

# PayLeap Reporting API Guide



Visit us at: <http://www.payleap.com>.

Copyright © 2009 PayLeap, LLC. All Rights Reserved

Congratulations on the selection of PayLeap Software Development Kit (SDK) Payment Processing Software, the most advanced solution in the industry for processing credit cards, debit cards and check services. This software provides you with a fast, easy, reliable way to authorize credit card, ATM/debit card, and check transactions on your PC. This guide prepares you with the detailed information that you will need to install, configure and test your payment processing application.

Your opinion is important to us. If you have any suggestions feel free to [email us](#).

**Thank you for choosing PayLeap!**

PayLeap, LLC  
3594 E. US Hwy 30  
Warsaw, Indiana 46580  
Email: [sales@payleap.com](mailto:sales@payleap.com)  
Phone: 877-472-9532

## Support

PayLeap is committed to providing the highest quality tools and customer support. If you have any questions, comments or suggestions please contact PayLeap by:

Email: [support@payleap.com](mailto:support@payleap.com)  
Phone: 877-472-9532

## **License Information**

### ***DISCLAIMER***

The PayLeap software and supporting services are licensed on an “as is” basis. There are no warranties, expressed or implied, including, but not limited to, warranties of merchantability or of fitness for a particular purpose, and all such warranties are expressly and specifically disclaimed. PayLeap shall have no liability or responsibility to you nor any other person or entity with respect to any liability, loss, or damage caused or alleged to be caused directly or indirectly by any PayLeap software and supporting services. Use of the PayLeap software and supporting services signifies agreement with this disclaimer and is subject to the license agreement provided with the installation of the software.

### **LICENSE**

Use of the PayLeap services requires a signed service agreement by the end user of the service or reseller of such services. PayLeap grants you the right to use this software and supporting software for the purpose of connecting to the PayLeap gateway or your processor, on any device of choice. You may physically transfer each license, without cost, to a different computer.

### **RESTRICTED USE**

Copying of the manual, interface specifications, or system files for use other than backing up files and/or to connect to systems other than the PayLeap gateway or your processor is prohibited. You may not remove any product identification, copyright, or other proprietary notices from the software or documentation. Reverse engineering is strictly prohibited.

This client software is provided for use in connecting with the PayLeap gateway or your processor only. Use of any components, code, or documentation to connect to other systems is strictly prohibited and a violation of this agreement and subject to all remedies available by law.

### ***EULA***

#### **END-USER LICENSE AGREEMENT FOR PAYLEAP SOFTWARE**

##### **IMPORTANT READ CAREFULLY:**

This PayLeap End-User License Agreement ( EULA ) is a legal agreement between you (either an individual or a single entity) and PayLeap, LLC for the PayLeap product accompanying this EULA, which includes computer software and may include associated media, printed materials, and online or electronic documentation (SOFTWARE). By installing, copying, or otherwise using the SOFTWARE, you agree to be bound by the

terms of this EULA. If you do not agree to the terms of this EULA, do not install, copy, or otherwise use the SOFTWARE.

#### SOFTWARE PRODUCT LICENSE

PayLeap (SOFTWARE) is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is licensed, not sold.

#### GRANT OF LICENSE

PayLeap,LLC hereby grants you ("Licensee") a limited, nonexclusive, non-transferable, royalty-free license to make and use a single copy an unlimited number of copies of the SOFTWARE accompanying this EULA to be installed on CPUs residing on Licensee's premises, solely for Licensee's internal use. PayLeap, LLC and its suppliers shall retain title and all ownership rights to the product, and this Agreement shall not be construed in any manner as transferring any rights of ownership or license to the SOFTWARE or to the features or information therein, except as specifically stated herein.

#### USE RESTRICTION

Licensee acknowledges that the SOFTWARE acquired hereunder can only be used for reseller and/or end merchant evaluation and/or product's intended use of payment processing. Use of this product for any other purposes such as Competitor Evaluation, Reverse Engineering, Decompilation, and Disassembly is violation of this License and Licensee agrees that such acts are a blatant and flagrant violation of this License and will be subject to any and all penalties and remedies available by Law for violation of intellectual property laws and treaties.

#### DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS

(a) Limitations on Modification, Reverse Engineering, Decompilation, and Disassembly. Licensee may not modify, reverse engineer, decompile, or disassemble the SOFTWARE, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

(b) No Rights to Sublicense or Rental. License may not rent, lease or lend the SOFTWARE.

(c) Termination. Without prejudice to any other rights, PayLeap may terminate this EULA if Licensee fails to comply with the terms and conditions of this EULA. In such event, Licensee must destroy all copies of the SOFTWARE and all of its component parts.

(d) Reservation of Rights. Licensee agrees that the SOFTWARE is owned PayLeap and all rights not expressly granted herein are reserved by PayLeap, LLC.

## PRODUCT MAINTENANCE

Licensee understands and agrees that PayLeap may provide updates to the SOFTWARE from time to time but PayLeap shall have no obligation to provide maintenance or updates to Licensee for SOFTWARE licensed under this Agreement.

## CONFIDENTIALITY

(a) Licensee understands that the SOFTWARE contains confidential and proprietary trade secret information of PayLeap that are not commercially available to the public. The Licensee agrees that, in partial consideration for PayLeap's allowing Licensee access to and use of the SOFTWARE pursuant to this EULA:

(i) Licensee shall treat the SOFTWARE in the same manner that it treats its most confidential and proprietary trade secret materials, and

(ii) Licensee shall take all measures necessary to prevent the SOFTWARE from falling into the possession of persons not bound to maintain the confidentiality of the trade secrets contained within the SOFTWARE. As such, Licensee shall only permit employees and contractors who have a need to use the SOFTWARE for the purposes stated in the license grant (Section 1) to have access to and use such SOFTWARE, and Licensee's contractors shall not use the SOFTWARE unless and until they have entered into written non-disclosure agreements with the Licensee that require them to maintain the confidentiality of SOFTWARE. Licensee shall promptly advise PayLeap, in writing, of any misappropriation or misuse of the SOFTWARE by any person which may come to the Licensee's attention.

(b) Licensee understands and agrees that disclosure or use of the SOFTWARE except as authorized above will result in irreparable harm to PayLeap and that monetary damages may be inadequate to compensate PayLeap for such breach. Accordingly, the Licensee agrees that PayLeap will, in addition to any other remedies available to it at law or in equity, be entitled to injunctive relief to enforce the terms of this Agreement.

## COPYRIGHT

All title and copyrights in and to the SOFTWARE (including but not limited to any images, photographs, animations, video, audio, music, text, and applets incorporated into the SOFTWARE), the accompanying printed materials, and any copies of the SOFTWARE are owned by PayLeap or its suppliers. The SOFTWARE is protected by copyright laws and international treaty provisions. Therefore, Licensee must treat the

SOFTWARE PRODUCT like any other copyrighted material except that Licensee may install the SOFTWARE on a single computer provided Licensee keep the original solely for backup or archival purposes. Licensee may not copy any printed materials accompanying the SOFTWARE without prior permission of PayLeap.

#### U.S. GOVERNMENT RESTRICTED RIGHTS

The SOFTWARE and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is PayLeap Payment Solutions, a division of PayLeap USA./3594 E. US Hwy 30/Warsaw, Indiana 46580.

#### EXPORT RESTRICTIONS

Licensee acknowledges that the SOFTWARE acquired hereunder are subject to the export control laws and regulations of the U.S.A., and any amendments thereof. Licensee confirms that with respect to these SOFTWARE, it will not export or re-export them, directly or indirectly, either to (i) any countries that are subject to U.S.A export restrictions (currently including, but not necessarily limited to, Cuba, the Federal Republic of Yugoslavia (Serbia and Montenegro), Iran, Iraq, Libya, North Korea, South Africa (military and police entities), Syria, and Vietnam); (ii) any end user who Licensee knows or has reason to know will utilize them in the design, development or production of nuclear, chemical or biological weapons; or (iii) any end user who has been prohibited from participating in the U.S.A. export transactions by any federal agency of the U.S.A. government. Licensee further acknowledges that the SOFTWARE may include technical data subject to export and re-export restrictions imposed by U.S.A. law.

#### DISCLAIMER OF WARRANTY

THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, TPI FURTHER DISCLAIMS ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE PRODUCT AND DOCUMENTATION REMAINS WITH LICENSEE.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL PAYLEAP BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, PUNITIVE, OR OTHER DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS

INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE PRODUCT OR DOCUMENTATION, EVEN IF PAYLEAP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO LICENSEE.

## GOVERNING LAW

This EULA shall be governed by the laws of the State of Washington.

Should you have any questions concerning this EULA, or if you desire to contact PayLeap for any reason, please write: PayLeap Payment Solutions, a division of PayLeap USA./3594 E. US Hwy 30/Warsaw, Indiana 46580

## Overview

### *Testing*

You can request a test account on our server to be established at our demo host server. To request this account, please send your email request to our technical support staff. Please include the following information with your test account request: company name, your name, phone number, email address associated with the test account, and which payment processor you would like to test. An email response will be sent with valid test information.

Testing can be performed with the following test cards:

Card Type	Number
MasterCard	5000300020003003
Visa	4005550000000019
Discover	6001111111111117
Diners	36999999999999
AMEX	374255312721002

## Web Services

### ePayment Web Services

The following operations are supported.

- [GetCardTrx](#) – Retrieves card transaction details for a merchant
- [GetCardTrxSummary](#) – Retrieves card transaction summary for a merchant

- [GetCheckTrx](#) – Retrieves check transaction details for a merchant
- [GetCardType](#) – Returns the name of the card issuer; such as Visa, MasterCard, AMEX, etc.
- [GetInfo](#) – Retrieves information from the web service
- [GetOpenBatchSummary](#) – Retrieves payment type transaction summary of the current open batch for a merchant
- [IsCommercialCard](#) – Returns (T/F) if the card is a known commercial card
- [GetNetworkID](#)- This web service allows for returning the debit network ID if the debit card number matches any of these network’s bin ranges

## GetCardTrx

This Web service operation retrieves card transaction details for a merchant. The URL to access this Web service is:

<https://test.payleap.com/admin/ws/trxdetail.aspx?op=GetCardTrx>. The text “test.payleap.com” in the URL should be replaced with the actual host name or static IP address of the payment server. Descriptions of the parameters are listed below.

