

# PayPoint®

## Merchant Integration

### Guide

Version: 2.1.1  
2/11/2008

First Data Government Solutions, LP  
Suite 300  
11311 Cornell Park Drive  
Cincinnati, Ohio 45242  
[www.fdgs.com](http://www.fdgs.com)

© 2007 First Data Government Solutions, LP. All Rights Reserved. This document contains unpublished, confidential, and proprietary information of First Data Government Solutions, LP. You may not disclose, copy, or use any part of these materials for any purpose without the express written consent of First Data Government Solutions, LP.

# PayPoint® Merchant Integration Guide

## Table of Contents

<b>Disclaimer</b>	<b>i</b>
<b>Revision History</b>	<b>i</b>
<b>1.0 Introduction</b>	<b>1</b>
Purpose	3
Key Assumptions	3
Best Practices	4
<b>2.0 Implementation Approach</b>	<b>5</b>
Overview	5
Pre-Implementation Preparation	6
Phase 1: Account Registration	6
PayPoint® Account	6
Credit Card/Debit Card Payments	7
E-Check Payments	7
PINless Debit Payments	7
PIN-based Debit Payments	7
Phase 2: Application Program Interface Design and Development	8
Web Services API	9
Secure HTTP API	9
Batch API	9
Phase 3: Test Mode	10
Function: MakePayment	10
Function: PaymentStatus	15
Function: Calculate Convenience Fees	17
Function: RegistrationCRD	18
Function: RegistrationInquiry	19
Function: RecurringPaymentCD (Create, Delete)	21
Function: RecurringPaymentInquiry	22
Phase 4: Certification Mode and Training	24
Training	25
Phase 5: Production Mode (Go Live)	25
<b>3.0 Integration and Development</b>	<b>26</b>
Real-Time Payment Interface Options	26
Web Service	26
Web Service Integration	28
Operational Objects	29
Functional Objects	41
Secure HTTP Integration	59
Standard Input Parameters	61
Standard Output Parameters	62
Client Side Usage Example	62
Server Side Usage	62
Reading the API Method Specifications	63
Batch Interface Option	80
Return Code Definitions	81
<b>4.0 Payment Posting File</b>	<b>87</b>
Select Extract File Type	87
Extract File Name Standard	87
Extract File Format Specifications	88
Sequence of Records in Extract File	89
Extract File Structure	90

# PayPoint® Merchant Integration Guide

---

<i>SH = Site Header</i>	90
<i>AH = Application Header</i>	90
<i>AD = Application Detail</i>	90
<i>AF = Application Footer</i>	92
<i>SF = Site Footer</i>	93
<b>5.0 Resources Available</b>	<b>94</b>
<b>6.0 Integration Timeline</b>	<b>95</b>
<b>7.0 Project Organization</b>	<b>96</b>
<b>8.0 Support Services</b>	<b>98</b>
<b>APPENDIX A: PayPoint® Account Application</b>	<b>99</b>
<b>APPENDIX B: CERTIFICATION CHECKLIST</b>	<b>99</b>
<b>APPENDIX C: ACRONYMS AND DEFINITIONS</b>	<b>100</b>

# PayPoint® Merchant Integration Guide

## DISCLAIMER

The material presented in this PayPoint® Integration Guide is for general guidance only. First Data Government Solutions, LP does not represent or warrant that this is the only information available or the only information that should be considered when deciding to implement an electronic payment processing solution. First Data Government Solutions, LP shall not be held liable for any losses caused by reliance on the accuracy, reliability or timeliness of this information. Portions of such information may not be useful or applicable to an entity's particular circumstance. Any person or entity that relies on any information obtained from this Guide does so at his or her own risk.

## REVISION HISTORY

Date	PayPoint® Version	Section	Integration Guide Update Description
	Prior to 1.2		2.0: Second version of PayPoint® Integration Guide, contains significant updates to the Integration Guide based on enhancements incorporated into the product.
	Prior to 1.2		2.1: This release contains additional information on best practices, clarification of functions, and updated tables to better explain the API mechanics. Added Transaction Date and State code to transaction detail record in the export file.
	Prior to 1.2		2.2: This release contains information relating to E-Checks, registration, and recurring payments.
	Prior to 1.2		2.4: Added new Return Code Definition sub-section to Section 3.
	Prior to 1.2		2.5: Modified Make A Payment Return Codes
	1.2		3.0 A variety of changes were made in coordination with PayPoint® 1.2 Release. See the release notes for what's new in the release.
	Prior to 1.5		4.1.1 Modified the Posting file output field in detail record called Card Type Code – Changed from 9(1) to A (1) and identified new card types.
	1.5		5.1.1 A variety of changes were made in coordination with PayPoint® 1.5 Release. See the release notes for what's new in the release. Affected sections include: Section 3.0 Integration and Development – Documentation was updated to include new PIN based debit support and additional returned value on the PaymentStatus.  Section 4.0 Payment Posting File – PaymentMethod and Cardtype

## PayPoint® Merchant Integration Guide

			documentation on the detail record was updated to include new PIN based debit functionality.
4/26/2007	2.0		Changed the versioning to match the current PayPoint® Release Number. Updated <b>EPayAddress</b> Object to include validations for First Name, Last Name, Full Name, Address and City for TeleCheck® and Credit Card processing.
8/15/2007	2.1	1.0 Introduction (Multi-Processor Credit Card Support)	Added support for Vital (TSYS)
8/15/2007	2.1	1.0 Introduction (Features)	Added description about Third Party Payments - ACH Credit, Point of Sale (POS) and Cash.
8/15/2007	2.1	3.0 Integration and Development (Operational Objects)	General Change: Added additional column called Validation Exceptions; Changed Control column title to Standard Validation.
8/15/2007	2.1	3.0 Integration and Development (Operational Objects)	Added support for Third Party Payments - ACH Credit, Point of Sale (POS) and Cash.
8/15/2007	2.1	3.0 Integration and Development (PaymentInfo Object)	Added that PaymentInfoCC Object is used for Cash and POS payments.
8/15/2007	2.1	3.0 Integration and Development (PaymentInfo Object)	Added that PaymentInfoEFT Object is used for ACH Credit payments.
8/15/2007	2.1	3.0 Integration and Development (PaymentInfoCC Object)	Added validation exceptions for Cash and POS.
8/15/2007	2.1	3.0 Integration and Development (PaymentInfoEFT Object)	Added validation exceptions for ACH Credit.
8/15/2007	2.1	3.0 Integration and Development (EPayAddress Object)	Updated validation exceptions for e-Check with TeleCheck® gateway (The second name of the Full Name can accept apostrophe).
8/15/2007	2.1	3.0 Integration and Development (MakePayment Object)	Added validation exceptions for ACH Credit, Cash, and POS including the exception that these payment mediums do not support Convenience Fees.
8/15/2007	2.1	3.0 Integration and Development (CalculateConvenienceFee Object)  (RegistrationCRD Object)  (RegistrationInquiry Object)  (RecurringPaymentCRD Object)	Added validation exceptions as not supported for ACH Credit, Cash, and POS.
8/15/2007	2.1	4.0 Payment Posting File (AD= Application Detail)	Added that third party payments = AD7.

# PayPoint® Merchant Integration Guide

2/11/2008	2.1	3.0 Integration and Development (Using the Page Pop capability)	Updated url to: The HTML Payment Collection Interface is accessed through the <a href="https://www.govone.com/epayadmin/http/pay.aspx">https://www.govone.com/epayadmin/http/pay.aspx</a> page on the PayPoint® website.
2/11/2008	2.1	3.0 Integration and Development (Batch Interface Option)	Updated url to: You can obtain a copy of the latest PayPoint® XML Request and Response XSD (Schema) files at the following URL location <a href="https://www.govone.com/epayadmin/validatebulkfile.aspx">https://www.govone.com/epayadmin/validatebulkfile.aspx</a> .

## PayPoint® Merchant Integration Guide

### 1.0 INTRODUCTION

PayPoint® facilitates electronic commerce by enabling governmental entities to accept credit card, pin-less debit card, and ACH/EFT payments for goods and services. PayPoint® is designed as an enterprise-wide payment system. This provides the flexibility to setup the organizational, administrative, and reporting structure for any size entity. The service can easily scale to handle both small and large-scale transaction volume. Our standard API interface can easily connect a business application to the financial institutions and payment networks that authorize and settle electronic payments. PayPoint's® administrative features and functionality enable consistent management and reporting on electronic payment activity across the enterprise, regardless of electronic payment type, through a common user interface.

Feature	Description
Enterprise Approach	PayPoint® is built to support enterprises with multiple sub-entities that need payment processing capabilities. Our enterprise approach allows financial administrators to delegate online line access reporting and payment history research to any level within the enterprise including the ability to perform reporting across the entire enterprise.
Online Reporting	PayPoint® provides the ability to produce on demand summary as well as detailed reports.
Online Payment Inquiry	PayPoint® provides dynamic payment search options for viewing payment history. Payments can be searched by dates, amounts, last 4 digits of an account, name, status, custom references, and more. Payment Administrators are able to view all details associated with payments including history of activity associated with payments.
Settlement Management	PayPoint® provides a unique ability to monitor settlement activity and perform reconciliation checks through online inquiry and reporting options.
Multi-Processor Credit Card Support	PayPoint® supports most major Credit Card processors including: First Data Nashville (Envoy), First Data North (CardNet), First Data South (Nabanco), Vital (including TSYS), PaymentTech, Concord, Visa USA, ECHO, First TeleCheck®, Nova, and BankServ
Fraud Detection	Fraud detection services for both Credit Card and E-Check. E-Check – Positive/Negative risk rating, Account and Bank Routing Validation, User Transaction Limits, Identity Verification, and duplicate payment detection. Credit Card – Address Verification, CVV2, Transaction Limits, and duplicate payment detection.
Application Programming Interface (API)	PayPoint® provides a set of open standard API interfaces for integrating payment processing into any environment. Our Web Service based API can be utilized by most all platforms. This support allows for multi-channel access through Web, IVR, PC, Kiosk, etc. PayPoint® also offers a Batch Processing API via a standard XML API request and response files for offline payment commands such as issuing payments in bulk.
Multiple Payment Options (Credit Card, E-Check, PINless Debit, PIN based)	Support for E-Check, Credit Card, and PINless Debit Payments. Includes the ability to issue and manage initial

## PayPoint® Merchant Integration Guide

Feature	Description
Debit)	payments, refunds, charge-backs, returns, voids non-sufficient funds, and notice of change.
Registered and Unregistered Payments	PayPoint® registration option allows integrators to register consumer payment information with PayPoint® limiting security risk and repeat consumer payments easy.
Recurring Payments	PayPoint® provides the ability to establish recurring payment schedules. Once establish PayPoint® will execute payments on defined schedule and provide Email confirmation of any authorization issues.
Convenience Fees	PayPoint® can automatically calculate and collect convenience fees on a per payment basis.
Custom Reference Management	Payments include the ability to assign a custom identifier linking the PayPoint® payment to your transactions. For example, if your application is accepting payments for parking tickets, the ticket number could be placed in our Custom Reference. When researching payments this Custom Reference can be used in locating payments.
E-Check Payment Warehousing	Payments can be post dated up to 365 days. PayPoint® manages payment execution based on the payment date provided.
Third Party Payments Reporting	PayPoint® accepts payment data relating to payments authorized, collected and/or settled through a separate system. The Third party payment mediums supported are ACH Credit, Point of Sale (POS), and Cash. This provides an enterprise approach to payment management, customer service and reconciliation. Even though these third party payments are not settled within PayPoint®, they are available to all other services of PayPoint payment management including payment searching, reporting and are exported in the posting file along with other PayPoint® payment transactions.
Secure Processing	PayPoint® is built on a highly secure infrastructure including; End to End encryption of communications, C2 Level Secure Data Centers, Data Security including encrypted storage of sensitive information, and Roles Based security access to payment management and reporting features.®

The PayPoint® Integration Guide describes the phased approach to integrating a business application to our standard Application Program Interface (API). PayPoint® can be integrated to business applications on a variety of platforms including Web sites, e-stores, Interactive Voice Response (IVR) systems, Point of Sale (POS) devices, and Kiosks. The API offers a programmable interface that is platform independent. Batch Processing is also offered. With Batch Processing you have the ability to execute payments and/or any other PayPoint® API calls offline through a batch interface provided with PayPoint®. Integrators can create an XML file with payment instructions to be executed by PayPoint®. The XML file includes the ability to do any of the existing Real-Time API functions including Make Payment, Cancel Payments, Payment Inquiry, Registration CRD, Registration Inquiry, Recurring Payment CRD, and Recurring Payment Inquiry. The XML file must adhere to the standards defined in our PayPoint® Integration Guide.

## PayPoint® Merchant Integration Guide

While PayPoint® can process many types of electronic payments, it is especially useful for processing non face-to-face payment transactions where the merchant and their customer are not in the same physical location. For a government entity, this allows interaction with citizens or businesses through a web/portal application or an IVR system. The business application transmits payment information for the transaction to PayPoint®. PayPoint® secures an authorization, and then returns a response to the business application for further processing. The business application communicates the success or failure of an authorization request to their customer. PayPoint® can support both registered and unregistered payees.

For credit card payments, PayPoint® can perform real-time credit card authorization with all the major processors. PayPoint® can accept payments made with numerous types of credit cards, including Visa®, MasterCard®, Discover®, American Express®, and other specialty cards such as Diners Club. PayPoint® can also accept payments made with debit cards displaying a Visa or MasterCard logo. PayPoint®'s daily batch settlement will process all authorizations processed and accepted during a given day.

For PINless Debit, PayPoint® can perform real-time authorizations through the STAR, PULSE or NYCE networks. PINless debit is an online payment process which requires no settlement processing within PayPoint®.

For PIN based Debit, PayPoint® can perform real-time authorizations. PIN based debit is an online payment process which requires no settlement processing within PayPoint®.

PayPoint® supports electronic check (E-Check) transactions. E-Check allows a government entity to accept payments from personal or business bank accounts online. The business application collects a bank account number and bank routing number from the payee to fund the electronic payment. E-Check authorization services range from checking the validity of the account information to other value added services including identity verification and guaranteed check processing. PayPoint® originates ACH transactions through the government entity's chosen clearing account for all E-Check payments authorized.

The intended audience for this integration guide includes project managers, programmer analysts, and testers. It explains in detail how electronic payments transactions are submitted to PayPoint® for processing.

### Purpose

This Integration Guide describes a **Phased Approach** to integrating electronic payment functionality into a new or existing business application. This guide has a prescribed list of implementation and deployment tasks and activities, best practice information, as well as technical options and detailed programming specifications for easy integration. The detailed programming specifications describe how to create the necessary environment for integrating with PayPoint®.

The intent of this guide is not to recommend how to code a web site or system interface. This guide presents the various technical options offered for integration with PayPoint®.

### Key Assumptions

- The government entity has the ability to build a real-time interface for presentment of payments regardless of the platform.
- The resources performing the development of the integration with PayPoint® have a working knowledge of the business application development (programming) environment.

## PayPoint® Merchant Integration Guide

- The government entity has a bank or financial institution and, if offering credit card payments, a merchant services agreement.

### Best Practices

These Best Practices are a courtesy to our clients. They are based on industry best practices and practical experience. You should find useful the information in the table as you begin to design and develop the interface to PayPoint®. Please be sure to contact your Project Manager or implementation support team for additional suggestions.

Best Practice	Description
Descriptive entity name	Ensure the name selected on the Account Application to be displayed on the payee's statement contains a descriptive name. A descriptive name helps eliminate confusion regarding the entity receiving the payment.
Always use AVS (Address Verification System)	When accepting credit cards, you should incorporate an appropriate level of fraud prevention techniques. AVS is a feature that verifies the cardholder's address and zip code at the time of payment authorization. The issuing bank verifies that the address entered by the cardholder matches the address they have on record. AVS has become an expected minimum level of fraud prevention. Eliminating AVS will nearly always result in higher interchange rates.
Electronic Payment Policies	Educate your customers. Make sure your electronic payment policy is available and accessible by your customer. Do not bury these policies deep within your web site or portal. Be open and up front.
Single Point of contact for Customer Service	If possible, set up a single point of contact. Make sure your customers know how to contact customer service and when customer service is available. Display the phone numbers and email addresses in prominent positions on all web pages. Be available for customers with questions about their payments. If you choose to outsource first line customer service, speak with your Project Manager. We offer a fully staffed customer service center around the clock which can support a fully outsourced operation or after-hour support for your own customer service organization.
Fraud prevention using CVV2 and/or CVC2	If possible, incorporate advanced fraud prevention techniques. The CVV2/CVC2 is a code printed on the back of a Visa or MasterCard credit card. This service helps validate that a genuine card is being used during a non face-to-face payment transaction. PayPoint® can validate the CVV2/CVC2 and return the results to the business application. The benefit of implementing this service is a possible reduction in fraud-related chargebacks.
Disaster Recovery / Business Continuity Planning	We recognize that service disruptions can greatly impact your business. For that reason, we continue to invest in redundant, fault-tolerant systems to minimize service disruptions. Unfortunately, bank processors sometimes experience outages that are entirely out of our control. When these disruptions occur, they can hamper your ability to complete sales in a timely fashion. To ensure a minimum of disruption to your business, we recommend only using processors that offer the same high-level of service availability as PayPoint®.

## 2.0 IMPLEMENTATION APPROACH

### Overview

The following is a summary of the implementation tasks required to ensure a successful PayPoint® implementation. Be sure to review the Implementation task detail section for more information about each task, as well as access to links to additional technical documentation.

**Phase 1: Account Registration.** Set up a PayPoint® account. Request a PayPoint® Application (Account) Identifier (ID) by filling out the PayPoint® Account Application Form. We will provide a unique Application Identifier and your own unique User ID and password. The ID and Password will allow your administrative users to securely access the administrative interface and manage payment transactions for your business applications.

**Phase 2: Application Program Interface Design and Development.** Construct an interface to PayPoint® using the API specifications (e.g. web service, Page POP, etc). After receiving an account, you can complete the design and development of your interface to PayPoint®. The PayPoint® API supports both web form pop integration as well as Web Service integration utilizing SOAP (Simple Object Access Protocol).

**Phase 3: Test Mode.** Test the interface between your business application and PayPoint®. Once you have an active account, the account will be placed in Test Mode. In Test Mode, we will provide you with basic guidance on how to verify that your business application can initiate payment transactions through the API. We encourage our clients to formulate and process as many test scenarios as possible. This phase will demonstrate that the business application can interact successfully with PayPoint®.

**Phase 4: Certification Mode and Training.** We require certification of your business application before activating your PayPoint® account for production use. Upon successful completion of the Phase 3 testing, we place your account into Certification Mode. Certification involves testing a set of payment transactions and validating the results. You will execute the certification transactions associated with the payment methods you have selected (i.e. Credit Card, E-Check, or both). We verify and validate the certification test results before activating your PayPoint® account.

During this phase, your operational personnel are introduced to the features of PayPoint®. Your staff will learn about PayPoint®, including the Administration, Search and Report modules. This training assists your personnel with successfully managing payment transactions.

**Phase 5: Production Mode.** We have specific requirements for activating PayPoint® accounts. It is important to review and understand the certification detailed in the Certification Checklist to ensure that your business application is deployed in a timely manner. Once we have certified your application, we mutually agree upon a schedule for placing the PayPoint® account into Production Mode for live payment processing.

## Pre-Implementation Preparation

The first step in building an interface to PayPoint® is to become familiar with common electronic payment terms, payment processing (external) entities, basic electronic payment functions, and the specific banking and/or merchant services contract.

### *Payment Processing (External) Entities*

A number of external entities may participate in the electronic payment process. They include all organizations that take part in the initiation, authorization, origination, and/or settlement of electronic payment transactions. See the table at right for the common external entities referenced in this guide. You should become familiar these terms in preparation for implementation.

### *Banking/Merchant Services Contract*

The agreement between you and your banking and/or merchant services provider is very important to the proper setup and configuration of PayPoint®. PayPoint® is flexible enough to accommodate most types of agreements. In some cases, the configuration of PayPoint® can have an impact on your banking or merchant services fees.

### *Policy and Procedures for Electronic Payments*

It is important to have well defined electronic payment policies and procedures regarding accepting credit card, debit card, and electronic check transactions over the Internet or from other input channels. These policies and procedures should define the operating guidelines related to electronic payments at the enterprise, work group or agency, and business application level. The Electronic Payment Policy should provide the structure for assigning responsibility and authority for controlling the configuration and usage of PayPoint®.

External Entity
<p><b>Acquiring (Merchant) Bank:</b> An entity must establish a merchant account with an Acquiring Bank, before accepting credit card payments. The merchant account must be configured to process card-not-present transactions.</p>
<p><b>Issuing (Consumer) Banks:</b> An Issuing Bank grants credit to a consumer and provides (issues) a credit card. Some cards may be co-branded, with the Issuing Bank, by another (non-bank) organization.</p>
<p><b>Credit Card Association:</b> Credit card associations manage the interchange between acquiring banks and issuing banks. They also develop industry standards, market their brands, and establish fees for acquiring merchants.</p>
<p><b>Payment (Acquiring) Processor:</b> Payment processors process electronic transactions through the payment network for authorization, clearing, and settlement on behalf of the Acquiring Bank.</p>
<p><b>PayPoint®:</b> Centralized electronic payment engine used by business applications to request authorization and settlement of funds for payments by either credit/debit card or electronic check. It provides a standard interface and consolidated processing environment.</p>
<p><b>PayPoint® Account:</b> An authorized Application ID on PayPoint® for accepting credit cards and/or electronic checks for all types of payments to government entities.</p>

## Phase 1: Account Registration

### PayPoint® Account

The first step in obtaining a PayPoint® account is to complete the “PayPoint® Account Application Form” (See *Appendix A*). We then establish a unique PayPoint® Application (Account) ID. Our PayPoint® project manager will provide you an electronic copy of the form prior to the implementation project. To successfully complete the form, you will need access to banking and/or merchant services provider information. Complete the form and return it to the PayPoint® project manager or Client Relationship Representative. We process the completed form and activate your new account. The sooner the form is completed and verified, the quicker we can activate your account.

It takes approximately 5 – 10 business days to process an application from the time of receipt. Once the application is processed, we issue a unique Application (Account) ID number. This number identifies your

## PayPoint® Merchant Integration Guide

---

account to PayPoint®. PayPoint® can then accept and process payment transactions originating from a business application using the Application ID. The PayPoint® project manager coordinates and facilitates the account registration process and provides all the key values created specifically for your application (e.g. Account ID, Security Key, etc.).

---

### Credit Card/Debit Card Payments

For Credit/Debit Card payments, a Merchant Account is required.

