

# PayFac™ API Reference Guide

June 2018

API Release: 13.0

Document Version: 2.5

PayFac API Reference Guide Document Version: 2.5

All information whether text or graphics, contained in this manual is confidential and proprietary information of Worldpay, Inc. and is provided to you solely for the purpose of assisting you in using a Worldpay, Inc. product. All such information is protected by copyright laws and international treaties. No part of this manual may be reproduced or transmitted in any form or by any means, electronic, mechanical or otherwise for any purpose without the express written permission of Worldpay, Inc.. The possession, viewing, or use of the information contained in this manual does not transfer any intellectual property rights or grant a license to use this information or any software application referred to herein for any purpose other than that for which it was provided. Information in this manual is presented "as is" and neither Worldpay, Inc. or any other party assumes responsibility for typographical errors, technical errors, or other inaccuracies contained in this document. This manual is subject to change without notice and does not represent a commitment on the part Worldpay, Inc. or any other party. Worldpay, Inc. does not warrant that the information contained herein is accurate or complete.

All trademarks are the property of their respective owners and all parties herein have consented to their trademarks appearing in this manual. Any use by you of the trademarks included herein must have express written permission of the respective owner.

Copyright © 2003-2018, Worldpay, Inc. ALL RIGHTS RESERVED.

---

# CONTENTS

---

## About This Guide

Intended Audience .....	ix
Revision History .....	ix
Document Structure .....	xiii
Documentation Set .....	xiii
Typographical Conventions .....	xiv
Contact Information.....	xv

## Chapter 1 Introduction

Legal Entity Processing .....	3
API Summary .....	5
Header Information .....	7

## Chapter 2 PayFac API Certification Tests

Testing Legal Entity Transactions .....	10
Creating Legal Entities .....	10
Creating Canadian Legal Entities.....	11
Updating Legal Entities .....	12
Updating Canadian Legal Entities.....	14
Retrieving Legal Entity Information .....	15
Retrieving Canadian Legal Entity Information.....	16
Testing Sub-Merchant Transactions .....	18
Creating Sub-Merchants .....	18
Creating Canadian Sub-merchants.....	19
Updating Sub-Merchant information .....	21
Updating Canadian Sub-Merchant Information.....	22
Retrieving Sub-Merchant Information .....	24
Retrieving Canadian Sub-Merchant Information .....	25
Retrieve Allowed MCC List .....	25

## Chapter 3 PayFac API Transaction Examples

Create Legal Entity Agreement.....	28
Create Legal Entity Agreement Request Message .....	28
Create Legal Entity Agreement Response Message .....	29
Legal Entity Agreement Retrieval Request .....	30
Legal Entity Agreement Retrieval Response .....	30
Create Legal Entity .....	33
Notes About Background Checks .....	33

Notes About Required Fields .....	34
Create Legal Entity Request Message.....	35
Create Legal Entity Response Message.....	36
Update Legal Entity.....	41
Update Legal Entity Request .....	42
Update Legal Entity Response.....	43
Retrieve Legal Entity .....	47
Retrieve Legal Entity Response.....	47
Create a Principal .....	52
Principal Create Request .....	52
Principal Create Response .....	53
Delete a Principal.....	55
Delete a Principal Response.....	55
Create Sub-Merchant.....	57
Create Sub-Merchant Request .....	57
Create Sub-Merchant Response.....	61
Update Sub-Merchant.....	64
Update a Sub-Merchant.....	65
Update a Sub-Merchant Response.....	66
Retrieve Sub-Merchant .....	67
Retrieve Sub-Merchant Response .....	67
Retrieve Allowed MCC List .....	71
Retrieve MCC List Response.....	71
Status Codes and Error Messages .....	73
Error Response Message .....	73

## Chapter 4 PayFac API XML Elements

acceptanceDateTime .....	76
address .....	77
addressVerified .....	78
agreements .....	79
agreementVersion.....	80
amexAcquired .....	81
amexMid .....	82
amexSellerId .....	83
annualCreditCardSalesVolume.....	84
approvedMcc .....	85
approvedMccResponse .....	86
approvedMccs.....	87
backgroundCheckDecisionNotes.....	88
backgroundCheckFields .....	89

backgroundCheckResults .....	90
bankAccountNumber .....	91
bankRoutingNumber .....	92
bankruptcyCount .....	93
bankruptcyData .....	94
bankruptcyType .....	95
business .....	96
businessToPrincipalAssociation .....	98
city .....	99
cityVerified .....	100
code .....	101
companyName .....	108
contactPhone .....	109
countryCode .....	110
createCredentials .....	111
credentials .....	112
customerServiceNumber .....	113
dateOfBirth .....	114
decisionDate .....	115
description .....	116
disable .....	117
disabled .....	118
discoverConveyedMid .....	119
dobVerified .....	120
doingBusinessAs .....	121
driversLicense .....	122
driversLicenseState .....	123
eCheck .....	124
eCheckBillingDescriptor .....	125
eCheckCompanyName .....	126
emailAddress .....	127
error .....	128
errorResponse .....	129
errors .....	130
feeProfile .....	131
filingDate .....	132
firstName .....	133
fraud .....	134
fundingSubmerchantId .....	135
hardCodedBillingDescriptor .....	136
hasAcceptedCreditCards .....	137

lastName .....	138
legalEntityAgreement .....	139
legalEntityAgreementCreateRequest .....	140
legalEntityAgreementCreateResponse .....	141
legalEntityAgreementRetrievalResponse .....	142
legalEntityAgreementType .....	143
legalEntityCreateRequest .....	144
legalEntityCreateResponse .....	145
legalEntityId .....	146
legalEntityName .....	147
legalEntityOwnershipType .....	148
legalEntityPrincipalCreateRequest .....	149
legalEntityPrincipalCreateResponse .....	150
legalEntityPrincipalDeleteResponse .....	151
legalEntityResponse .....	152
legalEntityRetrievalResponse .....	153
legalEntityType .....	154
legalEntityUpdateRequest .....	155
lienResult .....	156
lienType .....	157
manuallyEntered .....	158
maxTransactionAmount .....	159
merchantCategoryCode .....	160
merchantIdentString .....	161
merchantName .....	162
nameAddressPhoneAssociation .....	163
nameAddressSsnAssociation .....	164
nameAddressTaxIdAssociation .....	165
nameVerified .....	166
originalLegalEntityId .....	167
originalLegalEntityStatus .....	168
originalSubMerchant .....	169
overallScore .....	170
password .....	171
passwordExpirationDate .....	172
paypageCredential .....	173
paypageCredentials .....	174
paypageId .....	175
phone .....	176
phoneVerified .....	177
postalCode .....	178

primaryContact.....	179
principal.....	180
principalId.....	185
pspMerchantId .....	186
purchaseCurrency.....	187
releasedCount.....	188
response .....	189
responseCode.....	190
responseDescription .....	191
riskIndicator.....	192
riskIndicators.....	193
score .....	194
settlementCurrency.....	196
ssn .....	197
ssnVerified .....	198
stakePercent.....	199
state .....	200
stateVerified .....	201
stateProvince .....	202
streetAddress1, streetAddress2.....	203
subMerchantCreateRequest .....	204
subMerchantCreateResponse .....	205
subMerchantFunding .....	206
subMerchantId .....	207
sub_merchant_processing_status .....	208
subMerchantRetrievalResponse .....	209
subMerchantUpdateRequest .....	210
taxAuthority .....	211
taxAuthorityState.....	212
taxId .....	213
taxIdVerified .....	214
tinValidationStatus .....	215
title .....	216
transactionId .....	217
unreleasedCount.....	218
updateDate .....	219
url .....	220
userFullName.....	221
userIPAddress .....	222
userName .....	223
userSystemName .....	224

verificationIndicators .....	225
verificationResult.....	227
yearsInBusiness.....	230
zip .....	231
zip4 .....	232
zipVerified .....	233



# ABOUT THIS GUIDE

This manual serves as a reference to the PayFac Merchant Provisioner API. The guide provides information about the transaction formats used to create, update, and retrieve (information about) Legal Entities and Sub-Merchants. It also explains how to perform PayFac API certification testing with us.

## Intended Audience

This document is intended for technical personnel creating code to interface with the Vantiv eCommerce Merchant Provisioner API.

## Revision History

This document has been revised as follows:

**TABLE 1** Document Revision History

Doc. Version	Description	Location(s)
1.0	Initial Release of Document Note: This guide replaces the <i>PSP On-boarding API Quick Reference Guide</i> and the <i>PSP On-boarding API Certification Tests Guide</i> .	N/A
1.1	Updated for schema version 3.0. Added text to clarify some points raised by merchants and fixed errors in version 1.0. Added info about new <code>pspMerchantId</code> element.	All
1.2	Added / to calls in Table 1-1 and added section/flow chart about Legal Entity processing. Fixed typos in examples.	Chapter 1 Chapter 3

**TABLE 1** Document Revision History

Doc. Version	Description	Location(s)
1.3	Book name change - Merchant Provisioner to PayFac. Added information about behavior when duplicate Create Legal Entity and Create Sub-merchant transactions are submitted. Also added information about new elements added for duplicate behavior.	All All
1.4	Added note about error in schema - username vs. userName	Chapters 3 & 4
1.5	Updated document to reflect Version 5 of the schema.	All
1.6	Changed Flow charts in section 1.1. Minor corrections in various locations.	Chapter 1 All
1.7	Updated for Schema Version 6.0	Chapters 2, 3 and 4
1.8	Added Note/text to <code>hardCodedBillingDescriptor</code> and <code>amount</code> element definition to provide additional details. Also, fixed missing cross-reference for <code>purchaseCurrency</code> from <code>subMerchantretrievalResponse</code> .	Chapter 4.
1.9	Updated document for schema V7.0 - Removed <code>PCI</code> element and children; added <code>eCheck</code> element.	Chapters 3 & 4
1.10	Updated document for schema V8.0 - Added elements to designate the Fee Profile for PayFacs that use Sub-merchant Funding feature.	Chapter 4
1.11	Removed <code>&lt;PCI&gt;</code> section from one <code>updateLegalEntity</code> example.	Chapter 3
1.12	Removed reference to using DELETE method (to disable a Sub-merchant) from Table 1-1 Corrected definition of <code>primaryContact</code> element and its child elements.	Chapter 1 Chapter 4
1.13	Updated document to reflect schema V9.0 changes.	All
1.14	Updated document to reflect schema V10.0 changes. Added Canadian PayFac Cert test information.	All Chapter 2
1.15	Removed all references to the <code>legalEntityPrincipalUpdatable</code> transaction type. This should have been removed in V1.13. Removed erroneous steps shown for testing retrieving Legal Entity Cert tests.	Chapters 1, 2, 3, and 4 Chapter 2

**TABLE 1** Document Revision History

Doc. Version	Description	Location(s)
1.16	Updated doc for V10.1. Replaced Update Sub-merchant Response example. Fixed error in Update Legal Entity Request example. Fixed error in when <code>taxId</code> element is required. Re-branded document.	Chapters 3 & 4 Chapter 3 Chapter 3 Chapters 3 & 4 All
1.17	Added text to clarify use of <code>taxId</code> and <code>ssn</code> elements.	Chapter 3 & 4
1.18	Corrected note on use of <code>fundingSubmerchantId</code> element.	Chapter 4
1.19	Updated doc for schema V11.	Chapters 3 & 4
1.20	Miscellaneous updates and corrections.	Chapters 3 & 4
1.21	Added information about 2 new transaction types. Corrected LE updatable items list. Corrected an XML tag in the Update Legal Entity Response example.	Chapters 3 & 4 Chapter 2 Chapter 3
1.22	Removed references to Funds In/Out report Renamed Instruction-Based Dynamic Payout to Dynamic Payout and Profile-Based Dynamic Payout to Managed Payout.	All All
1.23	Clarified use/origin of <code>fundingSubmerchantId</code> element. Corrected errors (missing elements) in <code>subMerchantRetrievalResponse</code> structure and examples.	Chapter 4 Chapters 3 & 4
1.24	Added Note about the minimum wait time (two minutes) between creating a Sub-merchant and the ability for that Sub-merchant to process transactions.	Chapters 3 & 4
1.25	Miscellaneous corrections to text and examples.	All
1.26	Added element to support use of AmEx Opt Blue.	Chapters 3 & 4
1.27	Corrected structure/example of <code>createLegalEntityRequest</code>	Chapter 3
1.28	Added information and elements associated with V12 of the API. This version allows provisioning of multiple principals for a Legal Entity.	All
1.29	Fixed an error in the Notes about Required Fields section of Chapter 3.	Chapter 3
1.30	Fixed error in <code>legalEntityRetrievalResponse</code> example.	Chapter 3

**TABLE 1** Document Revision History

Doc. Version	Description	Location(s)
1.31	Fixed error in the structure of <legalEntityResponse>. Fixed error in the definition of <merchantIdString>. It was defined as Type = Long, but should have been Type = String.	Chapter 3 Chapter 4
1.32	Fixed miscellaneous typos in examples and definitions.	Chapters 3 & 4
1.33	Fixed Create and Update Legal Entity examples, as well as Legal Entity Retrieval example. All errors associated with the yearsInBusiness element.	Chapter 3
1.34	Changed response message for Test #10. Added Tests #2a and #2b Added Create Sub-merchant examples for DP and MP. Changed url element from required to optional. Update for V12.1, including new elements for MCC 9311.	Chapter 2 Chapter 2 Chapter 3 Chapter 4 Chapters 3 & 4
1.35	Added Note to title element - required if using AmEx Opt Blue.	Chapter 4
1.36	Create Sub-merchant examples were missing closing tag for </submerchantFunding>	Chapter 3
1.37	Replace LitleXML with cnpAPI.	All
2.0	Update for V13.0 of schema, which changes the namespace, as well as the Content-Type and Accept information in the HTTPS headers.	Chapters 1, 3, and 4
2.1	Added Notes specifying that you must designate the controlling Principal when creating a Legal Entity, and also that you must create additional Principals for owners with a 10% or greater interest in the Legal Entity.	Chapters 3 and 4
2.2	Changed “four Principals” to “eleven Principals” in section 3.7.	Chapter 3
2.3	Fixed miscellaneous typos. Removed all references to the legalEntityPrincipalUpdatable transaction type.	All Chapter 4
2.4	Correction: <url> element listed as optional, but required. Added info to amexMid note to specify it is not required if processing American Express Opt Blue. Also, fixed examples to reflect to include amexMid when amexAcquired enabled="false" and not include amexMid when amexAcquired enabled="true".	Chapter 4 Chapters 3 and 4
2.5	Changed the required fields for Legal Entity Principals.	Chapters 3 and 4

# Document Structure

This manual contains the following sections:

## Chapter 1, "Introduction"

This chapter provides an overview of PayFac Merchant Provisioner API functionality.

## Chapter 2, "PayFac API Certification Tests"

This chapter guides you through testing and certification process required prior to using the PayFac Merchant Provisioner API in the production environment.

## Chapter 3, "PayFac API Transaction Examples"

This chapter provides information concerning the required XML structure, as well as example of the available PayFac Merchant Provisioner API transaction messages.

## Chapter 4, "PayFac API XML Elements"

The chapter provides definitions of each element available in the PayFac Merchant Provisioner API.

# Documentation Set

Our documentation set also include the items listed below, as well as many other InfoSheets and FAQ documents. Please refer to the appropriate guide for information concerning other product offerings.

- *Vantiv PayFac™ Portal User Guide*
- *Vantiv cnpAPI Reference Guide*
- *Vantiv iQ Reporting and Analytics User Guide*
- *Vantiv PayFac™ Integration Overview (eComm)*
- *Vantiv PayPal Integration Guide*
- *Vantiv Chargeback API Reference Guide*
- *Vantiv Chargeback Process Guide*
- *Vantiv eProtect™ Integration Guide*
- *Vantiv Enterprise eProtect™ Integration Guide*
- *Vantiv cnpAPI Differences Guide*
- *Vantiv Scheduled Secure Reports Reference Guide*
- *Vantiv Account Updater Implementation and Operations Guide (Legacy)*

# Typographical Conventions

Table 2 describes the conventions used in this guide.

**TABLE 2** Typographical Conventions

Convention	Meaning
. . . . . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
. . .	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted.
< >	Angle brackets are used in the following situations: <ul style="list-style-type: none"><li>• user-supplied values (variables)</li><li>• XML elements</li></ul>
[ ]	Brackets enclose optional clauses from which you can choose one or more option.
<b>bold text</b>	Bold text indicates emphasis.
<i>Italicized text</i>	Italic type in text indicates a term defined in the text, the glossary, or in both locations.
<a href="#">blue text</a>	Blue text indicates a hypertext link.

