User not found</respmessage>
</LookupResult>
</CustomerLookupResultDS>
```

2.8 Real-Time G/L Validation

G/L validation is required to ensure that an item code's G/L or the G/L entered by the cashier is valid in the client system.

Cashnet offers both batch and real-time options for performing G/L validation. If the client decides to use real-time validation, this can be done using a real-time web service request.

2.8.1 Real-Time Validation Information

The table below lists the G/L validation information that Cashnet may send to the campus system.

Table 8: Real-Time Validation Information

PIECE OF INFORMATION	REQUIRED, OPTIONAL, OR CONDITIONAL?
G/L Code	Required
Segment Type	Required
G/L Code Description	Optional
Alt. G/L Code	Optional
G/L Attribute 1	Optional
G/L Attribute 2	Optional
G/L Attribute 3	Optional
G/L Attribute 4	Optional

2.8.2 Example G/L Validation Request

Below is an example G/L validation request.

```
<Cashnet>
   <command>glvalidate</command>
   <glaccount>4261070IT61070CAFELO      MAINTEMAINTE-JUL 2016 CAF CHK#1228
</glaccount>
</Cashnet>
```

2.8.3 Example G/L Validation Response

Below are example XML responses returned to the Cashnet application.

Example Successful Validation Response:

```
<Cashnet>
   <result>0</result> <!-- Where '0' indicates success and non-zero is associated with a
specific error -->
   <message>Valid G/L account</message>
</Cashnet>
```

Example Failed Validation Response:

```
<Cashnet>
   <result>100</result> <!-- A specific error code that is associated with a detailed
error message -->
   <message>Error message</message>
</Cashnet>
```

3 Real-Time Post Data

This section lists the information that Cashnet's real-time posts may write to the client's campus system.

The post requests from Cashnet will be HTTPS Post or Get to the client's application. The request will contain the web service data, as shown in the following examples.

3.1 Background: Post Types

There are two main categories of posts: **transaction posts**, which record information about payments made in Cashnet, and **status posts**, which record information about students' status in Cashnet modules.

3.1.1 Transaction Post Types

There are four types of transaction posts used for different transactional scenarios.

Table 9: Types of Transaction Posts

POST TYPE	USE CASE	PURPOSE IN CAMPUS SYSTEM
Payment	Used when a student makes a payment against a balance loaded into Cashnet from the ERP system.	Negates the balance, or portion thereof, which exists in the campus system.
Charge	Used when a student purchases an optional item or service that is not listed as an existing balance in the campus system.	Records the charge on the student account and sends the payment details to the campus system.
Refund	Used when an existing transaction is refunded by an administrator.	Reverses the original transaction (or creates a new, negative transaction) in the campus system.
Deposit	Used when a student deposits funds to an account.	Adds a balance to an account which exists in the campus system

3.1.2 Status Post Types

There are three types of status posts that may be used depending on the Cashnet modules that the client has purchased.

Note: In certain scenarios, the status may be posted to the campus system using a batch file as opposed to a real-time web service request.

Table 10: Types of Status Posts

POST TYPE	USE CASE	PURPOSE IN CAMPUS SYSTEM
1098-T Status	Post the new status when a student's opt-in/opt-out status changes for the 1098-T module.	Allows Cashnet to know which students are opted into electronic statements only.
eRefund Status	Post the new status when a student's enrollment in the eRefund module changes.	Allows Cashnet to know which students are enrolled in the eRefund module in order to retrieve the required refund information from the campus system when needed.
Payment Plan Status	Post the new status when a student's status in a payment plan changes.	Allows Cashnet to know students' statuses in payment plans in order to retrieve the required payment plan information from the campus system when needed.

3.2 Transaction Post Information

The table below lists the transaction information that may be written by Cashnet to the campus system in any of the various transaction posts.

Important Notes:

- Cashnet posts only one line item per transaction post. As such, Cashnet transactions that contain multiple item codes will post transactional information in separate payment/charge posts for each item.
- The information posted will vary depending on the type of transaction post.
- The tags for post parameters are configurable in Cashnet and can therefore be configured according to the client's preferences.
- Transaction posts are flexible and can be configured to post most information that exists in Cashnet (whether imported or stored in Cashnet's data structure). Work with your Deployment Team to determine what information to include in your transaction posts.

Table 11: Transaction Post Information

PIECE OF INFORMATION	MAX LENGTH	REMARKS
Student ID	50	
Payment Code	12	Used for payment posts

PIECE OF INFORMATION	MAX LENGTH	REMARKS
Payment Code Attribute 1	50	Payment Code Attributes can optionally be configured in Cashnet and can be used to send specific business requirement values from Cashnet to the client system.
Payment Code Attribute 2	50	
Payment Code Attribute 3	50	
Payment Code Attribute 4	50	
Item Code	12	Varies depending on the campus system and type of post. For details, discuss with your Deployment Team.
Item Code Attribute 1	50	Item Code Attributes can optionally be configured in Cashnet and can be used to send specific business requirement values from Cashnet to the client system.
Item Code Attribute 2	50	
Item Code Attribute 3	50	
Item Code Attribute 4	50	
G/L Code	100	
Transaction Description	50	
Term Code	12	
Bill Number	50	
Transaction Number		
Item Line Number		
Quantity		
Item Type	1	O: Open item or balance N: New charge I: Installment
Transaction Date		
Transaction Business Date		

PIECE OF INFORMATION	MAX LENGTH	REMARKS
Transaction Type	1	S: Sale R:Refund A: Reversal C: Dishonor
Transaction Source	3	WEB: Online CR: Cashier ER: eRefund
Amount		Amount of line item
IPP Code	12	Only for IPP transactions
Operator	50	
Station	10	
Station Attribute 1	50	Station Attributes can optionally be configured in Cashnet and can be used to send specific business requirement values from Cashnet to the client system.
Station Attribute 2	50	
Station Attribute 3	50	
Station Attribute 4	50	
Location	10	
Location Attribute 1	50	Location Attributes can optionally be configured in Cashnet and can be used to send specific business requirement values from Cashnet to the client system.
Location Attribute 2	50	
Location Attribute 3	50	
Location Attribute 4	50	
Batch Number		
Credit Card Type	1	Only for CC transactions

PIECE OF INFORMATION	MAX LENGTH	REMARKS
Payment Type	2	CC: Credit card AC: ACH PD: Pin-less Debit FX: Foreign currency Blank: Non-electronic payment
Last Four Digits of CC	4	Only for CC transactions
Original Transaction Number		Only for reversals & refund transactions
Additional Information		Additional name/value pairs can be configured as negotiated during the deployment process.

3.2.1 Example Transaction Post

Below is an example transaction post request.