A Merchant Account is necessary to accept online credit card/debit card payments. Your merchant account number is used by PayPoint® to identify all your credit card related electronic payment transactions. A merchant account establishes a relationship with a Merchant Bank (or its Processor) authorized to settle credit card payments through a card association interchange.

The terms and conditions of your Merchant Account are between you and your bank or merchant service provider. You should allow three to four weeks to establish your merchant account if you don't have an existing account. If you plan to use an existing account, you may be required to contact your Merchant Bank to inform them about accepting payments through the PayPoint® payment engine. The Merchant Account must allow Card Not Present transactions.

---

### E-Check Payments

For E-Check payments, a Bank Account is required.

To accept E-Check payments, you are required to establish a bank account with a financial institution before filling out the PayPoint® Account Application form. The bank account number and routing number are used by PayPoint® to process all ACH related electronic payment transactions.

---

### PINless Debit Payments

For PINless Debit payments, a Bank Account is required.

If you're interested in implementing a PINless debit option, please contact our Customer Service.

---

### PIN-based Debit Payments

For PIN-based debit payments, a Bank Account is required.

If you're interested in implementing a PINless debit option, please contact our Customer Service.

---

## PayPoint® Merchant Integration Guide

### Phase 2: Application Program Interface Design and Development

PayPoint® has a standard Application Programming Interface (API), which supports three methods for processing transactions. The three methods are Web Services, Secure HTTP, and Batch. Technical details of each method can be found within this section. All real-time PayPoint® API's interactions are performed using Secure Sockets Layer (SSL) 128-bit encryption.

PayPoint® has a standard set of payment functions, regardless of the type of API integration method selected. The following table describes the payment functions available in this release. The specifications for the inputs and outputs can be found later in this document.

Function	Description
<b>MakePayment</b>	This function initiates a payment transaction. Inputs to this function include Payment Amount, Account Number (Credit Card, Checking), and other information required to authorize a payment. The function responds with a return code identifying the status of the payment and a unique confirmation number associated with the payment transaction in PayPoint®.
<b>PaymentStatus</b>	This function returns the status of an existing payment.
<b>CancelPayment</b>	This function voids or refunds a payment previously authorized by the 'Make Payment' function.
<b>CalculateConvenienceFee</b>	This function calculates a convenience fee for a given payment amount. This function can be used by your business application to present the payer with the convenience fee prior to submitting the payment transaction for authorization.
<b>RegistrationCRD</b>	This function enables any business application to register a payer's financial account (either bank account or credit/debit card). PayPoint® securely stores the financial account information, eliminating the security risk associated with storing sensitive financial data in multiple locations. PayPoint® allows an application to manage account data without needing to store the data. This function provides a unique Registration ID for inquiring on, updating, or deleting the registered account information.
<b>RegistrationInquiry</b>	This function allows a business application to request registration data stored in PayPoint®. For example, a web application may request registration information through this inquiry and present it to the user for confirmation and/or update of their financial account information.
<b>RecurringPaymentCD</b>	This function allows a business application to create a schedule of automatically recurring payment requests. In order to establish a Recurring Payment, the payer must have a registered account. This function allows an application to add (create) or delete a recurring payment schedule.
<b>RecurringPaymentInquiry</b>	This function allows a business application to request recurring payment details (e.g. schedule)

## PayPoint® Merchant Integration Guide

---

### Web Services API

The PayPoint® Web Service API enables a business application to access the electronic payment services using XML and SOAP (Simple Object Access Protocol) standards for sending messages using the Internet.

Business applications can call on a PayPoint® WSDL (Web Service Description Language) document, which presents the functionality exposed by the Web Service. This standard integration method enables existing applications (i.e. Web, IVR, Desktop applications, etc.) to integrate with PayPoint® to manage payment processing. The advantage of this method is that it allows your technical staff (i.e. developers, programmers, etc.) to develop in their preferred programming language including Java, Perl, or Visual Studio.

Your technical resources responsible for PayPoint® integration need a working knowledge of Web Services, SOAP, and XML, as well as an understanding of how to integrate data into an application or Web site. In addition, the technical staff performing integration must have all required software to support web services already installed in a development environment and be proficient in their preferred programming language.

---

### Secure HTTP API

PayPoint® allows access to its web service methods through an alternative interface API known as the “Secure HTTP API.” It uses secure HTTP (HTTPS) to communicate between the business application and PayPoint®.

The Secure HTTP API interface accepts two types of inputs and can return two types of output. The business application may send data to the interface either through the query string or as FORM POST data. Depending on the application request, the API returns its results either via the query string or as the HTTP GET results of the application’s query.

In addition, the Secure HTTP API can perform a “page pop” for collecting payment data. Using this HTML Payment Collection Interface method, the business application uses the query string to pass specific information about the payment transaction and PayPoint® presents (‘pops’) pages prompting the user for input, as well as presenting payment transaction results on a confirmation page. The HTML Payment Collection Interface does allow for minimal customization of the presented pages. For example, parameters allow the calling application to specify style sheets, header and footer text, and page titles.

---

### Batch API

The PayPoint® batch interface allows a business application to initiate and access payment transactions in an off-line mode. The application accumulates a number of payment transaction requests (a batch) in a single file, uploads them to PayPoint® for processing, and then receives a results file.

## PayPoint® Merchant Integration Guide

### Phase 3: Test Mode

During Test Mode, you will test the technical interface between your business application and PayPoint® to ensure you can execute each PayPoint® function. A limited set of scripted test cases is defined in this section. These test cases will ensure the business application can successfully connect to and communicate with the service. It is very important to extensively test the interaction between the business application and PayPoint® before entering Certification mode.

You can begin Phase 3 testing after your PayPoint® Project Manager has assigned a PayPoint® Application ID. Our support staff will provide the necessary information to get started. In test mode, use the test data (e.g. credit card #, expiration date, confirmation #, etc.) for each test case. Do not deviate from the test data found in the table. Unpredictable results may occur if you deviate from the scripted test cases.

These test transactions will simulate the production system. It is intended as quick and easy way to allow your development team to test the system interface to PayPoint® and ensure each function can be initiated and the results processed. All test transactions will appear to be processed as real transactions. Test transactions will not be authorized by your card processor or reach the bank for settlement. PayPoint® will simulate the interaction with a card processor or ODFI. In addition, the test transactions are not stored in PayPoint® and therefore will not be displayed in reports and/or by the administrative screens.

After you successfully verify and validate all test transactions, you may request a change to Certification Mode. The PayPoint® Project Manager or Support Group updates your account status from Test mode to Certification mode. The following information provides details on each test case. If you have any questions, please contact your PayPoint® Project Manager.

#### Function: MakePayment

The Make Payment function is used to initiate a payment. The test cases in the following table test the interface between your business application and PayPoint®. You are required to insert the required data values for each object, including information like Card Number, Expiration Date, Payment Amount and other data required to authorize a payment. For example, the Header Object is a required object for each Make A Payment request. The Header object contains important information such as the Application ID and Security Key, specifically assigned to your application.

This function will provide a return code through the response object member. The Return Code gives the status of the transaction. The ReturnCode value is either a sting or integer based on the type of interface you select. If you deploy a Web Service interface, the return code will be a string value. If you deploy a Web HTTP (Form POP or Query String) interface, the return code will be an integer value. Your PayPoint® Project Manager is prepared to answer any questions you may have.

To successfully execute these test cases through PayPoint®, the application's PayPoint® account set up must be complete.

## TEST CASES: MAKEPAYMENT

Test #	Description	
	Field Name	Field Value
1	Process a credit card transaction with no errors.	
	CardNumber	1890886512515590
	ExpirationDate	1205
	Web Service ReturnCode	Payment_Success
	HTTP Return Code	2
	ResultMessage	<blank>
	Total Amount	<amount requested>
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	Date/Time when the payment will be submitted for settlement.
	Card Type	TEST
2	Process a credit card transaction that generates a technical difficulty error message.	
	CardNumber	1890886512515591
	ExpirationDate	1205
	Web Service ReturnCode	Error
	HTTP Return Code	4
	Result Message	Error in Payment Gateway, unable to process request.
	Total Amount	0
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	<blank>
	Card Type	TEST
3	Process a credit card transaction that generates a Communication error.	
	CardNumber	1890886512515592
	ExpirationDate	1205
	Web Service ReturnCode	Communication_Error
	HTTP Return Code	7
	Result Message	Error in Payment Gateway, unable to process request.
	Total Amount	0
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	<blank>
	Card Type	TEST
4	Process a credit card transaction that generates an authorization declined response.	
	CardNumber	1890886512515593

## PayPoint® Merchant Integration Guide

Test #	Description	
	Field Name	Field Value
	ExpirationDate	1205
	Web Service ReturnCode	Declined
	HTTP Return Code	5
	Result Message	Lost/Stolen Card
	Total Amount	0
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	<blank>
	Card Type	TEST
5	Process a credit card transaction that generates an authentication fails error message – this message is for those customers using AVS, CVV2, or other fraud detection schemes.	
	CardNumber	1890886512515594
	ExpirationDate	1205
	Web Service ReturnCode	Verification_Failed
	HTTP Return Code	6
	Result Message	Verification Failed
	Total Amount	0
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	<blank>
	Card Type	TEST
6	Process a credit card transaction that generates an invalid card type error message.	
	CardNumber	1890886512515595
	ExpirationDate	1205
	Web Service ReturnCode	Unaccepted_Card_Type
	HTTP Return Code	13
	Result Message	This application does not allow payments to be made using your type of card. Please select another card and retry.
	Total Amount	123.45
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	<blank>
	Card Type	TEST
7	Process an E-Check transaction that generates successful E-Check payment.	
	Account Number	8745720
	Routing Number	123123123
	Web Service ReturnCode	Payment_Success
	HTTP Return Code	2
	Result Message	<blank>
	Total Amount	<amount requested>

## PayPoint® Merchant Integration Guide

Test #	Description	
	Field Name	Field Value
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	Date/Time when the payment will be submitted for settlement.
8	Process an E-Check transaction that generates a technical difficulty error message.	
	Account Number	8745721
	Routing Number	123123123
	Web Service ReturnCode	Error
	HTTP Return Code	4
	Result Message	Unable to process Payment Gateway request: (TEST MODE) Error in Payment Gateway while processing the requested transaction.
	Total Amount	0
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	<blank>
9	Process an E-Check transaction that generates a communications error message.	
	Account Number	8745723
	Routing Number	123123123
	Web Service ReturnCode	Communication_Error
	HTTP Return Code	7
	Result Message	Communication Error in Payment Gateway
	Total Amount	0
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	<blank>
10	Process an E-Check transaction that generates a Declined payment	
	Account Number	8745724
	Routing Number	123123123
	Web Service ReturnCode	Declined
	HTTP Return Code	5
	Result Message	We are sorry that we cannot accept your check at this time. Our decision is based, in whole or in part, on information provided to us by TeleCheck®.
	Total Amount	0
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	<blank>
11	Process an E-Check Transaction that generates a verification failure.	
	Account Number	8745725
	Routing Number	123123123
	Web Service ReturnCode	Verification_Failed
	HTTP Return Code	6

## PayPoint® Merchant Integration Guide

Test #	Description	
	Field Name	Field Value
	Result Message	Customer information not valid
	Total Amount	0
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	<blank>
12	Process a PIN-less Debit transaction that generates successful E-Check payment.	
	Account Number	1890886512515590
	Expiration Date	<Any Valid future Month & Year>
	Web Service ReturnCode	Payment_Success
	HTTP Return Code	2
	Result Message	<blank>
	Total Amount	<amount requested>
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	Date/Time when the payment will be submitted for settlement.
13	Process a PIN-less Debit transaction that generates a communications error message.	
	Account Number	1890886512515592
	Expiration Date	<Any Valid future Month & Year>
	Web Service ReturnCode	Communication_Error
	HTTP Return Code	7
	Result Message	Error in Payment Gateway, unable to process request
	Total Amount	0
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	<blank>
14	Process a PINless Debit transaction that generates a Declined payment	
	Account Number	1890886512515593
	Expiration Date	<Any Valid future Month & Year>
	Web Service ReturnCode	Declined
	HTTP Return Code	5
	Result Message	Declined
	Total Amount	0
	Confirmation Number	<confirmation num>
	SettlementSubmissionDate	<blank>

## PayPoint® Merchant Integration Guide

### Function: CancelPayment

The Cancel Payment function is used to cancel a payment. The test cases in the following table test the interface between your business application and PayPoint®.

This function will provide a return code through the response object member. The Return Code gives the status of the transaction. The ReturnCode value is either a string or integer based on the type of interface you select. If you deploy a Web Service interface, the return code will be a string value. If you deploy a Web HTTP (Form POP or Query String) interface, the return code will be an integer value. Your PayPoint® Project Manager is prepared to answer any questions you may have.

To successfully execute these test cases through PayPoint®, the application's PayPoint® account set up must be complete.

### TEST CASES: CANCELPAYMENT

Test #	Description	Field Name	Field Value
1	Process a credit card transaction with no errors.		
	CardNumber	Any 14 Digit Number	
	Convenience Fee Refund Amount	245.00	
	Web Service ReturnCode	Cancel_Success	
	HTTP Return Code	3	
	ResultMessage	<blank>	
	Total Amount	245	
	Confirmation Number	<confirmation num>	
2	Process a credit card transaction that generates an error.		
	CardNumber	Any 14 Digit Number	
	Convenience Fee Refund Amount	Any amount other than 245.00	
	Web Service ReturnCode	Error	
	HTTP Return Code	4	
	Result Message	Unable to process PayPoint® request: Unable to cancel the payment. The requested refund Amount 'xxx.xx' does not match the transaction amount '245.0'. Partial refunds are not allowed on payments which have not been settled.	

### Function: PaymentStatus

PayPoint® will return the status of an existing transaction in PayPoint® when you execute the Payment Status function. For every transaction stored in the system there is a unique confirmation number associated with it. The confirmation number is the primary data key in the request object to search for a transaction in the database and return a status. PayPoint® will always issue a confirmation number that can be used to reference a specific transaction in the database. It is very important that you store and manage this number carefully in your business application for future use.

## TEST CASES: PAYMENTSTATUS

Test #	Description	
	Field Name	Field Value
1	Retrieve status of a <i>successful payment</i> transaction.	
	Confirmation Number	00000000000008
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
	Payment Status	Payment_Success
2	Retrieve status of a <i>successful cancellation</i> transaction.	
	Confirmation Number	00000000000009
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
	Payment Status	Cancel_Success
3	Retrieve status of a <i>settled payment</i> transaction.	
	Confirmation Number	00000000000010
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
	Payment Status	Settled
4	Retrieve status of a <i>declined payment</i> transaction.	
	Confirmation Number	00000000000011
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
	Payment Status	Declined
5	Retrieve status of an <i>erroneous payment</i> transaction.	
	Confirmation Number	00000000000012
	Web Service <i>ReturnCode</i>	Success



Test #	Description	
	Field Name	Field Value
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
	Payment Status	Error
6	Retrieve status of an invalid confirmation number.	
	Confirmation Number	11111111111113
	Web Service <i>ReturnCode</i>	Undefined_item
	HTTP <i>ReturnCode</i>	18
	Result Message	<i>Unable to process payment status request, invalid confirmation number.</i>
	Payment Status	Unknown

Function: Calculate Convenience Fees

The calculate convenience fee function can be used to compute a convenience fee based on a particular dollar amount. The calculation is based on the rules you provided us in on the PayPoint® Account Application Form. You pass us the payment amount in decimal format and we return a convenience fee.

No test numbers are needed for testing the calculate convenience fee function. You can pass any amount and the system will automatically respond with the appropriate convenience fee amount. You should test to ensure the convenience fee is calculated correctly based on the payment amount sent in the Request object.

TEST CASES: CALCULATE CONVENIENCE FEE

Test #	Description	
	Field Name	Field Value
1	If you activate convenience fees, First Data Government Solutions will configure your application ID to accept this fee based on the convenience fee information/schedule you pass to us on the PayPoint® Account Application Form.	
	PaymentAmount	<Decimal>
	Web Service <i>ReturnCode</i>	Unknown, Success, Error, Undefined_item
	HTTP <i>ReturnCode</i>	0,1,4,18
	Result Message	<blank>
	ConvenienceFee	<Decimal>

## PayPoint® Merchant Integration Guide

Function: RegistrationCRD

The RegistrationCRD function enables your business application to register payers on the PayPoint® system. PayPoint® will store the registration information and allow you to manage the registration data sent to the system. This process will centralize the registration information for the enterprise and off-load the security risk associated with storing sensitive account holder information in multiple locations. This function will allow you to create, update, or delete a registration ID. No test numbers are needed to create a registration number. To test the *update* and *delete* functions, you are required to pass the Registration ID defined in the table. For the delete function, a successful delete will return a <blank> Registration ID in the response object.

### TEST CASES: REGISTRATIONCRD

Test #	Description	
	Field Name	Field Value
1	Perform a 'create' new Register ID within PayPoint®.	
	Action	Create
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
	Register ID	439
2	Perform an 'update' function against an existing Registration ID.	
	Action	Update
	RegisterID	439
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
3	Perform a 'delete' function against an existing Registration ID.	
	Action	Delete
	RegisterID	439
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
4	Perform an 'update' or a 'delete' function against an invalid Registration ID.	
	Action	Update or Delete
	RegisterID	587
	Web Service <i>ReturnCode</i>	Undefined_Item

## PayPoint® Merchant Integration Guide

	HTTP <i>ReturnCode</i>	18
	Result Message	Unable to process Registration request, invalid Registration ID.

### Function: RegistrationInquiry

The RegistrationInquiry function will allow your business application to request registration data for a specific Registration ID. For example, your web application may request registration information through this inquiry and present it to the user for confirmation and/or update. The test cases will let you test retrieving Credit card information, e-check information, and an unidentified item. The registration ID is the key value in the request object. Please pay close attention to this value. A Registration ID of less than 500 will provide credit card information in the response object. A registration ID of greater than 500 will provide E-check information in the response object and a registration ID equal to 500 will provide an error message. All registration ID must be greater than zero. The error message is 'ID not found' in the database.

### TEST CASES: REGISTRATIONINQUIRY

Test #	Description	
	Field Name	Field Value
1	Perform a registration inquiry on a registration ID setup for credit card payments.	
	Register ID	475
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
	Card Number	5590 (Last 4)
	Expiration Date	12 + current year
	Card Status Flag	Not_Present
	Card Type	TEST
	Purchase ID	<blank>
	User IP Address	<blank>
	<b>Address Information</b>	
	Name Full	John W Smith
	Email	<a href="mailto:jwsmith@epay.com">jwsmith@epay.com</a>
	Phone 1	5134899599
	Phone 2	5134896521
	Street 1	11311 Cornell Park Drive
	Street 2	Suite 300
	City	Cincinnati
	State	OH
	Zip	45242

Test #	Description	
	Field Name	Field Value
2	Perform a registration inquiry on a registration ID setup for EFT payments.	
	RegisterID	525
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
	Account Holder Type	Consumer
	Account Type	Checking
	Bank Account Number	8744 (Last 4)
	Bank Routing Number	798562148
	Bank Name	BankOne
	Bank State	OH
	Driver's License Num	1589 (Last 4)
	Driver's License State	KY
	AuthorizationMedium	Web
	SSN	2145 (Last 4)
	<b>Billing Address Information</b>	
	Name Full	John W Smith
	Email	<a href="mailto:jwsmith@epay.com">jwsmith@epay.com</a>
	Phone 1	5134899599
	Phone 2	513489652
	Street 1	11311 Cornell Park Drive
	Street 2	Suite 300
	City	Cincinnati
	State	OH
	Zip	45242
	<b>Shipping Address Information</b>	
	Name Full	John W Smith
	Email	<a href="mailto:jwsmith@epay.com">jwsmith@epay.com</a>
	Phone 1	8597812569
	Phone 2	<blank>
	Street 1	123 My Street
	Street 2	<blank>
	City	Florence
	State	KY
	Zip	41079

## PayPoint® Merchant Integration Guide

Test #	Description	
	Field Name	Field Value
3	Perform a registration inquiry on an invalid registration ID.	
	RegisterID	500
	Web Service <i>ReturnCode</i>	Undefined_Item
	HTTP <i>ReturnCode</i>	18
	Result Message	Unable to process Registration inquiry, invalid Registration ID.

## Function: RecurringPaymentCD (Create, Delete)

The RecurringPaymentCD function will allow your application to initiate a payment request that will occur automatically based on a schedule. You define the payment schedule within PayPoint®. In order to establish a Recurring Payment you must have first created a registration ID associated with the recurring payment transaction. This function will allow you to delete and recreate a recurring payment schedule. You must use Registration ID 345 to create a new recurring payment ID. Use Recurring ID 123 to delete a recurring payment schedule.

## TEST CASES: RECURRINGPAYMENTCD

Test #	Description	
	Field Name	Field Value
1	Perform a 'create' new Recurring Payment using a valid Registration ID.	
	Action	Create
	RegistrationID	475 or 525
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
	RecurringID	415 or 416
2	Perform a 'create' new Recurring Payment with an invalid Registration ID.	
	Action	Create
	RegistrationID	589
	Web Service <i>ReturnCode</i>	Undefined_Item
	HTTP <i>ReturnCode</i>	18
	Result Message	Unable to create recurring payment request: invalid Registration ID.
	RecurringID	<blank>
3	Perform a 'delete' function against an existing Recurring Payment ID.	

## PayPoint® Merchant Integration Guide

Test #	Description	
	Field Name	Field Value
	Action	Delete
	RecurringID	415 or 416
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
4	Perform a 'delete' function on an invalid Recurring Payment ID.	
	Action	Delete
	RecurringID	874
	Web Service <i>ReturnCode</i>	Undefined_Item
	HTTP <i>ReturnCode</i>	18
	Result Message	Unable to create recurring payment request: invalid Recurring Payment ID.

### Function: RecurringPaymentInquiry

The RecurringPaymentInquiry function will allow your business application to request payment schedule data for a specific RegistrationID. For example, your web application may request a payment schedule through this inquiry and present the data to the user for confirmation. The recurring payment ID is the key value in the request object. The response object will return the same schedule information found in the table for any recurring ID of less than or greater than 500. All recurring IDs must be greater than zero. A recurring ID of 500 will give you an undefined\_item error message. This error message indicates the recurring ID sent in the request object is not found in the database.