## Contact Information

This section provides contact information for organizations within Vantiv.

**Implementation** - For technical assistance to resolve issues encountered during the on-boarding process, including certification testing.

### Implementation Contact Information

<b>E-mail</b>	<a href="mailto:implementation@vantiv.com">implementation@vantiv.com</a>
<b>Hours Available</b>	Monday – Friday, 8:30 A.M.– 5:30 P.M. EST

**Chargebacks** - For business-related issues and questions regarding financial transactions and documentation associated with chargeback cases, contact the Chargebacks Department.

### Chargebacks Department Contact Information

<b>Telephone</b>	1-844-843-6111 (option 4)
<b>E-mail</b>	<a href="mailto:chargebacks@vantiv.com">chargebacks@vantiv.com</a>
<b>Hours Available</b>	Monday – Friday, 7:30 A.M.– 5:00 P.M. EST

**Technical Support** - For technical issues such as file transmission errors, email Technical Support. A Technical Support Representative will contact you within 15 minutes to resolve the problem.

### Technical Support Contact Information

<b>E-mail</b>	<a href="mailto:eCommerceSupport@vantiv.com">eCommerceSupport@vantiv.com</a>
<b>Hours Available</b>	24/7 (seven days a week, 24 hours a day)

**Relationship Management/Customer Service** - For non-technical issues, including questions concerning iQ Reporting and Analytics, help with passwords, modifying merchant details, and changes to user account permissions, contact the Relationship Management/Customer Service Department. If you are a Payment Facilitator (PayFac), refer to the second table.

### Relationship Management/Customer Service Contact Information - Merchants

<b>Telephone</b>	1-844-843-6111 (Option 3)
<b>E-mail</b>	<a href="mailto:ecc@vantiv.com">ecc@vantiv.com</a>
<b>Hours Available</b>	Monday – Friday, 8:00 A.M.– 6:00 P.M. EST

## Relationship Management/Customer Service Contact Information - Payment Facilitators

<b>Telephone</b>	1-844-843-6111 (Option 5)
<b>E-mail</b>	<a href="mailto:PayFacEComm@vantiv.com">PayFacEComm@vantiv.com</a>
<b>Hours Available</b>	Monday – Friday, 8:00 A.M.– 5:00 P.M. EST

**Technical Publications** - For questions or comments about this document, please address your feedback to the Technical Publications Department. All comments are welcome.

## Technical Publications Contact Information

<b>E-mail</b>	<a href="mailto:TechPubs@vantiv.com">TechPubs@vantiv.com</a>
---------------	--

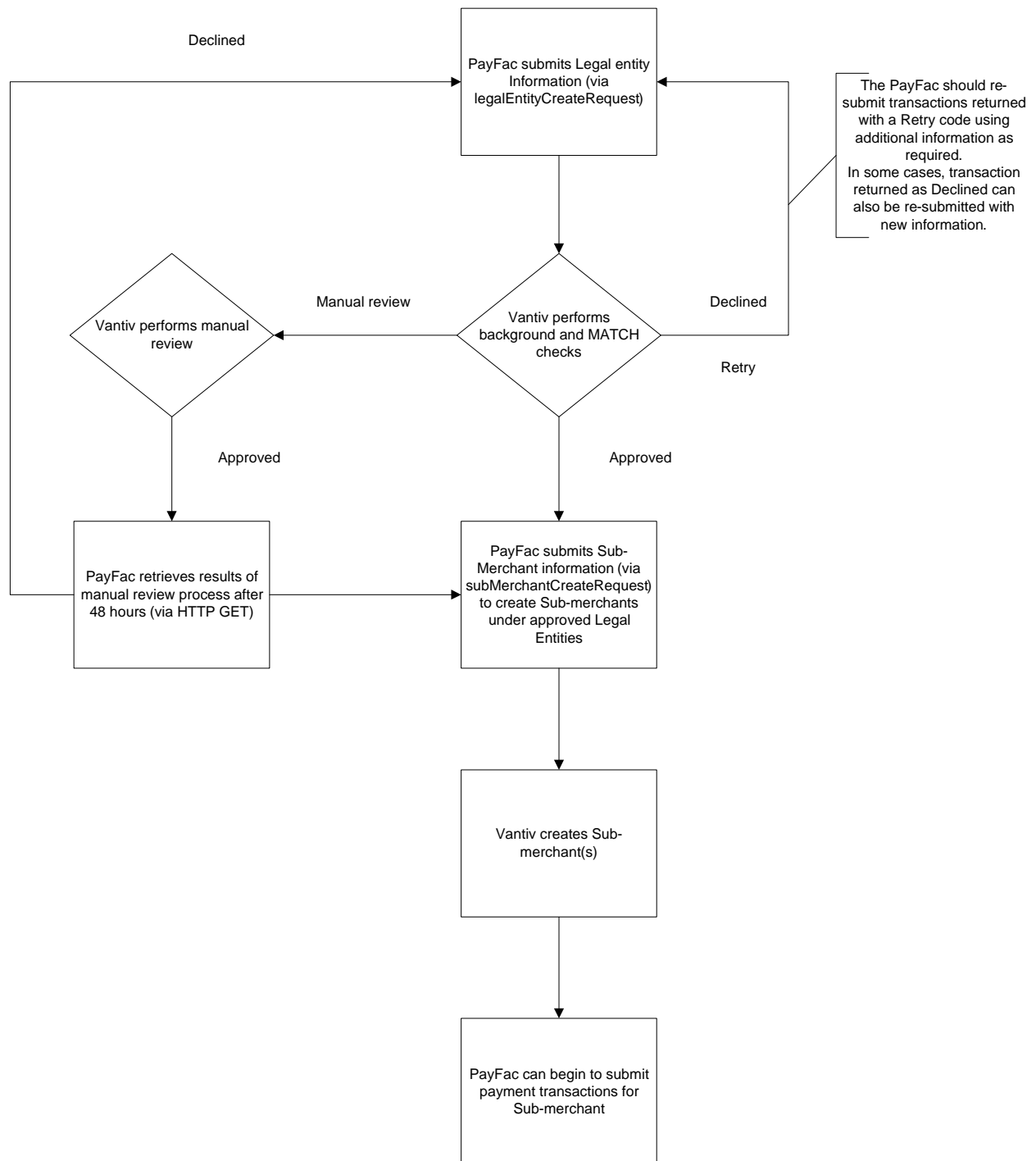


---

## INTRODUCTION

This document provides an overview of the PayFac API. This is an XML based, RESTful API that allows you to create and update Legal Entities and Sub-merchants, as well as retrieve information about existing Legal Entities and Sub-merchants in near real-time. Additionally, you can disable Sub-merchants should the need arise.

As shown in [Figure 1-1](#), before you can begin processing payment transactions for a Sub-merchant, you must first create the Legal Entity. A Legal Entity describes the parent of one or more Sub-merchants. After the successful creation of a Legal Entity, you create Sub-merchants under that Legal Entity.

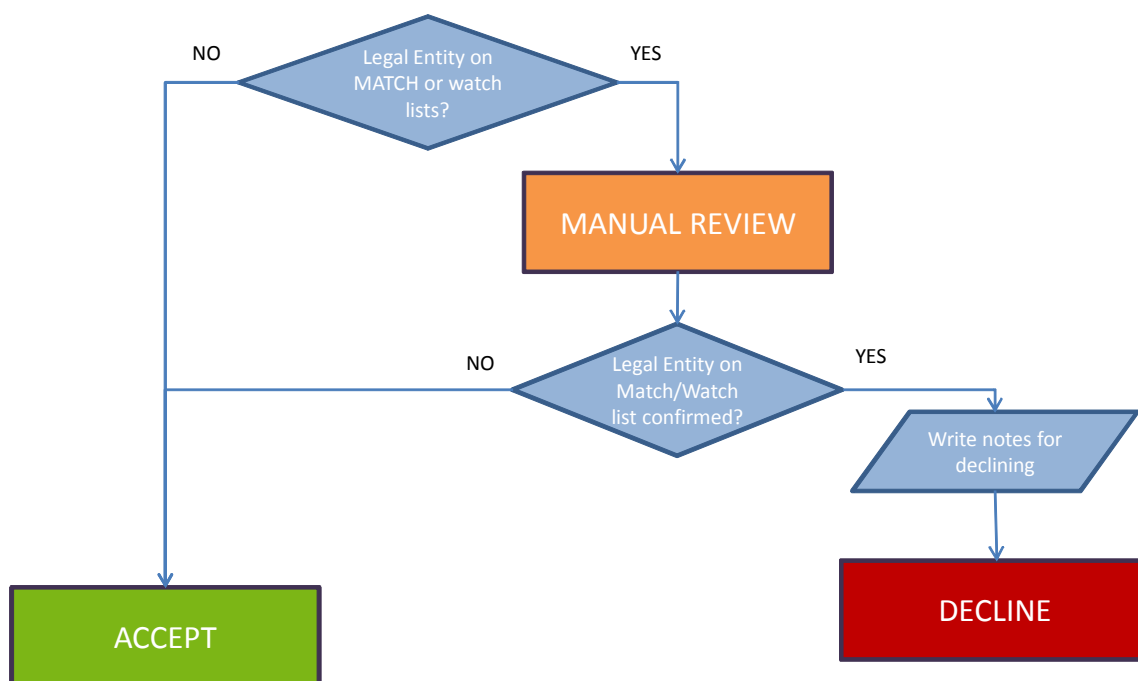
**FIGURE 1-1** Legal Entity and Sub-merchant Creation Flow

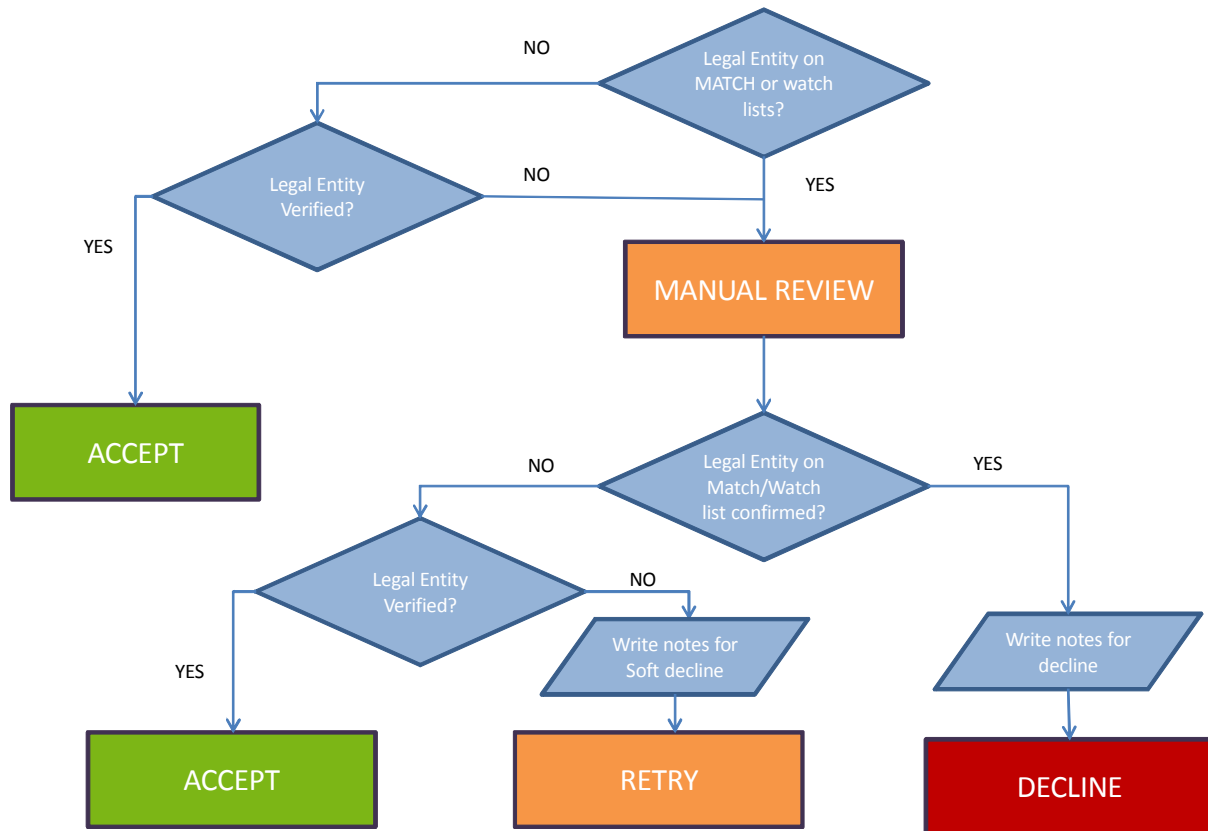
## 1.1 Legal Entity Processing

When you use the PayFac API to create a Legal Entity, add a Principal to a Legal Entity, and/or create a Sub-merchant, the system performs a series of background checks. The extent and type of check performed depend upon your needs and contract. There are three levels of service available: Basic, Premium, and Premium Plus.

As illustrated in [Figure 1-2](#) and [Figure 1-3](#), depending upon your service level, the result of the background checks, and subsequent analysis if necessary, there are three possible outcomes: Accept, Decline, or Soft Decline/Retry

**FIGURE 1-2** Legal Entity Acceptance Flow - Basic and Premium



**FIGURE 1-3** Legal Entity Acceptance Flow - Premium Plus

## 1.2 API Summary

You use the following methods to create, retrieve, update, or disable Sub-merchants, as well as list Merchant Category Codes (MCC). The associated data is submitted in various XML messages, the structure of which is defined in the other sections of this document.

---

**NOTE:** When creating a URI for a particular method, replace any **Id** object with the value returned in the create response XML message. For example, if you were updating a Legal Entity and the `legalEntityId = 3` (from the `legalEntityCreateResponse` message), the update call would be: **PUT /legalentity/3**

---

**TABLE 1-1** API Methods Summary (replace **bold** items with values)

Resource	Description
POST /legalentity/ <b>legalEntityId</b> /agreement	Request to designate the version of the agreement along with information about the agreeing party. (See <a href="#">Create Legal Entity Agreement</a> on page 28)
GET /legalentity/ <b>legalEntityId</b> /agreement	Request to retrieve all agreements associated with a specified Legal Entity. (See <a href="#">Legal Entity Agreement Retrieval Request</a> on page 30)
POST /legalentity	Request to create a new Legal Entity. (See <a href="#">Create Legal Entity</a> on page 33)
PUT /legalentity/ <b>legalEntityId</b>	Request to update data associated with the designated Legal Entity. (See <a href="#">Update Legal Entity</a> on page 41)
GET /legalentity/ <b>legalEntityId</b>	Request for information and/or status of the designated Legal Entity. Typically this is use to check status of a Legal Entity in Manual Review. (See <a href="#">Retrieve Legal Entity</a> on page 47)
POST /legalentity/ <b>legalEntityId</b> /principal	Request to create a new Principal associated with a Legal Entity. (See <a href="#">Create a Principal</a> on page 52)
DELETE /legalentity/ <b>legalEntityId</b> /principal/ <b>principalId</b>	Request to delete a Principal associated with a Legal Entity. (See <a href="#">Delete a Principal</a> on page 55)
POST /legalentity/ <b>legalEntityId</b> /submerchant	Request to create a Sub-merchant associated with the designated Legal Entity. (See <a href="#">Create Sub-Merchant</a> on page 57)
PUT /legalentity/ <b>legalEntityId</b> /submerchant/ <b>subMerchantId</b>	Request to update data associated with the designated Sub-merchant. (See <a href="#">Update Sub-Merchant</a> on page 64)

**TABLE 1-1** API Methods Summary (replace **bold** items with values)

Resource	Description
GET /legalentity/ <b>legalEntityId</b> /submerchant/ <b>subMerchantId</b>	Request to retrieve information about the designated Sub-merchant. (See <a href="#">Retrieve Sub-Merchant</a> on page 67)
GET /mcc	Returns the list of merchant category codes that are currently approved for the Payment Service Provider. (See <a href="#">Retrieve Allowed MCC List</a> on page 71)

## 1.3 Header Information

To authenticate, you will need a merchant ID (merchant identity), username and password. We uses an HTTP Basic Authentication scheme in combination with SSL to guarantee the protection of your authentication information. You must include your username and password in the HTTP header according to the HTTP Basic Authentication (i.e. base-64 encoding) on every API method invocation. The tables below contains additional information concerning the header.