Parameter	Description
<b>UserName</b>	Required. User name assigned in the payment server
<b>Password</b>	Required. Password for the user name assigned in the payment server
<b>RPNum</b>	Required. The merchant's RP Number in order to uniquely identify merchant's account for the query
<b>PNRef</b>	Optional. The unique payment reference number assigned to the transaction. <i>If this field is provided, all other query fields will be ignored when using <b>PNRef</b> parameter to query the system.</i>
<b>BeginDt</b>	Required, except when PNRef is provided. The begin date of the date range in MM/DD/YYYY (or YYYY-MM-DD or YYYY-MM-DDThh:mm:ss) format. This date will be converted to YYYY-MM-DDThh:mm:ss (time is in 24-hour format). If the passed-in BeginDt does not contain time information, BeginDt will default to midnight on the given date. For example: 2005-08-19T12:00:12 is kept as is 2005-08-19 becomes 2005-08-19T00:00:00 2005/08/19 becomes 2005-08-19T00:00:00 08/19/2005 becomes 2005-08-19T00:00:00 The query to obtain transactions in the date range is constructed as follows: (Date DT >=BeginDt) AND (Date DT<EndDt) where Date DT is the transaction timestamp
<b>EndDt</b>	Required, except when <b>PNRef</b> is provided. The end date of the date range in MM/DD/YYYY (or YYYY-MM-DD or YYYY-MM-DDThh:mm:ss) format. This date will be converted to YYYY-MM-DDThh:mm:ss (time is in 24-hour format). If the passed-in EndDt does not contain time information, EndDt will be incremented to the <i>next day</i> at midnight such that no transaction

	<p>on the desired end date will be excluded based on its time. For example:  2005-08-19T12:00:12 is kept as is  2005-08-19 becomes 2005-08-20T00:00:00  08/19/2005 becomes 2005-08-20T00:00:00</p>
<p><b>PaymentType</b></p>	<p>Optional. If provided, only those transactions matching the <b>PaymentType</b> will be included. Valid values are:</p> <ul style="list-style-type: none"> <li>• <b>'AMEX'</b> American Express card</li> <li>• <b>'CARTBLANCH'</b> Carte Blanch card</li> <li>• <b>'DEBIT'</b> Debit card</li> <li>• <b>'DINERS'</b> Diners Club card</li> <li>• <b>'DISCOVER'</b> Novus Discover card</li> <li>• <b>'EBT'</b> Electronic Benefit Transfer</li> <li>• <b>'JAL'</b> JAL card</li> <li>• <b>'JCB'</b> Japanese Commercial Bank card</li> <li>• <b>'MASTERCARD'</b> Master card</li> <li>• <b>'VISA'</b> Visa card</li> <li>• <b>'EGC'</b> Gift card</li> <li>• <b>'PAYRECEIPT'</b> to retrieve receipt images that were uploaded to the payment server</li> <li>• <b>'SETTLE'</b> to retrieve requests to settle transactions</li> </ul> <p>Or any permutation of the above values, e.g. <b>""PAYRECEIPT','SETTLE""</b> will pull all transactions with either PayReceipt or Settle payment types.</p>
<p><b>ExcludePaymentType</b></p>	<p>Optional. If provided, any transaction matching the <b>ExcludePaymentType</b> will be excluded</p>
<p><b>TransType</b></p>	<p>Optional. If provided, only those transactions matching the <b>TransType</b> will be included. Valid values are</p> <ul style="list-style-type: none"> <li>• <b>'Authorization'</b> to retrieve previously-authorized (pre-auth) transactions</li> <li>• <b>'Capture'</b> to retrieve captured transactions</li> <li>• <b>'Credit'</b> to retrieve return transactions</li> <li>• <b>'ForceCapture'</b> to retrieve force-auth transactions</li> <li>• <b>'GetStatus'</b> to make an inquiry to the EBT or gift card's balance</li> <li>• <b>'PostAuth'</b> to retrieve post-auth transactions</li> <li>• <b>'Purged'</b> to remove a transaction from the current batch due to an error</li> <li>• <b>'Receipt'</b> to retrieve receipt images that were uploaded to the payment server</li> <li>• <b>'RepeatSale'</b> to retrieve repeat-sale transactions</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>'Sale'</b> to retrieve sale transactions</li> <li>• <b>'Void'</b> to retrieve void transactions</li> </ul> <p>Or any permutation of the above values, e.g. <b>""Credit','Sale""</b> will pull all transactions with either Credit or Sale transaction types.</p>
<b>ExcludeTransType</b>	Optional. If provided, any transaction matching the <b>ExcludeTransType</b> will be excluded
<b>ApprovalCode</b>	Optional. If provided, only those transactions matching the <b>ApprovalCode</b> parameter will be included
<b>Result</b>	Optional. If provided, only those transactions matching the <b>Result</b> will be included. Valid values are: <ul style="list-style-type: none"> <li>• <b>0</b> is Approved</li> <li>• Anything else is declined; if you want all the declined transactions, you should leave this field empty and use the <b>ExcludeResult=0</b> instead.</li> </ul>
<b>ExcludeResult</b>	Optional. If provided, any transactions matching the <b>ExcludeResult</b> will be excluded.
<b>NameOnCard</b>	Optional. Cardholder's name as it appears on the card. If provided, only those transactions with cardholder's name matching <b>NameOnCard</b> will be included. Matching is done using wild cards: e.g. "test" will match "test", "1test" and "1test234"
<b>CardNum</b>	Optional. A card number. If provided, only those transactions with the cardholder's name matching <b>CardNum</b> will be included. Matching is done using wild cards
<b>CardType</b>	<p>If provided, only those transactions matching the <b>CardType</b> will be included. Valid values are:</p> <ul style="list-style-type: none"> <li>• <b>'AMEX'</b> American Express card</li> <li>• <b>'CARTBLANCH'</b> Carte Blanch card</li> <li>• <b>'DEBIT'</b> Debit card</li> <li>• <b>'DINERS'</b> Diners Club card</li> <li>• <b>'DISCOVER'</b> Novus Discover card</li> <li>• <b>'EBT'</b> Electronic Benefit Transfer</li> <li>• <b>'JAL'</b> JAL card</li> <li>• <b>'JCB'</b> Japanese Commercial Bank card</li> <li>• <b>'MASTERCARD'</b> Master card</li> <li>• <b>'VISA'</b> Visa card</li> <li>• <b>'EGC'</b> Gift card</li> </ul> <p>Or any permutation of the above values, <b>""VISA','MASTER','DISCOVER""</b> will pull all transactions with either VISA, MASTER and DISCOVER card type</p> <p>There are cases when Cardtype needs to be set to ALL, for example</p>

	CardType=ALL
<b>ExcludeCardType</b>	Optional. If provided, any transaction matching the <b>ExcludeCardType</b> will be excluded
<b>ExcludeVoid</b>	Required, except when <b>PNRef</b> is provided. An option to exclude voided transactions or not; must either be <b>TRUE</b> or <b>FALSE</b>
<b>User</b>	Optional. The user who originated the transactions. If provided, only those transactions created by the matching <b>User</b> will be included. Matching is done using wild cards
<b>InvoiceId</b>	Optional. The invoice ID that was included in the original transaction. If provided, only those transactions with matching <b>invoiceId</b> will be included. Matching is done using wild cards
<b>SettleFlag</b>	Optional. An option to retrieve the settled transactions or unsettled transactions; must either be <b>1</b> for true or <b>0</b> for false
<b>SettleMsg</b>	Optional. The settlement ID or message returned from the host
<b>SettleDt</b>	Optional. The date of the settlement
<b>TransformType</b>	Optional. The type of format to transform the data into. Leave the field blank to default to <b>XML</b> <ul style="list-style-type: none"> <li>• <b>XML</b> will output the plain XML string</li> <li>• <b>XSL</b> will use XSL to transform the XML output</li> <li>• <b>DELIM</b> uses <b>ColDelim</b> and <b>RowDelim</b> to format the output</li> </ul>
<b>Xsl</b>	Optional. This field is used only if the <b>TransformType</b> is <b>XSL</b> . The XSL to transform the resulting dataset. If provided, the resulting dataset will be transformed using this XSL. You may pass in a URL to the XSL file, or the XSL string itself. If this field is not empty, the Web Services will try to locate the file from the URL. If that also fails, it will treat it as an XSL string. In any case, the final XSL string will be loaded and validated against the XSL schema; if it passes, then that XSL will be used for transformation. A sample predefined XSL is included with this Web Services: <ul style="list-style-type: none"> <li>• <a href="http://test.payleap.com/admin/ws/TabDelim.xsl">http://test.payleap.com/admin/ws/TabDelim.xsl</a> for a tab delimited transformation</li> </ul>
<b>ColDelim</b>	Optional. This field is used only if the <b>TransformType</b> is <b>DELIM</b> . This defines the string that separates each column
<b>RowDelim</b>	Optional. This field is used only if the <b>TransformType</b> is <b>DELIM</b> . This defines the string that separates each row
<b>IncludeHeader</b>	Optional. This field is used only if the <b>TransformType</b> is <b>DELIM</b> . If <b>TRUE</b> , then field headers will be included in the first row using the same delimiter strings; must either be <b>TRUE</b> or <b>FALSE</b>
<b>ExtData</b>	Optional. Extended data in XML format. Valid values are: <ul style="list-style-type: none"> <li>• <code>&lt;IMAGE_TYPE&gt;NO_IMAGE&lt;/IMAGE_TYPE&gt;</code> for no image</li> <li>• <code>&lt;IMAGE_TYPE&gt;ONLY_IMAGE&lt;/IMAGE_TYPE&gt;</code> for only the image</li> <li>• <code>&lt;IMAGE_TYPE&gt;ALL&lt;/IMAGE_TYPE&gt;</code> for all images</li> </ul>

	<ul style="list-style-type: none"> <li>• <code>&lt;CustomerID&gt;CustomerID&lt;/CustomerID&gt;</code> for customer ID</li> <li>• <code>&lt;Amount&gt;Amount&lt;/Amount&gt;</code> Total amount to search transactions for in DDDD.CC format.</li> <li>• <code>&lt;RegisterNum&gt;RegisterNum&lt;/RegisterNum&gt;</code> Register number, originally passed with the transaction, to search transactions for.</li> </ul>
--	---

## Example

The following table contains sample data you can use to test this Web service. The **UserName**, **Password**, and **RPNum** parameters should be changed when testing this example yourself. The example data shown will create a list of sale card transactions that were processed between 1/1/2000 and 1/1/3000. The format of the output will show descriptive headers and will be similar to a Comma Separate Values (CSV) format because of the use of the "," character as the column delimiter.

Parameter	Value
<b>UserName</b>	test
<b>Password</b>	123
<b>RPNum</b>	1
<b>BeginDt</b>	1/1/2000
<b>EndDt</b>	1/1/3000
<b>TransType</b>	'Sale'
<b>ExcludeVoid</b>	TRUE
<b>TransformType</b>	DELIM
<b>ColDelim</b>	,
<b>RowDelim</b>	
<b>IncludeHeader</b>	TRUE

Result: The following is the result of the using the Web Services with the sample data from the table above.