## TEST CASES: RECURRINGPAYMENTINQUIRY

Test #	Description	
	Field Name	Field Value
1	Perform a Recurring Payment Inquiry on a valid Recurring Payment ID.	
	RecurringID	415 or 416
	Web Service <i>ReturnCode</i>	Success
	HTTP <i>ReturnCode</i>	1
	Result Message	<blank>
	Begin Date	Today – 30 days
	End Date	Begin Date + 1 year
	Next Payment Date	Today + 15 days
	Registration ID	875
	Interval Param 1	2
	Interval Param 2	<blank>
	Interval Param 3	<blank>
	Interval Parm 4	<blank>
	Payment Amount	175.00
	Recurring Interval Type	Monthly
	Registration ID	475 or 525
2	Perform a Recurring Payment Inquiry on an invalid Recurring Payment ID.	
	RecurringID	500
	Web Service <i>ReturnCode</i>	Undefined_item
	HTTP <i>ReturnCode</i>	18
	Result Message	Unable to process Recurring Payment inquiry, invalid Recurring Payment ID.

#### Phase 4: Certification Mode and Training

Before activating a PayPoint® Account, we must certify the business application. Credit/Debit Cards and E-Check may be certified independently. After successfully completing the Testing Mode phase, you must request that your PayPoint® Account be moved from Testing Mode to Certification Mode. We will update the account status allowing you to send transactions to the production server to be processed live. We will provide a list of transactions for you to perform to certify your application. All transactions processed in Certification Mode are restricted by PayPoint® to a payment amount of \$1.00. We validate the actual test results against the expected results. All of the certification test results *must* pass. If there are any discrepancies between the expected and actual test results, PayPoint® Account will be returned to Testing Mode. Once the discrepancies are resolved, we will change the account status back into Certification Mode.

The certification process will accomplish the following:

- 1) Test and verify that the business application has implemented each PayPoint® function properly,
- 2) Test and verify that your network/Internet connection is working,
- 3) Test and verify that your configuration settings are correct,
- 4) Test and verify that your reporting is set up correctly,
- 5) Test and verify that payment transactions are credited to the correct account.

At the completion of the Certification Tests, complete the Certification Checklist and forward it to the PayPoint® Project Manager. The Certification Checklist requires that you verify proper execution of your application source code, proper response and error handling, and proper processing of return codes. We will activate the PayPoint® Account for production use, at a time scheduled with you, after successful certification.

## PayPoint® Merchant Integration Guide

### Training

Training plays a critical role in knowledge transfer. Training your staff is essential for effective management, administration, maintenance, and support of your program. During training, your personnel will be introduced to the operations of our PayPoint®, including the Administration, Security, and Report modules. We will facilitate the training program with the purpose of developing skills necessary for the successful operation of PayPoint®.

The PayPoint® Training curriculum includes the following subjects:

Subject	Role	Delivery Options
PayPoint® Introduction and Overview	PayPoint® Functional End User/ PayPoint® Administrator	Training Guide and documentation, onsite classroom training, or remote Web-based training using Live Meeting Tutorial.
PayPoint® Security	PayPoint® Functional End User/ PayPoint® Administrator	Training Guide and documentation, onsite classroom training, or remote Web-based training using Live Meeting Tutorial.
PayPoint® Reporting Tool	PayPoint® Functional End User/ PayPoint® Administrator	Training Guide and documentation, onsite classroom training, or remote Web-based training using Live Meeting Tutorial.
PayPoint® Administration Tool	PayPoint® Administrator	Training Guide and documentation, onsite classroom training, or remote Web-based training using Live Meeting Tutorial.

Our support staff will assist with resolving specific technical issues with site integration, error codes and features of the various processing methods. We can provide additional training and development support upon request.

### Phase 5: Production Mode (Go Live)

This phase completes the transition from Certification mode to Production mode. Production mode allows an agency business application to begin accepting real payment transactions. *(Please note that your PayPoint® Account must be in Production Mode before you begin to accept payments).* After successfully completing certification testing, request that the PayPoint® Project Manager update your PayPoint® Account status from Certification mode to Production mode.

Make this request by sending an email or fax to the PayPoint® Project Manager or Client Relationship Representative stating your desire to transition your site to Production Mode. Include the target date for the transition and the Certification Checklist for verification. We will validate the checklist and contact you with the results. The results will include a scheduled production date.

Once the account has been enabled for production transaction processing, our Support Group will send an email confirming the successful transition to Production Mode. At this point, all transactions will be processed as live transactions resulting in a transfer of funds.

### 3.0 INTEGRATION AND DEVELOPMENT

We offer several interface alternatives, including Web Services, Secure HTTP, and Batch. The best alternative is driven by the capabilities of the business application and whether or not a real-time and/or off-line (batch) interface is required. Regardless of the type of API integration method you select, the same set of functions is available to your application. Our common interface approach makes it easy to integrate electronic payment processing into any business applications operating on a variety of platforms. PayPoint® offers the following integration methods:

- Real-Time API
  - Web Services
  - Secure HTTP
- Batch API

You need to determine how PayPoint® can best be integrated into your business application environment. Consider the type of input channels you have available today and your goals for electronic payment processing. For example, if you have existing Web, IVR, and Kiosk/POS business applications from which customers can purchase products or services online, then the Web Services interface may be the best alternative. If you have new or existing Web applications, then you can consider either the Web Services or the Secure HTTP integration methods. The alternative selected is also dependent on whether your business application has access to web services and the skills of your technical staff. Consider these things before you make a decision. Also, consult with your PayPoint® Project Manager for a recommendation based on your specific operating environment.

A complete listing of the Web Services Description Language (WSDL) document for the Web Services API is accessible from the following URL

<https://www.govone.com/epay/epaywebservicesdl.aspx>

#### Real-Time Payment Interface Options

The Real-time interface allows you to integrate electronic payment functionality, such as Credit Card, PINless debit, PIN based and E-Check acceptance, into your business application. Both real-time processing alternatives use a secure Web interface (HTTPS) with 128-bit encryption. The choice of integration method is independent for each business application. A technical evaluation of each business application should help determine which method is best in each situation.

#### Web Service

Web Services are based on a programming concept that makes distributed computing simple and effective. With complementary technologies of XML and SOAP, Web Services allow program components to be distributed across a local network and/or remotely over the Internet. Web Services are applications whose logic and functions are accessible and reusable using standard Internet protocols and XML data formats. PayPoint® is designed with Web Services as the underlying technology to support server-to-server real-time capabilities.

A Web Service interface is defined in terms of the XML messages that the service accepts and generates as an acknowledgement. Applications using Web Services can be implemented on any platform in any programming language, as long as they can create and process the messages through this interface. PayPoint® uses Simple Object Access Protocol (SOAP) as the solution for initiating Remote Procedure Call (RPC) requests. SOAP is a lightweight protocol intended for exchanging structured information in a distributed environment. It uses XML technologies to define an extensible messaging framework that provides a message construct that can be exchanged over a variety of underlying protocols. The

## PayPoint® Merchant Integration Guide

framework has been designed to be independent of any particular programming language, platform, and other technical criteria.

The Web Service API uses a SOAP interface over a Secure Hyper Text Transfer Protocol (HTTPS) connection to exchange information with the service. A Web Services Description Language (WSDL) document is provided describing all supported request/response objects. This document is accessible on the Internet. Ask your PayPoint® Project Manager for the URL to gain access this information. To expedite the integration of the PayPoint® service, application programmers may wish to use the information available on the Internet to assist in the development of a SOAP request (see table below). Note: We do not control these URLs, so some may have changed since we published this document.

ToolKit / Information	Location
Java Apache AXIS	<a href="http://xml.apache.org/axis">http://xml.apache.org/axis</a>
Python Web Services	<a href="http://Pywebsvcs.sourceforge.net">http://Pywebsvcs.sourceforge.net</a>
Perl SOAP	<a href="http://www.soaplite.com">http://www.soaplite.com</a>
PHP NuSOAP	<a href="http://www.sourceforge.net/projects/nusoap/">http://www.sourceforge.net/projects/nusoap/</a>
XML	<a href="http://www.xml.org/">http://www.xml.org/</a>
COM Microsoft SOAP	<a href="http://msdn.microsoft.com/code/default.asp">http://msdn.microsoft.com/code/default.asp</a>
NET .NET Framework SDK	<a href="http://msdn.microsoft.com/downloads/default.asp?URL=/code/sample.asp?url=/MSDN-FILES/027/000/976/msdncompositedoc">http://msdn.microsoft.com/downloads/default.asp?URL=/code/sample.asp?url=/MSDN-FILES/027/000/976/msdncompositedoc</a>
C++	<a href="http://www.sqldata.com/soapclient/soapclient30.htm">http://www.sqldata.com/soapclient/soapclient30.htm</a>

Additional SOAP resources to assist with development are described below..

Soap Resource	Location
Developmentor	<a href="http://www.develop.com/">http://www.develop.com/</a>
SoapWare.org	<a href="http://www.soapware.org/">http://www.soapware.org/</a>
SoapLite.com	<a href="http://www.soaplite.com/">http://www.soaplite.com/</a>
W3Schools tutorial	<a href="http://www.w3schools.com/soap/default.asp">http://www.w3schools.com/soap/default.asp</a>

## Web Service Integration

The PayPoint® API provides functions to make a payment, calculate convenience fees, cancel payments, get a payment's status, register accounts, and create and delete recurring payment schedules. For each function available through the API, there is a **request object** and a corresponding **response object**.

Interacting with the PayPoint® Web Service is as simple as populating and submitting the request object, and then examining the response object for the results.

The typical payment authorization process is described below:

1. A consumer selects a credit card or e-check service as the payment option from the agency application's checkout page,
2. The agency application formats an authorization request based on the data provided by the consumer by setting fields in the MakePayment request object.
3. The agency application passes the transaction data to PayPoint® by invoking the SOAP request and blocking (waiting) for a response,
4. PayPoint® requests authorization for the payment and returns the results of the request to the agency application,
5. The agency application evaluates the response and processes the results based on its business rules. Typically, the agency application continues with a confirmation to the consumer on a successful authorization. If an authorization failure occurs, the application may ask the consumer to correct the data provided, terminate the transaction, or prompt the consumer to select another payment method.

The SOAP request for service contains both Operational Objects and Functional Objects. Both categories of objects are defined below.

The address of PayPoint's Web Service is

<https://www.govone.com/epay/epaywebservice.asmx>

The a WSDL can be found at the following Address

<https://www.govone.com/epay/epaywebservicesdl.aspx>

# PayPoint® Merchant Integration Guide

## Operational Objects

The request and response object for each of the functions available within the PayPoint® Web Service are described below. The primary purpose of these objects is to group together related data in a structure to be easily incorporated into a PayPoint® request.

PayPoint® processes, authorizes, and settles payments within the gateway using the following payment mediums:

- Business Check
- Commercial Credit Card
- Credit Card
- e-Check
- PINless Debit
- PIN-based Debit

The operational objects described below, unless listed as a validation exception, apply to these six payment mediums.

There are additional payment mediums that can be stored in PayPoint that are made outside of PayPoint® by third party processes or gateways. These third party type payment mediums are:

- ACH Credit
- Cash
- Point of Sale (POS).

Since these three payment mediums are not processed, authorized and settled within PayPoint®, some of the objects and members do not apply to these third party payments. In these cases, these exceptions are also listed as a validation exception.

If there are no validation exceptions described for a member, this applies to all the payment mediums.

### HEADER OBJECT:

The Header Object will be present in every request for each of the available functions. The following members make up the header object and all are required items.

Member	Description	Standard Validation	Validation Exception				
<u>ApplicationID</u>	The Application Identifier assigned to the PayPoint® Account during Account Registration.	Required (String)	None				
<u>PaymentChannel</u>	<p>Payment Channel describes the means through which the payment was made.</p> <p>Note: If the payment data is collected via an operator who in turn keys in the data via a web page, the AuthorizationMedium would be "Voice".</p> <p>Valid Payment Channels values are:</p> <table border="1" data-bbox="435 1797 997 1875"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Web</td> <td>1</td> </tr> </tbody> </table>	String Value	Numeric Value	Web	1	Required (String)	None
String Value	Numeric Value						
Web	1						



	<table border="1"> <tr><td>IVR</td><td>2</td></tr> <tr><td>Walkin</td><td>3</td></tr> <tr><td>Voice</td><td>4</td></tr> <tr><td>FAX</td><td>5</td></tr> <tr><td>Mail</td><td>6</td></tr> </table>	IVR	2	Walkin	3	Voice	4	FAX	5	Mail	6		
IVR	2												
Walkin	3												
Voice	4												
FAX	5												
Mail	6												
SecurityKey	Unique password token assigned to the PayPoint® Account. An initial token is assigned during Account Registration. The password token can only be obtained or updated through the customer support center.	Required (String)	None										

PAYMENTINFO OBJECT:

The PaymentInfo object indicates to PayPoint® whether you are requesting a credit card or EFT payment service. The credit card and EFT information are mutually exclusive members in this object.

Member	Description	Standard Validation	Validation Exception																				
<u>PaymentInfoCC</u>	A reference to the PaymentInfoCC object (see definition below)	Required for Credit Card Commercial Credit Card PINless Debit, PIN-based Debit POS Cash	Not Used for: e-Check, Business Check ACH Credit																				
<u>PaymentInfoEFT</u>	A reference to the PaymentInfoEFT object (see definition below)	Required for e-Check Business Check ACH Credit	Not Used for Credit Card Commercial Credit Card PINless Debit PIN-based Debit POS Cash																				
<u>PaymentMedium</u>	Indicates the Payment Medium used for this transaction. <table border="1"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr><td>e-Check</td><td>1</td></tr> <tr><td>CreditCard</td><td>2</td></tr> <tr><td>CommercialCreditCard</td><td>3</td></tr> <tr><td>BusinessCheck</td><td>5</td></tr> <tr><td>PinlessDebit</td><td>6</td></tr> <tr><td>PINDebit</td><td>7</td></tr> <tr><td>POS</td><td>8</td></tr> <tr><td>Cash</td><td>9</td></tr> <tr><td>ACHCredit</td><td>10</td></tr> </tbody> </table>	String Value	Numeric Value	e-Check	1	CreditCard	2	CommercialCreditCard	3	BusinessCheck	5	PinlessDebit	6	PINDebit	7	POS	8	Cash	9	ACHCredit	10	Required	None
String Value	Numeric Value																						
e-Check	1																						
CreditCard	2																						
CommercialCreditCard	3																						
BusinessCheck	5																						
PinlessDebit	6																						
PINDebit	7																						
POS	8																						
Cash	9																						
ACHCredit	10																						



Member	Description	Standard Validation	Validation Exception
	<b>Note: The difference between e-Check and BusinessCheck is e-Check is for Personal Consumer Accounts and BusinessCheck is for business/corporate accounts. You should ask users which type of account they are making the payment toward. This ultimately affects the standard entry class sent through the ACH network. E-Check sends a standard entry class of WEB and the BusinessCheck sends a standard entry class of CCD.</b>		

PAYMENTINFOCC OBJECT

This object provides details associated with all card-based payments including Credit Card, PINless Debit and PIN based debit. When processing credit card transactions, providing additional data about the cardholder (including address information) and performing verifications checks, such as address verification and CVV2 verification may result in lower interchange fees and chargeback rates. Consult with your Merchant Bank to determine the optimal authorization process for your application. When processing a PINless debit card many times your application will want to ask the user to provide their credit card information and then determine if its eligible for PINless debit. If eligible may want to ask the user if they wish to processes this transactions as a “Debit” or “Credit Card” transaction. The value of the PaymentMedium controls whether the transaction is authorized against the Credit Card or Pinless Debit Networks configured for your application.

The PaymentInfoCC Object is used for the following payment mediums:

- Credit Card
- Commercial Credit Card
- Point of Sale (POS)
- Cash
- PINless Debit
- PIN-based Debit

Member	Description	Standard Validation	Validation Exceptions
BillingAddress	Billing Address of the card holder. See ePayAddress object	Optional	Required for applications boarded with: <ul style="list-style-type: none"> <li>• Fraud and Risk checking including Address Verification (AVS) for Credit Card and Commercial Credit Card</li> </ul>



Member	Description	Standard Validation	Validation Exceptions																														
			<ul style="list-style-type: none"> <li>Email Address for consumer notification support.</li> </ul>																														
ShippingAddress	Shipping Address of the card holder. This information is only utilized in the storage of registration data and is not utilized as a part of a make a payment request. It is used for storage of full address information if you are utilizing the registration. See ePayAddress object	Optional	None																														
CardNumber	Credit/Debit Card number.	Required (String)	Not Used for Cash and Optional for POS Payments																														
CardStatusFlag	<p>Provides details on how the card number is collected. (String) NOTE: This field is not used when performing RegistrationCRD or RegistrationInquiry functions. Valid CardStatusFlag Values are:</p> <table border="1"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Not_Present</td> <td>1</td> </tr> <tr> <td>Present</td> <td>2</td> </tr> <tr> <td>Swiped</td> <td>3</td> </tr> </tbody> </table>	String Value	Numeric Value	Not_Present	1	Present	2	Swiped	3	Required. (PIN-based debit method must be "Swiped")	Not Used for Cash and Optional for POS Payments																						
String Value	Numeric Value																																
Not_Present	1																																
Present	2																																
Swiped	3																																
CardType	<p>Type of Credit Card. Valid values CardType values are:</p> <table border="1"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Test</td> <td>1</td> </tr> <tr> <td>VISA</td> <td>2</td> </tr> <tr> <td>MC</td> <td>3</td> </tr> <tr> <td>AMEX</td> <td>4</td> </tr> <tr> <td>DISC</td> <td>5</td> </tr> <tr> <td>DCCB</td> <td>6</td> </tr> <tr> <td>CBLN</td> <td>7</td> </tr> <tr> <td>JAL</td> <td>8</td> </tr> <tr> <td>JCB</td> <td>9</td> </tr> <tr> <td>ENRT</td> <td>10</td> </tr> <tr> <td>STAR</td> <td>12</td> </tr> <tr> <td>Pulse</td> <td>13</td> </tr> <tr> <td>NYCE</td> <td>14</td> </tr> <tr> <td>PIN Based Debit Card</td> <td>15</td> </tr> </tbody> </table>	String Value	Numeric Value	Test	1	VISA	2	MC	3	AMEX	4	DISC	5	DCCB	6	CBLN	7	JAL	8	JCB	9	ENRT	10	STAR	12	Pulse	13	NYCE	14	PIN Based Debit Card	15	Optional	Not Used for Cash and Optional for POS Payments
String Value	Numeric Value																																
Test	1																																
VISA	2																																
MC	3																																
AMEX	4																																
DISC	5																																
DCCB	6																																
CBLN	7																																
JAL	8																																
JCB	9																																
ENRT	10																																
STAR	12																																
Pulse	13																																
NYCE	14																																
PIN Based Debit Card	15																																
CVV2	The CVV2 field for Visa (CVC2 for MasterCard and CID for American Express). The 3 or 4 digit verification identifier that is typically found on the back of credit cards. This is required	Optional (String)	Not Used for Cash and Optional for POS Payments																														

## PayPoint® Merchant Integration Guide

Member	Description	Standard Validation	Validation Exceptions
	only if set that way in the application parameter set. NOTE: This field is not used when performing RegistrationCRD or RegistrationInquiry functions.		
ECommerceGoodsFlag	Indicates whether or not the transaction falls under the Ecommerce Goods specification, according to the credit card associations. NOTE: This field is not used when performing RegistrationCRD or RegistrationInquiry functions.	Optional ( <i>Boolean</i> ) Default is "False"	None
ExpirationDate	Expiration Date of the Credit Card. This is a combination of expiry month and year.	Required ( <i>String MMY</i> <i>zero fill values</i> <i>i.e. 0103</i> )	Not Used for Cash and Optional for POS Payments
ExpirationMonth	Month of the year in which the credit card expires.	* Required; if no Expiration Date member. ( <i>String MM zero fill</i> )	Not Used for Cash and Optional for POS Payments
ExpirationYear	Year in which the credit card expires. This should be the last two digits of the year (the century is left off)	* Required; if no Expiration Date member. ( <i>String zero fill</i> )	Not Used for Cash and Optional for POS Payments
PurchaseID	Purchase ID such as a purchase number associated with this payment. NOTE: This field is not used when performing RegistrationCRD or RegistrationInquiry functions.	Optional ( <i>String</i> )	None
TrackData	When card data is collected via a card swiped device the TrackData should be passed along in the request. NOTE: This field is not used when performing RegistrationCRD or RegistrationInquiry functions.	Optional ( <i>String</i> )	Required for PIN-based Debit
UserIPAddress	IP address of the user submitting this transaction. ( <i>String</i> ) NOTE: This field is not used when performing RegistrationCRD or RegistrationInquiry functions.	Optional ( <i>String</i> )	
PINData	Encrypted PIN captured by the PIN device.	Not Used	Required for PIN-based Debit



Member	Description	Standard Validation	Validation Exceptions
			(String)
PINKeySerialNumber	Serial number used to encrypt PIN with DUKPT (Derived Unique Key Per Transaction) encryption..	Not Used	Required for PIN-based Debit (String)

PAYMENTINFOEFT OBJECT

This object provides details associated with e-Check Payments.