**TABLE 1-2** HTTP Request Header

Authorization	Basic (with user name and password)
Content-Type	application/com.vantivcnppayfac-v13+xml
Accept	application/com.vantivcnppayfac-v13+xml

---

**IMPORTANT:** The Content-Type and Accept may not be required for all methods. When included, the version number in Content-Type and Accept should always reflect the API version you use. For example, if you use API V14, use application/com.vantivcnppayfac-v14+xml for Content-Type and Accept.

---

**TABLE 1-3** Header Information

Description	Example/Info	Comments
Authorization	username:password = merchant1:password Base-64 encoded = bWVyY2hhbnQxOnBhc3N3b3Jk	Required for authentication purpose.

**Example: HTTP Header Example - Create Sub-merchant**

```
POST /legalentity/1/submerchant
Host: example_only.vantiv.com
Authorization: Basic bWVyY2hhbnQxOnBhc3N3b3Jk=
Content-Type: application/com.vantivcnppayfac-v13+xml
Accept: application/com.vantivcnppayfac-v13+xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
.
.
.
```





---

## PAYFAC API CERTIFICATION TESTS

You must complete a number of certification tests prior to submitting transactions to the Merchant Provisioning system. This testing process allows you to verify that your system not only submits correctly formatted transaction data, but also correctly parses the data returned to you in the response messages. To facilitate the certification process, we have established a certification environment that simulates the production environment.

During certification testing, an Implementation Consultant will guide you through each required test scenario. For each transaction type, specific data is supplied that you must use in your transactions. Use of this data allows the validation of your transaction structure/syntax, as well as the return of a response file containing known data. Please refer to [Chapter 4, "PayFac API XML Elements"](#) for element definitions and XML message structure.

---

**IMPORTANT:** The test data supplied does not account for all data fields/xml elements in a particular request. Where we do not supply data, you should provide appropriate information. You should never override your own system to enter supplied data. If you cannot enter the supplied data without overriding your system, please consult your Implementation Consultant concerning the test and how to proceed.

---

## 2.1 Testing Legal Entity Transactions

The following sections provide information used to test the creation and updating of Legal Entities, as well as the retrieval of information for already existing Legal Entities.

### 2.1.1 Creating Legal Entities

---

**NOTE:** For Canadian Legal Entity tests see section [Creating Canadian Legal Entities](#) on page 11.

---

You must create a Legal entity prior to creating Sub-merchants. Each Legal entity can have one or more associated Sub-Merchants.

To test the creation of Legal Entities:

1. Verify that your Create Legal Entity template is coded correctly (see [Create Legal Entity](#) on page 33).
2. Submit `legalEntityCreateRequest` transaction using the data provided for Test # 1, 2, and 3. You must provide the data for all other required elements (see [Notes About Required Fields](#) on page 34). The response of the simulator for these test cases is tied to the contents of the `streetAddress1` element.
3. Verify that your system parses the response file correctly and that the value are those shown in the Key Response Elements of the table.
4. Wait a minimum of two hours after submitting Test #2 and submit a Legal Entity Retrieval (see [Retrieve Legal Entity](#) on page 47), using the `legalEntityId` returned in Test #2.
5. Follow the instructions returned in the Retrieval response and submit an Update Legal Entity request. Verify the approval of the Legal Entity.

**TABLE 2-1** Create Legal Entity Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
1	<legalEntityType>	INDIVIDUAL_SOLE_PROPRIETORSHIP	<b>HTTP Status Code:</b>	201
	<streetAddress1>	900 Chelmsford St	<b>XML Response:</b>	
		<b>Note:</b> For this <code>legalEntityType</code> you must include the <code>ssn</code> element.	<responseCode>	10
			<responseDescription>	Approved

**TABLE 2-1** Create Legal Entity Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
2	<legalEntityType>  <streetAdress1>	INDIVIDUAL_SOLE_PROPRIETORSHIP  912 Chelmsford St <b>Note:</b> For this legalEntityType you must include the <code>ssn</code> element.	<b>HTTP Status Code:</b>  <b>XML Response:</b> <responseCode> <responseDescription>	201  20 Manual Review
2a	legalEntityId	Value from in Test #2.	<backgroundCheckDecisionNotes>	Notes for resubmission.
2b	legalEntityId	Value from in Test #2.	<b>HTTP Status Code:</b>  <b>XML Response:</b> <responseCode> <responseDescription>	201  10 Approved
3	<legalEntityType>  <streetAdress1>	LIMITED_LIABILITY_COMPANY  914 Chelmsford St <b>Note:</b> For this legalEntityType you must include the <code>taxId</code> element.	<b>HTTP Status Code:</b>  <b>XML Response:</b> <responseCode> <responseDescription>	201  10 Approved

## 2.1.2 Creating Canadian Legal Entities

To test the creation of Legal Entities for Canada:

- Verify that your Create Legal Entity template is coded correctly (see [Create Legal Entity](#) on page 33).
- Submit `legalEntityCreateRequest` transaction using the data provided for Test # C.1.1, C.1.2, and C.1.3. You must provide the data for all other required elements (see [Notes About Required Fields](#) on page 34). The response of the simulator for these test cases is tied to the contents of the `streetAddress1` element.
- Verify that your system parses the response file correctly and that the value are those shown in the Key Response Elements of the table.

**TABLE 2-2** Create Canadian Legal Entity Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
C.1.1	<legalEntityType> <doingBusinessAs> <streetAdress1>	CORPORATION Canada Cert Test Legal Entity A 900 Chelmsford St	<b>HTTP Status Code:</b>  <b>XML Response:</b> <responseCode> <responseDescription>	201  10 Approved
C.1.2	<legalEntityType>  <streetAdress1>	INDIVIDUAL_SOLE_PROPRIETORSHIP 900 Chelmsford St <b>Note:</b> For this legalEntityType you must include the <code>ssn</code> element.	<b>HTTP Status Code:</b>  <b>XML Response:</b> <responseCode> <responseDescription>	201  10 Approved
C.1.3	<legalEntityType>  <streetAdress1>	GENERAL_PARTNERSHIP 912 Chelmsford St <b>Note:</b> For this legalEntityType you must include the <code>taxId</code> element.	<b>HTTP Status Code:</b>  <b>XML Response:</b> <responseCode> <responseDescription>	201  20 Manual Review

## 2.1.3 Updating Legal Entities

---

**NOTE:** For Canadian Legal Entity updating tests see section [Updating Canadian Legal Entities](#) on page 14.

---

You can update the several items associated with a Legal Entity. To update information associated with a Legal Entity, you use a PUT method with the appropriate `legalEntityId` along with a `legalEntityUpdateRequest` message containing the updated information.

You can update the following items:

- streetAddress1 (entity and/or principal)
- streetAddress2 (entity and/or principal)
- city (entity and/or principal)
- stateProvince (entity and/or principal)
- postalCode (entity and/or principal)
- doingBusinessAs
- annualCreditCardSalesVolume
- hasAcceptedCreditCards
- contactPhone (entity and/or principal)
- title (principal)
- emailAddress (principal)
- firstName (principal)
- lastName (principal)
- ssn (principal)
- dateOfBirth (principal)
- driversLicense (principal)
- driversLicenseState (principal)
- legalEntityName
- legalEntityType
- taxId
- yearsInBusiness

To test the `legalEntityUpdateRequest` transaction:

1. Verify that your Update Legal Entity template is coded correctly (see [Update Legal Entity](#) on page 41).
2. Submit `legalEntityUpdateRequest` transaction using the data provided for Test # 4 and 5. You must provide the data for all other required elements.
3. Verify that your system parses the response file correctly and that the value are those shown in the Key Response Elements of the table.

**TABLE 2-3** Update Legal Entity Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
4	legalEntityId	Use the value returned in the response message from Test #1.	HTTP Status Code:	200
5	legalEntityId	Submit any invalid value	HTTP Status Code:  XML Response:  <error>	400  Error in request: Could not find requested object.

## 2.1.4 Updating Canadian Legal Entities

To test the `legalEntityUpdateRequest` transaction:

1. Verify that your Update Legal Entity template is coded correctly (see [Update Legal Entity](#) on page 41).
2. Submit `legalEntityUpdateRequest` transaction using the data provided in [Table 2-4](#). You must provide the data for all other required elements.
3. Verify that your system parses the response file correctly and that the value are those shown in the Key Response Elements of the table.

**TABLE 2-4** Update Canadian Legal Entity Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
C2.2.1	legalEntityId  <doingBusinessAs>	Use the value returned in the response message from Test #C.1.1.  Canada A	HTTP Status Code:	200
C2.2.2	legalEntityId  <principal> <address> <stateProvince>	Use the value returned in the response message from Test #C.1.1.   XX	HTTP Status Code:  XML Response:  <error>	400  Legal Entity Principal stateProvince: "XX" is not valid for Legal Entity Principal country.
C2.2.3	legalEntityId  <address> <postalCode>	Use the value returned in the response message from Test #C.1.1.   01730	HTTP Status Code:  XML Response:  <error>	400  Postal Code is not valid for country "CAN".
C2.2.4	legalEntityId  <principal> <address> <postalCode>	Use the value returned in the response message from Test #C.1.3.   01730	HTTP Status Code:  XML Response:  <error>	400  Postal Code "01730" is not valid for country "CAN".

**TABLE 2-4** Update Canadian Legal Entity Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
C2.2.5	legalEntityId  <doingBusinessAs>	Use the value returned in the response message from Test #C.1.1.  Canada Cert Test C - Update	HTTP Status Code:  XML Response:  <error>	200   Due to its current status, requested Legal Entity is not updatable
C2.2.6	legalEntityId  <backgroundCheckFields>  <taxId>	Use the value returned in the response message from Test #C.1.1.   123456789	HTTP Status Code:  XML Response:  <error>	400   Background check fields cannot be updated after background check.

## 2.1.5 Retrieving Legal Entity Information

---

**NOTE:** For Canadian Legal Entity updating tests see section [Retrieving Canadian Legal Entity Information](#) on page 16.

---

If you need to retrieve the on-boarding information related to a Legal Entity for verification purposes or to check the status of cases requiring manual intervention, you must perform a GET operation specifying the legalEntityId.

To test the retrieval of Legal Entity information use the information supplied in [Table 2-5](#).

Verify that your system parses the response file correctly and that the value are those shown in the Key Response Elements of the table.

**TABLE 2-5** Retrieve Legal Entity Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
6	legalEntityId	Use the value returned in the response message from Test #2.	<b>HTTP Status Code:</b>  <b>XML Response:</b> <responseCode> <responseDescription>	200  20 Manual Review <b>Note:</b> Response message will include other Legal Entity information submitted with Test #2.
7	legalEntityId	Submit any invalid value	<b>HTTP Status Code:</b>  <b>XML Response:</b> <error>	400  Error in request: Could not find requested object.

## 2.1.6 Retrieving Canadian Legal Entity Information

To test the retrieval of Canadian Legal Entity information use the information supplied in [Table 2-6](#).

Verify that your system parses the response file correctly and that the values returned are those shown in the Key Response Elements of the table.



**TABLE 2-6** Retrieve Legal Entity Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
C.3.1	legalEntityId	Use the value returned in the response message from Test #C.1.2.	<b>HTTP Status Code:</b>  <b>XML Response:</b> <responseCode> <responseDescription>	200   10 Approved <b>Note:</b> Response message will include other Legal Entity information submitted with Test #C.1.2.
C.3.2	legalEntityId	Submit any invalid value	<b>HTTP Status Code:</b>  <b>XML Response:</b> <error>	400   Error in request: Could not find requested object.

## 2.2 Testing Sub-Merchant Transactions

The following sections provide information used to test the creation and updating of Sub-merchants, as well as the retrieval of information for already existing Sub-merchants.

### 2.2.1 Creating Sub-Merchants

---

**NOTE:** For Canadian Sub-merchant creation tests see section [Creating Canadian Sub-merchants](#) on page 19.

---

After creating a Legal Entity, you can create one or more Sub-merchants associated with the Legal Entity. Create transactions that succeed return a `subMerchantCreateResponse` message, while transactions that fail return an `errorResponse` message.

To test the creation of Sub-merchants:

1. Verify that your Create Sub-merchant template is coded correctly (see [Create Sub-Merchant](#) on page 57).
2. Submit `subMerchantCreateRequest` transaction using the data provided for Test # 8, 9, and 10. You must provide the data for all other required elements.
3. Verify that your system parses the response file correctly and that the value are those shown in the Key Response Elements of the table.

**TABLE 2-7** Create Sub-merchant Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
8	legalEntityId	Use the value returned in the response message from Test #1.	HTTP Status Code:	201
9	legalEntityId	Submit any invalid value	HTTP Status Code:  XML Response:  <error>	400  Error in request: Could not find requested object.

**TABLE 2-7** Create Sub-merchant Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
10	legalEntityId	Use the value returned in the response message from Test #2.	<b>HTTP Status Code:</b>  <b>XML Response:</b> <div>&lt;error&gt;</div>	400  Error in request: Legal entity [legalEntityName] is in inactive state. You cannot add/update a submerchant.

## 2.2.2 Creating Canadian Sub-merchants

To test the creation of Canadian Sub-merchants:

1. Verify that your Create Sub-merchant template is coded correctly (see [Create Sub-Merchant](#) on page 57).
2. Submit `subMerchantCreateRequest` transaction using the data provided in [Table 2-8](#). You must provide the data for all other required elements.
3. Verify that your system parses the response file correctly and that the value are those shown in the Key Response Elements of the table.

**TABLE 2-8** Create Canadian Sub-merchant Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
C.4.1	legalEntityId	Use the value returned in the response message from Test #C.1.1.	<b>HTTP Status Code:</b>	201

**TABLE 2-8** Create Canadian Sub-merchant Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
C.4.2	legalEntityId  <purchaseCurrency> <settlementCurrency>	Use the value returned in the response message from Test #C.1.1.  USD CAD	HTTP Status Code:  XML Response:  <error>	400  Error in request: No processing group defined with purchaseCurrencyCode <840> and settlementCurrencyCode <124>
C.4.3	legalEntityId  <address>  <countryCode>	Use the value returned in the response message from Test #C.1.1.  USA	HTTP Status Code:  XML Response:  <error>	400  Error in request: Submerchant country code "USA" does not match Legal Entity country code "CAN"
C.4.4	legalEntityId	Use the value returned in the response message from Test #C.1.3.	HTTP Status Code:  XML Response:  <error>	400  Error in request: Legal Entity "Legal Entity Name" has not been approved
C.4.5	legalEntityId  <address>  <postalCode>	Use the value returned in the response message from Test #C.1.1.  01970	HTTP Status Code:  XML Response:  <error>	400  Postal Code "01970" is not valid for country "CAN".

## 2.2.3 Updating Sub-Merchant information

---

**NOTE:** For Canadian Sub-merchant update tests see section [Updating Canadian Sub-Merchant Information](#) on page 22.

---

You can update the several items associated with a Sub-Merchant. To update information associated with a Sub-merchant, you use a PUT method with the appropriate `legalEntityId`, `subMerchantId`, and a `subMerchantUpdatable` message containing the information to be updated.

You can update the following items:

- `url`
- `customerServiceNumber`
- `hardCodedBillingDescriptor`
- `maxTransactionAmount`
- `bankRoutingNumber`
- `bankAccountNumber`
- `streetAddress1`
- `streetAddress2`
- `city`
- `stateProvince`
- `postalCode`

To test the `subMerchantUpdatable` transaction:

1. Verify that your Update Sub-Merchant template is coded correctly (see [Update Sub-Merchant](#) on page 64).
2. Submit `subMerchantUpdatable` transaction using the data provided for Test # 11, 12, and 13. You must to provide the data for all other required elements.
3. Verify that your system parses the response file correctly and that the value are those shown in the Key Response Elements of the table.

**TABLE 2-9** Update Sub-merchant Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
11	legalEntityId	Use the value returned in the response message from Test # 1.	<b>HTTP Status Code:</b>	200
	subMerchantId	Use the value returned in the response message from Test # 8.		
12	legalEntityId	Submit any invalid value	<b>HTTP Status Code:</b>	400
	subMerchantId	Submit any valid or invalid value.	<b>XML Response:</b>  <error>	Error in request: Could not find requested object.
13	legalEntityId	Use the value returned in the response message from Test #1.	<b>HTTP Status Code:</b>	400
	subMerchantId	Submit any invalid value.	<b>XML Response:</b>  <error>	Error in request: Could not find requested object.

## 2.2.4 Updating Canadian Sub-Merchant Information

To test the `subMerchantUpdatable` transaction:

1. Verify that your Update Sub-Merchant template is coded correctly (see [Update Sub-Merchant](#) on page 64).
2. Submit `subMerchantUpdatable` transaction using the data provided in [Table 2-10](#). You must to provide the data for all other required elements.
3. Verify that your system parses the response file correctly and that the value are those shown in the Key Response Elements of the table.