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<string xmlns="http://test.payleap.com/admin/ws ">
```

```
TRX_HD_Key,Invoice_ID,Seq_Num_CH,Date_DT,Merchant_Key,User_Name
_VC,Register_Number_CH,Reseller_Key,Payment_Type_ID,Trans_Type_ID,
Processor_ID,TRX_Settle_Key,TRX_Settle_Msg_VC,Void_Flag_CH,Settle_Fla
g_CH,Ref_Number_CH,Settle_Date_DT,Last_Update_DT,TRX_Card_Key,Card
_Info_Key,Auth_Amt_MN,Tip_Amt_MN,Total_Amt_MN,Cash_Back_Amt_MN,
```

SureCharge\_Amt\_MN,Account\_Type\_CH,Result\_CH,Result\_Txt\_VC,Approval  
 \_Code\_CH,Host\_Ref\_Num\_CH,AVS\_Resp\_CH,AVS\_Resp\_Txt\_VC,CV\_Resp\_C  
 H,CV\_Resp\_Txt\_VC,Host\_Date\_CH,Host\_Time\_CH,Acct\_Num\_CH,Exp\_CH,Ty  
 pe\_CH,Name\_on\_Card\_VC,Street\_CH,Zip\_CH,Track\_VC,Pin\_Block\_CH,TRX\_  
 Receipt\_key,Create\_Date\_DT,Receipt\_Type\_ID,IP\_VC|14,,,12/3/2003  
 10:47:48 AM,1,test , ,100,VISA ,Sale ,VITAL ,14,GB00004  
 ACCEPTED,,1,,12/3/2003 10:46:34 AM,12/3/2003 10:48:13  
 AM,14,14,2.5000,0,2.5000,0,0,VISA ,0 ,APPROVAL VITAL1 ,VITAL1 , ,0,,  
 ,,,,4266503700000247,1009 ,VISA ,VINCE VISA , , ,1,  
 ,,,,127.0.0.1|13,,,12/3/2003 10:04:29 AM,1,test , ,100,MASTERCARD,Sale  
 ,VITAL ,13,GB00004 ACCEPTED,,1,,12/3/2003 10:46:34 AM,12/3/2003  
 10:48:13 AM,13,13,2.8500,0,2.8500,0,0,MASTERCARD,0 ,APPROVAL VITAL9  
 ,VITAL9 , ,0,, ,,,,542400000000015,0905 ,MASTERCARD, , , , ,  
 ,,,,127.0.0.1|12,,,12/3/2003 9:45:59 AM,1,test , ,100,VISA ,Sale ,VITAL  
 ,12,GB00003 ACCEPTED,,1,,12/3/2003 9:46:16 AM,12/3/2003 9:46:16  
 AM,12,12,1.6800,0,1.6800,0,0,VISA ,0 ,APPROVAL VITAL3 ,VITAL3 , ,0,,  
 ,,,,4126196901499,0905 ,VISA , , , , , ,,,,127.0.0.1|10,,,12/3/2003 9:35:19  
 AM,1,test , ,100,VISA ,Sale ,VITAL ,10,GB00003 ACCEPTED,,1,,12/3/2003  
 9:46:16 AM,12/3/2003 9:46:16 AM,10,10,2.3500,0,2.3500,0,0,VISA ,0  
 ,APPROVAL VITAL8 ,VITAL8 , ,0,, ,,,,4055016727870315,0905 ,VISA , , , , ,  
 ,,,,127.0.0.1|7,,,12/3/2003 9:27:57 AM,1,test , ,100,VISA ,Sale ,VITAL  
 ,7,GB00002 ACCEPTED,,1,,12/3/2003 9:26:35 AM,12/3/2003 9:28:16  
 AM,7,7,1.2500,0,1.2500,0,0,VISA ,0 ,APPROVAL VITAL4 ,VITAL4 , ,0,,  
 ,,,,4007000000027,0905 ,VISA , , , , , ,,,,127.0.0.1|6,,,12/3/2003 9:27:44  
 AM,1,test , ,100,VISA ,Sale ,VITAL ,6,GB00002 ACCEPTED,,1,,12/3/2003  
 9:26:35 AM,12/3/2003 9:28:16 AM,6,6,2.0000,0,2.0000,0,0,VISA ,0  
 ,APPROVAL VITAL9 ,VITAL9 , ,0,, ,,,,4007000000027,0905 ,VISA , , , , ,  
 ,,,,127.0.0.1|5,,,12/3/2003 9:27:15 AM,1,test , ,100,MASTERCARD,Sale  
 ,VITAL ,5,,,,,12/3/2003 9:27:17  
 AM,5,5,1.2500,0,1.2500,0,0,MASTERCARD,12 , ACCT LENGTH ERR,EA , ,0,,  
 ,,,,5405001478615777532,0905 ,MASTERCARD, , , , ,  
 ,,,,127.0.0.1|2,,,12/3/2003 9:03:16 AM,1,test , ,100,MASTERCARD,Sale  
 ,VITAL ,2,GB00001 ACCEPTED,,1,,12/3/2003 9:03:40 AM,12/3/2003  
 9:03:40 AM,2,2,2.4500,0,2.4500,0,0,MASTERCARD,0 ,APPROVAL VITAL7  
 ,VITAL7 , ,0,, ,,,,542400000000015,0905 ,MASTERCARD, , , , ,  
 ,,,,127.0.0.1|1,,,12/3/2003 8:58:02 AM,1,test , ,100,VISA ,Sale ,VITAL  
 ,1,GB00001 ACCEPTED,,1,,12/3/2003 9:03:40 AM,12/3/2003 9:03:40  
 AM,1,1,1.0000,0,1.0000,0,0,VISA ,0 ,APPROVAL VITAL3 ,VITAL3 , ,0,,  
 ,,,,4111111111111111,0905 ,VISA , , , , , ,,,,127.0.0.1

</string>

## GetCardTrxSummary

This Web service operation retrieves a card transaction summary for a merchant. The URL to access this Web service is:

<https://test.payleap.com/admin/ws/trxdetail.aspx?op=GetCardTrxSummary>. The text “test.payleap.com” in the URL should be replaced with the actual host name or static IP address of the payment server. Descriptions of the parameters are listed below.

Parameter	Value
-----------	-------

<b>UserName</b>	Required. User name assigned in the payment server
<b>Password</b>	Required. Password for the user name assigned in the payment server
<b>RPNum</b>	Required. The merchant's RP Number in order to uniquely identify the merchant's account for the query
<b>BeginDt</b>	Required. The begin date of the date range in MM/DD/YYYY format. This date will be converted to MM/DD/YYYYT00:00:00:0000AM
<b>EndDt</b>	Required. The end date of the date range in MM/DD/YYYY format. This date will be converted to MM/DD/YYYYT12:59:59:9999PM
<b>ApprovalCode</b>	Optional. If provided, only those transactions matching the <b>ApprovalCode</b> parameter will be included
<b>Register</b>	Optional. The register that originated the transaction. If provided, only those transactions with the matching register will be included
<b>NameOnCard</b>	Optional. Cardholder's name as it appears on the card. If provided, only those transactions with cardholder's name matching <b>NameOnCard</b> will be included. Matching is done using wild cards: e.g. "test" will match "test", "1test" and "1test234"
<b>CardNum</b>	Optional. A card number. If provided, only those transactions with the cardholder's name matching <b>CardNum</b> will be included. Matching is done using wild cards
<b>CardType</b>	Optional. If provided, only those transactions matching the <b>CardType</b> will be included. Valid values are: <ul style="list-style-type: none"> <li>• <b>'AMEX'</b> American Express card</li> <li>• <b>'CARTBLANCH'</b> Carte Blanch card</li> <li>• <b>'DEBIT'</b> Debit card</li> <li>• <b>'DINERS'</b> Diners Club card</li> <li>• <b>'DISCOVER'</b> Novus Discover card</li> <li>• <b>'EBT'</b> Electronic Benefit Transfer</li> <li>• <b>'JAL'</b> JAL card</li> <li>• <b>'JCB'</b> Japanese Commercial Bank card</li> <li>• <b>'MASTERCARD'</b> Master card</li> <li>• <b>'VISA'</b> Visa card</li> <li>• <b>'EGC'</b> Gift card</li> </ul>
<b>ExcludeVoid</b>	Required. Whether to exclude voided transactions; must either be <b>TRUE</b> or <b>FALSE</b> . Default is <b>TRUE</b>
<b>User</b>	Optional. The user who originated the transactions. If provided, only those transactions created by the matching <b>User</b> will be included. Matching is done using wild cards
<b>SettleFlag</b>	Optional. An option to retrieve the settled transactions or unsettled transactions; must either be <b>TRUE</b> or <b>FALSE</b>
<b>SettleMsg</b>	Optional. The settlement ID or message returned from the host
<b>SettleDt</b>	Optional. The settlement timestamp

<b>TransformType</b>	<p>Optional. The type of format to transform the data into. Leave the field blank to default to <b>XML</b></p> <ul style="list-style-type: none"> <li><b>XML</b> will output the plain XML string</li> <li><b>XSL</b> will use XSL to transform the XML output</li> <li><b>DELIM</b> uses <b>ColDelim</b> and <b>RowDelim</b> to format the output</li> </ul>
<b>Xsl</b>	<p>Optional. This field is used only if the <b>TransformType</b> is <b>XSL</b>. The XSL to transform the resulting dataset; if provided, the resulting dataset will be transformed using this XSL. You may pass in a URL to the XSL file, or the XSL string itself. If this field is not empty, the Web Services will try to locate the file from the URL. If that also fails, it will treat it as a XSL string. In any case, the final XSL string will be loaded and validated against the XSL schema; if it passes, then that XSL will be used for transformation. A sample predefined XSL is included with this Web Services:</p> <ul style="list-style-type: none"> <li><b>http://test.payleap.com/admin/ws/TabDelim.xsl</b> for a tab delimited transformation</li> </ul>
<b>ColDelim</b>	Optional. This field is used only if the <b>TransformType</b> is <b>DELIM</b> . This defines the string that separates each column
<b>RowDelim</b>	Optional. This field is used only if the <b>TransformType</b> is <b>DELIM</b> . This defines the string that separates each row
<b>IncludeHeader</b>	Optional. This field is used only if the <b>TransformType</b> is <b>DELIM</b> . If <b>TRUE</b> , then field headers will be included in the first row using the same delimiter strings; must either be <b>TRUE</b> or <b>FALSE</b>
<b>ExtData</b>	<p>Optional. Extended data in XML format. Valid values are:</p> <ul style="list-style-type: none"> <li><b>&lt;IMAGE_TYPE&gt;NO_IMAGE&lt;/IMAGE_TYPE&gt;</b> for no image</li> <li><b>&lt;IMAGE_TYPE&gt;ONLY_IMAGE&lt;/IMAGE_TYPE&gt;</b> for only the image</li> <li><b>&lt;IMAGE_TYPE&gt;ALL&lt;/IMAGE_TYPE&gt;</b> for all images</li> </ul>

## Example

The following table contains sample data you can use to test this Web service. The **UserName**, **Password**, and **RPNum** parameters should be changed when testing this example yourself. The example data shown will create a summarized list of Visa card transactions that were processed between 1/1/2000 and 1/1/3000. The output data is in XML format because the **TransformType** parameter was not specified.

Parameter	Value
<b>UserName</b>	Test
<b>Password</b>	123
<b>RPNum</b>	1
<b>BeginDt</b>	1/1/2000
<b>EndDt</b>	1/1/3000
<b>CardType</b>	VISA

ExcludeVoid	FALSE
-------------	-------

Result:

```
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://test.payleap.com/Admin/ws">
  <CardTrxSummary> <PaymentMethod> <Payment_Type_ID>VISA
  </Payment_Type_ID> <Authorization>1.0000</Authorization>
  <Capture>0</Capture> <ForceCapture>0</ForceCapture>
  <PostAuth>0</PostAuth> <Return>0</Return> <Sale>2.0000</Sale>
  <Receipt>0</Receipt> <RepeatSale>0</RepeatSale>
  <Activate>0</Activate> <Deactivate>0</Deactivate>
  <Reload>0</Reload> <Authorization_Cnt>1</Authorization_Cnt>
  <Capture_Cnt>0</Capture_Cnt>
  <ForceCapture_Cnt>0</ForceCapture_Cnt>
  <PostAuth_Cnt>0</PostAuth_Cnt> <Return_Cnt>0</Return_Cnt>
  <Sale_Cnt>2</Sale_Cnt> <Receipt_Cnt>0</Receipt_Cnt>
  <RepeatSale_Cnt>0</RepeatSale_Cnt> <Activate_Cnt>0</Activate_Cnt>
  <Deactivate_Cnt>0</Deactivate_Cnt> <Reload_Cnt>0</Reload_Cnt>
  <Cnt>3</Cnt> </PaymentMethod> </CardTrxSummary>
</string>
```

## GetCheckTrx

This Web service operation retrieves checks transaction details for a merchant. The URL to access this Web service is:

<https://test.payleap.com/admin/ws/trxdetail.aspx?op=GetCheckTrx>. The text “test.payleap.com” in the URL should be replaced with the actual host name or static IP address of the payment server. Descriptions of the parameters are listed below.

Parameter	Value
<b>UserName</b>	Required. User name assigned in the payment server
<b>Password</b>	Required. Password for the user name assigned in the payment server
<b>RPNum</b>	Required. The merchant's RP Number in order to uniquely identify merchant's account for the query
<b>PNRef</b>	Optional. The unique payment reference number assigned to the transaction. If this field is provided, all other query fields will be ignored when using <b>PNRef</b> parameter to query the system
<b>BeginDt</b>	<p>Required, except when <b>PNRef</b> is provided. The begin date of the date range in MM/DD/YYYY (or YYYY-MM-DD or YYYY-MM-DDThh:mm:ss) format. This date will be converted to YYYY-MM-DDThh:mm:ss (time is in 24-hour format). If the passed-in BeginDt does not contain time information, BeginDt will default to midnight on the given date. For example:            2005-08-19T12:00:12 is kept as is            2005-08-19 becomes 2005-08-19T00:00:00            2005/08/19 becomes 2005-08-19T00:00:00            08/19/2005 becomes 2005-08-19T00:00:00</p> <p>The query to obtain transactions in the date range is constructed as follows:            (Date DT &gt;=BeginDt) AND (Date DT&lt;EndDt)            where Date DT is the transaction timestamp</p>
<b>EndDt</b>	<p>Required, except when <b>PNRef</b> is provided. The end date of the date range in MM/DD/YYYY (or YYYY-MM-DD or YYYY-MM-DDThh:mm:ss) format. This date will be converted to YYYY-MM-DDThh:mm:ss (time is in 24-hour format). If the passed-in EndDt does not contain time information, EndDt will be incremented to the <i>next day</i> at midnight such that no transaction on the desired end date will be excluded based on its time. For example:            2005-08-19T12:00:12 is kept as is            2005-08-19 becomes 2005-08-20T00:00:00            08/19/2005 becomes 2005-08-20T00:00:00</p>
<b>PaymentType</b>	<p>Optional. If provided, only those transactions matching the <b>PaymentType</b> will be included. Valid values are:</p> <ul style="list-style-type: none"> <li>• <b>'ACH'</b> Automated Clearing House</li> <li>• <b>'ECHECK'</b> Electronic Check</li> <li>• <b>'GUARANTEE'</b> Guarantee check</li> <li>• <b>'PAYRECEIPT'</b> to retrieve receipt images that were uploaded to the payment server</li> <li>• <b>'SETTLE'</b> to retrieve requests to settle transactions</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>'VERIFY'</b> to retrieve pre-authorized checks</li> </ul> <p>Or any permutation of the above values, e.g. <b>"ACH', 'ECHECK'"</b> will pull all transactions with either ACH or ECHECK payment types</p>
<b>ExcludePaymentType</b>	Optional. If provided, any transaction matching the <b>ExcludePaymentType</b> will be excluded
<b>TransType</b>	<p>Optional. If provided, only those transactions matching the <b>TransType</b> will be included. Valid values are:</p> <ul style="list-style-type: none"> <li>• <b>'Authorization'</b> to retrieve previously-authorized (pre-auth) transactions</li> <li>• <b>'Capture'</b> to retrieve captured transactions</li> <li>• <b>'Credit'</b> to retrieve return transactions</li> <li>• <b>'ForceCapture'</b> to retrieve force-auth transactions</li> <li>• <b>'GetStatus'</b> to make an inquiry to the EBT or gift card's balance</li> <li>• <b>'PostAuth'</b> to retrieve post-auth transactions</li> <li>• <b>'Purged'</b> to remove a transaction from the current batch due to an error</li> <li>• <b>'Receipt'</b> to retrieve receipt images that were uploaded to the payment server</li> <li>• <b>'RepeatSale'</b> to retrieve repeat-sale transactions</li> <li>• <b>'Sale'</b> to retrieve sale transactions</li> <li>• <b>'Void'</b> to retrieve void transactions</li> </ul> <p>Or any permutation of the above values, e.g. <b>"Credit', 'Sale'"</b> will pull all transactions with either Credit or Sale transaction types</p>
<b>ExcludeTransType</b>	Optional. If provided, any transaction matching the <b>ExcludeTransType</b> will be excluded
<b>ApprovalCode</b>	Optional. If provided, only those transactions matching the <b>ApprovalCode</b> parameter will be included
<b>Result</b>	<p>Optional. If provided, only those transactions matching the <b>Result</b> will be included. Valid values are:</p> <ul style="list-style-type: none"> <li>• <b>0</b> is Approved</li> <li>• Anything else is declined, if you want all the declined transactions, you should leave this field empty and use the <b>ExcludeResult=0</b> instead</li> </ul>
<b>ExcludeResult</b>	Optional. If provided, any transactions matching the <b>ExcludedResult</b> will be excluded
<b>NameOnCheck</b>	Optional. Check owner's name as it appear on the check, if provided. Only those transactions with check owner's name matching <b>NameOnCheck</b> will be included. Matching is done using wild cards: e.g. "test" will match "test", "1test" and "1test234"
<b>CheckNum</b>	Optional. Check number. If provided, only those transactions with

	matching <b>CheckNum</b> will be included
<b>AcctNum</b>	Optional. Check account number. If provided, only those transactions matching the <b>AcctNum</b> will be included. Matching is done using wild cards
<b>RouteNum</b>	Optional. If provided, any transactions matching the <b>RouteNum</b> (Transit Number) will be excluded. Matching is done using wild cards
<b>ExcludeVoid</b>	Optional. Whether to exclude voided transactions; must either be <b>TRUE</b> or <b>FALSE</b> . The default value is <b>TRUE</b>
<b>User</b>	Optional. The user who originated the transactions. If provided, only those transactions created by the matching <b>User</b> will be included. Matching is done using wild cards
<b>InvoiceId</b>	Optional. The invoice ID that was included in the original transaction. If provided, only those transactions with matching <b>invoiceId</b> will be included. Matching is done using wild cards
<b>SettleFlag</b>	Optional. Whether the transaction was settled; must either be <b>1</b> for true or <b>0</b> for false
<b>SettleMsg</b>	Optional. The settlement ID or message returned from the host
<b>SettleDt</b>	Optional. The settlement timestamp
<b>TransformType</b>	Optional. The type of format to transform the data into. Leave the field blank to default to <b>XML</b> <ul style="list-style-type: none"> <li>• <b>XML</b> will output the plain XML string</li> <li>• <b>XSL</b> will use XSL to transform the XML output</li> <li>• <b>DELIM</b> uses <b>ColDelim</b> and <b>RowDelim</b> to format the output</li> </ul>
<b>Xsl</b>	Optional. This field is used only if the <b>TransformType</b> is <b>XSL</b> . The XSL to transform the resulting dataset; if provided, the resulting dataset will be transformed using this XSL. You may pass in a URL to the XSL file, or the XSL string itself. If this field is not empty, the Web Services will try to locate the file from the URL. If that also fails, it will treat it as a XSL string. In any case, the final XSL string will be loaded and validated against the XSL schema; if it passes, then that XSL will be used for transformation. A sample predefined XSL is included with this Web Services: <ul style="list-style-type: none"> <li>• <a href="http://test.payleap.com/admin/ws/TabDelim.xsl">http://test.payleap.com/admin/ws/TabDelim.xsl</a> for a tab delimited transformation.</li> </ul>
<b>ColDelim</b>	Optional. This field is used only if the <b>TransformType</b> is <b>DELIM</b> . This defines the string that separates each column
<b>RowDelim</b>	Optional. This field is used only if the <b>TransformType</b> is <b>DELIM</b> . This defines the string that separates each row
<b>IncludeHeader</b>	Optional. This field is used only if the <b>TransformType</b> is <b>DELIM</b> . If <b>TRUE</b> , then field headers will be included in the first row using the same delimiter strings; must either be <b>TRUE</b> or <b>FALSE</b>
<b>ExtData</b>	Optional. Extended data in XML format. Valid values are: <ul style="list-style-type: none"> <li>• <code>&lt;IMAGE_TYPE&gt;NO_IMAGE&lt;/IMAGE_TYPE&gt;</code> for no image</li> </ul>

	<ul style="list-style-type: none"> <li>• <code>&lt;IMAGE_TYPE&gt;ONLY_IMAGE&lt;/IMAGE_TYPE&gt;</code> for only the image</li> <li>• <code>&lt;IMAGE_TYPE&gt;ALL&lt;/IMAGE_TYPE&gt;</code> for all images</li> <li>• <code>&lt;CustomerID&gt;CustomerID&lt;/CustomerID&gt;</code> for customer ID</li> <li>• <code>&lt;Amount&gt;Amount&lt;/Amount&gt;</code> Total amount to search transactions for in DDDD.CC format.</li> <li>• <code>&lt;RegisterNum&gt;RegisterNum&lt;/RegisterNum&gt;</code> Register number, originally passed with the transaction, to search transactions for</li> </ul>
--	---

## Example

The following table contains sample data you can use to test this Web service. The **UserName**, **Password**, and **RPNum** parameters should be changed when testing this example yourself. The example data shown will create a list of approved Verify check transactions that were processed between 1/1/2003 and 12/31/2003. The output data is in XML format because the **TransformType** parameter was not specified.

Parameter	Value
<b>UserName</b>	Test
<b>Password</b>	123
<b>RPNum</b>	1
<b>BeginDt</b>	1/1/2003
<b>EndDt</b>	12/31/2003
<b>PaymentType</b>	'VERIFY'
<b>Result</b>	0

Result:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<string xmlns=" http://test.payleap.com/admin/ws">
```

```

<RichDBDS> <TrxDetailCheck> <RichDBDS> <TrxDetailCheck>
<TRX_HD_Key>25</TRX_HD_Key> <Invoice_ID /> <Date_DT>2003-12-
05T13:47:15.1230000-08:00</Date_DT>
<Merchant_Key>1</Merchant_Key> <User_Name_VC>test
</User_Name_VC> <Register_Number_CH> </Register_Number_CH>
<Reseller_Key>100</Reseller_Key> <Payment_Type_ID>VERIFY
</Payment_Type_ID> <Trans_Type_ID>Authorization </Trans_Type_ID>
<Processor_ID>RMRS </Processor_ID> <Last_Update_DT>2003-12-
05T13:47:15.1230000-08:00</Last_Update_DT>
<TRX_Check_Key>5</TRX_Check_Key> <CheckNum_CH>1001