The PaymentInfoEFT Object is used for the following payment mediums:

- e-Check
- Business Check
- ACH Credit

Member	Description	Standard Validation	Validation Exceptions						
AccountType	The type of the bank account. Valid AccountType values are: <table border="1" data-bbox="488 1052 820 1178"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Checking</td> <td>1</td> </tr> <tr> <td>Savings</td> <td>2</td> </tr> </tbody> </table>	String Value	Numeric Value	Checking	1	Savings	2	Required	None
String Value	Numeric Value								
Checking	1								
Savings	2								
AddressBilling	Billing address for the consumer requesting the E-Check. See EPayAddress object.	Optional (String)	Required for applications boarded with: <ul style="list-style-type: none"> <li>• Fraud and Risk checking including First Name and Last Name for e-Check or Business Check using TeleCheck® Gateway.</li> <li>• Email Address for consumer notification support.</li> </ul>						

## PayPoint® Merchant Integration Guide

Member	Description	Standard Validation	Validation Exceptions
AddressShipping	Shipping address for the consumer requesting the E-Check. See EPayAddress object.	Optional (String)	None
BankAccountNumber	Account from which the E-Check payment is to be settled. Combined with the bank's routing number, this uniquely identifies a bank account.	Required; if not a registered account (payer). (String)	None
BankName	Name of the bank where the bank account is located. If you provide a Bank Name you must also provide the state that the bank resides in the BankState element.	Optional (String)Max Len 50	None
BankRoutingNumber	Routing number for the bank at which the account is located.	Required; if not a registered account (payer). (String)	None
BankState	State where the bank is located is physically located. You can normally find the state listed on your check as part of the name of the bank. If you provide a Bank State you must also provide a BankName.	Optional (String) Max Len 2	None
DriversLicenseNumber	Driver's License Number of the payer. . This field can be used as input to the identity verification process.	Optional (String) Max Len 32	None
DriversLicenseState	State for which the Driver's License was issued. This field can be used as input to the identity verification process.	Optional (String) Max Len 2	None
SSN	Social Security Number of the payer. This field can be used as input to the identity verification process.	Optional (String) Max Len 9	None



Member	Description	Standard Validation	Validation Exceptions												
PreNoteStatus	<p>Used only when performing Registration Inquiry on an e-Check Registered account. If your PayPoint® application has been configured to require Prenotes on every registered account you can utilize this value on Registration Inquiry API request to determine the status of the PreNote. Any other type of API call can ignore this value.</p> <p>Valid values returned on a Registration Inquiry API request include:</p> <table border="1" data-bbox="492 663 964 884"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Unknown</td> <td>0</td> </tr> <tr> <td>Prenote_Created</td> <td>1</td> </tr> <tr> <td>Prenote_Waiting</td> <td>2</td> </tr> <tr> <td>PreNote_Failed</td> <td>3</td> </tr> <tr> <td>PreNote_Sucess</td> <td>4</td> </tr> </tbody> </table>	String Value	Numeric Value	Unknown	0	Prenote_Created	1	Prenote_Waiting	2	PreNote_Failed	3	PreNote_Sucess	4	Optional	None
String Value	Numeric Value														
Unknown	0														
Prenote_Created	1														
Prenote_Waiting	2														
PreNote_Failed	3														
PreNote_Sucess	4														
AuthorizationMedium	<p>If the consumer is providing their registration information via a self service interface (i.e. Web) where they are directly inputting their registration data then AuthorizationMedium should be set to the same values as your PaymentChannel. However, if it's input via a third party on their behalf such as an operator, then this AuthorizationMedium should contain the method by which the authorization was obtained. The following are examples of how to set PaymentChannel vs. AuthorizationMedium:</p> <ol style="list-style-type: none"> <li>1. If Consumer keys their registration via a web interface both PaymentChannel and Authorization Medium would be "Web".</li> <li>2. If the consumer is calling an operator who is keying in the registration via a web page on behalf of the consumer the PaymentChannel would be "Voice" and the Authorization Medium would be "Voice".</li> </ol> <p>Valid AuthorizationMedium values are:</p>	Not Used(String)	Required on RegistrationCRD request (String)												



Member	Description	Standard Validation	Validation Exceptions														
	<table border="1"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Web</td> <td>1</td> </tr> <tr> <td>IVR</td> <td>2</td> </tr> <tr> <td>Walkin</td> <td>3</td> </tr> <tr> <td>Voice</td> <td>4</td> </tr> <tr> <td>FAX</td> <td>5</td> </tr> <tr> <td>Mail</td> <td>6</td> </tr> </tbody> </table>	String Value	Numeric Value	Web	1	IVR	2	Walkin	3	Voice	4	FAX	5	Mail	6		
String Value	Numeric Value																
Web	1																
IVR	2																
Walkin	3																
Voice	4																
FAX	5																
Mail	6																
BusinessName	If the EPayPaymentInfo.EpayPaymentMedium is BusinessCheck this value can be filled with the name of the business on the bank account. This field is used as input to identity verification process.	Optional (String) Max Len 50															
FederalTaxID	If the EPayPaymentInfo.EpayPaymentMedium is BusinessCheck this value can be filled with a Federal Tax ID This field can be used as input to the identity verification process.	Optional (String) Max Len 9															

EPAYADDRESS OBJECT

This object provides address details to the web service. We recommend you supply all address information. The more information you provide to the system, the easier the payment transaction will be to identify in the system. It also will allow your payment processors to perform additional fraud detection.

Member	Description	Standard Validation	Validation Exceptions
NameFirst	First Name	Optional (String) Max Len 25 The validation rule is that First Name may contain letters and/or numbers with spaces allowed and the following special characters which are period, comma, dash, and apostrophe.	Required for e-Check using TeleCheck® Gateway and First Name must contain only letters with no spaces, no numbers, and no special characters.  Optional but recommended for e-Check using client's

## PayPoint® Merchant Integration Guide

Member	Description	Standard Validation	Validation Exceptions
			<p>bank to clear ACH transactions</p> <p>Optional but recommended for ACH Credit</p>
NameLast	Last Name (String)	<p>Optional (String) Max Len 50</p> <p>The validation rule is that Last Name may contain letters and/or numbers with spaces allowed and the following special characters which are period, comma, dash, and apostrophe.</p>	<p>Required for e-Check using TeleCheck® Gateway and Last Name may contain numbers or letters with spaces allowed and the following special characters which are period, dash, and apostrophe.</p> <p>Optional but recommended for e-Check using client's bank to clear ACH transactions</p> <p>Optional but recommended for ACH Credit</p>
NameMiddle	Middle Initial	Optional (String)	None
NameFull	Full Name	<p>Optional (String) Max Len 100</p> <p>The validation is that Full Name may contain letters and/or numbers with spaces allowed and the following special characters which are period, comma, dash, and apostrophe.</p>	<p>Required for e-check using TeleCheck® Gateway, The validation for Full Name is First word must contain only letters. Remaining words may contain letters and/or numbers with spaces allowed and the following special characters which are period, dash, and apostrophe. This must contain a minimum of two words with a space between them.</p> <p>May be required for applications boarded with:</p>



Member	Description	Standard Validation		Validation Exceptions
				<ul style="list-style-type: none"> <li>• Fraud and Risk checking including Address Verification (AVS) for Credit Card and Commercial Credit Card</li> </ul>
		<b>US Phone Number</b>	<b>International Phone Number</b>	
Phone1	Primary phone number	Optional <i>(String)</i> Max Len 10 Must be numeric.	Optional <i>(String)</i> Max Len 20 The validation rule, if filled, is that Phone2 may contain letters, spaces, numbers, and any special characters. Supports standard and extended ascii character sets.	None
Phone2	Secondary phone number	Optional <i>(String)</i> Max Len 10 Must be numeric.	Optional <i>(String)</i> Max Len 20 The validation rule, if filled, is that Phone2 may contain letters, spaces, numbers, and any special characters. Supports standard and extended ascii character sets.	None
Email	Email address	Optional <i>(String)</i> Max Len 75 Must be formatted as a valid email address		

## PayPoint® Merchant Integration Guide

Member	Description	Standard Validation		Validation Exceptions
		US Address	International Address	
Street1	Street address	Optional <i>(String)</i> Max Len 50 The validation rule, if filled, is that Street1 may contain letters and/or numbers with spaces allowed and the following special characters which are period, comma, and the pound sign.	Optional <i>(String)</i> Max Len 50 The validation rule, if filled, is that Street1 may contain letters, spaces, numbers, and any special characters. Supports standard and extended ascii character sets.	<u><b>US Address and International Address</b></u>  May be required for applications boarded with: <ul style="list-style-type: none"> <li>• Fraud and Risk checking including Address Verification (AVS) for Credit Card and Commercial Credit Card</li> </ul> <u><b>International Address</b></u> Not Supported by_e-check using TeleCheck® Gateway,
Street2	Additional street address information	Optional <i>(String)</i> Max Len 50 The validation rule, if filled, is that Street1 may contain letters and/or numbers with spaces allowed and the following special characters which are period, comma, and the pound sign.	Optional <i>(String)</i> Max Len 50 The validation rule, if filled, is that Street1 may contain letters, spaces, numbers, and any special characters. Supports standard and extended ascii character sets.	<u><b>International Address</b></u> Not Supported by_e-check using TeleCheck® Gateway,
City	City	Optional <i>(String)</i> Max Len 50 The validation rule, if filled, is that City may contain letters with spaces allowed and only the special character period.	Optional <i>(String)</i> Max Len 50 The validation rule, if filled, is that City may contain letters, spaces, numbers, and any special characters. Supports standard and extended ascii character sets.	<u><b>US Address and International Address</b></u>  May be required for applications boarded with: <ul style="list-style-type: none"> <li>• Fraud and Risk checking including Address Verification (AVS) for Credit Card and Commercial Credit Card</li> </ul> <u><b>International Address</b></u>

## PayPoint® Merchant Integration Guide

Member	Description	Standard Validation		Validation Exceptions
				Not Supported by_e-check using TeleCheck® Gateway,
State/Province	State/Province	Optional (String) Max Len 2	Optional (String) Max Len 50 The validation rule, if filled, is that State/Province may contain letters, spaces, numbers, and any special characters. Supports standard and extended ascii character sets.	<b><u>US Address and International Address</u></b>  May be required for applications boarded with: <ul style="list-style-type: none"> <li>• Fraud and Risk checking including Address Verification (AVS) for Credit Card and Commercial Credit Card</li> </ul> <b><u>International Address</u></b> Not Supported by_e-check using TeleCheck® Gateway,
Zip	Zip Code	Optional (String) Max Len 9 The validation rule, if filled, is that Zip contains numbers only.	Optional (String) Max Len 20 The validation rule, if filled, is that Zip may contain letters, spaces, numbers, and any special characters. Supports standard and extended ascii character sets.	<b><u>US Address and International Address</u></b>  May be required for applications boarded with: <ul style="list-style-type: none"> <li>• Fraud and Risk checking including Address Verification (AVS) for Credit Card and Commercial Credit Card</li> </ul> <b><u>International Address</u></b> Not Supported by_e-check using TeleCheck® Gateway,
CountryCode	Country Code	Optional (String) Max Len 2 Default to US	Required for international addresses (String) Max Len 2 Uses Country Codes identified in ISO 3166.	<b><u>International Address</u></b> Not Supported by_e-check using TeleCheck® Gateway,

## Functional Objects

## PayPoint® Merchant Integration Guide

The following section provides details on each the of PayPoint® function available for processing payment transactions.

### MAKEPAYMENT

The MakePayment Object is used for all the payment mediums:

- ACH Credit
- Business Check
- Cash
- Commercial Credit Card
- Credit Card
- e-Check
- PINless Debit
- PIN-based Debit
- Point of Sale (POS)

#### Request Object Members

Member	Description	Standard Validation	Validation Exceptions
Header	PayPoint® Header (See Header Object)	Required	None
ConvenienceFee	Convenience Fee - Only if application requires it. The amount of money charged for this transaction. This value is generated through a call to the "CalculateConvenienceFee" function.	Optional (Decimal)	Not Supported for Cash, POS, or ACH Credit
RegisterID	Registered Account Identifier. This ID is used to identify a previously registered account.	Optional (String)	Not Supported for Cash, POS, or ACH Credit
GroupID	Group ID – An agency business application may set the Group ID to a unique value to group a set of payments as a single transaction set. When searching for payments in PayPoint® administrative site, grouped items can be easily displayed together	Optional (String)	Not Supported for Cash, POS, or ACH Credit
RecurringPaymentID	Not Used	Not Used	None
TransactionBatchID	Not Used	Not Used	None
PaymentAmount	The face value of the transaction amount, which includes taxes. If you're utilizing EPay to calculate and apply convenience fees, the convenience fee should NOT be included in the Payment Amount. .	Required (Decimal)	None

## PayPoint® Merchant Integration Guide

PaymentDate	Payment Date – This allows for post dating a payment. It is only available for eCheck transactions. <i>Previous dates are not accepted.</i>	Optional (String) Format Required MM/DD/YYYY).	None
PaymentInfo	Payment Info contains Card or E-Check account information. See PaymentInfoCC or PaymentInfoEFT depending on payment method.	Required	None
Reference	Reference – An agency business application may send application-specific information to the EPE to help identify the transaction. Reference data should contain identifiers that will help the agency cross-reference this payment to a specific business transaction. For example, a courts application may send a ticket number to make it easier to locate a payment for that ticket using the search feature and it will make it easier to match payments using the posting file to a transaction in a parking ticket application.	Optional (String)Max Len 254	None
TaxAmount	Tax Amount – For commercial cards only	Not Supported	Optional for Commercial Credit Cards only (Decimal)

### Response Object Members

Member	Description																		
ConfirmationNumber	ConfirmationNumber is a string containing a unique number assigned by PayPoint® for this payment. If the payment transaction failed, this value may be blank. (String)																		
ResultMessage	ResultMessage is a string containing a message describing the result of a method call. (String)																		
ReturnCode	Return code specifying the result of the request (String). Either a numeric or descriptive code will be returned depending on the integration method. Numeric for HTTP & String Value for Web Service. Both are listed as the possible values returned. <table border="1" data-bbox="553 1591 1344 1894"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Unknown</td> <td>0</td> </tr> <tr> <td>Payment_Success</td> <td>2</td> </tr> <tr> <td>Cancel_Success</td> <td>3</td> </tr> <tr> <td>Error</td> <td>4</td> </tr> <tr> <td>Declined</td> <td>5</td> </tr> <tr> <td>Verification_failed</td> <td>6</td> </tr> <tr> <td>Communication_Error</td> <td>7</td> </tr> <tr> <td>Network_Error</td> <td>10</td> </tr> </tbody> </table>	String Value	Numeric Value	Unknown	0	Payment_Success	2	Cancel_Success	3	Error	4	Declined	5	Verification_failed	6	Communication_Error	7	Network_Error	10
String Value	Numeric Value																		
Unknown	0																		
Payment_Success	2																		
Cancel_Success	3																		
Error	4																		
Declined	5																		
Verification_failed	6																		
Communication_Error	7																		
Network_Error	10																		



Member	Description																															
	Processor_Mismatch	11																														
	CreditCards_Disabled	12																														
	Unaccepted_Card_Type	13																														
	Payment_Exceeds_System_Limit	14																														
	Payment_Exceeds_Card_Limit	15																														
	Possible_Duplicate_Payment	16																														
	Unresolved_Cancellation	17																														
	Undefined_Item	18																														
	eChecks_Disabled	24																														
	Missing_Identification	25																														
	Waiting_On_PreNote	26																														
	PreNote_Failed	27																														
	Payment_Pending	32																														
	Post_Date_Too_Large	33																														
	PINLessDebit_Disabled	38																														
	PINDebit_Disabled	39																														
For a full description of Return Codes see the end of this section.																																
SettlementSubmissionDate	The date when the payment will be submitted for settlement. <i>(String – Format Required MM/DD/YYYY)</i>																															
TotalAmount	Total Amount equals the payment amount plus the convenience fee and any other charges associated with the transaction. <i>(Decimal)</i>																															
CardType	<table border="1"> <thead> <tr> <th data-bbox="537 947 1073 999">String Value</th> <th data-bbox="1073 947 1421 999">Numeric Value</th> </tr> </thead> <tbody> <tr><td data-bbox="537 999 1073 1031">Test</td><td data-bbox="1073 999 1421 1031">1</td></tr> <tr><td data-bbox="537 1031 1073 1062">VISA</td><td data-bbox="1073 1031 1421 1062">2</td></tr> <tr><td data-bbox="537 1062 1073 1094">MC</td><td data-bbox="1073 1062 1421 1094">3</td></tr> <tr><td data-bbox="537 1094 1073 1125">AMEX</td><td data-bbox="1073 1094 1421 1125">4</td></tr> <tr><td data-bbox="537 1125 1073 1157">DISC</td><td data-bbox="1073 1125 1421 1157">5</td></tr> <tr><td data-bbox="537 1157 1073 1188">DCCB</td><td data-bbox="1073 1157 1421 1188">6</td></tr> <tr><td data-bbox="537 1188 1073 1220">CBLN</td><td data-bbox="1073 1188 1421 1220">7</td></tr> <tr><td data-bbox="537 1220 1073 1251">JAL</td><td data-bbox="1073 1220 1421 1251">8</td></tr> <tr><td data-bbox="537 1251 1073 1283">JCB</td><td data-bbox="1073 1251 1421 1283">9</td></tr> <tr><td data-bbox="537 1283 1073 1314">ENRT</td><td data-bbox="1073 1283 1421 1314">10</td></tr> <tr><td data-bbox="537 1314 1073 1346">STAR</td><td data-bbox="1073 1314 1421 1346">12</td></tr> <tr><td data-bbox="537 1346 1073 1377">Pulse</td><td data-bbox="1073 1346 1421 1377">13</td></tr> <tr><td data-bbox="537 1377 1073 1409">NYCE</td><td data-bbox="1073 1377 1421 1409">14</td></tr> <tr><td data-bbox="537 1409 1073 1440">PIN Based Debit</td><td data-bbox="1073 1409 1421 1440">15</td></tr> </tbody> </table>	String Value	Numeric Value	Test	1	VISA	2	MC	3	AMEX	4	DISC	5	DCCB	6	CBLN	7	JAL	8	JCB	9	ENRT	10	STAR	12	Pulse	13	NYCE	14	PIN Based Debit	15	
String Value	Numeric Value																															
Test	1																															
VISA	2																															
MC	3																															
AMEX	4																															
DISC	5																															
DCCB	6																															
CBLN	7																															
JAL	8																															
JCB	9																															
ENRT	10																															
STAR	12																															
Pulse	13																															
NYCE	14																															
PIN Based Debit	15																															
AuthorizationCode	If this is a Credit Card payment and the return code is Payment_Success(2) this value will be filled with the Credit Card Processors Authorization Code. This code should be shown on any receipts presented to the consumer. <i>(String)</i>																															

# PayPoint® Merchant Integration Guide

## PINLESSDEBITCHECK

This object is only used to identify the status of a PINless debit payment and does not apply to all other payment mediums.

### Request Object Members

Member	Description	Standard Validation
Header	PayPoint® Header Object (See description above)	Required
CardNumber	PinlessDebit Card Number	Required

### Response Object Members

Member	Description												
CardType	<table border="1"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Test</td> <td>1</td> </tr> <tr> <td>STAR</td> <td>12</td> </tr> <tr> <td>Pulse</td> <td>13</td> </tr> <tr> <td>NYCE</td> <td>14</td> </tr> </tbody> </table>	String Value	Numeric Value	Test	1	STAR	12	Pulse	13	NYCE	14		
String Value	Numeric Value												
Test	1												
STAR	12												
Pulse	13												
NYCE	14												
ResultMessage	ResultMessage is a string containing a message describing the result of the request. ( <i>String</i> )												
ReturnCode	<p>Return code specifying the result of the request. Either a numeric or descriptive code (<i>string</i>) will be returned depending on the integration method. Numeric for HTTP &amp; String Value for Web Service. Both are listed as the possible values returned.</p> <table border="1"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Unknown</td> <td>0</td> </tr> <tr> <td>Eligible</td> <td>36</td> </tr> <tr> <td>Not_Eligible</td> <td>37</td> </tr> <tr> <td>PINLessDebit_Disabled</td> <td>38</td> </tr> <tr> <td>PINDebit_Disabled</td> <td>39</td> </tr> </tbody> </table> <p>For a full description of Return Codes see the end of this section.</p>	String Value	Numeric Value	Unknown	0	Eligible	36	Not_Eligible	37	PINLessDebit_Disabled	38	PINDebit_Disabled	39
String Value	Numeric Value												
Unknown	0												
Eligible	36												
Not_Eligible	37												
PINLessDebit_Disabled	38												
PINDebit_Disabled	39												

## CALCULATECONVENIENCEFEE

This object is used to determine the associated convenience fees and supports the following payment mediums:

- Business Check
- Commercial Credit Card
- Credit Card
- e-Check
- PINless Debit
- PIN-based Debit

*Request Object Members*

Member	Description	Standard Validation	Validation Exceptions
Header	PayPoint® Header Object (See description above)	Required	Not supported by ACHCredit, Cash. and POS.
PaymentAmount	Payment amount to use in the convenience fee calculation.	Required ( <i>Decimal</i> )	Not supported by ACHCredit, Cash. and POS.

*Response Object Members*

Member	Description										
ConvenienceFee	The dollar amount of the convenience fee associated with the payment amount passed. ( <i>Decimal</i> )										
PaymentAmount	Payment amount that corresponds to the provided convenience fee. ( <i>Decimal</i> )										
ResultMessage	ResultMessage is a string containing a message describing the result of the request. ( <i>String</i> )										
ReturnCode	<p>Return code specifying the result of the request. Either a numeric or descriptive code (<i>string</i>) will be returned depending on the integration method. Numeric for HTTP &amp; String Value for Web Service. Both are listed as the possible values returned.</p> <table border="1"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Unknown</td> <td>0</td> </tr> <tr> <td>Success</td> <td>1</td> </tr> <tr> <td>Error</td> <td>4</td> </tr> <tr> <td>Undefined_Item</td> <td>18</td> </tr> </tbody> </table> <p>For a full description of Return Codes see the end of this section.</p>	String Value	Numeric Value	Unknown	0	Success	1	Error	4	Undefined_Item	18
String Value	Numeric Value										
Unknown	0										
Success	1										
Error	4										
Undefined_Item	18										

## PayPoint® Merchant Integration Guide

### CANCELPAYMENT

This object is used to stop a payment from processing and supports all payment mediums:

- ACH Credit
- Business Check
- Cash
- Commercial Credit Card
- Credit Card
- e-Check
- PINless Debit
- PIN-based Debit
- Point of Sale (POS)

#### Request Object Members

Member	Description	Standard Validation	Validation Exceptions
Header	PayPoint® Header Object (See description above)	Required	None
ConfirmationNumber	Confirmation Number that was returned by the original "MakePayment" request.	Required (String)	None
RefundAmount	Amount of the payment that will be refunded upon completion of the CancelPayment request. The amount must be an amount less than or equal to the original amount of the payment excluding any convenience fees,	Required (Decimal)	None
ConvenienceFeeRefundAmount	Amount of the convenience fee will be refunded.	Not Used	Required if your application is configured with PayPoint® Convenience fee options. The amount must be less than or equal to the original convenience fee. (Decimal)
TrackData	When card data is collected via a card swipe device the TrackData should be passed along in the request.	Not Used	Required for PIN-based Debit (String)
PINData	Encrypted PIN captured by the PIN device	Not Used	Required for PIN-based Debit (String)

## PayPoint® Merchant Integration Guide

PINKeySerialNumber	Serial number used to encrypt PIN with DUKPT (Derived Unique Key Per Transaction) encryption.	Not Used	Required for PIN-based Debit (String)
--------------------	---	----------	--