**TABLE 2-10** Update Canadian Sub-merchant Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
C.5.1	legalEntityId	Use the value returned in the response message from Test # C.1.1.	<b>HTTP Status Code:</b>	200
	subMerchantId	Use the value returned in the response message from Test # C.4.1.		
C.5.2	legalEntityId	Submit any invalid value	<b>HTTP Status Code:</b>	400
	subMerchantId	Submit any valid or invalid value.	<b>XML Response:</b>  <error>	Error in request: Could not find requested object.
C.5.3	legalEntityId	Use the value returned in the response message from Test #C.1.1.	<b>HTTP Status Code:</b>	400
	subMerchantId	Submit any invalid value.	<b>XML Response:</b>  <error>	Error in request: Could not find requested object.

## 2.2.5 Retrieving Sub-Merchant Information

If you need to retrieve the on-boarding information related to a Sub-merchant for verification purposes, you must perform a GET operation specifying the `legalEntityId` and `subMerchantId`.

To test the retrieval of Sub-merchant information use the information supplied in [Table 2-11](#).

**TABLE 2-11** Retrieve Sub-merchant Information Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
14	<code>legalEntityId</code>	Use the value returned in the response message from Test # 1.	<b>HTTP Status Code:</b>	200
	<code>subMerchantId</code>	Use the value returned in the response message from Test # 8.		
15	<code>legalEntityId</code>	Submit any invalid value	<b>HTTP Status Code:</b>	400
	<code>subMerchantId</code>	Submit any valid or invalid value.	<b>XML Response:</b>  <code>&lt;error&gt;</code>	Error in request: Could not find requested object.
16	<code>legalEntityId</code>	Use the value returned in the response message from Test #1.	<b>HTTP Status Code:</b>	400
	<code>subMerchantId</code>	Submit any invalid value.	<b>XML Response:</b>  <code>&lt;error&gt;</code>	Error in request: Could not find requested object.



## 2.2.6 Retrieving Canadian Sub-Merchant Information

To test the retrieval of Canadian Sub-merchant information use the information supplied in [Table 2-12](#).

**TABLE 2-12** Retrieve Canadian Sub-merchant Information Test Data

Test #	Supplied Data Elements		Key Response Elements	
	Element	Value	Element	Value
14	legalEntityId	Use the value returned in the response message from Test # C.1.1.	<b>HTTP Status Code:</b>	200
	subMerchantId	Use the value returned in the response message from Test # C.4.1.		

## 2.2.7 Retrieve Allowed MCC List

When on-boarding a new Sub-merchant, a Payment Service Provider should only use one of the approved MCCs. To retrieve the allowed MCC list perform a GET operation with the parameter `mcc (GET/mcc)`.

Verify that you correctly parse the list returned. Please refer to [Retrieve MCC List Response](#) on page 71 for an example of the structure of the response message.



---

## PAYFAC API TRANSACTION EXAMPLES

This chapter contains examples showing the message structure used by the PayFac Merchant Provisioner API. For each message type, the chapter provides a high level overview of the message structure followed by one or more examples of request and response messages. Examples of Error messages are also provided. The following message types are discussed:

- [Create Legal Entity Agreement](#)
- [Legal Entity Agreement Retrieval Request](#)
- [Create Legal Entity](#)
- [Update Legal Entity](#)
- [Retrieve Legal Entity](#)
- [Create a Principal](#)
- [Delete a Principal](#)
- [Create Sub-Merchant](#)
- [Update Sub-Merchant](#)
- [Retrieve Sub-Merchant](#)
- [Retrieve Allowed MCC List](#)
- [Status Codes and Error Messages](#)

---

**NOTE:** Each high-level structural example shows the message parent element and its children (one element down). The parent and each child element links to the element definition page in Chapter 4. Where there is additional structure under the child element, the element definition page includes structural examples.

---

## 3.1 Create Legal Entity Agreement

You use the Create Legal Entity Agreement request transaction to designate the version of the agreement along with information about the agreeing party. You should use this transaction type only if your PayFac Account Manager directs you to do so.

To create a Legal Entity Agreement use an HTTP POST method to submit the legalEntityAgreementCreateRequest message.

**TABLE 3-1** Resource Information

HTTP Method	POST
Call	/legalentity/legalEntityId/agreement
Request Format	XML
Response Format	XML
Rate Limited	No

**TABLE 3-2** POST Parameters

Parameter	Required or Optional	Description
legalEntityId	Required	The Id of the Legal Entity

### 3.1.1 Create Legal Entity Agreement Request Message

You must structure a Create Legal Entity Agreement request as shown in the following examples.

```
<legalEntityAgreementCreateRequest>
  <legalEntityAgreement>
    <legalEntityAgreementType>MERCHANT_AGREEMENT</legalEntityAgreementType>
    <agreementVersion>Agreement Version Number</agreementVersion>
    <userFullName>Full Name of Signer</userFullName>
    <userSystemName>User Name of Signer of PayFac System</userSystemName>
    <userIPAddress>IP Address of Signer's system</userIPAddress>
    <manuallyEntered>true or false</manuallyEntered>
    <acceptanceDateTime>Date and Time Agreement Signed</acceptanceDateTime>
  </legalEntityAgreement>
</legalEntityAgreementCreateRequest>
```

### Example: Create Legal Entity Agreement Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<legalEntityAgreementCreateRequest
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard">
<legalEntityAgreement>
  <legalEntityAgreementType>MERCHANT_AGREEMENT</legalEntityAgreementType>
  <agreementVersion>agreementVersion1</agreementVersion>
  <userFullName>userFullName</userFullName>
  <userSystemName>systemUserName</userSystemName>
  <userIPAddress>196.198.100.100</userIPAddress>
  <manuallyEntered>false</manuallyEntered>
  <acceptanceDateTime>2017-02-11T12:00:00-06:00</acceptanceDateTime>
</legalEntityAgreement>
</legalEntityAgreementCreateRequest>
```

---

**NOTE:** If the value for the `<manuallyEntered>` element is false, you must include the `<userIPAddress>` element. If the value for the `<manuallyEntered>` element is true, you can omit the `<userIPAddress>` element.

---

## 3.1.2 Create Legal Entity Agreement Response Message

The structure for the response to a Create Legal Entity Agreement request has the following structure.

```
<legalEntityAgreementCreateResponse>
  <transactionId>1234567890123456789</transactionId>
</legalEntityAgreementCreateResponse>
```

---

**NOTE:** If the create request is a duplicate, the response message includes a `duplicate="true"` attribute.

If the request fails, the system returns an error response. See [Status Codes and Error Messages](#) on page 73.

---

## 3.2 Legal Entity Agreement Retrieval Request

You use the Legal Entity Agreement Retrieval request to retrieve all agreements associated with a specified Legal Entity.

To retrieve all agreements associated with a Legal Entity use an HTTP GET method to submit the legalEntityAgreementCreateRequest message.

**TABLE 3-3** Resource Information

HTTP Method	GET
Call	/legalentity/legalEntityId/agreement
Request Format	XML
Response Format	XML
Rate Limited	No

**TABLE 3-4** GET Parameters

Parameter	Required or Optional	Description
legalEntityId	Required	The Id of the Legal Entity

### 3.2.1 Legal Entity Agreement Retrieval Response

The structure of the response message is as follows:

---

**NOTE:** If there are more than one agreements associated with the Legal Entity, the Retrieval Response message will have an `agreements` element containing a `legalEntityArgeement` child for each agreement.

---

```
<legalEntityAgreementRetrievalResponse>
  <agreements>
    <legalEntityAgreement>
      <legalEntityAgreementType>MERCHANT_AGREEMENT</legalEntityAgreementType>
      <agreementVersion>Agreement Version Number</agreementVersion>
      <userFullName>Full Name of Signer</userFullName>
      <userSystemName>User Name of Signer of PayFac System</userSystemName>
      <userIPAddress>IP Address of Signer's system</userIPAdress>
```

```
<manuallyEntered>true or false</manuallyEntered>
<acceptanceDateTime>Date and Time Agreement Signed</acceptanceDateTime>
</legalEntityAgreement>
</agreements>
</legalEntityAgreementRetrievalResponse>
```

**Example: Legal Entity Agreement Response (no agreements)**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<legalEntityAgreementRetrievalResponse
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard">
  <legalEntityId>82826972905308177</legalEntityId>
  <transactionId>82826972904898673</transactionId>
  <agreements/>
</legalEntityAgreementRetrievalResponse>
```

**Example: Legal Entity Agreement Response (with agreements)**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<legalEntityAgreementRetrievalResponse
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard">
  <legalEntityId>82826972905308177</legalEntityId>
  <transactionId>46226972904898673</transactionId>
  <agreements>
    <legalEntityAgreement>
      <legalEntityAgreementType>MERCHANT_AGREEMENT</legalEntityAgreementType>
      <agreementVersion>agreementVersion1</agreementVersion>
      <userFullName>userFullName</userFullName>
      <userSystemName>systemUserName</userSystemName>
      <userIPAddress>196.198.100.100</userIPAddress>
      <manuallyEntered>false</manuallyEntered>
      <acceptanceDateTime>2017-06-11T13:00:00-05:00</acceptanceDateTime>
    </legalEntityAgreement>
    <legalEntityAgreement>
      <legalEntityAgreementType>MERCHANT_AGREEMENT</legalEntityAgreementType>
      <agreementVersion>agreementVersion4</agreementVersion>
      <userFullName>userFullNameFour</userFullName>
      <userSystemName>systemUserNameFour</userSystemName>
      <userIPAddress>199.12.15.201</userIPAddress>
      <manuallyEntered>false</manuallyEntered>
      <acceptanceDateTime>2017-06-24T02:59:30-05:00</acceptanceDateTime>
    </legalEntityAgreement>
    <legalEntityAgreement>

```

```
<legalEntityAgreementType>MERCHANT_AGREEMENT</legalEntityAgreementType>
<agreementVersion>agreementVersion3</agreementVersion>
<userFullName>userFullNameThree</userFullName>
<userSystemName>systemUserNameThree</userSystemName>
<userIPAddress>210.13.154.155</userIPAddress>
<manuallyEntered>false</manuallyEntered>
<acceptanceDateTime>2017-06-23T01:59:30-05:00</acceptanceDateTime>
</legalEntityAgreement>
<legalEntityAgreement>
  <legalEntityAgreementType>MERCHANT_AGREEMENT</legalEntityAgreementType>
  <agreementVersion>agreementVersion2</agreementVersion>
  <userFullName>userFullNameTwo</userFullName>
  <userSystemName>systemUserNameTwo</userSystemName>
  <manuallyEntered>true</manuallyEntered>
  <acceptanceDateTime>2017-06-19T23:59:30-05:00</acceptanceDateTime>
</legalEntityAgreement>
</agreements>
</legalEntityAgreementRetrievalResponse>
```



## 3.3 Create Legal Entity

Prior to creating new Sub-merchants, a PayFac (Payment Facilitator) must establish the Legal Entity. The Legal Entity is the business or taxable entity (for example, corporation, or LLC) controlling the Sub-merchant within the system. You create the Legal Entity using the `legalEntityCreateRequest` message.

To create a Legal Entity use an HTTP POST method to submit the `legalEntityCreateRequest` message.

**TABLE 3-5** Resource Information

HTTP Method	POST
Call	/legalentity
Request Format	XML
Response Format	XML
Rate Limited	No

### 3.3.1 Notes About Background Checks

As part of the Legal Entity creation process, Vantiv performs background and MasterCard MATCH checks as part of Risk Analysis and/or underwriting the Legal Entity. Depending upon the level of service configured for the Background Check operation (see [Legal Entity Processing](#) on page 3), different verification operations occur and the system returns different results in the response file.

Only when the MasterCard MATCH and/or other background checks have a positive outcome, or the Legal Entity was reviewed by a Vantiv risk analyst and approved, will you be able to register one or more Sub-merchants for the Legal Entity. You create and assign the sub-merchant through the use of the `subMerchantCreateRequest` (see [Create Sub-Merchant](#) on page 57) using the `legalEntityId` returned in the create legal entity response (see [Create Legal Entity Response Message](#) on page 36)

### 3.3.2 Notes About Required Fields

In order to create the Legal Entity and perform the background checks, there are a number of required fields in the `legalEntityCreateRequest` message. Please note, the required fields vary according to the `legalEntityType` designation or if you process as part of the Vantiv PayFac Assurance program.

---

**NOTE:** When creating a Legal Entity, you must define the controlling Principal of the Legal Entity. Use the `legalEntityPrincipalCreateRequest` (see [Create a Principal](#) on page 52) to add all other Principal with a 10% or greater ownership of the Legal Entity.

---

The following fields are required for all Legal Entity Types:

- `legalEntityName`
- `legalEntityType`
- `taxId` (EIN for Canada)
- `firstName` (Principal)
- `lastName` (Principal)
- `streetAddress1` (Legal Entity and Principal)
- `stateProvince` (Legal Entity and Principal)
- `city` (Legal Entity and Principal)
- `postalCode` (Legal Entity and Principal)
- `countryCode` (Legal Entity and Principal)
- `annualCreditCardSalesVolume`
- `hasAcceptedCreditCards`
- `ssn` (Principal)
- `dateOfBirth` (Principal)
- `stakePercent` (Principal)
- `title` (Principal) (Required only if using AmEx Opt Blue)
- `legalEntityOwnershipType`

---

**NOTE:** In addition to the required fields listed above, Vantiv recommends that you also include the contact phone number (`contactPhone` element) of the principal, as well as the business phone number of the Legal Entity and the DBA name (`doingBusinessAs` field).

---

If you process as part of the Vantiv PayFac Assurance program, in addition to the fields listed above, you must also include the following fields:

- title (Principal)
- yearsInBusiness (Legal Entity)
- url (Sub-merchant)

### 3.3.3 Create Legal Entity Request Message

You must structure a Create Legal Entity request as shown in the following examples.

```
<legalEntityCreateRequest>
  <legalEntityName>Name of Legal Entity</legalEntityName>
  <legalEntityType>Type of Legal Entity</legalEntityType>
  <legalEntityOwnershipType>PUBLIC or PRIVATE</legalEntityOwnershipType>
  <doingBusinessAs>Alternate Name</doingBusinessAs>
  <taxId>Tax Id/EIN</taxId>
  <contactPhone>Legal Entity Contact Number</contactPhone>
  <annualCreditCardSalesVolume>Annual Visa Sales</annualCreditCardSalesVolume>
  <hasAcceptedCreditCards>true or false</hasAcceptedCreditCards>
  <address>
  <principal>
  <yearsInBusiness>Number of Years in Business</yearsInBusiness>
</legalEntityCreateRequest>
```

#### Example: Create Legal Entity Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<legalEntityCreateRequest
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard">
  <legalEntityName>Legal Entity Name</legalEntityName>
  <legalEntityType>CORPORATION</legalEntityType>
  <legalEntityOwnershipType>PUBLIC</legalEntityOwnershipType>
  <doingBusinessAs>Alternate Business Name</doingBusinessAs>
  <taxId>12345</taxId>
  <contactPhone>7817659800</contactPhone>
  <annualCreditCardSalesVolume>80000000</annualCreditCardSalesVolume>
  <hasAcceptedCreditCards>true</hasAcceptedCreditCards>
  <address>
    <streetAddress1>Street Address 1</streetAddress1>
    <streetAddress2>Street Address 2</streetAddress2>
```

```
<city>City</city>
<stateProvince>MA</stateProvince>
<postalCode>01730</postalCode>
<countryCode>USA</countryCode>
</address>
<principal>
  <title>Chief Financial Officer</title>
  <firstName>p first</firstName>
  <lastName>p last</lastName>
  <emailAddress>emailAddress</emailAddress>
  <ssn>123459876</ssn>
  <contactPhone>7817659800</contactPhone>
  <dateOfBirth>1980-10-12</dateOfBirth>
  <driversLicense>892327409832</driversLicense>
  <driversLicenseState>MA</driversLicenseState>
  <address>
    <streetAddress1>p street address 1</streetAddress1>
    <streetAddress2>p street address 2</streetAddress2>
    <city>Boston</city>
    <stateProvince>MA</stateProvince>
    <postalCode>01890</postalCode>
    <countryCode>USA</countryCode>
  </address>
  <stakePercent>33</stakePercent>
</principal>
<yearsInBusiness>12</yearsInBusiness>
</legalEntityCreateRequest>
```

### 3.3.4 Create Legal Entity Response Message

The structure for the response to a Create Legal Entity message varies slightly depending upon if it is an original request, a duplicate request with all data matching (true duplicate), or a duplicate with data that does not entirely match, as well as the state of the original request, if the new request is a duplicate. The response message has the following structure with exceptions as noted:

```
<legalEntityCreateResponse duplicate="true or false">
  <transactionId>1234567890123456789</transactionId>
  <legalEntityId>ID assigned to Legal Entity</legalEntityId>(appears if first
request)
  <responseCode>Response Code</responseCode>
  <responseDescription>Description of Response Code</responseDescription>
```

```

    <originalLegalEntityId>Original ID assigned to Legal
    Entity</originalLegalEntityId> (appears if dupe)

    <originalLegalEntityStatus>Original Status of Legal
    Entity</originalLegalEntityStatus> (appears if dupe)

    <backgroundCheckResults> (part of originalLegalEntity if present; otherwise,
                                at this level)

</legalEntityCreateResponse>

```

---

**NOTE:** If you receive a Response Code 30 - Retry, the decision notes will contain additional details about the information that you should provide/correct on the retry.