```
<Cashnet>
    <command>post</command>
    <posttype>payment</posttype> <!-- The value for 'posttype' can be 'payment', 
'charge', 'deposit', 'refund', or blank, to indicate the type of transaction post. Blank 
is used for posts that are both payment AND charge. -->
    <studentid>P987654321</studentid>
    <custattr1></custattr1>
    <custattr2></custattr2>
    <custattr3></custattr3>
    <custattr4></custattr4>
    <group>S</group>
    <paymentcode>ACHWEB</paymentcode>
    <payattr1></payattr1>
    <payattr2></payattr2>
    <payattr3></payattr3>
    <payattr4></payattr4>
    <itemcode>Tuition</itemcode>
    <itemattr1></itemattr1>
    <itemattr2></itemattr2>
    <itemattr3></itemattr3>
    <itemattr4></itemattr4>
    <gl></gl>
    <description>Tuition Payment</description>
    <termcode>201620</termcode>
    <billno>8908AYZ</billno>
    <txnumber>1234891</txnumber>
    <lineno>1</lineno>
    <qty></qty>
    <itemtype>N</itemtype>
    <txdate>20161005</txdate>
    <busdate>20161006</busdate>
    <txtype>S</txtype>
    <txsource>WEB</txsource>
    <amount>100.00</amount>
    <ippcode></ippcode>
    <operator>BURSAR</operator>
    <station>000</station>
    <stationattr1></stationattr1>
    <stationattr2></stationattr2>
    <stationattr3></stationattr3>
    <stationattr4></stationattr4>
    <location>main</location>
    <locationattr1></locationattr1>
    <locationattr2></locationattr2>
    <locationattr3></locationattr3>
    <locationattr4></locationattr4>
    <batchno>10</batchno>
    <cctype></cctype>
    <paymenttype>AC</paymenttype>
    <last4digits></last4digits>
    <origtxno></origtxno>
</Cashnet>
```

3.3 Status Post Information

The following sections list the information that may be written to the campus system in the three types of status posts, and present examples of each status post using web services.

3.3.1 1098-T Status

The table below lists the transaction information that may be written by Cashnet to the campus system in the 1098-T status post.

Table 12: 1098-T Status Post Information

PIECE OF INFORMATION	MAX LENGTH	REMARKS
Student ID	50	

PIECE OF INFORMATION	MAX LENGTH	REMARKS
Status	1	A: Active D: Declined
Note Code		This code should be valid in campus system

3.3.1.1 Example 1098-T Status Post

Below is an example 1098-T status post.