```

```

</CheckNum_CH> <MICR_VC />
<AccountNum_VC>*****7890</AccountNum_VC>
<TransitNum_VC>123456780</TransitNum_VC> <RawMICR_VC />
<DL_VC /> <SS_CH /> <DOB_CH> </DOB_CH> <StateCode_CH>DE
</StateCode_CH> <NameOnCheck_VC>Jane Doe</NameOnCheck_VC>
<EMail_VC /> <Phone_VC /> <Amount_MN>1.0000</Amount_MN>
<Result_CH>0 </Result_CH> <Result_Txt_VC>&lt;INPUT TYPE=HIDDEN
NAME=PARAM_RESPONSE_CODE VALUE="AA"&gt;&lt;INPUT TYPE=HIDDEN
NAME=PARAM_AUTH_SOURCE VALUE="0"&gt;&lt;INPUT TYPE=HIDDEN
NAME=PARAM_CAP_CODE VALU</Result_Txt_VC>
<Host_Approval_CH>963852 </Host_Approval_CH> <Host_Ref_Num_CH>
</Host_Ref_Num_CH> <Result_Msg_VC>APPROVAL
963852</Result_Msg_VC> <Result_Msg1_VC /> <Result_Msg2_VC />
<Host_Date_CH>120503 </Host_Date_CH> <Host_Time_CH>164502
</Host_Time_CH> <IP_VC>192.168.2.22</IP_VC> </TrxDetailCheck>
<TrxDetailCheck> <TRX_HD_Key>24</TRX_HD_Key> <Invoice_ID />
<Date_DT>2003-12-05T13:46:50.0770000-08:00</Date_DT>
<Merchant_Key>1</Merchant_Key> <User_Name_VC>test
</User_Name_VC> <Register_Number_CH> </Register_Number_CH>
<Reseller_Key>100</Reseller_Key> <Payment_Type_ID>VERIFY
</Payment_Type_ID> <Trans_Type_ID>Authorization </Trans_Type_ID>
<Processor_ID>RMRS </Processor_ID> <Last_Update_DT>2003-12-
05T13:46:50.0770000-08:00</Last_Update_DT>
<TRX_Check_Key>4</TRX_Check_Key> <CheckNum_CH>4877
</CheckNum_CH> <MICR_VC />
<AccountNum_VC>*****7890</AccountNum_VC>
<TransitNum_VC>123456780</TransitNum_VC> <RawMICR_VC />
<DL_VC /> <SS_CH /> <DOB_CH> </DOB_CH> <StateCode_CH>WA
</StateCode_CH> <NameOnCheck_VC>John Doe</NameOnCheck_VC>
<EMail_VC /> <Phone_VC /> <Amount_MN>2.0000</Amount_MN>
<Result_CH>0 </Result_CH> <Result_Txt_VC>&lt;INPUT TYPE=HIDDEN
NAME=PARAM_RESPONSE_CODE VALUE="AA"&gt;&lt;INPUT TYPE=HIDDEN
NAME=PARAM_AUTH_SOURCE VALUE="0"&gt;&lt;INPUT TYPE=HIDDEN
NAME=PARAM_CAP_CODE VALU</Result_Txt_VC>
<Host_Approval_CH>963852 </Host_Approval_CH> <Host_Ref_Num_CH>
</Host_Ref_Num_CH> <Result_Msg_VC>APPROVAL
963852</Result_Msg_VC> <Result_Msg1_VC /> <Result_Msg2_VC />
<Host_Date_CH>120503 </Host_Date_CH> <Host_Time_CH>164437
</Host_Time_CH> <IP_VC>192.168.2.22</IP_VC> </TrxDetailCheck>
</RichDBDS>

```

</string>

## ***GetCardType***

This Web service operation returns the name of the card issuer, such as Visa, MasterCard, Amex. The URL to access this operation is: <https://test.payleap.com/SmartPayments/validate.aspx?op=GetCardType>. The text "test.payleap.com" in the URL should be replaced with the actual host name or static IP address of the payment server. Descriptions of the parameters are listed below.

Parameter	Description
<b>CardNumber</b>	Required. The number of a credit card

## Example

The following table contains sample data you can use to test this Web service.

Parameter	Value
<b>CardNumber</b>	5454545454545454

Result:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<string xmlns="http://test.payleap.com/SmartPayments/">MASTERCARD</string>
```

## GetInfo

This Web service retrieves information pertaining to the transaction type (TransType) specified. The URL to access this Web service is:

<https://test.payleap.com/SmartPayments/transact.aspx?op=GetInfo>. The text "test.payleap.com" in the URL should be replaced with the actual host name or static IP address of the payment server. Descriptions of the parameters are listed below.

Parameter	Description
<b>UserName</b>	Required. User name assigned in the payment server
<b>Password</b>	Required. Password for the user name assigned in the payment server
<b>TransType</b>	<p>Required. Valid values are:</p> <ul style="list-style-type: none"> <li> <b>BatchInquiry</b> returns a comma delimited list in a single XML tag that contains the summarized transaction dollar amount and transaction count for each payment method in the current batch. The list is in the following format:  <i>Payment_Method1=0.00, Payment_Method2=0.00</i> </li> <li> <b>Setup</b> returns a comma delimited list in a single XML tag that contains merchant setup information. The list is in the following format:  <i>Setup_Name1=Y N, Setup_Name2=Y N</i> </li> <li> <b>StatusCheck</b> returns "OK" if a connection can be made to the payment server with the supplied user name and password; otherwise, an error message is returned </li> <li> <b>Initialize</b> returns the merchant account setup, including Partner number, Merchant ID, credit card type, phone number, etc. </li> </ul>
<b>ExtData</b>	Optional. Extended data in XML format. Valid values are:

	<ul style="list-style-type: none"> <li>• <b>&lt;TrainingMode&gt;T&lt;/TrainingMode&gt;</b> an indicator that specifies transactions will be processed for local loop back testing</li> <li>• <b>&lt;TrainingMode&gt;F&lt;/TrainingMode&gt;</b> an indicator that specifies transactions will not be processed for local loop back testing</li> <li>• <b>&lt;BatchSequenceNum&gt;Number&lt;/BatchSequenceNum&gt;</b> used when the <b>TransType</b> is <b>BatchInquiry</b> and it is a number that indicates which previous batch or current batch the Payment Server should query from the processor in order to get information about the batch. Currently only support with Global Payments. Valid values are: <ul style="list-style-type: none"> <li>• <b>0</b> = Current open batch (default value if <b>BatchSequenceNum</b> is not specified in <b>ExtData</b>)</li> <li>• <b>1</b> = previous batch</li> <li>• <b>2</b> = the batch before the previous batch specified with the value <b>1</b></li> <li>• <i>N</i> = and so on...</li> </ul> </li> </ul>
--	--

## Examples

The following examples show the results of testing four **TransTypes**. When testing each example yourself, the **UserName** and **Password** parameters should be changed.

### Example 1: BatchInquiry

Parameter	Value
<b>UserName</b>	Test
<b>Password</b>	123
<b>TransType</b>	BatchInquiry

Result:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Response xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://test.payleap.com/SmartPayments/"><Result>0</Result><RespMSG>Approved</RespMSG><ExtData>Credit_Sale_Amount=28.00,Credit_Sale_Count=28,Credit_Return_Amount=7.00,Credit_Return_Count=7,Credit_Net_Amount=21.00,Credit_Net_Count=35,Debit_Sale_Amount=9.00,Debit_Sale_Count=6,Debit_Return_Amount=1.00,Debit_Return_Count=1,Debit_Net_Amount=8.00,Debit_Net_Count=7,Check_Sale_Amount=0.00,Check_Sale_Count=0,Check_Net_Amount=0.00,Check_Net_Count=0</ExtData></Response>
```

### Example 2: Setup

Parameter	Value
-----------	-------

<b>UserName</b>	Test
<b>Password</b>	123
<b>TransType</b>	Setup

Result:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Response xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://test.payleap.com/SmartPayments/"><Result>0</Result><Res
pMSG>Approved</RespMSG><ExtData>Force_Duplicates=N,Auto_Close_Batc
h=Y,Industry=R,DEBIT=Y,AMEX=Y,CARTBLANCH=Y,DINERS=Y,DISCOVER=
Y,JAL=Y,JCB=Y,MASTERCARD=Y,VISA=Y,EBT=Y</ExtData></Response>
```

### Example 3: StatusCheck

Parameter	Value
<b>UserName</b>	Test
<b>Password</b>	123
<b>TransType</b>	StatusCheck

Result:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Response xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://test.payleap.com/SmartPayments/"><Result>0</Result><Res
pMSG>Approved</RespMSG><ExtData>OK</ExtData></Response>
```

### Example 4: Initialize

Parameter	Value
<b>UserName</b>	Test
<b>Password</b>	123
<b>TransType</b>	Initialize

Result:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Response xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://test.payleap.com/SmartPayments/">
```