## Response Object Members

Member	Description																												
ConfirmationNumber	ConfirmationNumber is the confirmation number assigned by PayPoint® for a cancellation transaction. This confirmation number will differ from the one received when the original payment was made to ensure the system tracks each and every event. (String)																												
RefundAmount	Amount of the payment that will be refunded upon completion of the CancelPayment request. (Decimal)																												
ResultMessage	ResultMessage is a string containing a message describing the result of the request. (String)																												
ReturnCode	<p>Return code specifying the result of the request. Either a numeric or descriptive code (string) will be returned depending on the integration method. Numeric for HTTP &amp; String Value for Web Service. Both are listed as the possible values returned.</p> <table border="1"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Unknown</td> <td>0</td> </tr> <tr> <td>Cancel_Success</td> <td>3</td> </tr> <tr> <td>Error</td> <td>4</td> </tr> <tr> <td>Declined</td> <td>5</td> </tr> <tr> <td>Verification_failed</td> <td>6</td> </tr> <tr> <td>Communication_Error</td> <td>7</td> </tr> <tr> <td>Network_Error</td> <td>10</td> </tr> <tr> <td>Processor_Mismatch</td> <td>11</td> </tr> <tr> <td>CreditCards_Disabled</td> <td>12</td> </tr> <tr> <td>Unaccepted_Card_Type</td> <td>13</td> </tr> <tr> <td>Possible_Duplicate_Payment</td> <td>16</td> </tr> <tr> <td>Unresolved_Cancellation</td> <td>17</td> </tr> <tr> <td>Undefined_Item</td> <td>18</td> </tr> </tbody> </table> <p>For a full description of Return Codes see the end of this section.</p>	String Value	Numeric Value	Unknown	0	Cancel_Success	3	Error	4	Declined	5	Verification_failed	6	Communication_Error	7	Network_Error	10	Processor_Mismatch	11	CreditCards_Disabled	12	Unaccepted_Card_Type	13	Possible_Duplicate_Payment	16	Unresolved_Cancellation	17	Undefined_Item	18
String Value	Numeric Value																												
Unknown	0																												
Cancel_Success	3																												
Error	4																												
Declined	5																												
Verification_failed	6																												
Communication_Error	7																												
Network_Error	10																												
Processor_Mismatch	11																												
CreditCards_Disabled	12																												
Unaccepted_Card_Type	13																												
Possible_Duplicate_Payment	16																												
Unresolved_Cancellation	17																												
Undefined_Item	18																												
SettlementSubmissionDate	The date when the refund will be submitted for settlement. (String MM/DD/YYYY)																												

PAYMENTSTATUS

The PaymentStatus Object is used to identify information about a specific payment and supports all payment mediums:

- ACH Credit
- Business Check
- Cash
- Commercial Credit Card
- Credit Card
- e-Check
- PINless Debit
- PIN-based Debit
- Point of Sale (POS)

Request Object Members

Member	Description	Standard Validation
Header	PayPoint® Header Object (See description above)	Required
ConfirmationNumber	Confirmation Number that was returned by the original "MakePayment" request. <i>(String)</i>	See Note Below
Reference	Reference data can be used to search for the payment status. The reference data sent in the payment status object must match the reference data sent in the original payment transaction. <i>(String) The field can be very helpful for customer service. You can use the reference data to search for a payment status when your customer doesn't know the confirmation number.</i>	See Note Below

**\* Note: You must pass either a Confirmation Number or Reference ID. If you pass data in both fields, the payment status check will default to the confirmation number.**

Response Object Members

Member	Description																								
PaymentStatus	Results of the payment status inquiry will return of the following codes: <table border="1" data-bbox="500 1486 1409 1879"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Unknown</td> <td>0</td> </tr> <tr> <td>Payment_Success</td> <td>2</td> </tr> <tr> <td>Cancel_Success</td> <td>3</td> </tr> <tr> <td>Error</td> <td>4</td> </tr> <tr> <td>Declined</td> <td>5</td> </tr> <tr> <td>Verification_failed</td> <td>6</td> </tr> <tr> <td>Communication_Error</td> <td>7</td> </tr> <tr> <td>Settled</td> <td>8</td> </tr> <tr> <td>Settled_Error</td> <td>9</td> </tr> <tr> <td>Network_Error</td> <td>10</td> </tr> <tr> <td>Processor_Mismatch</td> <td>11</td> </tr> </tbody> </table>	String Value	Numeric Value	Unknown	0	Payment_Success	2	Cancel_Success	3	Error	4	Declined	5	Verification_failed	6	Communication_Error	7	Settled	8	Settled_Error	9	Network_Error	10	Processor_Mismatch	11
String Value	Numeric Value																								
Unknown	0																								
Payment_Success	2																								
Cancel_Success	3																								
Error	4																								
Declined	5																								
Verification_failed	6																								
Communication_Error	7																								
Settled	8																								
Settled_Error	9																								
Network_Error	10																								
Processor_Mismatch	11																								



Member	Description											
	CreditCards_Disabled	12										
	Unaccepted_Card_Type	13										
	Payment_Exceeds_System_Limit	14										
	Payment_Exceeds_Card_Limit	15										
	Possible_Duplicate_Payment	16										
	Unresolved_Cancellation	17										
	Undefined_Item	18										
	Chargeback	19										
	Chargeback_Reversal	20										
	Settlement_Incomplete	21										
	Partial_Settlement	22										
	Settlement_Pending	23										
	eChecks_Disabled	24										
	Missing_Identification	25										
	Waiting_On_PreNote	26										
	PreNote_Failed	27										
	Stop_Payment_Issued	28										
	Non_Sufficient_Funds	29										
	Final_Non_Sufficient_Funds	30										
	Account_Invalid	31										
	Payment_Pending	32										
	Post_Date_Too_Large	33										
	Refund_Settlement_Pending	34										
	Pre_Auth_Success	35										
	Not_Eligible	37										
	PINLessDebit_Disabled	38										
	PINDebit_Disabled	39										
For a full description of Return Codes see the end of this section.												
ResultMessage	ResultMessage is a string containing a message describing the result of a method call. ( <i>String</i> )											
ReturnCode	Return code specifying the result of the request. Either a numeric or descriptive code ( <i>string</i> ) will be returned depending on the integration method. Numeric for HTTP & String Value for Web Service. Both are listed as the possible values returned.											
	<table border="1"> <thead> <tr> <th data-bbox="500 1430 1021 1472">String Value</th> <th data-bbox="1021 1430 1292 1472">Numeric Value</th> </tr> </thead> <tbody> <tr> <td data-bbox="500 1472 1021 1503">Unknown</td> <td data-bbox="1021 1472 1292 1503">0</td> </tr> <tr> <td data-bbox="500 1503 1021 1535">Success</td> <td data-bbox="1021 1503 1292 1535">1</td> </tr> <tr> <td data-bbox="500 1535 1021 1566">Error</td> <td data-bbox="1021 1535 1292 1566">4</td> </tr> <tr> <td data-bbox="500 1566 1021 1598">Undefined_Item</td> <td data-bbox="1021 1566 1292 1598">18</td> </tr> </tbody> </table>	String Value	Numeric Value	Unknown	0	Success	1	Error	4	Undefined_Item	18	
String Value	Numeric Value											
Unknown	0											
Success	1											
Error	4											
Undefined_Item	18											
For a full description of Return Codes see the end of this section.												
ConfirmationNumber	Confirmation Number of the payment status requested. This value will be provided in the results anytime there was a match on the ConfirmationNumber or References provided on the input parameters. ( <i>String</i> )											

REGISTRATIONCRD

The RegistrationCRD Object supports the following payment mediums:

- Business Check
- Commercial Credit Card
- Credit Card
- e-Check
- PINless Debit
- PIN-based Debit

*Request Object Members*

Member	Description	Standard Validation	Validation Exceptions								
Header	PayPoint® Header (See Header Object)	Required	Not Supported for ACHCredit, Cash, and POS								
Action	Register Action to be performed.  Valid Action values include: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Create</td> <td>1</td> </tr> <tr> <td>Update</td> <td>2</td> </tr> <tr> <td>Delete</td> <td>3</td> </tr> </tbody> </table>	String Value	Numeric Value	Create	1	Update	2	Delete	3	Required	Not Supported for ACHCredit, Cash, and POS
String Value	Numeric Value										
Create	1										
Update	2										
Delete	3										
AgreedToTerms	Before sending registration data to the EPE, your application must ask the user for approval. If they agree, then this flag should be set to true. If this flag is not set to true, the registration cannot be stored with our system.	Required <i>(Boolean)</i>	Not Supported for ACHCredit, Cash, and POS								
RegisterID	Utilized for update or delete actions. This should contain the original RegisterID returned when creating your Registration.	Required <i>(String)</i>	Not Supported for ACHCredit, Cash, and POS								
PaymentInfo	Payment Info contains credit card or e-Check billing information. See PaymentInfoCC or PaymentInfoEFT depending on type of transaction.	Required	Not Supported for ACHCredit, Cash, and POS								

*Response Object Members*

Member	Description
RegisterID	For a Create request, the Registration ID assigned by PayPoint® is returned. For Replace or Delete requests, the Registration ID passed



	in the request object is returned. <i>(String)</i>														
ResultMessage	ResultMessage is a string containing a message describing the result of the request. <i>(String)</i>														
ReturnCode	<p>Return code specifying the result of the request. Either a numeric or descriptive code <i>(string)</i> will be returned depending on the integration method. Numeric for HTTP &amp; String Value for Web Service. Both are listed as the possible values returned.</p> <table border="1"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Unknown</td> <td>0</td> </tr> <tr> <td>Success</td> <td>1</td> </tr> <tr> <td>Error</td> <td>4</td> </tr> <tr> <td>Declined</td> <td>5</td> </tr> <tr> <td>Network_Error</td> <td>10</td> </tr> <tr> <td>Undefined_Item</td> <td>18</td> </tr> </tbody> </table> <p>For a full description of Return Codes see the end of this section.</p>	String Value	Numeric Value	Unknown	0	Success	1	Error	4	Declined	5	Network_Error	10	Undefined_Item	18
String Value	Numeric Value														
Unknown	0														
Success	1														
Error	4														
Declined	5														
Network_Error	10														
Undefined_Item	18														

REGISTRATIONINQUIRY

The RegistrationCRD Object is used to determine information about a registered account and supports the following payment mediums:

- Business Check
- Commercial Credit Card
- Credit Card
- e-Check
- PINless Debit
- PIN-based Debit

Request Object Members

Member	Description	Standard Validation	Validation Exceptions
RegisterID	The PayPoint® Registration ID returned by the original create registration ID request.	Required <i>(String)</i>	Not Supported for ACHCredit, Cash, and POS
Header	PayPoint® Header (See Header Object)	Required	Not Supported for ACHCredit, Cash, and POS



*Response Object Members*

Member	Description														
PaymentInfo	Payment Info contains Card or E-Check account information. See PaymentInfoCC or PaymentInfoEFT depending on the type of account registered. The financial information and other confidential data is truncated to ensure privacy.														
ResultMessage	ResultMessage is a string containing a message describing the result of request. ( <i>String</i> )														
ReturnCode	<p>Return code specifying the result of the request. Either a numeric or descriptive code (<i>string</i>) will be returned depending on the integration method. Numeric for HTTP &amp; String Value for Web Service. Both are listed as the possible values returned.</p> <table border="1" data-bbox="578 667 1369 915"> <thead> <tr> <th data-bbox="578 667 1101 716">String Value</th> <th data-bbox="1101 667 1369 716">Numeric Value</th> </tr> </thead> <tbody> <tr> <td data-bbox="578 716 1101 747">Unknown</td> <td data-bbox="1101 716 1369 747">0</td> </tr> <tr> <td data-bbox="578 747 1101 779">Success</td> <td data-bbox="1101 747 1369 779">1</td> </tr> <tr> <td data-bbox="578 779 1101 810">Error</td> <td data-bbox="1101 779 1369 810">4</td> </tr> <tr> <td data-bbox="578 810 1101 842">Declined</td> <td data-bbox="1101 810 1369 842">5</td> </tr> <tr> <td data-bbox="578 842 1101 873">Network_Error</td> <td data-bbox="1101 842 1369 873">10</td> </tr> <tr> <td data-bbox="578 873 1101 915">Undefined_Item</td> <td data-bbox="1101 873 1369 915">18</td> </tr> </tbody> </table> <p>For a full description of Return Codes see the end of this section.</p>	String Value	Numeric Value	Unknown	0	Success	1	Error	4	Declined	5	Network_Error	10	Undefined_Item	18
String Value	Numeric Value														
Unknown	0														
Success	1														
Error	4														
Declined	5														
Network_Error	10														
Undefined_Item	18														

## PayPoint® Merchant Integration Guide

### RECURRINGPAYMENTCRD

The RecurringPaymentCRD Object is used to create a recurring payment schedule and supports the following payment mediums:

- Business Check
- Commercial Credit Card
- Credit Card
- e-Check
- PINless Debit
- PIN-based Debit

#### Request Object Members

Member	Description	Standard Validation	Validation Exceptions								
Action	Recurring Payment Action requested.  Valid Action values include: <table border="1" data-bbox="483 825 951 982"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Create</td> <td>1</td> </tr> <tr> <td>Update</td> <td>2</td> </tr> <tr> <td>Delete</td> <td>3</td> </tr> </tbody> </table>	String Value	Numeric Value	Create	1	Update	2	Delete	3	Required (String)	Not Supported for ACHCredit, Cash, and POS
String Value	Numeric Value										
Create	1										
Update	2										
Delete	3										
Header	PayPoint® Header (See Header Object)	Required	Not Supported for ACHCredit, Cash, and POS								
RecurringID	The PayPoint® Recurring Payment ID returned to the original Create request is required to Delete. This member is omitted for Create requests. (String)	Required (String)	Not Used for Create requests.  Not Supported for ACHCredit, Cash, and POS								
RecurringPaymentInfo	PayPoint® Recurring Payment Schedule Object.	Required	Not Supported for ACHCredit, Cash, and POS								

#### RecurringPaymentInfo Subobject

Member	Description	Standard Validation	Validation Exceptions
RegistrationID	Registration ID to which the recurring payment schedule is associated.	Required (String)	Not Supported for ACHCredit, Cash, and POS
BeginDate	Define the Begin date of the recurring payments (String – Format Required MM/DD/YYYY)	Required (String MM/DD/YYYY)	Not Supported for ACHCredit, Cash, and POS
EndDate	Define the end date of the recurring payments (String – Format Required MM/DD/YYYY)	Required (String MM/DD/YYYY)	Not Supported for ACHCredit, Cash, and POS



Member	Description	Standard Validation	Validation Exceptions								
NextPayDate	Define the next payment date ( <i>String – Format Required MM/DD/YYYY</i> )	Reserved for internal use only.	Not Supported for ACHCredit, Cash, and POS								
RecurringIntervalType	<p>Defines the interval parameters of the recurring payment schedule object. See description below.</p> <p>Valid RecurringIntervalTypes values are:</p> <table border="1"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Daily</td> <td>1</td> </tr> <tr> <td>Monthly</td> <td>2</td> </tr> <tr> <td>Yearly</td> <td>3</td> </tr> </tbody> </table>	String Value	Numeric Value	Daily	1	Monthly	2	Yearly	3	Required	Not Supported for ACHCredit, Cash, and POS
String Value	Numeric Value										
Daily	1										
Monthly	2										
Yearly	3										
IntervalParam1	Define the parameters for schedule number 1. See description below.	Required Depending on type of recurring schedule being created. ( <i>String</i> )	Not Supported for ACHCredit, Cash, and POS								
IntervalParam2	Define the parameters for schedule number 2. See description below.	Required Depending on type of recurring schedule being created. ( <i>String</i> )	Not Supported for ACHCredit, Cash, and POS								
IntervalParam3	Define the parameters for schedule number 3. See description below.	Required Depending on type of recurring schedule being created. ( <i>String</i> )	Not Supported for ACHCredit, Cash, and POS								
IntervalParam4	Not Used	Not Used	Not Supported for ACHCredit, Cash, and POS								
PaymentAmount	The face value of the transaction amount, which includes taxes. If you're utilizing EPay to calculate and apply convenience fees the convenience fee should NOT be included in the Payment Amount. .	Required ( <i>Decimal</i> )	Not Supported for ACHCredit, Cash, and POS								
Reference	Reference – An agency business application may send application-specific information to the EPE to help identify the transaction.	Optional ( <i>String</i> ) Max Len 254	Not Supported for ACHCredit, Cash, and POS								

## PayPoint® Merchant Integration Guide

Member	Description	Standard Validation	Validation Exceptions
	Reference data should contain identifiers that will help the agency cross-reference this payment to a specific business transaction. For example, a courts application may send a ticket number to make it easier to locate a payment for that ticket using the search feature and it will make it easier to match payments using the posting file to a transaction in a parking ticket application.		

### Using Recurrence Patterns

To create a recurring payment pattern, you must specify the "Recurring Interval Type" along with three string parameters: IntervalParam1, IntervalParam2, and IntervalParam3. Please note that although the API currently contains a property for IntervalParam4, this value is not used.

### Setting up Daily recurring payments

An integer value of 1 is used when you specify a recurring interval type of **Daily**. The recurring pattern is defined by the first parameter (IntervalParam1).

*IntervalParam1 = <blank>*

<blank> = a payment will occur every day.

*IntervalParam1 = "Interval"*

<interval> = When IntervalParam1 is the string "Interval", IntervalParam2 must contain a string that contains numbers only. This string is the number of days between each payment.

*IntervalParam1 = "Days"*

<Days> When IntervalParam2 is the string "Days", IntervalParam2 must contain a seven-character string which contains days of the week payments will occur. The first character in the string represents Sunday. For example, the string "0100000" indicates that payments will occur on Mondays. The string "0111110" indicates that payments will occur on Monday, Tuesday, Wednesday, Thursday, and Friday. The string "0101010" indicates that payments will occur on Monday, Wednesday, and Friday. You may include IntervalParam3 as an integer specifying the number of weeks to skip between payment cycles.

#### Daily Examples

*Payment every other day*

IntervalParam1 = "Interval"

IntervalParam2 = "2"

IntervalParam3 = ""

*Payments every weekday*

IntervalParam1 = "Days"

IntervalParam2 = "0111110"

IntervalParam3 = ""

*Payments biweekly on Fridays*

IntervalParam1 = "Days"

IntervalParam2 = "0000010"

IntervalParam3 = "2"

## PayPoint® Merchant Integration Guide

### **Setting up Monthly recurring payments**

An integer value of 2 is used when you specify a recurring interval type of **Monthly**. The recurring pattern is defined by the first parameter (IntervalParam1).

*IntervalParam1 = "Days"*

<Days> When IntervalParam1 is "Days", IntervalParam2 should be a comma-delimited list of days of the month. Note that the days 29, 30, and 31 will automatically be converted to the first of the month. It is recommended that you not use these days. IntervalParam3 and IntervalParam4 are optional when using "Days" in IntervalParam1. IntervalParam3 can contain a number between 1 and 28 representing the number of days BEFORE the day identified in IntervalParam2. For example this can be used in situations where you want the payment executed 2 days prior to the first of the month. IntervalParam4 can contain a number between 1 and 12 representing the number of months between each payment, i.e. every other month would be 2. If this value is not supplied it's assumed to be executed every month.

*IntervalParam1 = "DayByWeek"*

<DayByWeek> When IntervalParam1 is "DayByWeek", you are specifying that recurring payments should occur in the form of "the first Monday" or "the third Wednesday" of each month. IntervalParam2 specifies the week number (1, 2, 3, or 4 for first, second, third, or fourth weeks). IntervalParam3 specifies the day of the week (1 is Sunday, 7 is Saturday). IntervalParam4 is optional when using "DaybyWeek" in IntervalParam1. IntervalParam4 can contain a number between 1 and 12 representing the number of months between each payment, i.e. every other month would be 2. If this value is not supplied it's assumed to be executed every month.

### Monthly Examples

*Payments on the 1st of each month*

IntervalParam1 = "Days"

IntervalParam2 = "1"

IntervalParam3 = ""

*Payments on the 2 days before the end of month*

IntervalParam1 = "Days"

IntervalParam2 = "1"

IntervalParam3 = "2"

*Payments Simi Monthly*

IntervalParam1 = "Days"

IntervalParam2 = "1"

IntervalParam3 = ""

IntervalParam4 = "2"

*Payments on the 1st, 10th, and 20th of each month*

IntervalParam1 = "Days"

IntervalParam2 = "1, 10, 20"

IntervalParam3 = ""

*Payments on the 1st Monday of each month*

IntervalParam1 = "DayByWeek"

IntervalParam2 = "1"

IntervalParam3 = "2"

*Payments on the 3rd Friday of each month*

IntervalParam1 = "DayByWeek"

IntervalParam2 = "3"

IntervalParam3 = "6"

## PayPoint® Merchant Integration Guide

### Setting up Yearly recurring payments

An integer value of 3 is used when you specify a recurring interval type of **Yearly**. The recurring pattern is defined by the first parameter (IntervalParam1).

*IntervalParam1 = "Days"*

<Days> When IntervalParam1 is "Days", IntervalParam2 should be a comma-delimited list of dates of the year. These dates must be specified in the format "MM/DD".

#### Yearly Examples

*Payment on April 15th of each year*

IntervalParam1 = "Days"

IntervalParam2 = "04/15"

IntervalParam3 = ""

*Payment on January 30th, April 30th, July 30th, and October 30th of each year*

IntervalParam1 = "Days"

IntervalParam2 = "01/30, 04/30, 07/30, 10/30"

IntervalParam3 = ""

### Response Object Members

Member	Description														
RecurringID	This contains the RecurringID returned by PayPoint® when creating your recurring payment. ( <i>String</i> )														
ResultMessage	ResultMessage is a string containing a message describing the result of a method call. ( <i>String</i> )														
ReturnCode	Return code specifying the result of the request. Either a numeric or descriptive code ( <i>string</i> ) will be returned depending on the integration method. Numeric for HTTP & String Value for Web Service. Both are listed as the possible values returned. <table border="1" data-bbox="578 1129 1370 1381"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Unknown</td> <td>0</td> </tr> <tr> <td>Success</td> <td>1</td> </tr> <tr> <td>Error</td> <td>4</td> </tr> <tr> <td>Declined</td> <td>5</td> </tr> <tr> <td>Network_Error</td> <td>10</td> </tr> <tr> <td>Undefined_Item</td> <td>18</td> </tr> </tbody> </table> <p>For a full description of Return Codes see the end of this section.</p>	String Value	Numeric Value	Unknown	0	Success	1	Error	4	Declined	5	Network_Error	10	Undefined_Item	18
String Value	Numeric Value														
Unknown	0														
Success	1														
Error	4														
Declined	5														
Network_Error	10														
Undefined_Item	18														

### RECURRINGPAYMENTINQUIRY

The RecurringPaymentInquiry Object is used to retrieve information about a recurring payment schedule and supports the following payment mediums:

- Business Check
- Commercial Credit Card
- Credit Card
- e-Check
- PINless Debit
- PIN-based Debit

*Request Object Members*

Member	Description	Standard Validation	Validation Exceptions
Header	PayPoint® Header Object (See description above)	Required	Not Supported for ACHCredit, Cash, and POS
RecurringID	The PayPoint® Recurring Payment ID returned by the original Create request.	Required (String)	Not Supported for ACHCredit, Cash, and POS

*Response Object Members*

Member	Description														
RecurringPaymentInfo	PayPoint® Recurring Payment Schedule Object. (See RecurringPaymentCRD above for object details)														
ResultMessage	ResultMessage is a string containing a message describing the result of the request. (String)														
ReturnCode	<p>Return code specifying the result of the request. Either a numeric or descriptive code (string) will be returned depending on the integration method. Numeric for HTTP &amp; String Value for Web Service. Both are listed as the possible values returned.</p> <table border="1"> <thead> <tr> <th>String Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Unknown</td> <td>0</td> </tr> <tr> <td>Success</td> <td>1</td> </tr> <tr> <td>Error</td> <td>4</td> </tr> <tr> <td>Declined</td> <td>5</td> </tr> <tr> <td>Network_Error</td> <td>10</td> </tr> <tr> <td>Undefined_Item</td> <td>18</td> </tr> </tbody> </table> <p>For a full description of Return Codes see the end of this section.</p>	String Value	Numeric Value	Unknown	0	Success	1	Error	4	Declined	5	Network_Error	10	Undefined_Item	18
String Value	Numeric Value														
Unknown	0														
Success	1														
Error	4														
Declined	5														
Network_Error	10														
Undefined_Item	18														

**Secure HTTP Integration**

Secure HTTP Integration uses standard HTTPS web technology to communicate via either the query string or the HTTP FORM POST method. This integration method allows an agency to integrate their existing or new Web-based applications without having to develop the pages to collect payment authorization data.