```
<Cashnet>
   <command>1098Tstatuspost</command>
   <studentid>P987654321</studentid>
   <status>A</status>
   <notecode></notecode>
</Cashnet>
```

3.3.2 eRefund Status

The table below lists the information that may be written by Cashnet to the campus system in the eRefund status post.

Table 13: eRefund Status Post Information

PIECE OF INFORMATION	MAX LENGTH	REMARKS
Student ID	50	
Status	1	A: Active S: Suspend D: Declined W: Withdraw
Enroll Date		
Suspend Date		
Declined Date		
Withdraw Date		
Note Code		This code should be valid in campus system

3.3.2.1 Example eRefund Post

Below is an example eRefund status post.

```
<Cashnet>
   <command>erefundstatuspost</command>
   <studentid>P987654321</studentid>
   <status>A</status>
   <enrolldate>20161001</enrolldate>
   <suspenddate></suspenddate >
   <declinedate></declinedate>
   <withdrawdate></withdrawdate>
   <notecode></notecode>
</Cashnet>
```

3.3.3 Payment Plan Status

The table below lists the information that may be written by Cashnet to the campus system in the payment plan status post.

Table 14: Payment Plan Status Post Information

PIECE OF INFORMATION	MAX LENGTH	REMARKS
Student ID	50	
IPP Code	12	Installment plan code
Term Code	12	Term code
Status		A: Active W: Withdraw
Enroll Date		Only used if status is A
Complete Date		Only used if status is A
Plan Amount		Only used if status is A
Enrolled ID	50	May be used if the parent enrolls in plan
Operator	50	Only used if status is W
Withdraw Date		Only used if status is W
Note Code		This code should be valid in campus system

3.3.3.1 Example Payment Plan Status Post

Below is an example payment plan status post.

```
<Cashnet>
   <command>ippstatuspost</command>
   <studentid>P987654321</studentid>
   <ippcode>5PAY</ippcode>
   <termcode>201630</termcode>
   <status>A</status>
   <enrolldate>20161001</enrolldate>
   <completedate>20170301</completedate>
   <amount>12000.00</amount>
   <enrollid>P987654321</enrollid>
   <operator></operator>
   <withdrawdate></withdrawdate>
   <notecode></notecode>
</Cashnet>
```

3.3.4 Example Status Post Responses

Below are example XML responses returned to the Cashnet application from any of the different status post types.

Example Successful Post Response:

```
<Cashnet>
   <result>0</result> <!-- Where '0' indicates success and non-zero is associated with a
specific error -->
   <message>Successfully posted.</message>
</Cashnet>
```

Example Failure Post Response:

```
<Cashnet>
   <result>100</result> <!-- A specific error code that is associated with a detailed
error message -->
   <message>Error message</message>
</Cashnet>
```

4 Certificate Authentication

If requested by the client, web service connections can be configured to use certificate authentication for both inbound and outbound requests. Cashnet's certificate authentication works as follows:

- Cashnet has an Organization Validation (OV) certificate from our preferred vendor, Geotrust, a major trusted Certificate Authority (CA).
- Cashnet's certificate is used for multiple clients and therefore has a common name on the certificate. The common name is generic and cannot not be chosen by the client.
- When the certificate is approaching expiration, Cashnet will determine the date and time when the certificate will be replaced with a new one, and will announce the details to the client several days in advance of the change.

Appendix A Lookup Response Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="CustomerLookupResultDS"
targetNamespace="http://tempuri.org/CustomerLookupResultDS.xsd"
elementFormDefault="qualified" attributeFormDefault="qualified"
xmlns="http://tempuri.org/CustomerLookupResultDS.xsd"
xmlns:mstns="http://tempuri.org/CustomerLookupResultDS.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-
msdata">
<xs:element name="CustomerLookupResultDS" msdata:IsDataSet="true">
<xs:complexType>
<xs:choice maxOccurs="unbounded">
<xs:element name="LookupResult">
<xs:complexType>
<xs:sequence>
<xs:element name="result" type="xs:string" minOccurs="0"/>
<xs:element name="respmessage" type="xs:string" minOccurs="0"/>
<xs:element name="cust_code" type="xs:string" minOccurs="0"/>
<xs:element name="last_name" type="xs:string" minOccurs="0"/>
<xs:element name="first_name" type="xs:string" minOccurs="0"/>
<xs:element name="cust_group" type="xs:string" minOccurs="0"/>
<xs:element name="cust_pin" type="xs:string" minOccurs="0"/>
<xs:element name="salt" type="xs:string" minOccurs="0"/>
<xs:element name="address_1" type="xs:string" minOccurs="0"/>
<xs:element name="address_2" type="xs:string" minOccurs="0"/>
<xs:element name="address_3" type="xs:string" minOccurs="0"/>
<xs:element name="city" type="xs:string" minOccurs="0"/>
<xs:element name="state" type="xs:string" minOccurs="0"/>
<xs:element name="zip" type="xs:string" minOccurs="0"/>
<xs:element name="country" type="xs:string" minOccurs="0"/>
<xs:element name="area_code" type="xs:string" minOccurs="0"/>
<xs:element name="phone_number" type="xs:string" minOccurs="0"/>
<xs:element name="email_id" type="xs:string" minOccurs="0"/>
<xs:element name="custattr_1" type="xs:string" minOccurs="0"/>
<xs:element name="custattr_2" type="xs:string" minOccurs="0"/>
<xs:element name="custattr_3" type="xs:string" minOccurs="0"/>
<xs:element name="custattr_4" type="xs:string" minOccurs="0"/>
<xs:element name="custotherdata" type="xs:string" minOccurs="0"/>
<xs:element name="filter_data" type="xs:string" minOccurs="0"/>
<xs:element name="alternateid_1" type="xs:string" minOccurs="0"/>
<xs:element name="alternateid_1_type" type="xs:string" minOccurs="0"/>
<xs:element name="alternateid_1_email" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="alternateid_1_activeflag" type="xs:string" minOccurs="0"/>
<xs:element name="alternateid_1_notifyflag" type="xs:string" minOccurs="0"/>
<xs:element name="alternateid_2" type="xs:string" minOccurs="0"/>
<xs:element name="alternateid_2_type" type="xs:string" minOccurs="0"/>
<xs:element name="alternateid_3" type="xs:string" minOccurs="0"/>
<xs:element name="alternateid_3_type" type="xs:string" minOccurs="0"/>
<xs:element name="alternateid_4" type="xs:string" minOccurs="0"/>
<xs:element name="alternateid_4_type" type="xs:string" minOccurs="0"/>
<xs:element name="note_code" type="xs:string" minOccurs="0"/>
<xs:element name="note_exp_date" type="xs:string" minOccurs="0"/>
<xs:element name="gl_code" type="xs:string" minOccurs="0"/>
<xs:element name="dob" type="xs:string" minOccurs="0"/>
<xs:element name="shared_secret" type="xs:string" minOccurs="0"/>
<xs:element name="t4_indicator" type="xs:string" minOccurs="0"/>
<xs:element name="t4_prior_term_auth" type="xs:string" minOccurs="0"/>
<xs:element name="t4_other_charges" type="xs:string" minOccurs="0"/>
<xs:element name="t4_credit_balance" type="xs:string" minOccurs="0"/>
<xs:element name="t4_choice_made_flag" type="xs:string" minOccurs="0"/>
<xs:element name="t4_auth_datetime" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="items">
<xs:complexType>
<xs:sequence>
<xs:element name="item_code" type="xs:string" minOccurs="0"/>
<xs:element name="bill_desc" type="xs:string" minOccurs="0"/>
<xs:element name="bill_no" type="xs:string" minOccurs="0"/>
<xs:element name="balance_due" type="xs:string" minOccurs="0"/>
<xs:element name="minimum_due" type="xs:string" minOccurs="0"/>
<xs:element name="balotherdata" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="recentactivity">
<xs:complexType>
<xs:sequence>
<xs:element name="activity_date" type="xs:string" minOccurs="0"/>
<xs:element name="reference_no" type="xs:string" minOccurs="0"/>
<xs:element name="activity_description" type="xs:string" minOccurs="0"/>
<xs:element name="activity_description_2" type="xs:string" minOccurs="0"/>
<xs:element name="amount" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
```

```
</xs:element>
<xs:element name="IPPSourceAmount">
<xs:complexType>
<xs:sequence>
<xs:element name="termcode" type="xs:string" minOccurs="0"/>
<xs:element name="status" type="xs:string" minOccurs="0"/>
<xs:element name="gradelevel" type="xs:string" minOccurs="0"/>
<xs:element name="ippsourceamountcode" type="xs:string" minOccurs="0"/>
<xs:element name="amount" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="notes">
<xs:complexType>
<xs:sequence>
<xs:element name="multi_note_code" type="xs:string" minOccurs="0"/>
<xs:element name="multi_note_exp_date" type="xs:string" minOccurs="0"/>
<xs:element name="multi_note_std_view" type="xs:string" minOccurs="0"/>
<xs:element name="multi_note_cashier_view" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
```