```

<Result>0</Result><RespMSG>Approved</RespMSG><ExtData><Partner>110<
/Partner> <Vendor>206</Vendor> <MerchantID /> <PinPadKeyManagement
/><LiveURL>https://www.YourWebSite.com</LiveURL><LiveURL1>https://w
ww1.YourWebSite.com</LiveURL1><TestURL>https://test.YourWebSite.com
</TestURL><TestURL1>https://test.YourWebSite.com</TestURL1><EPSPay>/
pay/payxml.aspx</EPSPay><EPSlogin>/Admin/login.aspx</EPSlogin><Phone
1>425-123-1234</Phone1><Phone2>425-123-
1234</Phone2><Auto_Close_Batch>N</Auto_Close_Batch><eCheck>Y</eCheck>
<CreditCard>Y</CreditCard><PaymentTypes><CardType>Amex</CardType><Car
dType>CartBlanch</CardType><CardType>Diners</CardType><CardType>Disc
over</CardType><CardType>JAL</CardType><CardType>JCB</CardType><Card
Type>MasterCard</CardType><CardType>Visa</CardType></PaymentTypes><E
xpressPay><CardType>Visa</CardType><LTAmount>25</LTAmount><Am
ount>0</Amount><EntryMethod>S</EntryMethod><ProcessingRule>On-
Line</ProcessingRule><PrintReceipt>N</PrintReceipt></ExpressPay></Ext
Data>

```

</Response>

## GetOpenBatchSummary

This Web service operation retrieves a payment type transaction summary of the current open batch for a merchant. The URL to access this Web service is: <https://test.payleap.com/admin/ws/trxdetail.aspx?op=GetOpenBatchSummary>. The text “test.payleap.com” in the URL should be replaced with the actual host name or static IP address of the payment server. Descriptions of the parameters are listed below.

Parameter	Value
<b>UserName</b>	Required. User name assigned in the payment server
<b>Password</b>	Required. Password for the user name assigned in the payment server
<b>RPNum</b>	Required. The number uniquely identifies each merchant
<b>BeginDt</b>	Optional. The begin date of the date range in MM/DD/YYYY format. This date will be converted to: MM/DD/YYYYT00:00:00:0000AM
<b>EndDt</b>	Optional. The end date of the date range in MM/DD/YYYY format. This date will be converted to: MM/DD/YYYYT12:59:59:9999PM
<b>ExtData</b>	Currently there is no data value available

## Example

The following table contains sample data you can use to test this Web service. The **UserName**, **Password**, and **RPNum** parameters should be changed when testing this example yourself. The example data shown will create a categorized summary list of all transactions in the current open batch.

Parameter	Value
UserName	Test
Password	123
RPNum	1

Result:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<string xmlns="http://test.payleap.com/Admin/ws"><OpenBatchSummary> <Table>
  <Payment_Type_ID>DEBIT </Payment_Type_ID>
  <Authorization>0</Authorization> <Capture>0</Capture>
  <ForceCapture>0</ForceCapture> <PostAuth>0</PostAuth>
  <Return>7.0000</Return> <Sale>61.0000</Sale>
  <Receipt>0</Receipt> <RepeatSale>0</RepeatSale>
  <Authorization_Cnt>0</Authorization_Cnt>
  <Capture_Cnt>0</Capture_Cnt>
  <ForceCapture_Cnt>0</ForceCapture_Cnt>
  <PostAuth_Cnt>0</PostAuth_Cnt> <Return_Cnt>7</Return_Cnt>
  <Sale_Cnt>58</Sale_Cnt> <Receipt_Cnt>0</Receipt_Cnt>
  <RepeatSale_Cnt>0</RepeatSale_Cnt> <Cnt>65</Cnt> </Table>
  <Table> <Payment_Type_ID>EBT </Payment_Type_ID>
  <Authorization>0</Authorization> <Capture>0</Capture>
  <ForceCapture>0</ForceCapture> <PostAuth>0</PostAuth>
  <Return>132.0000</Return> <Sale>150.0000</Sale>
  <Receipt>0</Receipt> <RepeatSale>0</RepeatSale>
  <Authorization_Cnt>0</Authorization_Cnt>
  <Capture_Cnt>0</Capture_Cnt>
  <ForceCapture_Cnt>0</ForceCapture_Cnt>
  <PostAuth_Cnt>0</PostAuth_Cnt> <Return_Cnt>15</Return_Cnt>
  <Sale_Cnt>24</Sale_Cnt> <Receipt_Cnt>0</Receipt_Cnt>
  <RepeatSale_Cnt>0</RepeatSale_Cnt> <Cnt>39</Cnt> </Table>
  <Table> <Payment_Type_ID>MASTERCARD</Payment_Type_ID>
  <Authorization>0</Authorization> <Capture>0</Capture>
  <ForceCapture>0</ForceCapture> <PostAuth>0</PostAuth>
  <Return>0</Return> <Sale>4.0000</Sale> <Receipt>0</Receipt>
  <RepeatSale>7.0000</RepeatSale>
  <Authorization_Cnt>0</Authorization_Cnt>
  <Capture_Cnt>0</Capture_Cnt>
  <ForceCapture_Cnt>0</ForceCapture_Cnt>
  <PostAuth_Cnt>0</PostAuth_Cnt> <Return_Cnt>0</Return_Cnt>
  <Sale_Cnt>4</Sale_Cnt> <Receipt_Cnt>0</Receipt_Cnt>
  <RepeatSale_Cnt>7</RepeatSale_Cnt> <Cnt>11</Cnt> </Table>
  <Table> <Payment_Type_ID>VISA </Payment_Type_ID>
  <Authorization>0</Authorization> <Capture>0</Capture>
  <ForceCapture>1.0000</ForceCapture> <PostAuth>1.0000</PostAuth>
  <Return>0</Return> <Sale>19.0000</Sale> <Receipt>0</Receipt>
  <RepeatSale>0</RepeatSale>
```

```

<Authorization_Cnt>0</Authorization_Cnt>
<Capture_Cnt>0</Capture_Cnt>
<ForceCapture_Cnt>1</ForceCapture_Cnt>
<PostAuth_Cnt>1</PostAuth_Cnt> <Return_Cnt>0</Return_Cnt>
<Sale_Cnt>19</Sale_Cnt> <Receipt_Cnt>0</Receipt_Cnt>
<RepeatSale_Cnt>0</RepeatSale_Cnt> <Cnt>21</Cnt> </Table>
</OpenBatchSummary></string>

```

## ***IsCommercialCard***

This Web service operation checks whether the card number entered is for a commercial card or not. The URL to access this Web service is:

<https://test.payleap.com/SmartPayments/validate.asmx?op=IsCommercialCard>. The text “test.payleap.com” in the URL should be replaced with the actual host name or static IP address of the payment server. Please note due to ever-changing card bin-ranges and other factors, not all Commercial/Purchase cards can be definitely determined by this method. Developers should design their applications while keeping this fact in mind. Descriptions of the parameters are listed below.

Parameter	Description
<b>CardNumber</b>	Required. The number of a credit card

## **Examples**

The following tables contain sample data you can use to test this Web service. The return result will be either true or false.

**Example 1:** The example data shown will result in return false.

Parameter	Value
<b>CardNumber</b>	5454545454545454

Result:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<boolean xmlns="http://test.payleap.com/SmartPayments/">false</boolean>
```

**Example 2:** The example data shown will result in return true.

Parameter	Value
<b>CardNumber</b>	4055011111111111

Result:

<?xml version="1.0" encoding="utf-8" ?>

<boolean xmlns="http://test.payleap.com/SmartPayments/" >true</boolean>

## GetNetworkID

This web service allows for returning the debit network ID if the debit card number matches any of these network's bin ranges. If there is a match, the card can likely be used as a debit card and processed through the Debit Network. This BIN range is stored in %EPSROOT%\xmlfiles\CardBin.txt which needs to be periodically updated MANUALLY probably every month or so.

This web service can be accessed through this URL:  
<http://test.payleap.com/smartpayments/validate.asmx?op=GetNetworkID>

Note: This was tested with a live Debit card number.

## Example

Parameter	Value
<b>Username</b>	Required. The username of the admin user.
<b>Password</b>	Required. The password of the admin user
<b>CardNumber</b>	Required. The customer's debit card number.

Sample: <http://test.payleap.com/smartpayments/validate.asmx?op=GetNetworkID>

```
POST /smartpayments/validate.asmx/GetNetworkID HTTP/1.1
Host: test.payleap.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length
```

**UserName=string&Password=string&CardNumber=string**

Parameter	Value
UserName:	<input type="text" value="test"/>
Password:	<input type="text" value="123"/>
CardNumber:	<input type="text" value="XXXXXXXXXXXX2557"/>

## Response

```
<?xml version="1.0" encoding="utf-8" ?>  
  
= <Response xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xmlns="http://test.payleap.com/SmartPayments/">  
  