---

### Example: Create Legal Entity Response - for original

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<legalEntityResponse
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard">
  <transactionId>82820200338801014</transactionId>
  <legalEntityId>8</legalEntityId>
  <responseCode>10</responseCode>
  <responseDescription>Approved</responseDescription>
  <principal>
    <principalId>1</principalId>
    <firstName>p first</firstName>
    <lastName>p last</lastName>
  </principal>
  <backgroundCheckResults>
    <business>
      <verificationResult>
        <overallScore>
          <score>Overall results for Business</score>
          <description>Text Description of Score</description>
        </overallScore>
        <nameAddressTaxIdAssociation>
          <code>Name_Address_TIN_info</code>
          <description>Text Description of Code</description>
        </nameAddressTaxIdAssociation>
        <nameAddressPhoneAssociation>
          <code>Name_Address_Phone_info</code>
          <description>Text Description of Code</description>
        </nameAddressPhoneAssociation>
        <verificationIndicators>
          <nameVerified>true or false</nameVerified>
          <addressVerified>true or false</addressVerified>

```

```
<cityVerified>true or false</cityVerified>
<zipVerified>true or false</zipVerified>
<phoneVerified>true or false</phoneVerified>
<taxIdVerified>true or false</taxIdVerified>
</verificationIndicators>
<riskIndicators>
  <riskIndicator>
    <code>Risk Indicator Info</code>
    <description>Text Description of Code</description>
  </riskIndicator>
</riskIndicators>
</verificationResult>
</business>
<principal>
  <verificationResult>
    <overallScore>
      <score>Overall Results for Principal</score>
      <description>Text Description of Score</description>
    </overallScore>
    <nameAddressSsnAssociation>
      <code>Name_Address_SSN_info</code>
      <description>Text Description of Code</description>
    </nameAddressSsnAssociation>
    <nameAddressPhoneAssociation>
      <code>Name_Address_Phone_info</code>
      <description>Text Description of Code</description>
    </nameAddressPhoneAssociation>
    <verificationIndicators>
      <nameVerified>true or false</nameVerified>
      <addressVerified>true or false</addressVerified>
      <phoneVerified>true or false</phoneVerified>
      <ssnVerified>true or false</ssnVerified>
      <dobVerified>true or false</dobVerified>
    </verificationIndicators>
    <riskIndicators>
      <riskIndicator>
        <code>Risk Indicator results</code>
        <description>Text Description of Code</description>
      </riskIndicator>
    </riskIndicators>
  </verificationResult>
</principal>
<businessToPrincipalAssociation>
```

```

    <score>Business_To_Principal_info</score>
    <description>Text Description of Score</description>
  </businessToPrincipalAssociation>
  <backgroundCheckDecisionNotes>Additional Info About
Decision</backgroundCheckDecisionNotes>
  <bankruptcyData>
    <bankruptcyType>Sub-category of bankruptcy</bankruptcyType>
    <bankruptcyCount>1</bankruptcyCount>
    <companyName>Company Name</companyName>
    <streetAddress1>100 Main Street</streetAddress1>
    <streetAddress2>Suite 2</streetAddress2>
    <city>Boston</city>
    <state>MA</state>
    <zip>01150</zip>
    <zip4>2202</zip4>
    <filingDate>2011-05-13</filingDate>
  </bankruptcyData>
  <lienResult>
    <lienType>Subtype of Lien</lienType>
    <releasedCount>2</releasedCount>
    <unreleasedCount>1</unreleasedCount>
    <companyName>Company Name</companyName>
    <streetAddress1>100 Main Street</streetAddress1>
    <streetAddress2>Suite 2</streetAddress2>
    <city>Boston</city>
    <state>MA</state>
    <zip>01150</zip>
    <zip4>2202</zip4>
    <filingDate>2011-05-13</filingDate>
  </lienResult>
</backgroundCheckResults>
</legalEntityResponse>

```

### Example: Create Legal Entity Response - for duplicate

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<legalEntityCreateResponse
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard" duplicate="true">
  <transactionId>82821203948749101</transactionId>
  <originalLegalEntityId>82821203948912667</originalLegalEntityId>
  <originalLegalEntityStatus>Approved</originalLegalEntityStatus>
</legalEntityCreateResponse>

```

---

---

**NOTE:** If the Legal Entity is in Manual Review the response would be identical to the above response except the `originalLegalEntityStatus` element would have a value of Manual Review.

---

---



## 3.4 Update Legal Entity

You can update the several items associated with a Legal Entity, including information about a Principal. To update information associated with a Legal Entity, you submit a `legalEntityUpdateRequest` message with the appropriate `legalEntityId` and the updated information.

You can update the following items:

- `streetAddress1` (entity and/or principal)
- `streetAddress2` (entity and/or principal)
- `city` (entity and/or principal)
- `stateProvince` (entity and/or principal)
- `postalCode` (entity and/or principal)
- `doingBusinessAs`
- `annualCreditCardSalesVolume`
- `hasAcceptedCreditCards`
- `contactPhone` (entity and/or principal)
- `title` (principal)
- `emailAddress` (principal)
- `firstName` (principal)
- `lastName` (principal)
- `ssn` (principal)
- `dateOfBirth` (principal)
- `driversLicense` (principal)
- `driversLicenseState` (principal)
- `legalEntityName`
- `legalEntityType`
- `taxId`
- `yearsInBusiness`

---

---

**NOTE:** If the Legal Entity is approved and the TIN validation status is either approved or not required, you can not update the following elements: `legalEntityName`, `legalEntityType`, `taxId`, `firstName`, `lastName`, `ssn`, `dateOfBirth`, `driversLicense`, `driversLicenseState`. Prior to the TIN validation, updating any of these elements triggers a background check.

---

---

**TABLE 3-6** Resource Information

HTTP Methods	PUT
Call	<code>/legalentity/legalEntityId</code>
Request Format	XML
Response Format	XML

**TABLE 3-6** Resource Information

Rate Limited	Yes
--------------	-----

**TABLE 3-7** PUT Parameters

Parameter	Required or Optional	Description
legalEntityId	Required	The Id of the Legal Entity

## 3.4.1 Update Legal Entity Request

You must structure an Update Legal Entity request as shown in the following examples.

```
<legalEntityUpdateRequest>
  <address>
    <contactPhone>9785551234</contactPhone>
    <doingBusinessAs>DBA Name</doingBusinessAs>
    <annualCreditCardSalesVolume>400000</annualCreditCardSalesVolume>
    <hasAcceptedCreditCards>true</hasAcceptedCreditCards>
    <principal>
      <backgroundCheckFields>
        <legalEntityOwnershipType>PUBLIC</legalEntityOwnershipType>
        <yearsInBusiness>3</yearsInBusiness>
      </backgroundCheckFields>
    </principal>
  </address>
</legalEntityUpdateRequest>
```

### Example: Update Legal Entity Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<legalEntityUpdateRequest
  xmlns="http://payfac.vantivcnp.com/api/merchant/onboard">
  <address>
    <streetAddress1>LE Street Address 1</streetAddress1>
    <streetAddress2>LE Street Address 2</streetAddress2>
    <city>LE City</city>
    <stateProvince>MA</stateProvince>
    <postalCode>01730</postalCode>
    <countryCode>USA</countryCode>
  </address>
  <contactPhone>9785550101</contactPhone>
  <doingBusinessAs>Other Name Co.</doingBusinessAs>
```

```
<annualCreditCardSalesVolume>10000000</annualCreditCardSalesVolume>
<hasAcceptedCreditCards>true</hasAcceptedCreditCards>
<principal>
  <title>CEO</title>
  <emailAddress>jdoe@mail.net</emailAddress>
  <contactPhone>9785551234</contactPhone>
  <address>
    <streetAddress1>p street address 1</streetAddress1>
    <streetAddress2>p street address 2</streetAddress2>
    <city>Boston</city>
    <stateProvince>MA</stateProvince>
    <postalCode>01890</postalCode>
    <countryCode>USA</countryCode>
  </address>
  <backgroundCheckFields>
    <firstName>p first</firstName>
    <lastName>p last</lastName>
    <ssn>123459876</ssn>
    <dateOfBirth>1980-10-12</dateOfBirth>
    <driversLicense>892327409832</driversLicense>
    <driversLicenseState>MA</driversLicenseState>
  </backgroundCheckFields>
</principal>
<backgroundCheckFields>
  <legalEntityName>Company Name</legalEntityName>
  <legalEntityType>LE Type Enum</legalEntityType>
  <taxId>123456789</taxId>
</backgroundCheckFields>
<legalEntityOwnershipType>PUBLIC</legalEntityOwnershipType>
<yearsInBusiness>10</yearsInBusiness>
</legalEntityUpdateRequest>
```

### 3.4.2 Update Legal Entity Response

The system replies to an Update Legal Entity request with a generic response message. The structure of the response message is as follows:

```
<legalEntityResponse>
  <transactionId>1234567890123456789</transactionId>
  <legalEntityId> (appears if first request of dupe with all matching data)
```

```
<responseCode>
<responseDescription>
<backgroundCheckResults> (appears if background check triggered by update)
</legalEntityResponse>
```

### Example: Update Legal Entity Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<legalEntityResponse
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard">
  <transactionId>82820200338801014</transactionId>
  <legalEntityId>8</legalEntityId>
  <responseCode>10</responseCode>
  <responseDescription>Approved</responseDescription>
  <principal>
    <principalId>1</principalId>
    <firstName>p first</firstName>
    <lastName>p last</lastName>
  </principal>
  <backgroundCheckResults>
    <business>
      <verificationResult>
        <overallScore>
          <score>Overall results for Business</score>
          <description>Text Description of Score</description>
        </overallScore>
        <nameAddressTaxIdAssociation>
          <code>Name_Address_TIN_info</code>
          <description>Text Description of Code</description>
        </nameAddressTaxIdAssociation>
        <nameAddressPhoneAssociation>
          <code>Name_Address_Phone_info</code>
          <description>Text Description of Code</description>
        </nameAddressPhoneAssociation>
        <verificationIndicators>
          <nameVerified>true or false</nameVerified>
          <addressVerified>true or false</addressVerified>
          <cityVerified>true or false</cityVerified>
          <zipVerified>true or false</zipVerified>
          <phoneVerified>true or false</phoneVerified>
          <taxIdVerified>true or false</taxIdVerified>
        </verificationIndicators>
        <riskIndicators>
```

```
<riskIndicator>
  <code>Risk Indicator Info</code>
  <description>Text Description of Code</description>
</riskIndicator>
</riskIndicators>
</verificationResult>
</business>
<principal>
  <verificationResult>
    <overallScore>
      <score>Overall Results for Principal</score>
      <description>Text Description of Score</description>
    </overallScore>
    <nameAddressSsnAssociation>
      <code>Name_Address_SSN_info</code>
      <description>Text Description of Code</description>
    </nameAddressSsnAssociation>
    <nameAddressPhoneAssociation>
      <code>Name_Address_Phone_info</code>
      <description>Text Description of Code</description>
    </nameAddressPhoneAssociation>
    <verificationIndicators>
      <nameVerified>true or false</nameVerified>
      <addressVerified>true or false</addressVerified>
      <phoneVerified>true or false</phoneVerified>
      <ssnVerified>true or false</ssnVerified>
      <dobVerified>true or false</dobVerified>
    </verificationIndicators>
    <riskIndicators>
      <riskIndicator>
        <code>Risk Indicator results</code>
        <description>Text Description of Code</description>
      </riskIndicator>
    </riskIndicators>
  </verificationResult>
</principal>
<businessToPrincipalAssociation>
  <score>Business_To_Principal_info</score>
  <description>Text Description of Score</description>
</businessToPrincipalAssociation>
<backgroundCheckDecisionNotes>Additional Info About
Decision</backgroundCheckDecisionNotes>
<bankruptcyData>
```

```
<bankruptcyType>Sub-category of bankruptcy</bankruptcyType>
<bankruptcyCount>1</bankruptcyCount>
<companyName>Company Name</companyName>
<streetAddress1>100 Main Street</streetAddress1>
<streetAddress2>Suite 2</streetAddress2>
<city>Boston</city>
<state>MA</state>
<zip>01150</zip>
<zip4>2202</zip4>
<filingDate>2011-05-13</filingDate>
</bankruptcyData>
<lienResult>
  <lienType>Subtype of Lien</lienType>
  <releasedCount>2</releasedCount>
  <unreleasedCount>1</unreleasedCount>
  <companyName>Company Name</companyName>
  <streetAddress1>100 Main Street</streetAddress1>
  <streetAddress2>Suite 2</streetAddress2>
  <city>Boston</city>
  <state>MA</state>
  <zip>01150</zip>
  <zip4>2202</zip4>
  <filingDate>2011-05-13</filingDate>
</lienResult>
</backgroundCheckResults>
</legalEntityResponse>
```

## 3.5 Retrieve Legal Entity

If you need to retrieve the latest information on file related to a Legal Entity for verification purposes or to check the status of cases requiring manual intervention, you must perform a GET operation specifying the `legalEntityId`. In the case of a retrieval request, the system does not require an accompanying XML message.

---

**NOTE:** Vantiv makes the final results for the manual review of Legal entity information available within 2 business days of the initial response to the create request.