The business application in all cases performs all calculations, item totaling, and tax calculations to determine the final payment amount. PayPoint® collects the payment amount, the credit card number or bank account number and other financial data depending on the payment method. PayPoint® processes the payment transaction in a secure dialog with the payment processor. Developers responsible for the business application integration must be proficient in HTML.

## PayPoint® Merchant Integration Guide

---

The business application may send data to PayPoint® either through a query string or as FORM POST data. Depending on the method requested, the API returns its results to the application either via a query string or as the HTTP GET results of the query.

With Secure HTTP integration, the business application can pass all payment data or allow PayPoint® to collect the account information with a “Page Pop” web page. Using the Page Pop method, the business application passes transaction-specific information about the payment and PayPoint® “pops” a web page to capture the payment account data. This is not a pop-up page similar to web advertisements. This page will prompt the payer for financial data and present the results of the authorization on a confirmation page. The Page Pop method allows the business application to pass parameters to customize the look of the web page (including references to the agency’s style sheets).

Internet Explorer limits the query string to 2,048 characters. While most other browsers support more than this, as does the FORM POST method, PayPoint® is designed to work with the lower limit. Thus, all input and output parameter names are cryptically short to save space in the query string. For example, the Returncode is a numeric value instead of a string value.

The URL’s referenced in this example utilize “localhost”. You will want to replace this reference with PayPoint® URL provided after successfully registering for an PayPoint® account.

To ensure proper interpretation of your parameters you should URL encode each of the parameter values.

Secure HTTP Integration supports the following payment mediums:

- ACH Credit
- Business Check
- Cash
- Commercial Credit Card
- Credit Card
- e-Check
- PINless Debit
- PIN-based Debit
- Point of Sale (POS).

## PayPoint® Merchant Integration Guide

### Standard Input Parameters

The following parameters are standard for all PayPoint® HTTP API requests.

Parameter	Description	Standard Validation
<b>m</b>	Return Mode. This value must be either “q” for query string returns, or “r” for response mode.	Required
<b>r</b>	Return Page. This is the URL of the page that the API should return its results to via the query string. This value is required when the Return Mode is set to “q.”	Optional
<b>a</b>	Application ID. This is the numeric Application ID for your specific enterprise/agency/application.	Required
<b>s</b>	Security Key. The unique token assigned to the PayPoint® Account. An initial token is assigned during Account Registration. The token can be updated through the PayPoint® administrative site.	Required
<b>p</b>	Payment Channel. This integer specifies the payment channel used for a particular payment or interaction. The acceptable values are 0 (unknown), 1 (Web), 2 (IVR), 3 (walk-in), 4 (fax), 5 (voice), 6 (mail) and 7 (recurring).	Required
<b>i</b>	Reference Identifier. You may pass in a reference identifier to help tie your request to a particular user or session within your application. This id will simply be returned to you as part of the results. Note: this parameter is optional for developer use in maintaining state between calls to the HTTP API. It is not stored in the database with the payment, and should not be confused with MakePayment’s “Custom Reference” field.	Optional

## PayPoint® Merchant Integration Guide

### Standard Output Parameters

The following parameters are standard for the HTTP API responses.

Parameter	Description
<b>c</b>	Return Code. This is a numeric code specifying the results of the API method call. Please see the PayPoint® Web Service API documentation for the list of these values and their meanings.
<b>m</b>	Result Message. English text describing the results for the API call. The value is <blank> for successful and will contain data for failures and errors.
<b>i</b>	Reference Identifier. This value is the custom reference identifier that you passed as optional input parameter.

### Client Side Usage Example

Using the client-side API requires information to be passed between the business application and the PayPoint® via a user's browser. The following ASP code makes a call to the API's method to calculate the convenience fee for a particular payment.

```
Response.Redirect
"https://www.govone.com/epay/http/ccf.aspx?m=q&r=http://localhost/myapp/respons
e.asp&i=1234&a=1&y=10.75"
```

In the example above, the API page for calculating the convenience fee (ccf.aspx) is called for application 1 for a payment amount of \$10.75. The API page will process the request and return its result to your specified page (in this case, it's "<http://localhost/myapp/response.asp>") on the query string. The following is an example of the page redirection that may occur from a successful call:

```
http://localhost/myapp/response.asp?c=1&m=&i=1234&o=2.50&y=10.75&t=13.25
```

The return code (c) is 1, and the message (m) is empty. The reference identifier (i) is the same as the one passed in. The convenience fee (o), payment amount (y), and total amount (t) are all returned to you.

#### Notes:

- To ensure proper interpretation of your parameters you should URL encode each of the parameter values.

### Server Side Usage

Using the server-side API allows all information to be passed between your application's server and the PayPoint® API server without using the user's browser. In the example below, uses the WinHTTP object to post the data to the server.

```
sFormData = "m=r&i=1234&a=1&n=MyApp&p=1&y=10.75"
Set oHTTP = CreateObject("WinHttp.WinHttpRequest.5")
oHTTP.Open "POST", "https://www.govone.com/epay/http/ccf.aspx"
oHTTP.SetRequestHeader "Content-Type", "application/x-www-form-
urlencoded"
```

## PayPoint® Merchant Integration Guide

```
oHTTP.Send sFormData
```

```
'oHTTP.ResponseText contains the response data
```

The server-side API can either return the data via the query string to a referring page, or it can send it directly back to you via results of the FORM POST. The latter is the preferred technique, as it is more straightforward and simple to use (the example request above uses this technique). The results that come back from this method are in the form of *name=value*<br>. For example, the results from the above query would be:

```
c=1<br>m=<br>i=1234<br>o=2.50<br>y=10.75<br>t=13.25<br>
```

### Notes:

- The order of return parameters is not guaranteed!
- Although "WinHttp.WinHttpRequest.5.1" is the standard object Microsoft recommends using, on older machines with just the WinHTTP SDK installed, a Prog ID of "WinHttp.WinHttpRequest.5" may work instead.
- If you are converting the above code to VB6, make sure to declare the sFormData variable as a Variant – not a String. A bug in the WinHTTP API will cause the request to fail if you pass a String variable to the Send method.
- To ensure proper interpretation of your parameters you should URL encode each of the parameter values.

### Reading the API Method Specifications

The input and output parameters are listed in the API specifications below. Specified with each parameter is the full name of the property that this parameter represents within the PayPoint® Web Service API documentation. The expected data type of the parameter is also provided. Please refer to the Web Service API documentation (above) for more information on specific properties.

## PayPoint® Merchant Integration Guide

### USING THE PAGE POP CAPABILITY

Through the Secure HTTP API, a business application may also have PayPoint® “pop” an HTML page to collect payment information from the user to process a payment authorization request.

- The HTML Payment Collection Interface uses the same standard input parameters and output parameters as the Secure HTTP API methods. However, the mode (m) must be “q” (query string) and the return page (“r”) must be specified. (Note: To ensure proper interpretation of your parameters you should URL encode each of the parameter values)

The HTML Payment Collection Interface is accessed through the <https://www.govone.com/epayadmin/http/pay.aspx> page on the PayPoint® website. Redirecting users to this page will take them to the PayPoint® website for gathering payment information and then return the results of the transaction to the specified return page.

In addition to the standard input parameters, the following input parameters may be used when using the Page Pop feature:

Parameter	Description	Standard Validation
<b>Pt</b>	Page Title. This is the <title> of the HTML page.	Optional
<b>Ht</b>	Header Text. This text is displayed at the top of the payment collection page. You may customize this text to suit your needs. This text can contain HTML tags (i.e., include image tags that reference images on your website).	Optional
<b>Ft</b>	Footer Text. This text is displayed at the bottom of the payment collection page. You may customize this text to suit your needs. This text can contain HTML tags (i.e., include image tags that reference images on your website).	Optional
<b>Cht</b>	Confirmation Header Text. This text is displayed at the top of the confirmation page (if enabled). This text can contain HTML tags (i.e., include image tags that reference images on your website).	Optional
<b>Su</b>	Stylesheet URL. The URL to the stylesheet you want used for this page. If this value is not specified, “pay.css” is used. To see what classes and styles are used by pay.aspx, download pay.css from your PayPoint® website.	Optional
<b>Tc</b>	Terms and Conditions URL. The URL to the HTML page containing the terms and conditions for using this payment collection interface. If this value is not specified, “terms.htm” is used.	Optional
<b>Pm</b>	Payment Medium Flag. This is an integer value that allows multiple Boolean properties to be set with one value. Calculate the value of Flag by adding all of the properties you wish to set. The default value for this property is 1. <ol style="list-style-type: none"> <li>1. Allow Credit Card Payments</li> <li>2. 2 – Allow EFT Payments</li> </ol>	Optional
<b>Gi</b>	Group ID*	Optional
<b>Re</b>	Reference*	Optional
<b>Pa</b>	Payment Amount*	Required

## PayPoint® Merchant Integration Guide

Parameter	Description	Standard Validation
<b>Cf</b>	Convenience Fee* This value is required if convenience fees are being charged for the application.	Optional
<b>Ta</b>	Tax Amount*	Optional
<b>Pd</b>	Payment Date* Post dating a payment is only allowed with eChecks.	Optional
<b>Pi</b>	Purchase ID*	
<b>Ct</b>	Card Type Flag. This flag will allow you to identify the types of cards that a user can select from on the pop form. This is an integer value that allows multiple Boolean properties to be set with one value. Calculate the value of Flag by adding all of the properties you wish to set. The default value for this property (if not specified) is 15 (the first four card types). 1 – Allow Visa 2 – Allow Mastercard 4 – Allow American Express 8 – Allow Discover Card 16 – Allow DCCB 32 – Allow CBLN 64 – Allow JAL 128 – Allow JCB 256 – Allow ENRT	Optional
<b>F</b>	Flags. This is an integer value that allows multiple Boolean properties to be set with one value. Calculate the value of Flag by adding all of the properties you wish to set. The default value for this property is 0. 1 – Collect Address 2 – Collect CVV2 4 – Collect Verified by VISA password 8 – Show Confirmation Page 16 – Collect Drivers License Info (EFT Only) 32 – Collect SSN (EFT Only) 64 – Collection Bank Information (EFT Only) 128 - Collect Business Name and FID (EFT Only)	Optional
<b>ri</b>	Registration ID. If your storing registered account information with PayPoint® through the RegistrationCRD API interfaces you can pass a PayPoint® Registration ID which will pre-populate the payment window with the users registration information including Address, Card Number, etc.	Optional
<b>pit</b>	Pinless Introduction Text – allows you to provide help information regarding the option a user has to make a PINLess Debit transaction. If your PayPoint® Application account is enabled for PINless transactions and the user enters a card number that is valid for PINless transactions they will be presented with a radio box selection after entering their account information asking them to choose between Debit or Credit Card. This text appears above the radio card entry.	Optional

*\*This is a standard input parameter on the PayPoint® Web Service's Make Payment method. See the PayPoint® Web Service documentation for more information on this parameter.*

## PayPoint® Merchant Integration Guide

Once the payment information has been collected and processed, the results of the processing are returned via the query string to your specified return page. In addition to the standard output parameters defined above, the following output parameters may also be returned:

Parameter	Description
<b>O</b>	EPayMakePaymentResult . ConfirmationNumber ( <i>String</i> )
<b>T</b>	EPayMakePaymentResult . TotalAmount ( <i>Decimal</i> )
<b>S</b>	EPayMakePaymentResult . SettlementSubmissionDate ( <i>String</i> )
<b>D</b>	EPayMakePaymentResult . CardType ( <i>EPayCardType Enumeration</i> )
<b>Z</b>	EPayMakePaymentResult.AuthorizationCode ( <i>String</i> )

### CALCULATECONVENIENCEFEE

The Calculate Convenience Fee method is accessed via the <https://www.govone.com/epay/http/ccf.aspx> API page on the PayPoint® web site.

#### Input Parameters

Parameter Identifier	Corresponding Web Service Member
<b>Y</b>	EPayCalculateConvenienceFeeRequest.PaymentAmount ( <i>Decimal</i> )

#### Output Parameters

Parameter Identifier	Corresponding Web Service Member
<b>o</b>	EPayCalculateConvenienceFeeResult.ConvenienceFee ( <i>Decimal</i> )
<b>y</b>	EPayCalculateConvenienceFeeResult.PaymentAmount ( <i>Decimal</i> )
<b>t</b>	EPayCalculateConvenienceFeeResult.TotalAmount ( <i>Decimal</i> )

### PINLESSDEBITCHECK

The Pinless Debit Check method is accessed via the <https://www.govone.com/epay/http/pdc.aspx> API page on the PayPoint® web site.

#### Input Parameters

Parameter Identifier	Corresponding Web Service Member
<b>C</b>	EPayPINlessDebitCheck.CardNumber

## PayPoint® Merchant Integration Guide

### Output Parameters

Parameter Identifier	Corresponding Web Service Member
n	<u><a href="#">PinlessDebitCheck.CardType</a></u>

### CANCELPAYMENT

The Cancel Payment method is accessed via the <https://www.govone.com/epay/http/cp.aspx> API page on the PayPoint® web site.

### Input Parameters

Parameter Identifier	Corresponding Web Service Member
o	EPayCancelPaymentRequest.ConfirmationNumber ( <i>String</i> )
e	EPayCancelPaymentRequest.RefundAmount ( <i>Decimal</i> )
v	EPayCancelPaymentRequest.ConvenienceFeeRefundAmount ( <i>Decimal</i> )
t	EPayCancelPaymentRequest.TrackData ( <i>String</i> )
pd	EPayCancelPaymentRequest.PINData ( <i>String</i> )
ksn	EPayCancelPaymentRequest.PINKeySerialNumber ( <i>String</i> )

### Output Parameters

Parameter Identifier	Corresponding Web Service Member
o	EPayCancelPaymentResult.ConfirmationNumber ( <i>String</i> )
e	EPayCancelPaymentResult.RefundAmount ( <i>Decimal</i> )
t	EPayCancelPaymentResult.SettlementSubmissionDate ( <i>String</i> )

## PayPoint® Merchant Integration Guide

### REGISTRATIONCRD (CREATE, REPLACE, DELETE)

The Registration CRD method is accessed via the <https://www.govone.com/epay/http/e.aspx> API page on the PayPoint® web site.

#### Input Parameters

Parameter Identifier	Corresponding Web Service Member
T	EPayRegisterCRDRequest.Action ( <i>EPayCRDAction Enumeration</i> )
G	EPayRegistrationCRDRequest.AgreedToTerms ( <i>Boolean</i> )
E	EPayRegisterCRDRequest.RegisterID ( <i>String</i> )
Pp	EPayRegisterCRDRequest.PaymentInfo.PaymentMedium ( <i>EPayPaymentMedium Enumeration</i> )
ppa	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AccountType ( <i>EPayEFTAccountType Enumeration</i> )
ppb	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.BankRoutingNumber ( <i>String</i> )
ppn	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.BankAccountNumber ( <i>String</i> )
ppk	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.BankState ( <i>String</i> )
ppm	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.BankName ( <i>String</i> )
ppd	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.DriversLicenseNumber ( <i>String</i> )
ppr	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.DriversLicenseState ( <i>String</i> )
pps	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.SSN ( <i>String</i> )
ppu	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoEFT.BusinessName ( <i>String</i> )
ppt	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoEFT.AuthorizationMedium ( <i>EPayPaymentChannel Enumeration</i> )
ppp	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoEFT.PreNoteStatus ( <i>EPayPreNoteStatus Enumeration</i> )
ppan	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.NameFirst ( <i>String</i> )
ppaa	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.NameLast ( <i>String</i> )
ppam	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.NameMiddle ( <i>String</i> )
ppae	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.NameFull ( <i>String</i> )
ppap	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Phone1 ( <i>String</i> )
ppah	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Phone2 ( <i>String</i> )
ppai	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Email ( <i>String</i> )
ppas	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Street1

Parameter Identifier	Corresponding Web Service Member
	<i>(String)</i>
<b>ppat</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Street2 <i>(String)</i>
<b>ppac</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.City <i>(String)</i>
<b>ppast</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.State <i>(String)</i>
<b>ppaz</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Zip <i>(String)</i>
<b>ppdn</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.NameFirst <i>(String)</i>
<b>ppda</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.NameLast <i>(String)</i>
<b>ppdm</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.NameMiddle <i>(String)</i>
<b>ppde</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.NameFull <i>(String)</i>
<b>ppdp</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Phone1 <i>(String)</i>
<b>ppdh</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Phone2 <i>(String)</i>
<b>ppdi</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Email <i>(String)</i>
<b>ppds</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Street1 <i>(String)</i>
<b>ppdt</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Street2 <i>(String)</i>
<b>ppdc</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.City <i>(String)</i>
<b>ppdst</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.State <i>(String)</i>
<b>ppdz</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Zip <i>(String)</i>
<b>pac</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.CardNumber <i>(String)</i>
<b>pae</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.ExpirationDate <i>(String)</i>
<b>pax</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.ExpirationMonth <i>(String)</i>
<b>pap</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.ExpirationYear <i>(String)</i>
<b>pabn</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.NameFirst <i>(String)</i>
<b>paba</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.NameLast <i>(String)</i>
<b>pabm</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.NameMiddle <i>(String)</i>
<b>pabe</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.NameFull <i>(String)</i>
<b>pabp</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Phone1

Parameter Identifier	Corresponding Web Service Member
	<i>(String)</i>
<b>pabh</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Phone2 <i>(String)</i>
<b>pabi</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Email <i>(String)</i>
<b>pabs</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Street1 <i>(String)</i>
<b>pabt</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Street2 <i>(String)</i>
<b>pabc</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.City <i>(String)</i>
<b>pabst</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.State <i>(String)</i>
<b>pabz</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Zip <i>(String)</i>
<b>pasn</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.NameFirst <i>(String)</i>
<b>pasa</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.NameLast <i>(String)</i>
<b>pasmm</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.NameMiddle <i>(String)</i>
<b>pase</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.NameFull <i>(String)</i>
<b>paspp</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Phone1 <i>(String)</i>
<b>pasph</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Phone2 <i>(String)</i>
<b>paspi</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Email <i>(String)</i>
<b>pass</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Street1 <i>(String)</i>
<b>past</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Street2 <i>(String)</i>
<b>pascc</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.City <i>(String)</i>
<b>passt</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.State <i>(String)</i>
<b>pasz</b>	EPayRegistrationCRDRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Zip <i>(String)</i>
<b>paa</b>	EPayRegisterCRDRequest.PaymentInfo.PaymentInfoCC.CardType ( <i>EPayCardType Enumeration</i> )

*Output Parameters*

Parameter Identifier	Corresponding Web Service Member
<b>e</b>	EPayRegisterCRDResult.RegisterID <i>(String)</i>

## PayPoint® Merchant Integration Guide

### REGISTRATIONINQUIRY

The Register Inquiry method is accessed via the <https://www.govone.com/epay/http/ei.aspx> API page on the PayPoint® web site.

#### Input Parameters

Parameter Identifier	Corresponding Web Service Member
e	EPayRegisterInquiryRequest.RegisterID (String)

#### Output Parameters

Parameter Identifier	Corresponding Web Service Member
pp	EPayRegisterInquiryResult.PaymentInfo.PaymentMedium (EPayPaymentMedium Enumeration)
ppa	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AccountType (EPayEFTAccountType Enumeration)
ppb	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.BankRoutingNumber (String)
ppn	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.BankAccountNumber (String)
ppk	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.BankState (String)
ppm	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.BankName (String)
ppd	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.DriversLicenseNumber (String)
ppr	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.DriversLicenseState (String)
pps	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.SSN (String)
ppu	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoEFT.BusinessName (String)
ppf	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoEFT.FederalTaxID (String)
ppt	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoEFT.AuthorizationMedium (EPayPaymentChannel Enumeration)
ppp	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoEFT.PreNoteStatus (EPayPreNoteStatus Enumeration)
ppan	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.NameFirst (String)
ppaa	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.NameLast (String)
ppam	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.NameMiddle (String)
ppae	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.NameFull (String)
ppap	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.Phone1 (String)
ppah	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.Phone2 (String)
ppai	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.Email (String)

## PayPoint® Merchant Integration Guide

Parameter Identifier	Corresponding Web Service Member
<b>ppas</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.Street1 (String)
<b>ppat</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.Street2 (String)
<b>ppac</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.City (String)
<b>ppast</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.State (String)
<b>ppaz</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressShipping.Zip (String)
<b>ppdn</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.NameFirst (String)
<b>ppda</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.NameLast (String)
<b>ppdm</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.NameMiddle (String)
<b>ppde</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.NameFull (String)
<b>ppdp</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.Phone1 (String)
<b>ppdh</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.Phone2 (String)
<b>ppdi</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.Email (String)
<b>ppds</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.Street1 (String)
<b>ppdt</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.Street2 (String)
<b>ppdc</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.City (String)
<b>ppdst</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.State (String)
<b>ppdz</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoEFT.AddressBilling.Zip (String)
<b>pac</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.CardNumber (String)
<b>paе</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.ExpirationDate (String)
<b>pax</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.ExpirationMonth (String)
<b>pap</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.ExpirationYear (String)
<b>pabn</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.NameFirst (String)
<b>paba</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.NameLast (String)
<b>pabm</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.NameMiddle (String)
<b>pabe</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.NameFull (String)
<b>pabp</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.Phone1 (String)
<b>pabh</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.Phone2 (String)
<b>pabi</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.Email (String)

## PayPoint® Merchant Integration Guide

Parameter Identifier	Corresponding Web Service Member
<b>pabs</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.Street1 ( <i>String</i> )
<b>pabt</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.Street2 ( <i>String</i> )
<b>pabc</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.City ( <i>String</i> )
<b>pabst</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.State ( <i>String</i> )
<b>pabz</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.BillingAddress.Zip ( <i>String</i> )
<b>pasn</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.NameFirst ( <i>String</i> )
<b>pasa</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.NameLast ( <i>String</i> )
<b>pasm</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.NameMiddle ( <i>String</i> )
<b>pase</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.NameFull ( <i>String</i> )
<b>pasp</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.Phone1 ( <i>String</i> )
<b>pash</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.Phone2 ( <i>String</i> )
<b>pasi</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.Email ( <i>String</i> )
<b>pass</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.Street1 ( <i>String</i> )
<b>past</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.Street2 ( <i>String</i> )
<b>pasc</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.City ( <i>String</i> )
<b>passt</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.State ( <i>String</i> )
<b>pasz</b>	EPayRegistrationInquiryResult.PaymentInfo.PaymentInfoCC.ShippingAddress.Zip ( <i>String</i> )
<b>paa</b>	EPayRegisterInquiryResult.PaymentInfo.PaymentInfoCC.CardType ( <i>EPayCardType Enumeration</i> )

## PayPoint® Merchant Integration Guide

### MAKEPAYMENT

The Make Payment method is accessed via the <https://www.govone.com/epay/http/mp.aspx> API page on the PayPoint® web site.