  <Result>0</Result>  
  
  <RespMSG>Interlink</RespMSG>  
  
  <Message>Y</Message>  
  
  <Message1>NetworkID=ILK,NetworkAuthorizerNum=48</Message1>  
  
  <Message2>OfflineSupported=True</Message2>  
  
  <AuthCode>ILK</AuthCode>  
  
  <HostCode>48</HostCode>  
  
  </Response>
```

## BIN Management Network Table Values

Debit Network	Network ID	Network Authorization Number
Accel	ACL	69
AFFN	AFN	68
Alaska Option	AKO	61
CU24	C24	85
Interlink	ILK	48
Jeanie	JEN	86
Star Northeast (MAC)	MAC	17
Maestro	MAE	40

Nets	NET	83
NYCE	NYC	28
Pulse	PUL	06
Star Southeast	SES	07
Shazam	SHZ	58
Star West	STX	23
TYME	TYM	78

## Web Service Response Field

### *Transact.asmx*

Response Field	Data Type	Value Description	Remarks
<b>AuthCode</b>	A string value up to 50 characters	Returns the transaction result code from the payment processor	This value can be either an approval code, for approved transactions, or an error code, for declined transactions
<b>ExtData</b>	A string value up to 500 characters	Returns extra data from the processed transaction	The value of ExtData will be in a specific format. The format typically consists of the name of the data field, an equal sign, and then the value for the data field. Multiple data fields are separated with a comma. See the "Web Service ExtData Response Field Data Elements" for full description of data elements that can be returned. The following is an example of the format: <i>ExtName1=ExtValue1,ExtName2=ExtValue2</i>
<b>GetAVSResult</b>	A string value up to 1 character	Returns the overall address verification result code from the payment processor	When programmatically validating an AVS Result, this value should ALWAYS be used instead of any formatted response message describing the result
<b>GetAVS ResultTXT</b>	A string value up to 25 characters	Returns the formatted response message when address verification is performed	Do NOT use this when programmatically validating a transaction's AVS result; please see GetAVSResult field

<b>Get CommercialCard</b>	A string value representing a Boolean value	Returns the payment processor's response indicator that specifies if the card is a commercial card	This value is only applicable to credit card transactions. The card verification number is typically printed on the back of the card and not embossed on the front. It is used as an extra authentication method for "card not present" transactions. When programmatically validating a CV Result, this value should ALWAYS be used instead of any formatted response message describing the result
<b>GetCVResult</b>	A string value up to 1 character	Returns the card verification result code from the payment processor	This value is only applicable to credit card transactions. The card verification number is typically printed on the back of the card and not embossed on the front. It is used as an extra authentication method for "card not present" transactions. When programmatically validating a CV Result, this value should ALWAYS be used instead of any formatted response message describing the result
<b>GetCVResultTXT</b>	A string value up to 25 characters	Returns the formatted response message when card verification is performed	This value is only applicable to credit card transactions. Do NOT use this when programmatically validating a transaction's CV result; please see GetCVResult field
<b>GetStreetMatchTXT</b>	A string value up to 25 characters	Returns the formatted response message when street number address verification is performed	This value will typically be "Match", for correctly matching the street address, or "No Match", for an incorrect street address
<b>GetZipMatchTXT</b>	A string value up to 25 characters	Returns the formatted response message when zip code address verification is performed	This value will typically be "Match", for correctly matching the zip code, or "No Match", for an incorrect zip code
<b>HostCode</b>	A string value up to 30 characters	Typically returns a number which uniquely identifies the transaction in the payment processor	This value may not be returned for all payment processors
<b>Message</b>	A string value up to 50 characters	Returns a formatted response message concerning the processed transaction	This value will typically be "APPROVAL", for approved transactions, or an error message, for declined transactions. Do NOT use this when programmatically validating a transaction's result; please see Result field below
<b>Message1</b>	A string value up to 50 characters	Returns an extra formatted response message giving more information about the processed transaction	The Payment Server will only populate this field when there is applicable information from the payment processor to return
<b>Message2</b>	A string value up to 50 characters	Returns an extra formatted response message giving more information about the processed transaction	The Payment Server will only populate this field when there is applicable information from the payment processor to return
<b>PNRef</b>	A string value representing a signed 32-bit integer	Returns a number which uniquely identifies the transaction in the payment gateway	
<b>RespMSG</b>	A string value up to 50	Returns the response message concerning the	This value is typically either Approved or Declined. Do NOT use this when

	characters	processed transaction	programmatically validating a transaction's result; please see Result field below
<b>Result</b>	A string value representing a signed 32-bit integer	Returns the transaction result code from the payment gateway which signifies the result of the transaction (i.e. approved, decline, etc.)	When programmatically validating a transaction's result, this value should ALWAYS be used instead of any response message describing the result. See the "Result Response Fields Definitions" section for a full list of result values and descriptions

### ***Transact.asmx Web Service ExtData Response Field Data Elements***

Data Element Name	Value Description	Remarks
<b>BatchNum</b>	Returns the current batch number, returned by the payment processor, for transactions, settlement, and batch inquiries	Not all payment processors support returning this data element
<b>CardType</b>	Returns the credit card type (VISA, MASTERCARD, etc), payment method (Debit, EBT, or EGC) for card-based payments	This value is not returned for Check/ACH payments
<b>InvNum</b>	Returns the same invoice number for the transaction that was originally sent in the request to the Payment Server	

### ***TrxDetail.asmx***

Response Field	Data Type	Value Description	Remarks
<b>Account_Type_CH</b>	A string value up to 10 characters	Returns the card type of the transaction, e.g. VISA, Diners, EBT	
<b>AccountNum_VC</b>	A string value up to 200 characters	Returns the check account number	This field will be masked out with asterisk (*) characters except for the last 4 digits if the System Security Level of the user is set to 1
<b>Acct_Num_CH</b>	A string value up to 200 characters	Returns credit card number	This field will be masked out with asterisk (*) characters except for the last 4 digits if the System Security Level of the user is set to 1
<b>Amount_MN</b>	A string value	Returns the check's	

	representing a signed 64-bit real number	total amount	
<b>Approval_Code_CH</b>	A string value up to 50 characters	Returns the response code from the payment processor	
<b>Auth_Amt_MN</b>	A string value representing a signed 64-bit real number	Returns the authorized amount of a card transaction	
<b>Authorization</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all Authorization (PreAuth) transactions	
<b>Authorization_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all Authorization (PreAuth) transactions	
<b>AVS_Resp_CH</b>	A string value up to 1 character	Returns the address verification result code from the payment processor	
<b>AVS_Resp_Txt_VC</b>	A string value up to 25 characters	Returns the formatted response message when address verification is performed	
<b>Batch_Number</b>	A string value up to 10 characters	Returns the batch number for the transaction that was returned by the payment processor	Not all payment processors support returning this data element
<b>Capture</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all Capture transactions	This value will always return "0"
<b>Capture_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all Capture transactions	This value will always return "0"
<b>Card_Info_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the CC_Info_T table in the database	
<b>Cash_Back_Amt_MN</b>	A string value representing a signed 64-bit real number	Returns the cash back amount for a debit or EBT transaction	
<b>CheckNum_CH</b>	A string value up to 10 characters	Returns the check number	
<b>Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all transactions	
<b>CustomerID</b>	A string value up to 50 characters	Returns the Customer ID of a customer to which the transaction belongs to	

<b>CV_Resp_CH</b>	A string value up to 1 character	Returns the card verification result code from the payment processor	
<b>CV_Resp_Txt_VC</b>	A string value up to 25 characters	Returns the formatted response message when card verification is performed	
<b>Date_DT</b>	A string value representing a date and time	Returns the date on which the transaction is first made	
<b>ERROR</b>	A string value up to 200 characters	Returns an error message when a problem occurs during the transaction processing	
<b>Exp_CH</b>	A string value up to 10 characters	Returns the credit card expiration date	
<b>ForceCapture</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all ForceCapture (ForceAuth) transactions	
<b>ForceCapture_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all ForceCapture (ForceAuth) transactions	
<b>Host_Date_CH</b>	A string value up to 10 characters	Returns the payment processor's date on which the transaction is performed	
<b>Host_Ref_Num_CH</b>	A string value up to 30 characters	Returns a number which uniquely identifies the transaction for the payment processor	
<b>Host_Time_CH</b>	A string value up to 10 characters	Returns the payment processor's time at which the transaction was performed	
<b>Invoice_ID</b>	A string value up to 100 characters	Returns the transaction's Invoice number	
<b>IP_VC</b>	A string value up to 15 characters	Returns the IP address of the client machine from which the transaction was processed	
<b>Last_Update_DT</b>	A string value representing a date and time	Returns the date and time on which the transaction is last	

		modified	
<b>Manual</b>	A string value representing a Boolean value	Returns the card was swiped or not	
<b>Merchant_Key</b>	A string value representing a signed 32-bit integer	Returns a number which uniquely identifies a merchant	
<b>Name_on_Card_VC</b>	A string value up to 25 characters	Returns the name of the cardholder	
<b>NameOnCheck_VC</b>	A string value up to 25 characters	Returns the check payer's name on the check	
<b>Orig_TRX_HD_Key</b>	A string value representing a signed 32-bit integer	Returns the TRX_HD_Key on which the current transaction is based	
<b>Payment_Type_ID</b>	A string value up to 10 characters	Returns the payment type, e.g. ECHECK	
<b>PostAuth</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all PostAuth transactions	
<b>PostAuth_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all PostAuth transactions	
<b>Processor_ID</b>	A string value up to 10 characters	Returns the name the payment processor, e.g. Vital	
<b>Receipt</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all transactions with a Receipt	This value will always return "0"
<b>Receipt_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all transactions with a Receipt	This value will always return "0"
<b>Ref_Number_CH</b>	A string	Not currently used	This field is not the unique transaction identifier (also called PNRef) of the Payment Server. See the field TRX_HD_Key for the PNRef value
<b>Register_Number_CH</b>	A string value up to 10 characters	Returns the register number of a transaction	
<b>RepeatSale</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all RepeatSale (Recurring Billing/Installment) transactions	
<b>RepeatSale_Cnt</b>	A string value	Returns the transaction	

	representing a signed 32-bit integer	count of all RepeatSale (Recurring Billing/Installment) transactions	
<b>Reseller_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the Reseller_T table in the database	
<b>Result_CH</b>	A string value up to 50 characters	Returns the transaction processing result, e.g. 0, 12. "0" for approval, "12" for decline	
<b>Result_Msg_VC</b>	A string value up to 50 characters	Returns the check transaction's processing result	
<b>Result_Msg1_VC</b>	A string value up to 50 characters	Returns an extra formatted response message giving more information about the processed transaction	
<b>Result_Msg2_VC</b>	A string value up to 50 characters	Returns an extra formatted response message giving more information about the processed transaction	
<b>Result_Txt_VC</b>	A string value up to 150 characters	Returns the text message of either approval or decline for the transaction processing result	
<b>Return</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all Return (Credit) transactions	
<b>Return_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all Return (Credit) transactions	
<b>Sale</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all Sale transactions	
<b>Sale_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all Sale transactions	
<b>Settle_Date_DT</b>	A string value representing a date and time	Returns the date on which the transaction is settled	
<b>Settle_Flag_CH</b>	A string value representing a Boolean value	Returns if the transaction is settled or not	
<b>StateCode_CH</b>	A string value up to 10 characters	Returns the state code	
<b>Street_CH</b>	A string value up to 25	Returns the billing	

	characters	street address of the credit card	
<b>SureCharge_Amt_MN</b>	A string value representing a signed 64-bit real number	Returns the sure charge amount of a transaction	
<b>Tip_Amt_MN</b>	A string value representing a signed 64-bit real number	Returns the tip amount of a transaction	
<b>Trans_Type_ID</b>	A string value up to 20 characters	Returns the transaction type, e.g. Sale, Credit	
<b>Transport_Method</b>	A string	Returns the Transportation Method	Only for use with Dial-up transactions
<b>Transport_EndPoint</b>	A string	Returns the Transportation's Ending Destination	Only for use with Dial-up transactions
<b>TransitNum_VC</b>	A string value up to 200 characters	Returns the transit/routing number	This field will be masked out with asterisk (*) characters except for the last 4 digits if the System Security Level of the user is set to 1
<b>TRX_Card_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the TRX_Card_T table in the database	
<b>TRX_Check_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the TRX_Check_T table in the database	
<b>TRX_HD_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the TRX_Header_T table in the database	This field is the unique transaction identifier (also called PNRef) of the Payment Server. Use its value when submitting transactions based on a previous transaction (i.e. Voids) through the Transact.aspx Web Service
<b>TRX_Settle_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the TRX_Settle_T table in the database	
<b>TRX_Settle_Msg_VC</b>	A string value up to 25 characters	Returns the transaction's settlement message	
<b>Type_CH</b>	A string value up to 10 characters	Returns the credit card type, e.g. VISA, MASTERCARD	
<b>User_Name_VC</b>	A string value up to 25 characters	Returns the username, under which the transactions were made	

<b>Void_Flag_CH</b>	A string value representing a Boolean value	Returns the transaction is voided or not	
<b>Zip_CH</b>	A string value up to 10 characters	Returns the billing zip code of the credit card	