---

**TABLE 3-8** Resource Information

HTTP Methods	GET
Call	/legalentity/legalEntityId
Request Format	XML
Response Format	XML
Rate Limited	Yes

**TABLE 3-9** GET Parameters

Parameter	Required or Optional	Description
legalEntityId	Required	The Id of the legalEntity

### 3.5.1 Retrieve Legal Entity Response

The `legalEntityRetrieveResponse` message has the following format:

```
<legalEntityRetrievalResponse overallStatus="Approved">
  <legalEntityName>Name of Legal Entity</legalEntityName>
  <legalEntityType>Type of Legal Entity</legalEntityType>
  <doingBusinessAs>Alternate Name</doingBusinessAs>
  <taxId>Tax Id/EIN</taxId>
  <contactPhone>Legal Entity Contact Number</contactPhone>
  <annualCreditCardSalesVolume>Annual Visa Sales</annualCreditCardSalesVolume>
  <hasAcceptedCreditCards>true or false</hasAcceptedCrreditCards>
```

```
<address>
<principal>
<legalEntityId>12345678901234567</legalEntityId>
<responseCode>3-digit Code</responseCode>
<responseDescription>Description of Response Code</responseDescription>
<backgroundCheckResults>
<transactionId>1234567890123456789</transactionId>
<updateDate>YYYY-MM-DDTHH:MM:SS+/-HH:MM</updateDate>
<decisionDate>YYYY-MM-DDTHH:MM:SS+/-HH:MM</decisionDate>
<tinValidationStatus>Pending</tinValidationStatus>
<sub_merchant_processing_status>true or false</sub_merchant_processing_status>
</legalEntityRetrievalResponse>
<
```

### Example: Legal Entity Retrieval Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<legalEntityRetrievalResponse
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard"
  overallStatus="Approved">
  <legalEntityName>Legal Entity Name</legalEntityName>
  <legalEntityType>CORPORATION</legalEntityType>
  <doingBusinessAs>Alternate Name</doingBusinessAs>
  <taxId>X-2345</taxId>
  <contactPhone>7817659800</contactPhone>
  <annualCreditCardSalesVolume>80</annualCreditCardSalesVolume>
  <hasAcceptedCreditCards>true</hasAcceptedCreditCards>
  <address>
    <streetAddress1>12 Norton St</streetAddress1>
    <city>City</city>
    <stateProvince>NH</stateProvince>
    <postalCode>03064</postalCode>
    <countryCode>USA</countryCode>
  </address>
  <principal>
    <principalId>1</principalId>
    <title>CEO</title>
    <firstName>p first</firstName>
    <lastName>p last</lastName>
    <emailAddress>emailAddress</emailAddress>
    <ssn>XXXXX-9876</ssn>
```



```
<contactPhone>7817659800</contactPhone>
<dateOfBirth>1980-10-12</dateOfBirth>
<driversLicense>XXXXXXXX-9832</driversLicense>
<driversLicenseState>MA</driversLicenseState>
<address>
  <streetAddress1>p street address 1</streetAddress1>
  <streetAddress2>p street address 2</streetAddress2>
  <city>Boston</city>
  <stateProvince>MA</stateProvince>
  <postalCode>01890</postalCode>
  <countryCode>USA</countryCode>
</address>
</principal>
<legalEntityId>3</legalEntityId>
<responseCode>10</responseCode>
<responseDescription>Approved</responseDescription>
<backgroundCheckResults>
  <business>
    <verificationResult>
      <overallScores>
        <score>Overall results for Business</score>
        <description>Text Description of Score</description>
      </overallScores>
      <nameAddressTaxIdAssociation>
        <code>Name_Address_TIN_info</code>
        <description>Text Description of Code</description>
      </nameAddressTaxIdAssociation>
      <nameAddressPhoneAssociation>
        <code>Name_Address_Phone_info</code>
        <description>Text Description of Code</description>
      </nameAddressPhoneAssociation>
      <verificationIndicators>
        <nameVerified>true or false</nameVerified>
        <addressVerified>true or false</addressVerified>
        <cityVerified>true or false</cityVerified>
        <zipVerified>true or false</zipVerified>
        <phoneVerified>true or false</phoneVerified>
        <taxIdVerified>true or false</taxIdVerified>
      </verificationIndicators>
      <riskIndicators>
        <riskIndicator>
          <code>Risk Indicator Info</code>
          <description>Text Description of Code</description>
```

```
        </riskIndicator>
      </riskIndicators>
    </verificationResult>
  </business>
</principal>
<verificationResult>
  <overallScores>
    <score>Overall Results for Principal</score>
    <description>Text Description of Score</description>
  </overallScores>
  <nameAddressSsnAssociation>
    <code>Name_Address_SSN_info</code>
    <description>Text Description of Code</description>
  </nameAddressSsnAssociation>
  <nameAddressPhoneAssociation>
    <code>Name_Address_Phone_info</code>
    <description>Text Description of Code</description>
  </nameAddressPhoneAssociation>
  <verificationIndicators>
    <nameVerified>true or false</nameVerified>
    <addressVerified>true or false</addressVerified>
    <phoneVerified>true or false</phoneVerified>
    <ssnVerified>true or false</ssnVerified>
    <dobVerified>true or false</dobVerified>
  </verificationIndicators>
  <riskIndicators>
    <riskIndicator>
      <code>Risk Indicator results</code>
      <description>Text Description of Code</description>
    </riskIndicator>
  </riskIndicators>
</verificationResult>
</principal>
<businessToPrincipalAssociation>
  <score>Business_To_Principal_info</score>
  <description>Text Description of Score</description>
</businessToPrincipalAssociation>
<backgroundCheckDecisionNotes>Additional Info About
Decision</backgroundCheckDecisionNotes>
<bankruptcyData>
  <bankruptcyType>Sub-category of bankruptcy</bankruptcyType>
  <bankruptcyCount>1</bankruptcyCount>
  <companyName>Company Name</companyName>
```

```
<streetAddress1>100 Main Street</streetAddress1>
<streetAddress2>Suite 2</streetAddress2>
<city>Boston</city>
<state>MA</state>
<zip>01150</zip>
<zip4>2202</zip4>
<filingDate>2011-05-13</filingDate>
</bankruptcyData>
<lienResult>
  <lienType>Subtype of Lien</lienType>
  <releasedCount>2</releasedCount>
  <unreleasedCount>1</unreleasedCount>
  <companyName>Company Name</companyName>
  <streetAddress1>100 Main Street</streetAddress1>
  <streetAddress2>Suite 2</streetAddress2>
  <city>Boston</city>
  <state>MA</state>
  <zip>01150</zip>
  <zip4>2202</zip4>
  <filingDate>2011-05-13</filingDate>
</lienResult>
</backgroundCheckResults>
<transactionId>82820200338801030</transactionId>
<tinValidationStatus>Status of TIN Validation</tinValidationStatus>
<sub_merchant_processing_status>true</sub_merchant_processing_status>
</legalEntityRetrievalResponse>
```

## 3.6 Create a Principal

When you create a Legal Entity, you also create a Principal. You must create a Principal for each person that owns 10% or more of the Legal Entity. To create additional Principals associated with the Legal Entity you use a `legalEntityPrincipalCreateRequest`.

**TABLE 3-10** Resource Information

HTTP Methods	POST
Call	/legalentity/legalEntityId/principal
Request Format	XML
Response Format	XML
Rate Limited	Yes

**TABLE 3-11** POST Parameters

Parameter	Required or Optional	Description
legalEntityId	Required	The Id of the Legal Entity.

---

**NOTE:** If the `legalEntityType` is `SOLE_PROPRIETORSHIP`, you cannot delete or add Principals. The only Principal allowed is the one created with the Legal Entity. Also, if you already defined multiple Principals, the system does not allow you to change the `legalEntityType` to `SOLE_PROPRIETORSHIP` until you delete all but one of the Principals.

---

### 3.6.1 Principal Create Request

You must structure an Principal Create request as shown in the following examples.

```
<legalEntityPrincipalCreateRequest>
  <principal>
</legalEntityPrincipalCreateRequest>
```

**Example: Create Principal Request**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<legalEntityPrincipalCreateRequest
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard">
  <principal>
```

```

<title>Mr.</title>
<firstName>First</firstName>
<lastName>Last</lastName>
<emailAddress>abc@gmail.com</emailAddress>
<ssn>123450015</ssn>
<dateOfBirth>1980-10-12</dateOfBirth>
<address>
  <streetAddress1>p2 street address 1</streetAddress1>
  <streetAddress2>p2 street address 2</streetAddress2>
  <city>Boston2</city>
  <stateProvince>MA</stateProvince>
  <postalCode>01892</postalCode>
  <countryCode>USA</countryCode>
</address>
<stakePercent>31</stakePercent>
</principal>
</legalEntityPrincipalCreateRequest>

```

### 3.6.2 Principal Create Response

The structure of the response message is as follows:

---

**IMPORTANT:** Currently, the actual response message does not reflect the structure shown in the schema file for versions 12.x and 13.x. The parent of the response message is `principalCreateResponse`, not `legalEntityPrincipalCreateResponse`.

---

```

<legalEntityPrincipalCreateResponse> (actual parent is principalCreateResponse)
  <transactionId>Transaction ID</transactionId>
  <principalId>ID of New Principal</principalId>
  <firstName>First Name of Principal</firstName>
  <lastName>Last Name of Principal</lastName>
  <responseCode>Response Code</responseCode>
  <responseDescription>Response Description</responseDescription>
</legalEntityPrincipalCreateResponse>

```

### Example: Create Principal Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<principalCreateResponse
  xmlns="http://payfac.vantivcnf.com/api/merchant/onboard">
  <transactionId>8282985303203</transactionId>
  <principalId>2</principalId>
  <firstName>Princ First</firstName>
  <lastName>Princ Last</lastName>
  <responseCode>10</responseCode>
  <responseDescription>Approved</responseDescription>
</principalCreateResponse>
```

## 3.7 Delete a Principal

You can create and associate as many as eleven Principals with a Legal Entity. In the event that the Principals change and you need to remove one or more from association with the Legal Entity, you must delete a Principal. To delete a Principal perform a DELETE operation, specifying the `legalEntityId` with the parameter `principal` and the `principalId`. The system does not require an accompanying XML message.

---

**NOTE:** If the `legalEntityType` is `SOLE_PROPRIETORSHIP`, you cannot add or delete Principals.

---

**TABLE 3-12** Resource Information

HTTP Methods	DELETE
Call	/legalentity/legalEntityId/principal/principalId
Request Format	XML
Response Format	XML
Rate Limited	Yes

**TABLE 3-13** POST Parameters

Parameter	Required or Optional	Description
legalEntityId	Required	The Id of the Legal Entity.
principalId	Required	The Id of the Principal to be deleted.

### 3.7.1 Delete a Principal Response

The structure of the response message is as follows:

---

**IMPORTANT:** Currently, the actual response message does not reflect the structure shown in the schema file for versions 12.x and 13.x. The parent of the response message is `principalDeleteResponse`, not `legalEntityPrincipalDeleteResponse`.

---

```
<legalEntityPrincipalDeleteResponse> (actual parent is principalDeleteResponse)
  <transactionId>82820200338801014</transactionId>
  <legalEntityId>ID of Legal entity</legalEntityId>
  <principalId>Id of Deleted Principal</principalId>
  <responseDescription>Description of Response</responseDescription>
</legalEntityPrincipalDeleteResponse>
```

### Example: Delete a Principal Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<legalEntityPrincipalDeleteResponse
  xmlns="http://payfac.vantivcnf.com/api/merchant/onboard">
  <transactionId>82820200338801014</transactionId>
  <legalEntityId>8</legalEntityId>
  <principalId>2</principalId>
  <responseDescription>Legal Entity Principal successfully
  deleted</responseDescription>
</legalEntityPrincipalDeleteResponse>
```



## 3.8 Create Sub-Merchant

Once you have established a Legal Entity, you can create one or more associated Sub-merchants. For example, if the Legal Entity controlled three different business at different locations, you would establish three Sub-merchants under the Legal Entity. You create new Sub-merchants by submitting a `subMerchantCreateRequest` message.

**TABLE 3-14** Resource Information

HTTP Methods	POST
Call	/legalentity/legalEntityId/submerchant
Request Format	XML
Response Format	XML
Rate Limited	Yes

**TABLE 3-15** POST Parameters

Parameter	Required or Optional	Description
legalEntityId	Required	The Id of the Legal Entity.

---

**NOTE:** After the creation of a Sub-merchant, wait a minimum of two minutes before attempting to process transactions for the Sub-merchant. This is the minimum amount of time required for information about the newly created Sub-merchant to propagate through our system. Attempts to process transactions for a new Sub-merchant sooner than two minutes will result in system errors.

---

### 3.8.1 Create Sub-Merchant Request

You must structure a Create Sub-merchant request as shown in the following examples.

```
<subMerchantCreateRequest>
  <merchantName>My Company Name</merchantName>
  <amexMid>Amex Merchant Id</amexMid>
  <discoverConveyedMid>Discover Merchant Id</discoverConveyedMid>
  <url>MyCompany.com</url>
  <customerServiceNumber>9785551234</customerServiceNumber>
  <hardCodedBillingDescriptor>Default Descriptor<hardCodedBillingDescriptor>
```

```
<maxTransactionAmount>100000</maxTransactionAmount>
<purchaseCurrency>USD</purchaseCurrency>
<merchantCategoryCode>MCC Number</merchantCategoryCode>
<taxAuthority>Name of Tax Authority</taxAuthority> (MCC 9311 only)
<taxAuthorityState>State of Tax Authority</taxAuthorityState> (MCC 9311 only)
<bankRoutingNumber>123456789</bankRoutingNumber>
<bankAccountNumber>1234</bankAccountNumber>
<pspMerchantId>012345</pspMerchantId>
<fraud>
<amexAcquired>
<address>
<primaryContact>
<createCredentials>true</createCredentials>
<eCheck>
<subMerchantFunding>
<settlementCurrency>USD</settlementCurrency>
</subMerchantCreateRequest>
```

### Example: Create Sub-merchant Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<subMerchantCreateRequest
  xmlns="http://payfac.vantivcnp.com/api/merchant/onboard">
  <merchantName>Merchant Name</merchantName>
  <amexMid>1234567890</amexMid>
  <discoverConveyedMid>123456789012345</discoverConveyedMid>
  <url>http://merchantUrl</url>
  <customerServiceNumber>8407809000</customerServiceNumber>
  <hardCodedBillingDescriptor>billing
  Descriptor</hardCodedBillingDescriptor>
  <maxTransactionAmount>8400</maxTransactionAmount>
  <purchaseCurrency>USD</purchaseCurrency>
  <merchantCategoryCode>5964</merchantCategoryCode>
  <bankRoutingNumber>840123124</bankRoutingNumber>
  <bankAccountNumber>84012312415</bankAccountNumber>
  <pspMerchantId>123456</pspMerchantId>
  <fraud enabled="true"></fraud>
  <amexAcquired enabled="false"></amexAcquired>
  <address>
    <streetAddress1>Street Address 1</streetAddress1>
```

```

    <streetAddress2>Street Address 2</streetAddress2>
    <city>City</city>
    <stateProvince>MA</stateProvince>
    <postalCode>01970</postalCode>
    <countryCode>USA</countryCode>
  </address>
  <primaryContact>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <emailAddress>John.Doe@company.com</emailAddress>
    <phone>9785552222</phone>
  </primaryContact>
  <createCredentials>true</createCredentials>
  <eCheck enabled="true">
    <eCheckCompanyName>Your Company Name</eCheckCompanyName>
    <eCheckBillingDescriptor>9785552222</eCheckBillingDescriptor>
  </eCheck>
  <subMerchantFunding enabled="false"></subMerchantFunding>
  <settlementCurrency>USD</settlementCurrency>
</subMerchantCreateRequest>

```

### Example: Create Sub-merchant Request with Dynamic Payout

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<subMerchantCreateRequest
  xmlns="http://psp.vantivcn.com/api/merchant/onboard">
  <merchantName>Merchant Name</merchantName>
  <discoverConveyedMid>123456789012345</discoverConveyedMid>
  <url>http://merchantUrl</url>
  <customerServiceNumber>8407809000</customerServiceNumber>
  <hardCodedBillingDescriptor>billing
Descriptor</hardCodedBillingDescriptor>
  <maxTransactionAmount>8400</maxTransactionAmount>
  <purchaseCurrency>USD</purchaseCurrency>
  <merchantCategoryCode>5964</merchantCategoryCode>
  <bankRoutingNumber>840123124</bankRoutingNumber>
  <bankAccountNumber>84012312415</bankAccountNumber>
  <pspMerchantId>123456</pspMerchantId>
  <fraud enabled="true"></fraud>
  <amexAcquired enabled="true"></amexAcquired>
  <address>
    <streetAddress1>Street Address 1</streetAddress1>
    <streetAddress2>Street Address 2</streetAddress2>

```

```
<city>City</city>
<stateProvince>MA</stateProvince>
<postalCode>01970</postalCode>
<countryCode>USA</countryCode>
</address>
<primaryContact>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <emailAddress>John.Doe@company.com</emailAddress>
  <phone>9785552222</phone>
</primaryContact>
<createCredentials>true</createCredentials>
<eCheck enabled="true">
  <eCheckCompanyName>Your Company Name</eCheckCompanyName>
  <eCheckBillingDescriptor>9785552222</eCheckBillingDescriptor>
</eCheck>
<subMerchantFunding enabled="true">
  <fundingSubmerchantId>Unique Identifier</fundingSubmerchantId>
</subMerchantFunding>
<settlementCurrency>USD</settlementCurrency>
</subMerchantCreateRequest>
```

### Example: Create Sub-merchant Request with Managed Payout

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<subMerchantCreateRequest
  xmlns="http://psp.vantivcn.com/api/merchant/onboard">
  <merchantName>Merchant Name</merchantName>
  <amexMid>1234567890</amexMid>
  <discoverConveyedMid>123456789012345</discoverConveyedMid>
  <url>http://merchantUrl</url>
  <customerServiceNumber>8407809000</customerServiceNumber>
  <hardCodedBillingDescriptor>billing
  Descriptor</hardCodedBillingDescriptor>
  <maxTransactionAmount>8400</maxTransactionAmount>
  <purchaseCurrency>USD</purchaseCurrency>
  <merchantCategoryCode>5964</merchantCategoryCode>
  <bankRoutingNumber>840123124</bankRoutingNumber>
  <bankAccountNumber>84012312415</bankAccountNumber>
  <pspMerchantId>123456</pspMerchantId>
  <fraud enabled="true"></fraud>
  <amexAcquired enabled="false"></amexAcquired>
  <address>
```