#### Input Parameters

Parameter Identifier	Corresponding Web Service Member
<b>g</b>	EPayMakePaymentRequest.GroupID ( <i>String</i> )
<b>e</b>	EPayMakePaymentRequest.RegisterID ( <i>String</i> )
<b>f</b>	EPayMakePaymentRequest.Reference ( <i>String</i> )
<b>y</b>	EPayMakePaymentRequest.PaymentDate ( <i>String</i> )
<b>o</b>	EPayMakePaymentRequest.PaymentAmount ( <i>Decimal</i> )
<b>v</b>	EPayMakePaymentRequest.ConvenienceFee ( <i>Decimal</i> )
<b>x</b>	EPayMakePaymentRequest.TaxAmount ( <i>Decimal</i> )
<b>pp</b>	EPayMakePaymentRequest.PaymentInfo.PaymentMedium ( <i>EPayPaymentMedium Enumeration</i> )
<b>ppa</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AccountType ( <i>EPayEFTAccountType Enumeration</i> )
<b>ppb</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.BankRoutingNumber ( <i>String</i> )
<b>ppn</b>	EpayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.BankAccountNumber ( <i>String</i> )
<b>ppk</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.BankState ( <i>String</i> )
<b>ppm</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.BankName ( <i>String</i> )
<b>ppd</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.DriversLicenseNumber ( <i>String</i> )
<b>ppr</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.DriversLicenseState ( <i>String</i> )
<b>pps</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.SSN ( <i>String</i> )
<b>ppu</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.BusinessName ( <i>String</i> )
<b>ppf</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.FederalTaxID ( <i>String</i> )
<b>ppan</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.NameFirst ( <i>String</i> )
<b>ppaa</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.NameLast ( <i>String</i> )
<b>ppam</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.NameMiddle ( <i>String</i> )
<b>ppae</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.NameFull ( <i>String</i> )
<b>ppap</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Phone1 ( <i>String</i> )
<b>ppah</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Phone2 ( <i>String</i> )
<b>ppai</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Email ( <i>String</i> )

## PayPoint® Merchant Integration Guide

Parameter Identifier	Corresponding Web Service Member
<b>ppas</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Street1 (String)
<b>ppat</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Street2 (String)
<b>ppac</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.City (String)
<b>ppast</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.State (String)
<b>ppaz</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressShipping.Zip (String)
<b>ppdn</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.NameFirst (String)
<b>ppda</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.NameLast (String)
<b>ppdm</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.NameMiddle (String)
<b>ppde</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.NameFull (String)
<b>ppdp</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Phone1 (String)
<b>ppdh</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Phone2 (String)
<b>ppdi</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Email (String)
<b>ppds</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Street1 (String)
<b>ppdt</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Street2 (String)
<b>ppdc</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.City (String)
<b>ppdst</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.State (String)
<b>ppdz</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoEFT.AddressBilling.Zip (String)
<b>pac</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.CardNumber (String)
<b>pae</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ExpirationDate (String)
<b>pax</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ExpirationMonth (String)
<b>pap</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ExpirationYear (String)
<b>pau</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.PurchaseID (String)
<b>pat</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.TrackData (String)
<b>pav</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.CVV2 (String)
<b>pas</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.UserIPAddress (String)
<b>pabn</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.NameFirst (String)
<b>paba</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.NameLast (String)
<b>pabm</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.NameMiddle (String)
<b>pabe</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.NameFull

## PayPoint® Merchant Integration Guide

Parameter Identifier	Corresponding Web Service Member
	<i>(String)</i>
<b>pabp</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Phone1 <i>(String)</i>
<b>pabh</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Phone2 <i>(String)</i>
<b>pabi</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Email <i>(String)</i>
<b>pabs</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Street1 <i>(String)</i>
<b>pabt</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Street2 <i>(String)</i>
<b>pabc</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.City <i>(String)</i>
<b>pabst</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.State <i>(String)</i>
<b>pabz</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.BillingAddress.Zip <i>(String)</i>
<b>pasn</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.NameFirst <i>(String)</i>
<b>pasa</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.NameLast <i>(String)</i>
<b>pasm</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.NameMiddle <i>(String)</i>
<b>pase</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.NameFull <i>(String)</i>
<b>pasp</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Phone1 <i>(String)</i>
<b>pash</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Phone2 <i>(String)</i>
<b>pasi</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Email <i>(String)</i>
<b>pass</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Street1 <i>(String)</i>
<b>past</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Street2 <i>(String)</i>
<b>pasc</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.City <i>(String)</i>
<b>passt</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.State <i>(String)</i>
<b>pasz</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ShippingAddress.Zip <i>(String)</i>
<b>paa</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.CardType <i>(EPayCardType Enumeration)</i>
<b>par</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.CardStatusFlag <i>(EPayCreditCardStatusFlag Enumeration)</i>
<b>pao</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.ECommerceGoodsFlag <i>(Boolean)</i>
<b>pai</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.Installment <i>(Boolean)</i>
<b>papd</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.PINData <i>(String)</i>
<b>pan</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.InstallmentSequence <i>(Integer)</i>
<b>pal</b>	EPayMakePaymentRequest.PaymentInfo.PaymentInfoCC.InstallmentCount <i>(Integer)</i>

*Output Parameters*

## PayPoint® Merchant Integration Guide

Parameter Identifier	Corresponding Web Service Member
<b>o</b>	EPayMakePaymentResult.ConfirmationNumber ( <i>String</i> )
<b>t</b>	EPayMakePaymentResult.TotalAmount ( <i>Decimal</i> )
<b>e</b>	EPayMakePaymentResult.SettlementSubmissionDate ( <i>String</i> )
<b>d</b>	EPayMakePaymentResult.CardType ( <i>EPayCardType Enumeration</i> )
<b>z</b>	EPayMakePaymentResult.AuthorizationCode ( <i>String</i> )

### PAYMENTSTATUS

The Payment Status method is accessed via the <https://www.govone.com/epay/http/ps.aspx> API page on the PayPoint® web site.

#### Input Parameters

Parameter Identifier	Corresponding Web Service Member
<b>o</b>	EPayPaymentStatusRequest.ConfirmationNumber ( <i>String</i> )
<b>E</b>	EPayPaymentStatusRequest.Reference ( <i>String</i> )

#### Output Parameters

Parameter Identifier	Corresponding Web Service Member
<b>y</b>	EPayPaymentStatusResult.PaymentStatus ( <i>EPayResultCode Enumeration</i> )
<b>O</b>	EPayPaymentStatusResult.ConfirmationNumber ( <i>String</i> )

## PayPoint® Merchant Integration Guide

### RECURRINGPAYMENTCRD (CREATE, DELETE)

The Recurring Payment CRD method is accessed via the <https://www.govone.com/epay/http/r.aspx> API page on the PayPoint® web site.

#### Input Parameters

Parameter Identifier	Corresponding Web Service Member
<b>t</b>	EPayRecurringPaymentCRDRequest.Action ( <i>EPayCRDAction Enumeration</i> )
<b>rr</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.RecurringPaymentID ( <i>String</i> )
<b>re</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.RegisterID ( <i>String</i> )
<b>rb</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.BeginDate ( <i>String</i> )
<b>rn</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.EndDate ( <i>String</i> )
<b>rx</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.NextPayDate ( <i>String</i> )
<b>rc</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.RecurringIntervalType ( <i>EPayRecurringIntervalType Enumeration</i> )
<b>ri</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.IntervalParam1 ( <i>String</i> )
<b>rt</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.IntervalParam2 ( <i>String</i> )
<b>rv</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.IntervalParam3 ( <i>String</i> )
<b>ra</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.IntervalParam4 ( <i>String</i> )
<b>rp</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.PaymentAmount ( <i>Decimal</i> )
<b>rf</b>	EPayRecurringPaymentCRDRequest.RecurringPaymentInfo.Reference ( <i>String</i> )
<b>e</b>	EPayRecurringPaymentCRDRequest.RecurringID ( <i>String</i> )
<b>f</b>	EPayRecurringPaymentCRDRequest.Reference ( <i>String</i> )

#### Output Parameters

Parameter Identifier	Corresponding Web Service Member
<b>e</b>	EPayRecurringPaymentCRDResult.RecurringID ( <i>String</i> )

## PayPoint® Merchant Integration Guide

### RECURRINGPAYMENTINQUIRY

The Recurring Payment Inquiry method is accessed via the <https://www.govone.com/epay/http/ri.aspx> API page on the PayPoint® web site.

#### Input Parameters

Parameter Identifier	Corresponding Web Service Member
<b>e</b>	EPayRecurringPaymentInquiryRequest.RecurringID ( <i>String</i> )

#### Output Parameters

Parameter Identifier	Corresponding Web Service Member
<b>rr</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.RecurringPaymentID ( <i>String</i> )
<b>re</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.RegisterID ( <i>String</i> )
<b>rb</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.BeginDate ( <i>String</i> )
<b>rn</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.EndDate ( <i>String</i> )
<b>rx</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.NextPayDate ( <i>String</i> )
<b>rc</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.RecurringIntervalType ( <i>EPayRecurringIntervalType Enumeration</i> )
<b>ri</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.IntervalParam1 ( <i>String</i> )
<b>rt</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.IntervalParam2 ( <i>String</i> )
<b>rv</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.IntervalParam3 ( <i>String</i> )
<b>ra</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.IntervalParam4 ( <i>String</i> )
<b>rp</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.PaymentAmount ( <i>Decimal</i> )
<b>rf</b>	EPayRecurringPaymentInquiryResult.RecurringPaymentInfo.Reference ( <i>String</i> )

## PayPoint® Merchant Integration Guide

### Batch Interface Option

The PayPoint® Batch interface allows you to integrated payment functionality in an off-line mode. This interface can be used by any integrator who does not need a real time response to an authorization request. The same API functions discussed in the Web Service section above are supported through PayPoint® Batch Processing option. PayPoint® will accept batch XML files that conform to the XML Schema defined by PayPoint® through its Secure FTP site. All XML files received prior to 12:00 AM EST can be picked up by 6:00 AM EST. In order to transmit batch files to and from PayPoint® you must first contact PayPoint® customer service organization and request a FTP Account for sending and receiving files to our Secure FTP (SFTP) site. You will need to obtain a SFTP client. PayPoint® recommends F-Secure.

In order to integrate with PayPoint's batch interface you must be familiar with building and reading XML files. PayPoint® has a developed XML Schema for the XML Request and XML Response files. The request file will contain all of the commands you wish Point to execute for you. When completed PayPoint® will return a XML Response file that contains the API results of each of your request commands.

When creating Request files there are a couple critical rules that you must follow:

**One File Per Application ID** – PayPoint® will only accept commands for a single Application Identifier. If you are developing a solution that utilizes multiple PayPont Application Identifiers you must separate your API commands for each application into separate XML request Files.

**XML Format** – If you do not provide well formed XML you will not receive a response XML file back. Reasons for not receiving a response file back can include malformed XML, invalid Application Identifier or no Application Identifier.

**Unique File Naming** - You should attempt to name your request files with a unique name. To ensure uniqueness within PayPoint's batch processing environment your file will automatically assign a unique suffix including a unique date/time stamp.

**Dropping and Picking Up Files** – You will be assigned a unique id for access to a PayPoint® Secure FTP site. There will be two directories under the FTP account assigned to you. There will be a "Request" directory where you will need to drop files you want to be processed. And there will be a "Response" directory where you will pickup the results of your batch process. Request received by 12:00 AM EST can be picked up by 6:00 AM EST.

You can obtain a copy of the latest PayPoint® XML Request and Response XSD (Schema) files at the following URL location <https://www.govone.com/epayadmin/validatebulkfile.aspx>. This page also contains an option to have PayPoint® validate your XML against our schemas and a sample file that shows various API request in a single file for a single application.

The naming convention utilized in the XML Schema follows the same naming of the web service object model described earlier in this section. When developing the content of your XML data please refer to the Web Service section of this document for the available API request / response elements and their definitions.

The only difference is there are a couple of extra elements available for you to pass custom data. This is the data that goes within the "CustomData" "Custom" element. Data within the comments element are ignored by PayPoint®. This custom element not required. For example, it can be used to application or batch identifiers unique to this transmission that may be useful to the integrated application when it receives a response file back in associating it with an original transmission or set of transactions.

## PayPoint® Merchant Integration Guide

### Return Code Definitions

The result of actions through API calls or batch processing such as settlement with in PayPoint® result in a specific return code associated with the action. These return codes are visible within the PayPoint® Administrative interface when researching specific payments. In addition API request also receive additional details with the ResultMessage provided with the result of any API request. The Result Message will contain more descriptive text. For example you may receive a result of Undefined\_Item (18) which indicates that you've provided invalid data. The Result Message would describe which element did not meet data validation requirements. Below is a list of the return code enumerations and numeric values associated with possible return codes.

The return codes provide information on that status of a given API request or results associated with post processing activities like settlement. Refer to the specific API request to determine the possible return codes for a given API request.

Return Code	Numeric ReturnCode	Description
Success	1	Request was received and successfully processed Returned on API calls which don't involve executing a payment or payment cancellation.
Payment_Success	2	Specific to Make a Payment. Indicates the payment was accepted and successfully issued for authorization.
Cancel_Success	3	Specific to Cancel Payment. Indicates the request to cancel a payment was received and successfully issued to the processor.
Error	4	Indicates an error occurred in processing your request. Any number of problems could produce an error condition. You must refer to the Result Message for more details. In some cases the Result Message will contain a reference number that can be used by PayPoint® support staff to determine the source of a problem. In normal processing conditions Error after integration testing are rare, however your application still must be prepared to alert the consumer that a problem has occurred and provide options for dealing payment submission through other forms
Declined	5	Indicates that a payment authorization has been denied by the processor. There can be multiple reasons for a Decline by a processor including insufficient funds, fraudulent activity against the card or account. The Result Message may contain additional details that come back from the processor. Your application will need to deal with alerting the consumer that their payment authorization has been declined. You may want to ask them to verify their information and/or utilize a different payment method.
Verification_Failed	6	Indicates that that consumer identity information did not match the processors records. If you are enforcing checks on CVV2 or Address verification and the consumers input does not match what

## PayPoint® Merchant Integration Guide

Return Code	Numeric ReturnCode	Description
		the process has on file the payment will be declined. This is a form of payment Decline and your application must provide options for the user to verify their information and re-submit the payment and/or provide an alternative payment method.
Communication_Error	7	Indicates that the PayPoint® system is experiencing issues communicating with in its internal systems. PayPoint® connections are built on a high level of redundancy which means the likelihood of getting this return code is rare. If you application experiences these errors you should contact PayPoint® Support personnel.
Settled	8	Indicates that the transaction was issued for Settlement and was successfully accepted. When issue payment actions such as Make A Payment or Cancel Payment your initial request is sent through an authorization process.
Settlement_Error	9	Indicates that when the payment action was issued for settlement an error occurred. If a payment action receives a Settlement Error you should contact PayPoint® Support personnel for additional details.
Network_Error	10	Indicates that the PayPoint® system is experiencing issues communicating with third party processing. For example PayPoint® relies on credit card processors such as Vital, PaymentTech, FDMS, etc to process payment authorizations. If the processor communications is not available you'll receive this return code. Your application needs to deal with provide consumer feedback that their payment cannot currently be processed and provide them with options to return later and/or provide other methods of payment. PayPoint® and its third party connections are built on a high level of redundancy which means the likely hood of getting the return code are very rare. If you application experiences these errors you should contact PayPoint® Support personnel.
CreditCards_Disabled	12	Indicates that the application you are attempting to issue a credit card payment action against is currently not enabled for Credit Cards. If you wish to start accepting credit cards contact PayPoint® Support personnel.
Unaccepted_Card_Type	13	Indicates the Make A Payment request was rejected because the card type used is not valid for this application. PayPoint® enables only cards being accepted by your Payment Processor.
Payment_Exceeds_System_Limit	14	PayPoint® has the ability to set a daily limit on

## PayPoint® Merchant Integration Guide

Return Code	Numeric ReturnCode	Description
		payments received and processed for a given account. This limit is determined at the time you fill out your PayPoint® application. The default setting is unlimited. Unless you explicitly request a limitation on your PayPoint® application you will never see this return code.
Payment_Exceeds_Allowable_Limit	15	PayPoint® has the ability to set a single payment limit on payments received and processed for a given account. This limit is determined at the time you fill out your PayPoint® application. The default setting is unlimited. Unless you explicitly request a limitation on your PayPoint® application you will never see this return code.
Possible_Duplicate_Payment	16	PayPoint® has the ability to track duplicate payments received and can reject them as a part of the Make A Payment request. This feature is determined at the time you fill out your PayPoint® Application. The default setting is set to not check for duplicates. Unless you explicitly request a duplicate payment check option on your PayPoint® application you will never see this return code.
Undefined_Item	18	Prior to processing a request within PayPoint® the data you send is run through a data validation process. This process checks to ensure that your request conforms to the specification. For example that you have provide all required data elements in your request, or that you are passing proper data types (i.e. not passing characters in numeric values). If your request fails this validation process PayPoint® will reject your request with this return code. The Result Message will provide more specific details on what data failed validation.
Chargeback	19	Indicates that the payment was successfully charged back.
Chargeback_Reversal	20	Indicates that a charge back request was successfully reversed.
Settlement_Incomplete	21	Catch all error for technical problems encountered during settlement processing. You should never see this result code. If this error is ever seen contact PayPoint® Support personnel.
Partial_Settlement	22	Anytime there are multiple payments actions under a single transaction where one payment is settled but the other has not you will see a Partial_Settlement Result Code. For example if you have a primary payment and a convenience fee payment under the same transaction and only the primary payment has been issued for settlement. Another possible example is a E-Check payment which is refunded, the original

## PayPoint® Merchant Integration Guide

Return Code	Numeric ReturnCode	Description
		payment would be settled but the refund may not have settled yet.
Settlement_Pending	23	This result code is specific to E-Check payments which have a longer settlement process than Credit Cards. This result code means a payment has been sent off for settlement, but we have not received back confirmation of the settlement result. Settlement results within ACH are typically updated as a result of no negative activity within the first 6 days from the settlement issuance.
eChecks_Disabled	24	Indicates that the application you are attempting to issue a E-Check payment action against is currently not enabled for E-Checks. If you wish to start accepting E-Checks contact PayPoint® Support personnel.
Missing_Identification	25	The result code is specific to E-Checks. PayPoint® provides fraud detection services. PayPoint® provides the ability to enable identity verification services for an additional cost. These services are offered to allow clients to be NACHA compliant with consumer identity requirements. If enabled this result code indicates that the payment is being rejected because you did not provide the required identity data such as Drivers License Number or SSN.
Waiting_On_PreNote	26	This result code is specific to E-Checks. PayPoint® supports the ability to require the issuance of Pre-Notes for registered accounts. If your application is enabled to require Pre-Notes it may also be set to require a successful pre-note before accepting payments. You will receive this result code for any payments received prior to completion of the pre-note process for a given registered account.
PreNote_Failed	27	This result code is specific to E-Checks. PayPoint® supports the ability to require the issuance of Pre-Notes for registered accounts. If your application is enabled to require Pre-Notes it may also be set to require a successful pre-note before accepting payments. Indicates that the Pre-Note request for a registered account has failed. No Payments can be applied against the registered account until the account information is updated and a new pre-note is issued by PayPoint®. You will receive this result code for any payments received prior to correcting the registration information and successful issuance of the pre-note.
Stop_Payment_Issued	28	This result code is specific to E-Checks. Consumers have the right to issue a stop payment up to 60 days after making an original payment. This result code indicates that we

## PayPoint® Merchant Integration Guide

Return Code	Numeric ReturnCode	Description
		received a stop payment request through the ACH network and have reversed the original payment transaction within PayPoint® and a reversal has been issued against the merchants account.
Non_Sufficient_Funds	29	This result code is specific to E-Checks. This indicates that a E-Check which was issued for settlement resulted in a Non-Sufficient Funds return. Depending on your configuration the payment will be re-presented through the ACH network up to 2 additional times. If you see this message it indicates 1 of 2 possible attempts. If after re-presentation the payment still results in Non-Sufficient funds in the consumers account a Final_Non_Sufficient_Funds result code is returned.
Final_Non_Sufficient_Funds	30	This result code is specific to E-Checks. This indicates that the consumers account has insufficient funds to process the original payment request. As a result PayPoint® will reverse the transaction. In addition a reversal of the original payment request will be issued against the merchants account.
Account_Invalid	31	This result code is specific to E-Checks. When a E-Check payment request is received PayPoint® will perform a basic check of the account and routing numbers provided against known data sources such as Thomson account files to ensure the account number is valid. However after a payment is issued for settlement their can be other conditions that result in invalidating the use of the account number. One example is consumers who may have debit blocks on their accounts which would result in a denial by the consumers back to allow the debit to take place. If a Invalid account result is receive PayPoint® will reverse the transaction within PayPoint®. In addition a reversal of the original payment request will be issued against the merchants account.
Payment_Pending	32	Indicates that the original Payment was issued with a post dated payment. PayPoint® only support post dated transactions for E-Check payments. Once the payment date is reached the payment request will be sent through the normal payment authorization and settlement processing.
Post_Date_Too_Large	33	This result code is specific to E-Checks. This result indicates that at the time the Make A Payment request sent the payment was posted dated beyond the acceptable limits. PayPoint® can support post dated payments up to 365 days.