<b>Auth_Amt_MN</b>	A string value representing a signed 64-bit real number	Returns the authorized amount of a card transaction	
<b>Authorization</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all Authorization (PreAuth) transactions	
<b>Authorization_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all Authorization (PreAuth) transactions	
<b>AVS_Resp_CH</b>	A string value up to 1 character	Returns the address verification result code from the payment processor	
<b>AVS_Resp_Txt_VC</b>	A string value up to 25 characters	Returns the formatted response message when address verification is performed	
<b>Batch_Number</b>	A string value up to 10 characters	Returns the batch number for the transaction that was returned by the payment processor	Not all payment processors support returning this data element
<b>Capture</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all Capture transactions	This value will always return "0"
<b>Capture_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all Capture transactions	This value will always return "0"
<b>Card_Info_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the CC_Info_T table in the database	
<b>Cash_Back_Amt_MN</b>	A string value representing a signed 64-bit real number	Returns the cash back amount for a debit or EBT transaction	
<b>CheckNum_CH</b>	A string value up to 10 characters	Returns the check number	
<b>Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all transactions	
<b>CustomerID</b>	A string value up to 50 characters	Returns the Customer ID of a customer to which the transaction belongs to	

<b>CV_Resp_CH</b>	A string value up to 1 character	Returns the card verification result code from the payment processor	
<b>CV_Resp_Txt_VC</b>	A string value up to 25 characters	Returns the formatted response message when card verification is performed	
<b>Date_DT</b>	A string value representing a date and time	Returns the date on which the transaction is first made	
<b>ERROR</b>	A string value up to 200 characters	Returns an error message when a problem occurs during the transaction processing	
<b>Exp_CH</b>	A string value up to 10 characters	Returns the credit card expiration date	
<b>ForceCapture</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all ForceCapture (ForceAuth) transactions	
<b>ForceCapture_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all ForceCapture (ForceAuth) transactions	
<b>Host_Date_CH</b>	A string value up to 10 characters	Returns the payment processor's date on which the transaction is performed	
<b>Host_Ref_Num_CH</b>	A string value up to 30 characters	Returns a number which uniquely identifies the transaction for the payment processor	
<b>Host_Time_CH</b>	A string value up to 10 characters	Returns the payment processor's time at which the transaction was performed	
<b>Invoice_ID</b>	A string value up to 100 characters	Returns the transaction's Invoice number	
<b>IP_VC</b>	A string value up to 15 characters	Returns the IP address of the client machine from which the transaction was processed	
<b>Last_Update_DT</b>	A string value representing a date and time	Returns the date and time on which the transaction is last	

		modified	
<b>Manual</b>	A string value representing a Boolean value	Returns the card was swiped or not	
<b>Merchant_Key</b>	A string value representing a signed 32-bit integer	Returns a number which uniquely identifies a merchant	
<b>Name_on_Card_VC</b>	A string value up to 25 characters	Returns the name of the cardholder	
<b>NameOnCheck_VC</b>	A string value up to 25 characters	Returns the check payer's name on the check	
<b>Orig_TRX_HD_Key</b>	A string value representing a signed 32-bit integer	Returns the TRX_HD_Key on which the current transaction is based	
<b>Payment_Type_ID</b>	A string value up to 10 characters	Returns the payment type, e.g. ECHECK	
<b>PostAuth</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all PostAuth transactions	
<b>PostAuth_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all PostAuth transactions	
<b>Processor_ID</b>	A string value up to 10 characters	Returns the name the payment processor, e.g. Vital	
<b>Receipt</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all transactions with a Receipt	This value will always return "0"
<b>Receipt_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all transactions with a Receipt	This value will always return "0"
<b>Ref_Number_CH</b>	A string	Not currently used	This field is not the unique transaction identifier (also called PNRef) of the Payment Server. See the field TRX_HD_Key for the PNRef value
<b>Register_Number_CH</b>	A string value up to 10 characters	Returns the register number of a transaction	
<b>RepeatSale</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all RepeatSale (Recurring Billing/Installment) transactions	
<b>RepeatSale_Cnt</b>	A string value	Returns the transaction	

	representing a signed 32-bit integer	count of all RepeatSale (Recurring Billing/Installment) transactions	
<b>Reseller_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the Reseller_T table in the database	
<b>Result_CH</b>	A string value up to 50 characters	Returns the transaction processing result, e.g. 0, 12. "0" for approval, "12" for decline	
<b>Result_Msg_VC</b>	A string value up to 50 characters	Returns the check transaction's processing result	
<b>Result_Msg1_VC</b>	A string value up to 50 characters	Returns an extra formatted response message giving more information about the processed transaction	
<b>Result_Msg2_VC</b>	A string value up to 50 characters	Returns an extra formatted response message giving more information about the processed transaction	
<b>Result_Txt_VC</b>	A string value up to 150 characters	Returns the text message of either approval or decline for the transaction processing result	
<b>Return</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all Return (Credit) transactions	
<b>Return_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all Return (Credit) transactions	
<b>Sale</b>	A string value representing a signed 64-bit real number	Returns the dollar amount of all Sale transactions	
<b>Sale_Cnt</b>	A string value representing a signed 32-bit integer	Returns the transaction count of all Sale transactions	
<b>Settle_Date_DT</b>	A string value representing a date and time	Returns the date on which the transaction is settled	
<b>Settle_Flag_CH</b>	A string value representing a Boolean value	Returns if the transaction is settled or not	
<b>StateCode_CH</b>	A string value up to 10 characters	Returns the state code	
<b>Street_CH</b>	A string value up to 25	Returns the billing	

	characters	street address of the credit card	
<b>SureCharge_Amt_MN</b>	A string value representing a signed 64-bit real number	Returns the sure charge amount of a transaction	
<b>Tip_Amt_MN</b>	A string value representing a signed 64-bit real number	Returns the tip amount of a transaction	
<b>Total_Amt_MN</b>	A string value representing a signed 64-bit real number	Returns the total amount of a transaction	
<b>Trans_Type_ID</b>	A string value up to 20 characters	Returns the transaction type, e.g. Sale, Credit	
<b>Transport_Method</b>	A string	Returns the Transportation Method	Only for use with Dial-up transactions
<b>Transport_EndPoint</b>	A string	Returns the Transportation's Ending Destination	Only for use with Dial-up transactions
<b>TransitNum_VC</b>	A string value up to 200 characters	Returns the transit/routing number	This field will be masked out with asterisk (*) characters except for the last 4 digits if the System Security Level of the user is set to 1
<b>TRX_Card_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the TRX_Card_T table in the database	
<b>TRX_Check_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the TRX_Check_T table in the database	
<b>TRX_HD_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the TRX_Header_T table in the database	This field is the unique transaction identifier (also called PNRef) of the Payment Server. Use its value when submitting transactions based on a previous transaction (i.e. Voids) through the Transact.asmx Web Service
<b>TRX_Settle_Key</b>	A string value representing a signed 32-bit integer	Returns the primary key of the TRX_Settle_T table in the database	
<b>TRX_Settle_Msg_VC</b>	A string value up to 25 characters	Returns the transaction's settlement message	
<b>Type_CH</b>	A string value up to 10 characters	Returns the credit card type, e.g. VISA, MASTERCARD	

<b>User_Name_VC</b>	A string value up to 25 characters	Returns the username, under which the transactions were made	
<b>Void_Flag_CH</b>	A string value representing a Boolean value	Returns the transaction is voided or not	
<b>Zip_CH</b>	A string value up to 10 characters	Returns the billing zip code of the credit card	

## Response Values

### ***Result Response Field Definitions (Error Codes)***

The list below contains result codes returned in the Result response field of the XMLPayResponse when using a transaction processing Transact.asmx web service operation (i.e. ProcessCreditCard, ProcessCheck, etc). A decline returned by the payment processor for this response field is value twelve (12) or thirteen (13). An approval is value zero (0). Any other value is an error code, which is returned by the payment gateway and not by the payment processor. Please note that when programmatically validating a transaction's result, this value should be used instead of any response message describing the result. I.e. do *not* use RespMSG or Message response fields, as these values may vary. Please note that this list is subject to change without prior notice.

<b>Value</b>	<b>Description</b>
-100	Transaction NOT Processed; Generic Host Error
0	Approved
1	User Authentication Failed
2	Invalid Transaction
3	Invalid Transaction Type
4	Invalid Amount
5	Invalid Merchant Information
7	Field Format Error
8	Not a Transaction Server
9	Invalid Parameter Stream
10	Too Many Line Items
11	Client Timeout Waiting for Response

12	Decline
13	Referral
14	Transaction Type Not Supported In This Version
19	Original Transaction ID Not Found
20	Customer Reference Number Not Found
22	Invalid ABA Number
23	Invalid Account Number
24	Invalid Expiration Date
25	Transaction Type Not Supported by Host
26	Invalid Reference Number
27	Invalid Receipt Information
28	Invalid Check Holder Name
29	Invalid Check Number
30	Check DL Verification Requires DL State
40	Transaction did not connect (to NCN because SecureNCIS is not running on the web server)
50	Insufficient Funds Available
99	General Error
100	Invalid Transaction Returned from Host
101	Timeout Value too Small or Invalid Time Out Value
102	Processor Not Available
103	Error Reading Response from Host
104	Timeout waiting for Processor Response
105	Credit Error
106	Host Not Available
107	Duplicate Suppression Timeout
108	Void Error
109	Timeout Waiting for Host Response
110	Duplicate Transaction
111	Capture Error
112	Failed AVS Check
113	Cannot Exceed Sales Cap

1000	Generic Host Error
1001	Invalid Login
1002	Insufficient Privilege or Invalid Amount
1003	Invalid Login Blocked
1004	Invalid Login Deactivated
1005	Transaction Type Not Allowed
1006	Unsupported Processor
1007	Invalid Request Message
1008	Invalid Version
1010	Payment Type Not Supported
1011	Error Starting Transaction
1012	Error Finishing Transaction
1013	Error Checking Duplicate
1014	No Records To Settle (in the current batch)
1015	No Records To Process (in the current batch)

## ***AVS Response Codes***

The following table contains the possible response values returned for address verification (AVS).

<b>Value</b>	<b>Description</b>
X	Exact: Address and nine-digit Zip match
Y	Yes: Address and five-digit Zip match
A	Address: Address matches, Zip does not
Z	5-digit Zip: 5-digit Zip matches, address doesn't
W	Whole Zip: 9-digit Zip matches, address doesn't
N	No: Neither address nor Zip matches
U	Unavailable: Address information not available
G	Unavailable: Address information not available for international transaction

R	Retry: System unavailable or time-out
E	Error: Transaction unintelligible for AVS or edit error found in the message that prevents AVS from being performed
S	Not Supported: Issuer doesn't support AVS service
B	* Street Match: Street addresses match for international transaction, but postal code doesn't
C	* Street Address: Street addresses and postal code not verified for international transaction
D	* Match: Street addresses and postal codes match for international transaction
I	* Not Verified: Address Information not verified for International transaction
M	* Match: Street addresses and postal codes match for international transaction
P	* Postal Match: Postal codes match for international transaction, but street address doesn't
0	** No response sent
5	Invalid AVS response

\* These values are Visa specific.

\*\* These values are returned by the Payment Server and not the processor.

## ***CV Response Codes***

The following table contains the possible response values returned for a CVV2/CVC2/CID check.

<b>Value</b>	<b>Description</b>
M	CVV2/CVC2/CID Match
N	CVV2/CVC2/CID No Match
P	Not Processed
S	Issuer indicates that the CV data should be present on the card, but the merchant has indicated that the CV data is not present on the card.
U	Unknown / Issuer has not certified for CV or issuer has not provided Visa/MasterCard with the CV encryption keys.
X	Server Provider did not respond

## Valid Parameter Input Characters

The table below displays all allowable characters (unless otherwise noted) that are accepted by the Payment Server. Characters are displayed in Courier New font. All other characters may cause undesirable results.

**Table 1. Valid Data Characters**

DEC	HEX	Character	DEC	HEX	Character	DEC	HEX	Character
32	20	Space	63	3F	?	96	60	`
33	21	!	64	40	@	97	61	a
34	22	"	65	41	A	98	62	b
35	23	#	66	42	B	99	63	c
36	24	\$	67	43	C	100	64	d
37	25	%	68	44	D	101	65	e
38	26	&	69	45	E	102	66	f
39	27	'	70	46	F	103	67	g
40	28	(	71	47	G	104	68	h
41	29	)	72	48	H	105	69	i
42	2A	*	73	49	I	106	6A	j
43	2B	+	74	4A	J	107	6B	k
44	2C	,	75	4B	K	108	6C	l
45	2D	-	76	4C	L	109	6D	m
46	2E	.	77	4D	M	110	6E	n
47	2F	/	78	4E	N	111	6F	o
48	30	0	79	4F	O	112	70	p
49	31	1	80	50	P	113	71	q
50	32	2	81	51	Q	114	72	r
51	33	3	82	52	R	115	73	s
52	34	4	83	53	S	116	74	t
53	35	5	84	54	T	117	75	u
54	36	6	85	55	U	118	76	v
55	37	7	86	56	V	119	77	w
56	38	8	87	57	W	120	78	x
57	39	9	88	58	X	121	79	y
58	3A	:	89	59	Y	122	7A	z
59	3B	;	90	5A	Z	123	7B	{
60	3C	<	92	5C	\	124	7C	
61	3D	=	94	5E	^	125	7D	}
62	3E	>	95	5F	_	126	7E	~

## Character Removal

The table below displays all acceptable characters that must be removed by the Payment Server before submitting information to the Web Service operations. See each input parameter for each Web Service operation in order to know which input parameters will have these characters removed. This character removal ensures that the Payment Servers' internal XML parsers can properly read the information of the Web Service operation. Characters in the table are displayed in Courier New font.

Many XML Parsers will encode these characters for you. In this case, the characters *will not* be converted back to their proper values by the Payment Server; they will be taken literally. Also, if you pass the encoded character through an input parameter that removes the characters listed in the table below, then certain characters may be removed (see examples below). However, if you are not using a parser, or if the parser does not handle this encoding, then the characters in the table listed below may still be removed, depending on the input parameter for the Web Service operation you are using.

**Table 2. XML Character Removal**

Character	XML Parser Encoding
<	&lt;
>	&gt;
&	&amp;
'	&apos;
"	&quot;

### Example

The following example shows how characters would be removed if the data was passed through the NameOnCard parameter of the ProcessCreditCard operation.

Valid: *John James*

Invalid: *John & James* becomes *John James*

Invalid: *John &amp; James* becomes *John amp; James*