```
<streetAddress1>Street Address 1</streetAddress1>
<streetAddress2>Street Address 2</streetAddress2>
<city>City</city>
<stateProvince>MA</stateProvince>
<postalCode>01970</postalCode>
<countryCode>USA</countryCode>
</address>
<primaryContact>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <emailAddress>John.Doe@company.com</emailAddress>
  <phone>9785552222</phone>
</primaryContact>
<createCredentials>true</createCredentials>
<eCheck enabled="true">
  <eCheckCompanyName>Your Company Name</eCheckCompanyName>
  <eCheckBillingDescriptor>9785552222</eCheckBillingDescriptor>
</eCheck>
<subMerchantFunding enabled="true">
  <feeProfile>Fee Profile Id</feeProfile>
</subMerchantFunding>
<settlementCurrency>USD</settlementCurrency>
</subMerchantCreateRequest>
```

### 3.8.2 Create Sub-Merchant Response

The response message to the Create Sub-merchant request has the following structure:

```
<subMerchantCreateResponse>
  <transactionId>0123456789</transactionId>
  <subMerchantId>1</subMerchantId>
  <merchantIdentString>123456789</merchantIdentString>
  <originalSubMerchant> (included only in some duplicate cases)
  <credentials> (included only if credentials requested)
  <paypageCredentials> (only if credentials requested and PayPage enabled)
  <amexSellerId>1234567890</amexSellerId>
</subMerchantCreateResponse>
```

---

**NOTE:** If the submission is a duplicate with changed data, the `submerchantId` and `merchantIdentString` elements appear as children of the `originalSubmerchant` element.

---

#### Example: Create Sub-merchant Response with Credentials

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<subMerchantCreateResponse
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard">
  <transactionId>82821240469914003</transactionId>
  <subMerchantId>1100003</subMerchantId>
  <merchantIdentString>01100003</merchantIdentString>
  <credentials>
    <userName>JDoe123</userName>
    <password>MyPassword</password>

    <passwordExpirationDate>2016-06-30T23:59:59-05:00</passwordExpirationDate>
  </credentials>
  <paypageCredentials>
    <paypageCredential>
      <userName>JDoe123</userName>
      <paypageId>1234567890123456</paypageId>
    </paypageCredential>
  </paypageCredentials>
  <amexSellerId>1234567890</amexSellerId>
</subMerchantCreateResponse>
```

#### Example: Create Sub-merchant Response - Duplicate with All Data Matching

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<subMerchantCreateResponse
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard" duplicate="true">
  <transactionId>82821240469914003</transactionId>
  <subMerchantId>1100003</subMerchantId>
  <merchantIdentString>01100003</merchantIdentString>
</subMerchantCreateResponse>
```

#### Example: Create Sub-merchant Response - Duplicate with Data Not Matching

```
<subMerchantCreateResponse
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard" duplicate="true">
  <transactionId>82821240469914003</transactionId>
  <originalSubMerchant>
```

```
<merchantName>Merchant Name</merchantName>
<amexMid>1234567890</amexMid>
<discoverConveyedMid>123456789012345</discoverConveyedMid>
<url>http://merchantUrl</url>
<customerServiceNumber>8407809000</customerServiceNumber>
<hardCodedBillingDescriptor>billing
Descriptor</hardCodedBillingDescriptor>
<maxTransactionAmount>8400</maxTransactionAmount>
<purchaseCurrency>USD</purchaseCurrency>
<merchantCategoryCode>5964</merchantCategoryCode>
<bankRoutingNumber>840123124</bankRoutingNumber>
<bankAccountNumber>84012312415</bankAccountNumber>
<pspMerchantId>123456</pspMerchantId>
<address>
  <streetAddress1>Street Address 1</streetAddress1>
  <streetAddress2>Street Address 2</streetAddress2>
  <city>City</city>
  <stateProvince>MA</stateProvince>
  <postalCode>01970</postalCode>
  <countryCode>USA</countryCode>
</address>
<primaryContact>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <phone>9785552222</phone>
  <emailAddress>John.Doe@company.com</emailAddress>
</primaryContact>
<subMerchantId>1002</subMerchantId>
<merchantIdentString>01100002</merchantIdentString>
</originalSubMerchant>
</subMerchantCreateResponse>
```

## 3.9 Update Sub-Merchant

You can update the several items associated with a Sub-merchant. To update information associated with a Sub-merchant, you submit a `subMerchantUpdateRequest` message with the appropriate `legalEntityId`, `subMerchantId`, and the updated information.

---

**NOTE:** If/when the bank account and routing number of the Sub-merchant changes, it is important that you notify us as soon as possible, so that our required records are always up to date.

Also, please consult your Implementation Consultant before updating the American Express Merchant Id (`amexMid` element) or the Discover Merchant Id (`discoverConveyedMid` element), since this information has direct impact on transaction processing.

---

You can update the following items:

- `url`
- `customerServiceNumber`
- `hardCodedBillingDescriptor`
- `maxTransactionAmount`
- `bankRoutingNumber`
- `bankAccountNumber`
- `discoverConveyedMid`
- `eCheckCompanyName`
- `streetAddress1`
- `streetAddress2`
- `city`
- `stateProvince`
- `postalCode`
- `amexMid`
- `eCheckBillingDescriptor`

**TABLE 3-16** Resource Information

HTTP Methods	PUT
Call	/legalentity/legalEntityId/submerchant/subMerchantId
Request Format	XML
Response Format	XML
Rate Limited	Yes



**TABLE 3-17** PUT Parameters

Parameter	Required or Optional	Description
legalEntityId	Required	The Id of the Legal Entity
subMerchantId	Required	The Id of the Sub-merchant

### 3.9.1 Update a Sub-Merchant

You must structure an Update Sub-merchant message as shown in the following examples:

```
<subMerchantUpdateRequest>
  <amexMid>Amex Merchant Id</amexMid>
  <discoverConveyedMid>Discover Merchant Id</discoverConveyedMid>
  <url>MyCompany.com</url>
  <customerServiceNumber>9785551234</customerServiceNumber>
  <hardCodedBillingDescriptor>Default Descriptor</hardCodedBillingDescriptor>
  <maxTransactionAmount>100000</maxTransactionAmount>
  <bankRoutingNumber>123456789</bankRoutingNumber>
  <bankAccountNumber>1234</bankAccountNumber>
  <pspMerchantId>123456</pspMerchantId>
  <purchaseCurrency>USD</purchaseCurrency>
  <address>My Company Name</address>
  <primaryContact>
  <disable>true or false</disable>
  <fraud>
  <amexAcquired>
  <eCheck>
  <subMerchantFunding>
  <taxAuthority>Name of Tax Authority</taxAuthority> (MCC 9311 only)
  <taxAuthorityState>State of Tax Authority</taxAuthorityState> (MCC 9311 only)
</subMerchantUpdateRequest>
```

#### Example: Update a Sub-Merchant Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<subMerchantUpdateRequest
  xmlns="http://payfac.vantivcnp.com/api/merchant/onboard">
```

```
<amexMid>1234567890</amexMid>
<discoverConveyedMid>123456789012345</discoverConveyedMid>
<url>http://merchantUrl</url>
<customerServiceNumber>8407809000</customerServiceNumber>
<hardCodedBillingDescriptor>Descriptor</hardCodedBillingDescriptor>
<maxTransactionAmount>8400</maxTransactionAmount>
<bankRoutingNumber>840123124</bankRoutingNumber>
<bankAccountNumber>84012312415</bankAccountNumber>
<pspMerchantId>785412365</pspMerchantId>
<purchaseCurrency>USD</purchaseCurrency>
<address>
  <streetAddress1>Street Address 1</streetAddress1>
  <streetAddress2>Street Address 2</streetAddress2>
  <city>City</city>
  <stateProvince>MA</stateProvince>
  <postalCode>01970</postalCode>
</address>
<primaryContact>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <phone>9785552222</phone>
  <emailAddress>John.Doe@company.com</emailAddress>
</primaryContact>
<fraud enabled="true"></fraud>
<amexAcquired enabled="true"></amexAcquired>
<eCheck enabled="true">
  <eCheckBillingDescriptor>9785552222</eCheckBillingDescriptor>
  <eCheckCompanyName>Your Company Name</eCheckCompanyName>
</eCheck>
</subMerchantUpdateRequest>
```

### 3.9.2 Update a Sub-Merchant Response

The system replies to an Update Sub-merchant request with a generic response message. The structure of the response message is as follows:

```
<response>
  <transactionId>1234567890123456789</transactionId>
</response>
```

**Example: Update Sub-Merchant Response**

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<response xmlns="http://payfac.vantivcn.com/api/merchant/onboard">
  <transactionId>82820200338801022</transactionId>
</response>
```

## 3.10 Retrieve Sub-Merchant

If you need to retrieve the latest information on file related to a Sub-merchant for verification purposes, you must perform a GET operation specifying the `legalEntityId` and `subMerchantId`. A retrieval request does not require an accompanying XML message.

---

**NOTE:** If you are processing transactions only on the Vantiv eCommerce platform and using Dynamic Payout, after creating a Sub-merchant using a value of `AUTO_GENERATE` for the `fundingSubmerchantId`, retrieve the assigned `fundingSubmerchantId` value by submitting a Sub-merchant Retrieval Request.

---

**TABLE 3-18** Resource Information

HTTP Methods	GET
Call	/legalentity/legalEntityId/submerchant/subMerchantId
Request Format	XML
Response Format	XML
Rate Limited	Yes

**TABLE 3-19** GET Parameters

Parameter	Required or Optional	Description
legalEntityId	Required	The Id of the Legal Entity
subMerchantId	Required	The Id of the Sub-merchant

### 3.10.1 Retrieve Sub-Merchant Response

The retrieve Sub-merchant response message has the following structure:

```
<subMerchantRetrievalResponse>
```

```
<merchantName>My Company Name</merchantName>
<amexMid>Amex Merchant Id</amexMid>
<discoverConveyedMid>Discover Merchant Id</discoverConveyedMid>
<url>MyCompany.com</url>
<customerServiceNumber>9785551234</customerServiceNumber>
<hardCodedBillingDescriptor>Default Descriptor<hardCodedBillingDescriptor>
<maxTransactionAmount>100000</maxTransactionAmount>
<purchaseCurrency>USD</purchaseCurrency>
<merchantCategoryCode>MCC Number</merchantCategoryCode>
<bankRoutingNumber>123456789</bankRoutingNumber>
<bankAccountNumber>XXXXX-1234</bankAccountNumber>
<pspMerchantId>012345</pspMerchantId>
<fraud>
<amexAcquired>
<address>
<primaryContact>
<createCredentials>true</createCredentials>
<eCheck>
<subMerchantFunding>
<settlementCurrency>USD</settlementCurrency>
<subMerchantId>1</subMerchantId>
<amexSellerId>1234567890</amexSellerId>
<disabled>true or false</disabled>
<transactionId>0123456789</transactionId>
<merchantIdentString>011000022</merchantIdentString>
<credentials>
<paypageCredentials>
<updateDate>2013-04-30T11:18:23.127-04:00</updateDate>
</subMerchantretrievalResponse>
```

### Example: Retrieval Sub-Merchant Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<subMerchantRetrievalResponse
  xmlns="http://payfac.vantivcnp.com/api/merchant/onboard">
  <merchantName>Merchant Name</merchantName>
  <amexMid>1234567890</amexMid>
```

```
<discoverConveyedMid>123456789012345</discoverConveyedMid>
<url>http://merchantUrl.com</url>
<customerServiceNumber>8407809000</customerServiceNumber>
<hardCodedBillingDescriptor>billing
Descriptor</hardCodedBillingDescriptor>
<maxTransactionAmount>100000</maxTransactionAmount>
<purchaseCurrency>USD</purchaseCurrency>
<merchantCategoryCode>5964</merchantCategoryCode>
<bankRoutingNumber>840123124</bankRoutingNumber>
<bankAccountNumber>XXXXX-3124</bankAccountNumber>
<pspMerchantId>123456</pspMerchantId>
<fraud enabled="false" />
<address>
  <streetAddress1>Street Address 1</streetAddress1>
  <streetAddress2>Street Address 2</streetAddress2>
  <city>City</city>
  <stateProvince>MA</stateProvince>
  <postalCode>01970</postalCode>
  <countryCode>USA</countryCode>
</address>
<primaryContact>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <phone>9785552222</phone>
  <emailAddress>John.Doe@company.com</emailAddress>
</primaryContact>
<eCheck enabled="true">
  <eCheckCompanyName>Your Company Name</eCheckCompanyName>
  <eCheckBillingDescriptor>9785552222</eCheckBillingDescriptor>
</eCheck>
<subMerchantFunding enabled="true">
  <fundingSubmerchantId>12345678901234</fundingSubmerchantId>
</subMerchantFunding>
<subMerchantId>88899966655544433</subMerchantId>
<amexSellerId>12345678901234</amexSellerId>
<disabled>true or false</disabled>
<transactionId>82820200338801105</transactionId>
<merchantIdentString>011000022</merchantIdentString>
<credentials>
  <userName>UserName</userName>
  <password>Password</password>

  <passwordExpirationDate>2017-10-03T11:18:23.127-04:00</passwordExpirationDate>
```

```
ate>
</credentials>
<paypageCredentials>
  <paypageCredential>
    <userName>PSPxmlV8</userName>
    <paypageId>Asd23thI974Jpk32</paypageId>
  </paypageCredential>
  <paypageCredential>
    <userName>PSPxmlV8Two</userName>
    <paypageId>odzhgcbQX3e3EaKV</paypageId>
  </paypageCredential>
  <paypageCredential>
    <userName>PSPxmlV8Three</userName>
    <paypageId>qmnpUBM6G47YJAcq</paypageId>
  </paypageCredential>
</paypageCredentials>
<updateDate>2017-09-30T11:18:23.127-04:00</updateDate>
</subMerchantRetrievalResponse>
```

## 3.11 Retrieve Allowed MCC List

You can retrieve the list of Merchant Category Codes, which are pre-approved for your use. When on-boarding a new Sub-merchant, a Payment Service Provider should only use one of the approved MCCs, appropriate to the Sub-merchant. To retrieve the allowed MCC list perform a GET operation, specifying the `legalEntityId` with the parameter `mcc`. In the case of a request to retrieve the allowed MCC list, the system does not require an accompanying XML message.

---

**NOTE:** Should you anticipate the need for new or different MCCs beyond those approved, please contact your Customer Experience Manager or our Underwriting team for additional information.