## PayPoint® Merchant Integration Guide

Return Code	Numeric ReturnCode	Description
		The default value is to not accept post dated payments. When you fill out your PayPoint® Application form you must identify the number of days your application will accept posted dated payments.
Refund_Settlement_Pending	34	E-Check Only result code. This result code indicates that a E-Check Payment which is currently in Settlement_Pending status was refunded. (i.e. before the original payment fully settled through ACH). The refund stays in a Refund_Settlement_Pending status until the original payment fully settles and the refund can be settled against the original payment.
Pre_Auth_Success	35	This result code is specific to Credit Card payments and is only seen when PayPoint® is responsible for doing account verifications when new registrations are created. Account Verification is a feature that can be enabled on your PayPoint® account to make a verification request of the account data associated with the registration data being created or updated. To verify a credit card a Pre-Authorization for \$1.00 is made to verify account is valide.
Eligible	36	Result code is specific to PinLess debit cards. The result is returned as a result of a call to the PinlessDebitCheck API call. This result code indicates that the card is eligible to be processed through the PINless debit network.
Not_Eligible	37	Result code is specific to PinLess debit cards. The result is returned as a result of a call to the PinlessDebitCheck API call. This result code indicates that the card is NOT eligible to be processed through the PINless debit network.
PINLessDebit_Disabled	38	Result code is specific to PINless debit cards. This result is returned when your PayPoint® account is not enabled to process PINless debit transactions.
PINDebit_Disabled	39	Result is specific to PIN based debit cards. This result is returned when your PayPont account is not enabled to process PIN debit transactions.

## 4.0 PAYMENT POSTING FILE

PayPoint® provides a batch interface for transmitting daily payment activity. This process is referred to as the PayPoint® Posting File Process. Posting files contain details about payments, cancellations, and returns processed throughout the day by PayPoint®. These details can be used by business applications for a variety of purposes including balancing, audit checks against transactional systems, and posting to backend accounting systems. The customer will download their payment data from the secure PayPoint® FTP server.

PayPoint®

### Select Extract File Type

PayPoint® extracts payment data from the database on a nightly basis. You can specify how you want to aggregate and/or segregate the data into extract files. The following three options are available:

1. Receive a single file at the Site level. PayPoint® aggregates and reports payment data for all business applications into a single extract file.
2. Receive a separate file for each Agency (e.g. Department or Business Unit) defined under a Site. PayPoint® aggregates and reports payment data for all business applications defined under an Agency into a single extract file for that Agency.
3. Receive a separate file for each business application defined under a Site. PayPoint® aggregates and reports payment data for each business application into a separate extract file for that application.

The PayPoint® Project Manager can advise and assist you during the selection process. We will generate the appropriate site, agency, and business application identifiers and configure PayPoint® to support your selection.

### Extract File Name Standard

The PayPoint® extract file name standard uses the first 2 nodes to uniquely identify the payment data. The first node will always be the unique Site Identifier. This number is assigned during the initial PayPoint® Account Registration process. The PayPoint® Project Manager will provide this information when the initial PayPoint® Account is registered. The second node will be either the Agency identifier or the Business Application identifier. This number will either identify an agency or business application, based on the selection of the payment extract file type (see above). The format for the filename is as follows:

Site Identifier + Agency or Business Application Identifier + Year (YY) + Month (MM) + Day (DD) + Hour (HH) + Minutes (MM).

Note: If you select Payment Extract File Option 1, the Agency or Business Application Identifier node of the filename will contain zeros.

## Extract File Format Specifications

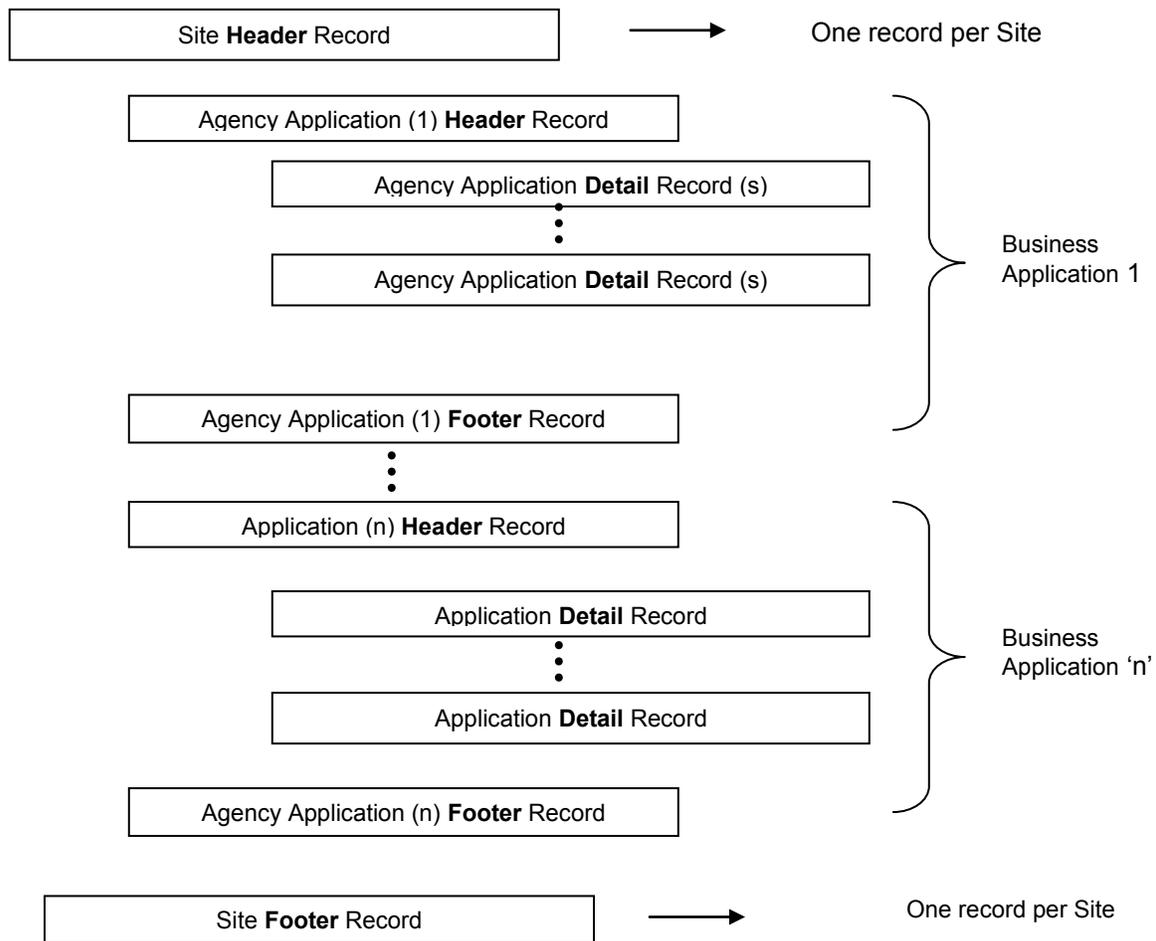
The extract file format and data specifications are listed below. The character set used in these records contains the ASCII characters A-Z, 0-9, Space, and null.

- ASCII
- Numeric Fields are **right justified** and space filled.
- Alphanumeric and Alphabetic Fields are **left justified** and space or null filled.
- Fixed length format for all fields and records
- Payment extract files are encrypted with a 256-bit AES compliant encryption algorithm. Encrypted files are placed on the PayPoint® FTP server. The tool used for encryption is PKWare. To decrypt extract files, you must have either PKWare or PKWare Reader software installed and a password assigned for your Site.

The PayPoint® Project Manager will assist or answer any questions related to the file specifications.

**Sequence of Records in Extract File**

Each extract file begins with a Site Header Record. After the Site Header, the file contains one or more sets of Agency Application records. The first record in an Agency Application set is the Agency Application header record. One or more Entity Application Detail records follow the Agency Application Header record. Each detail record contains details for a single payment processed by PayPoint®. The Application Detail records are followed by an Agency Application Footer record. This record signifies an end to the set of records for an Agency Application. After the last set of Agency Application records, the file ends with a Site Footer record. The Site Footer record contains a record count for verification that all records were transmitted successfully.



### Extract File Structure

A brief definition of each Extract File record type is listed below:

SH = Site Header

The Site Header record indicates the header record for the enterprise. There will be only one header record associated with an extract file and it will always be the first record in the extract file.

Field Name	Length	Data Type	Comments
Record Type	2	A (2)	SH
Site	13	9 (13)	Site Identifier

AH = Application Header

The Agency Application Header indicates the start of payment data associated with a specific business application within the enterprise (Site). There will be only one AH record type for each entity application associated with a given Site. Depending on the number of business applications and the extract option selected, each file may contain either one or multiple AH records.

Field Name	Length	Data Type	Comments
Record Type	2	A (2)	AH
Site	13	9 (13)	Site Identifier
Entity	13	9 (13)	Unique Number that identifies a specific Entity.
Entity Application	13	9 (13)	Unique Number that identifies a specific Entity Application.

AD = Application Detail

Each Application Detail record contains details for a single payment processed by PayPoint® for a specific Agency Application. AD records contain payment details for all types of payment activity -- including original payments, cancellations, returns, and charge backs -- processed for a given Agency Application since the last time an extract was performed.

Field Name	Length	Data Type	Comments
Record Type	2	A (2)	AD
Payment Method	1	9 (1)	1 = Credit Card (General) 2 = E-Check (TeleCheck®) 3 = PINless Debit, Credit Card, Debit (Concord) 4 = E-Check 6 = Credit Card, Debit (Vital) 7 = Third Party Payments (ACH Credit, POS, Cash)

## PayPoint® Merchant Integration Guide

Field Name	Length	Data Type	Comments
Payment ID	13	9 (13)	Unique ID assigned to a payment within PayPoint
Transaction Date	16	A (16)	MM/DD/YYYY HH:MM
Payment Code	2	9 (2)	1 = Primary Payment 2 = Convenience Fee 4 = ChargeBack Primary 5 = PreNote 6 = ChargeBack Convenience Fee
Payment Command Code	2	9 (2)	1 = Payment 3 = Sale with Pre-Auth 5 = Refund 6 = Void 9 = Chargeback 10 = Chargeback Reversal 11 = Stop Payment (E-check Only) 12 = Non Sufficient Funds (E-check Only) 13 = Invalid Account (E-check Only) 14 = Partial Refund
Payment Amount	13	9 (13)	Dollar amount for this payment. Decimal point is imbedded. If there is no decimal point entered, the field will show up as whole dollars. Ex. if you enter 15.00, then 15.00 will be in the file. If you enter 15, then 15 will be in the file.
Confirmation Number	14	A (14)	Unique PayPoint® Confirmation number.
Registered Account ID	13	9 (13)	Unique identifier assigned to a registered account. This field will only contain a value if the payment was completed using a registered account (a recurring payment, for example).
Recurring Id	13	9 (13)	Unique identifier assigned to a recurring payment schedule defined for a registered account. This field will only contain a value if this payment was generated by a recurring payment schedule.
Card Type Code	1	A(1)	2 = Visa 3 = MasterCard 4=American Express 5=Discover C=PINless Debit - Star Debit D = PINless Debit - Pulse Debit E = PINless Debit - NYCE F= Pin Based Debit
Routing Number	9	9 (9)	Bank Routing Number. This filed is only used for E-Check payments.
Account Number	9	9 (4)	Last four digits of the account number used.

## PayPoint® Merchant Integration Guide

Field Name	Length	Data Type	Comments
First Name	25	A (25)	First Name of person making the payment.
Middle Initial	1	A (1)	Middle initial of person making the payment.
Last Name	50	A (50)	Last name of person making the payment.
Account Holder Name	100	A (100)	Full name of person making the payment.
Email Address	75	A (75)	Email address for person making the payment.
Street Name 1	50	A (50)	Street name for person making the payment.
Street Name 2	50	A (50)	Secondary street name for person making the payment.
City	50	A (50)	City for person making the payment
State	2	A (2)	State for person making the payment
Zip	10	A (10)	Zip for person making the payment
Payment Channel	1	9 (1)	0 = Unknown 1=Web 2=IVR 3=Walk-in 4=Voice 5=Fax 6=Mail 7=Recurring Payment
Custom Reference Data	254	A (254)	Custom data sent by the Agency's business application.

## AF = Application Footer

The Application Footer record indicates the completion of payment details for a given business application. There will be one AF record for each application header record in a given extract file. The entity application footer will also provide record count information for a given business application.

Field Name	Length	Data Type	Comments
Record Type	2	A (2)	AF
Site	13	9 (13)	Site Identifier
Entity	13	9 (13)	Unique Number that identifies a specific Entity.
Entity Application	13	9 (13)	Unique Number that identifies a specific Entity Application.
Total Records for Entity Application	13	9 (13)	Count of all records for the Entity Application.
Total Amount of Records for Entity Application	13	9 (13)	Total Amount of all records for the Entity Application.

SF = Site Footer

The Site Footer record indicates the end-of-file for the entire extract file. There will be one SF record in a given file and it will always be the last record in the file. It will contain record total information for the entire extract file.

Field Name	Length	Data Type	Comments
Record Type	2	A (2)	SF
Site	13	9 (13)	Site Identifier
Total Records for Site	13	9 (13)	Count of all records in the file including headers and footers.

## 5.0 RESOURCES AVAILABLE

This section provides a list of resources for Electronic Payment Processing standards and best practices. The following sites offer information about e-commerce/payment issues, trends, and risks, as well as useful details about Web site privacy. The resources presented here are available through the Internet as of the publication date of this guide. Whether you have previous experience or are new to offering electronic payments, these resources may help you learn more about the e-commerce market, ensure the security of your Web site, and explore the opportunities of eCommerce.

- *The Electronic Payments Journal*. National Automated Clearing House Association (NACHA); The Electronic Payments Association. NACHA is the standards organization that oversees rules governing the ACH network. <http://www.nacha.org/>.
- *Electronics Payment Primer, 2002*. National Electronic Commerce Coordinating Council (NECCC). Government based National Electronic Commerce Coordinating Council. <http://www.ec3.org/>.
- *Federal Reserve System 12 CFR Part 205 [Regulation E Docket No. R1074]*. Federal Reserve Board. Federal Reserve Board. This regulation establishes the basic rights, liabilities, and responsibilities for consumers who use electronic funds transfer services. The primary objective of this regulation is consumer protection. <http://www.federalreserve.gov/>.



**6.0 INTEGRATION TIMELINE**

This template integration schedule is prepared by the PayPoint® Project Team as a means to outline the timeframe necessary for integrating the various PayPoint® components into an Agency business application. The PayPoint® Project Manager will assist an Agency in creating a custom timeline for the Agency's particular situation and environment. The template schedule and the recommended project organization include a number of key factors such as:

- Standard project milestones and estimated duration,
- Recommended Agency resources, and
- Roles and responsibilities.

The project schedule should be followed as closely as possible, while remaining flexible enough to accommodate adjustments as needed.

Integration Milestone	⇄Schedule⇄		
	Start Date	End Date	Notes
<b>PHASE 1</b>			
Get Merchant Account			* 2 – 3 Weeks (* to acquire new merchant number for credit card payments)
PayPoint® Account Registration			2 weeks
<b>PHASE 2</b>			
API Installation and Integration			(40 – 60 hours)
<b>PHASE 3</b>			
Test Mode			(40 - 60 hours)
<b>PHASE 4</b>			
Certification Mode			(40 – 60 hours)
Training			(4 – 8 hours)
<b>PHASE 5</b>			
Production Mode (Go Live)			(4 hours)

## 7.0 PROJECT ORGANIZATION

This section describes the minimum staffing requirements, including the major roles and responsibilities of the key individuals for each team. The project organization may evolve throughout the integration lifecycle resulting in the assignment of additional team members by either team.

Role	Responsibility
Agency Executive Sponsor	<ul style="list-style-type: none"> <li>▪ Business champion</li> <li>▪ Facilitate and expedite implementation</li> <li>▪ Issue resolution</li> </ul>
Agency Project Manager	<ul style="list-style-type: none"> <li>▪ Day-to-Day Project Management for Agency Integration</li> <li>▪ Single Point of Contact</li> <li>▪ Status Monitoring</li> <li>▪ Issue resolution</li> <li>▪ Submits request for transition to Certification Mode</li> <li>▪ Complete and submit Certification Checklist</li> <li>▪ Submits request for transition to Production Mode</li> <li>▪ Subject Matter Expert</li> </ul>
Agency Business/Financial Analyst	<ul style="list-style-type: none"> <li>▪ Business Requirements of Application</li> <li>▪ Merchant Account / Banking relationship</li> <li>▪ Testing and Validation</li> <li>▪ Certification Testing</li> <li>▪ Subject Matter Expert</li> </ul>
Agency Programmer	<ul style="list-style-type: none"> <li>▪ Develop interface with PayPoint® (API)</li> <li>▪ Testing and Validation</li> </ul>
Agency Security Specialist	<ul style="list-style-type: none"> <li>▪ Security policy</li> <li>▪ Role-based Security for PayPoint® Application</li> <li>▪ Security testing and validation</li> </ul>
PayPoint® Account Executive	<ul style="list-style-type: none"> <li>▪ Initial Contact with Client</li> <li>▪ Contract Negotiation</li> <li>▪ Contract Monitoring</li> <li>▪ On-going relationship with Client</li> </ul>
PayPoint® Project Director	<ul style="list-style-type: none"> <li>▪ Oversight of Integration Project</li> <li>▪ Support for Project Team</li> <li>▪ Risk Management Oversight</li> <li>▪ Issue Resolution</li> </ul>
PayPoint® Project Manager	<ul style="list-style-type: none"> <li>▪ Day-to-Day Project Management for PayPoint® tasks</li> <li>▪ Single Point of Contact</li> <li>▪ Risk Management</li> <li>▪ Status Reporting</li> <li>▪ Issue Resolution</li> <li>▪ Subject Matter Expert</li> <li>▪ Contract Administration and Management</li> </ul>
PayPoint® Business Analyst	<ul style="list-style-type: none"> <li>▪ Secondary Contact</li> </ul>



Role	Responsibility
	<ul style="list-style-type: none"> <li>▪ Business Requirements of Application</li> <li>▪ Testing Requirements</li> <li>▪ Training</li> <li>▪ Subject Matter Expert</li> </ul>
PayPoint® Executive Sponsor	<ul style="list-style-type: none"> <li>▪ Executive Level Project Support</li> <li>▪ Steering Committee</li> <li>▪ Interface with PayPoint® Account Executive</li> <li>▪ Contract Management</li> </ul>



## 8.0 SUPPORT SERVICES

At the conclusion of the PayPoint® implementation, each Agency will use our central support group as the primary point of contact for all support services. A Customer Support Representative will answer your questions. Questions are answered quickly and reliably thanks to our training program and dedicated staff. Issues do not linger. We have a well-defined internal escalation plan that describes in detail the path to get results when needed. The PayPoint® support contact information will be provided to you as part of your PayPoint® integration process.

If you have any questions or concerns, please contact the PayPoint® support desk.



**APPENDIX A: PAYPOINT® ACCOUNT APPLICATION**

Before you can begin processing payments with PayPoint® for your application you must complete an Account Application form. Please contact your Project Manager or Client Relationship Representative for a copy of the electronic version of the form.

**APPENDIX B: CERTIFICATION CHECKLIST**

**Instructions:** The Certification process requires an Agency to verify proper execution of their business application code, proper response and error handling, processing of return codes, and data storage within the Agency environment. The Certification Checklist must be completed by the Agency at the conclusion of the Certification phase and forwarded to the PayPoint® Project Manager. The certification tasks are listed below. Specify which certification tasks are complete or incomplete by entering the date of verification in the “Complete” and “Incomplete” columns. In addition, provide the initials of the individual who performed the verification in the “Verified By” column.

Certification Checklist for [Application ID]	Completed	Verified
Authorization using https: (secure connection)	<input type="checkbox"/>	<input type="checkbox"/>
Authorization request	<input type="checkbox"/>	<input type="checkbox"/>
Recurring authorizations, if required	<input type="checkbox"/>	<input type="checkbox"/>
Response Handling for User Error	<input type="checkbox"/>	<input type="checkbox"/>
Response Handling after successful completion of authorization	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Storage	<input type="checkbox"/>	<input type="checkbox"/>
Update Authorization, if required	<input type="checkbox"/>	<input type="checkbox"/>
Retrieve PayPoint® Extract (Posting) File	<input type="checkbox"/>	<input type="checkbox"/>

**APPENDIX C: ACRONYMS AND DEFINITIONS**

In the course of this document, several terms related to electronic payment processes and the PayPoint® solution are used. Definitions are as follows:

Term	Definition
<b>ACH</b>	Automated Clearing House. A funds transfer system governed by the rules of the National Automated Clearing House Association which provides for the interbank clearing of electronic entries for participating financial institutions. This will typically be the Federal Reserve.
<b>Acquirer</b>	A financial institution, or third-party processor, authorized to accept credit card transactions from merchants and enter the transactions into interchange for settlement. The acquirer may also be called a Merchant Bank or Merchant Services Provider.
<b>Authorization</b>	An authorization indicates the availability of the cardholder's credit at the time the authorization is requested.
<b>Authorization (E-Check)</b>	The process by which an E-Check processor approves a transaction. Authorization of an E-Check transaction does not guarantee that funds will be available when the transaction is settled.
<b>AVS</b>	Address Verification Service. This service compares the billing address on record with the billing address provided by the cardholder completing a transaction.
<b>Card Issuer</b>	Financial organization authorized by a card association to issue credit or debit cards to individual cardholders.
<b>Chargeback</b>	A credit card transaction that is returned as a financial liability to the Acquirer (merchant bank) by the Card Issuer, usually because of a disputed transaction. The Acquirer may return the transaction to the merchant and debit the merchant account. Chargebacks are normally the result from the failure to adhere to card association regulations and/or operating procedures.
<b>Clearing</b>	The process of exchanging financial transaction details between Acquirers and a Card Issuers (for credit card transactions) or Originating Banks and Receiving Banks (for ACH transactions) to facilitate settlement of payment transactions.
<b>Credit Card</b>	A card that enables the cardholder to purchase goods or services against a line of credit established by the Card Issuer.
<b>CVV2</b>	Card Verification Value 2 for Visa (CVC2 for MasterCard and CID for American Express). A 3- or 4-digit verification identifier that is typically found on the back of credit cards and used to verify the presence of the card in a non face-to-face transaction.
<b>FTP</b>	File Transfer Protocol. A standard Internet protocol that provides a simple way to exchange files between computers on the Internet. Like the Hypertext Transfer Protocol (HTTP), which transfers displayable Web pages and related files, and the Simple Mail Transfer Protocol (SMTP), which transfers e-mail, FTP is an application protocol that uses the Internet's TCP/IP protocols. It is commonly used to download programs and other files.