---

**TABLE 3-20** Resource Information

HTTP Methods	GET
Call	/mcc
Request Format	XML
Response Format	XML
Rate Limited	Yes

### 3.11.1 Retrieve MCC List Response

The structure of the response message is as follows:

```
<approvedMccResponse>
  <transactionId>1234567890123456789</transactionId>
  <approvedMccs>
    <approvedMcc>Merchant Category Code</approvedMcc>
  </approvedMccResponse>
```

#### Example: Retrieve MCC List Response

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<approvedMccResponse
  xmlns="http://payfac.vantivcn.com/api/merchant/onboard">
  <transactionID>82820200338801014</transactionID>
  <approvedMccs>
    <approvedMcc>4890</approvedMcc>
    <approvedMcc>4891</approvedMcc>
```

```
...  
    <approvedMcc>4896</approvedMcc>  
  </approvedMccs>  
</approvedMccResponse>
```



## 3.12 Status Codes and Error Messages

The HTTP Status Codes provide information about the success or failure of a transaction. In the case of a failure, an `errorResponse` message will contain additional information.

**TABLE 3-21** HTTP Status Codes

Code	Description
200	Update/Retrieval/Disable is successful
201	Create call is successful
400	Invalid Request. For example: <ul style="list-style-type: none"><li>• requested legal entity/sub merchant doesn't exist</li><li>• invalid field data (data is too long, required data is not sent)</li><li>• extra fields found in request</li><li>• attempt to create sub merchant for unapproved legal entity</li></ul> The response message will contain more details.
401	Failed Authentication
500	Internal error. Vantiv is investigating the issue. Please contact Vantiv Customer Support for additional information.
503	Returns for a soft decline or if the background checking service is unavailable. For a soft decline, please verify the submitted information and retry. If the service was unavailable, the Retry-After header contains a suggested retry time.

### 3.12.1 Error Response Message

The structure of the Error response message is as follows:

```
<errorResponse>
  <transactionId>1234567890123456789</transactionId>
  <errors>
    <error>Error Message</error>
  </errors>
</errorResponse>
```

#### Example: Error Response - Status Code 400

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<errorResponse>
  <transactionId>82820205828260878</transactionId>
```

```
<errors>
  <error>Error in Request:Legal Entity [Entity Name] has not been
  approved</error>
</errors>
</errorResponse>
```

or

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<errorResponse>
  <errors>
    <error>Error of [may not be null] on
    [subMerchantCreateRequest.merchantCategoryCode]</error>
    <error>Error of [size must be between 1 and 50] on
    [subMerchantCreateRequest.merchantName]</error>
  </errors>
</errorResponse>
```

#### Example: Error Response - Status Code 401

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<errorResponse>
  <errors>
    <error>You are not authorized to access this resource. Please check your
    credentials.</error>
  </errors>
</errorResponse>
```

#### Example: error Response - Status Code 500

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<errorResponse>
  <errors>
    <error>Internal Error. This error has already been escalated to Vantiv
    for resolution. Please contact support with questions.</error>
  </errors>
</errorResponse>
```

---

## PAYFAC API XML ELEMENTS

This chapter provides definitions for the elements used in PayFac API. This information is intended to assist you as you build the code necessary to submit messages creating and updating Legal Entities and Sub-merchants. Each section defines a particular element, its relationship to other elements (parents and children), as well as any attributes associated with the element.

For additional information on the structure of requests and responses using these elements, as well as XML examples, please refer to [Chapter 3, "PayFac API Transaction Examples"](#).

The XML elements defined in this chapter appear alphabetically.

## 4.1 acceptanceDateTime

The acceptanceDateTime element is a required child of the legalEntityAgreement element and defines date and time the party signing agreed to the Legal Entity Agreement.

**Type** = dateTime; **Format** = YYYY-MM-DDTHH:MM:SS+/-HH:MM

---

<b>NOTE:</b>	The +/-HH:MM portion of the dateTime format represents the offset from UTC time. For example, for Boston the value would be -05:00, meaning five hours behind UTC time.
--------------	---

---

**Parent Elements:**

[legalEntityAgreement](#)

**Attributes:**

None

**Child Elements:**

None

## 4.2 address

The parent element of several address related child elements. Although listed in the schema as an optional child of the `principal` element, you must include address information for principals.

### Parent Elements:

[legalEntityCreateRequest](#), [legalEntityUpdateRequest](#), [principal](#), [legalEntityRetrievalResponse](#), [subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

---

---

**NOTE:** When you include the `address` element in a `legalEntityUpdateRequest` message, you can not update the `countryCode` child element.

---

---

### Attributes:

None

### Child Elements:

[streetAddress1](#), [streetAddress2](#), [city](#), [stateProvince](#), [postalCode](#), [countryCode](#)

### Example: Address Structure

```
<address>
  <streetAddress1>100 Main Street</streetAddress1>
  <streetAddress2>Suite 400</streetAddress2>
  <city>Lowell</city>
  <stateProvince>MA</stateProvince>
  <postalCode>01852</postalCode>
  <countryCode>USA</countryCode>
</address>
```

## 4.3 addressVerified

The `addressVerified` element is an optional child of the `verificationIndicators` element. A value of **true** indicates that the address was verified, while a value of **false** indicates that the address could not be verified.

**Type** = Boolean; **Allowed values** = true or false

### Parent Elements:

[verificationIndicators](#)

### Attributes:

None

### Child Elements:

None

## 4.4 agreements

The `agreements` element is an optional child of the `legalEntityAgreementRetrievalResponse` message and contains a child element that define the agreements associated with the Legal Entity designated in the request. If this element is not returned in the response or is empty, the Legal Entity does not yet have any associated agreements. If the Legal Entity has multiple associated agreements, there will be multiple `legalEntityAgreement` child elements, one for each agreement.

### Parent Elements:

[legalEntityAgreementRetrievalResponse](#)

### Attributes:

None

### Child Elements:

[legalEntityAgreement](#)

### Example: agreements Structure

```
<agreements>
  <legalEntityAgreement>
    <legalEntityAgreementType>MERCHANT_AGREEMENT</legalEntityAgreementType>
    <agreementVersion>Agreement Version Number</agreementVersion>
    <userFullName>Full Name of Signer</userFullName>
    <userSystemName>User Name of Signer of PayFac System</userSystemName>
    <userIPAddress>IP Address of Signer's system</userIPAddress>
    <manuallyEntered>true or false</manuallyEntered>
    <acceptanceDateTime>Date and Time Agreement Signed</acceptanceDateTime>
  </legalEntityAgreement>
</agreements>
```

## 4.5 agreementVersion

The `agreementVersion` element is a required child of the `legalEntityAgreement` element and defines the version of Legal Entity Agreement.

**Type** = String; **minLength** = 1; **maxLength** = 50

**Parent Elements:**

[legalEntityAgreement](#)

**Attributes:**

None

**Child Elements:**

None



## 4.6 amexAcquired

The `amexAcquired` element is an optional child of the `subMerchantCreateRequest` and contains an attribute that indicates if the Sub-merchant is using the American Express Opt Blue program. The element is also a child of the `subMerchantUpdateRequest` and the `subMerchantRetrievalResponse` structures.

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

Attribute Name	Type	Required?	Description
enabled	Boolean	Yes	Defines if Amex Opt Blue processing is enabled for this sub-merchant. Valid values are either <b>true</b> or <b>false</b> .

### Child Elements:

None

## 4.7 amexMid

The `amexMid` element is an optional child of the `subMerchantCreateRequest` element and specifies the American Express Merchant Id of the Sub-merchant. If you accept American Express transactions, you must submit the Merchant Id provided by American Express.

---

**NOTE:** You must supply an American Express Merchant Id if you plan to process American express payments other than Opt Blue.

---

**Type** = String; **minLength** = 1; **maxLength** = 15

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.8 amexSellerId

The amexSellerId element is an optional child of the subMerchantCreateResponse and the subMerchantRetrievalResponse elements. This value is the American Express Seller Id assigned to the Sub-merchant. We return this value to you for your use when you communicate Sub-merchant profile information to American Express. We also includes this information in AmEx transactions, when we submit them for approval.

---

---

**NOTE:** Use of the American Express Seller Id becomes mandatory in October of 2014.

---

---

**Type** = String; **minLength** = N/A; **maxLength** = 32

**Parent Elements:**

[subMerchantCreateResponse](#), [subMerchantRetrievalResponse](#)

**Attributes:**

None

**Child Elements:**

None

## 4.9 annualCreditCardSalesVolume

The `annualCreditCardSalesVolume` element is a required child of the `legalEntityCreateRequest` element and defines the approximate annual credit card sales expected to be processed under this Legal Entity, in USD.

**Type** = Integer; **minLength** = 1; **maxLength** = 23

**Parent Elements:**

[legalEntityCreateRequest](#), [legalEntityRetrievalResponse](#), [legalEntityUpdateRequest](#)

**Attributes:**

None

**Child Elements:**

None

## 4.10 approvedMcc

The `approvedMcc` element is a child of the `approvedMccs` element and defines an approved Merchant Category Code for a Legal Entity/Sub-merchant.

**Parent Elements:**

[approvedMccs](#)

**Attributes:**

None

**Child Elements:**

None

## 4.11 approvedMccResponse

The `approvedMccResponse` element is the parent element for the XML message returned by the platform in response to a Approved MCC List retrieval request.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location you must specify it as: <code>http://payfac.vantivcnpp.com/api/merchant/onboard</code> <b>minLength = N/A maxLength = 50</b>

### Child Elements:

`transactionId`, `approvedMccs`

## 4.12 approvedMccs

The `approvedMccs` element is a required child of the `approvedMccResponse` message and contains a child element that define the list of Merchant Category Codes approved for use with Legal Entity/Sub-merchant.

### Parent Elements:

[approvedMccResponse](#)

### Attributes:

None

### Child Elements:

[approvedMcc](#)

### Example: approvedMccs Structure

```
<approvedMccs>
  <approvedMcc>4890</approvedMcc>
  <approvedMcc>4891</approvedMcc>
  <approvedMcc>4892</approvedMcc>
  ...
  <approvedMcc>4899</approvedMcc>
</approvedMccs>
```

## 4.13 backgroundCheckDecisionNotes

The `backgroundCheckDecisionNotes` element is an optional child of the `backgroundCheckResults` element and provides additional information about the background check performed by us.

---

**NOTE:** If the Legal Entity was declined after a manual review, this element will contain information relating to the reason for the decline.

---

**Type** = String; **minLength** = N/A; **maxLength** = 2000

### Parent Elements:

[backgroundCheckResults](#)

### Attributes:

None

### Child Elements:

None



## 4.14 backgroundCheckFields

The backgroundCheckFields element has multiple uses in the schema. This element is an optional child of the legalEntityUpdateRequest element, where it designates information about the Legal Entity. The element is also an optional child of the principal element, where it designates information about the Principal of the Legal Entity.

### Parent Elements:

[legalEntityResponse](#), [principal](#)

### Attributes:

None

### Child Elements (when used in legalEntityUpdateRequest):

All Optional: [legalEntityName](#), [legalEntityType](#), [taxId](#)

#### **Example:** backgroundCheckFields Structure when used in legalEntityUpdateRequest

```
<backgroundCheckFields>
  <legalEntityName>XYZ Co.</legalEntityName>
  <legalEntityType>LIMITED_LIABILITY_COMPANY</legalEntityType>
  <taxId>012345678</taxId>
</backgroundCheckFields>
```

### Child Elements (when used in principal):

All Optional: [firstName](#), [lastName](#), [ssn](#), [dateOfBirth](#), [driversLicense](#), [driversLicenseState](#),

#### **Example:** backgroundCheckFields Structure when used in principal

```
<backgroundCheckFields>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <ssn>012345678</ssn>
  <dateOfBirth>1958-12-12</dateOfBirth>
  <driversLicense>S123456789</driversLicense>
  <driversLicenseState>MA</driversLicenseState>
</backgroundCheckFields>
```

## 4.15 backgroundCheckResults

The backgroundCheckResults element is an optional child of both the legalEntityCreateResponse message and the legalEntityRetrievalResponse message. It contains child elements that provide scores and other information from the background checks performed by us.

### Parent Elements:

[legalEntityResponse](#), [legalEntityRetrievalResponse](#)

### Attributes:

None

### Child Elements:

[business](#), [principal](#), [businessToPrincipalAssociation](#), [backgroundCheckDecisionNotes](#), [bankruptcyData](#), [lienResult](#)

## 4.16 bankAccountNumber

The bankAccountNumber element is a required child of the subMerchantCreateRequest element and the subMerchantRetrievalResponse, where it specifies the account number of the merchant account used to fund the Sub-merchant. The element is also an optional child of the subMerchantUpdateRequest element.

---

**NOTE:** When returned in the submerchantRetrievalResponse, Vantiv masks the first five characters of the bankAccountNumber, as shown in the example below.

```
<bankAccountNumber>XXXXXX-1234</bankAccountNumber>
```

---

Type = String; minLength = 1; maxLength = 50

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.17 bankRoutingNumber

The `bankRoutingNumber` element is a required child of the `subMerchantCreateRequest` element and specifies the routing number of the merchant account used to fund the Sub-merchant. The element is also an optional child of the `subMerchantUpdateRequest` element.

**Type** = String; **minLength** = 1; **maxLength** = 50

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.18 bankruptcyCount

The `bankruptcyCount` element is an optional child of the `bankruptcyData` element and specifies the count of bankruptcy records found.

**Type** = int; **totalDigits** = N/A

**Parent Elements:**

[bankruptcyData](#)

**Attributes:**

None

**Child Elements:**

None

## 4.19 bankruptcyData

The `bankruptcyData` element is an optional child of the `backgroundCheckResults` element. It contains child elements that provides information about bankruptcy history of the company from the background checks performed by us.

### Parent Elements:

[backgroundCheckResults](#)

### Attributes:

None

### Child Elements:

[bankruptcyType](#), [bankruptcyCount](#), [companyName](#), [streetAddress1](#), [streetAddress2](#), [city](#), [state](#), [zip](#), [zip4](#), [filingDate](#)

### Example: bankruptcyData Structure

```
<bankruptcyData>
  <bankruptcyType>Sub-category of bankruptcy</bankruptcyType>
  <bankruptcyCount>1</bankruptcyCount>
  <companyName>Company Name</companyName>
  <streetAddress1>100 Main Street</streetAddress1>
  <streetAddress2>Suite 2</streetAddress2>
  <city>Boston</city>
  <state>MA</state>
  <zip>01150</zip>
  <zip4>2202</zip4>
  <filingDate>2011-05-13</filingDate>
</bankruptcyData>
```

## 4.20 bankruptcyType

The `bankruptcyType` element is an optional child of the `bankruptcyData` element and specifies the sub-type of bankruptcy discovered during the background check process.

**Type** = String; **minLength** = N/A; **maxLength** = 50

### Parent Elements:

[bankruptcyData](#)

### Attributes:

None

### Child Elements:

None

## 4.21 business

The `business` element is a child of the `backgroundCheckResults` element and contains child elements providing the results of background checks performed on the Legal Entity.

### Parent Elements:

[backgroundCheckResults](#)

### Attributes:

None

### Child Elements:

[verificationResult](#)

### Example: business Structure

```
<business>
  <verificationResult>
    <overallScores>
      <score>Overall results for Business</score>
      <description>Text Description of Score</description>
    </overallScores>
    <nameAddressTaxIdAssociation>
      <code>Name_Address_TIN_info</code>
      <description>Text Description of Code</description>
    </nameAddressTaxIdAssociation>
    <nameAddressPhoneAssociation>
      <code>Name_Address_Phone_info</code>
      <description>Text Description of Code</description>
    </nameAddressPhoneAssociation>
    <verificationIndicators>
      <nameVerified>true or false</nameVerified>
      <addressVerified>true or false</addressVerified>
      <cityVerified>true or false</cityVerified>
      <zipVerified>true or false</zipVerified>
      <phoneVerified>true or false</phoneVerified>
      <taxIdVerified>true or false</taxIdVerified>
    </verificationIndicators>
  </verificationResult>
</business>
```



```
</verificationIndicators>
<riskIndicators>
  <riskIndicator>
    <code>Risk Indicator Info</code>
    <description>Text Description of Code</description>
  </riskIndicator>
</riskIndicators>
</verificationResult>
</business>
```

## 4.22 businessToPrincipalAssociation

The `businessToPrincipalAssociation` element is an optional child of the `verificationResult` element and contains child elements providing background check information about how tightly coupled the principal is to the Legal Entity.

### Parent Elements:

[backgroundCheckResults](#)

### Attributes:

None

### Child Elements:

[score](#), [description](#)

### Example: businessToPrincipalAssociation Structure

```
<businessToPrincipalAssociation>
  <score>Results for Business to Principal</score>
  <description>Text Description of Score</description>
</businessToPrincipalAssociation>
```

## 4.23 city

The `city` element defines the city of either the Legal Entity, principal, or Sub-merchant. This element is always required for a Legal Entity and is also required for a principal if the `legalEntityType` is Sole Proprietorship. It is optional for all other `legalEntityType` values and for Sub-merchants.

The `city` element is also an optional child of both the `bankruptcyData` and `lienResults` elements, where it specifies the city of the company with bankruptcy or lien information.

**Type** = String; **minLength** = 1; **maxLength** = 20 (for address)

**Type** = String; **minLength** = 1; **maxLength** = 30 (for `bankruptcyData` or `lienResult`)

### Parent Elements:

`address`, `bankruptcyData`, `lienResult`

### Attributes:

None

### Child Elements:

None

## 4.24 cityVerified

The `cityVerified` element is an optional child of the `verificationIndicators` element, when `verificationIndicators` is a descendant of the `business` element. A value of **true** indicates that the city (of the business) was verified, while a value of **false** indicates that the city could not be verified.

**Type** = Boolean; **Allowed values** = true or false

### Parent Elements:

[verificationIndicators](#)

### Attributes:

None

### Child Elements:

None

## 4.25 code

The `code` element is used in several locations and its meaning depends upon the parent element. In each case the element value is selected from a set of enumerated values.

**Type** = String (Enum); **minLength** = N/A; **maxLength** = N/A

### Parent Elements:

[nameAddressPhoneAssociation](#), [nameAddressTaxIdAssociation](#), [riskIndicator](#),  
[nameAddressSsnAssociation](#),

### Attributes:

None

### Child Elements:

None

### Enumerations (as a child of [nameAddressTaxIdAssociation](#)):

Enumeration	Description
NOT_VERIFIED	Supplied information could not be not verified.
WRONG_TAX_ID	Supplied Tax Id is wrong.
NAME_OR_ADDRESS	The name or the address is verified.
BAD_NAME	The name as submitted can not be verified.
BAD_ADDRESS	The address as submitted can not be verified.
MISSING_ADDRESS	The address information is missing from the submitted data.
NAME_AND_ADDRESS_BAD_TAX_ID	Name and address verified; bad Tax Id
NAME_AND_ADDRESS_NO_TAX_ID	Name and address verified; no Tax Id information
NAME_ADDRESS_TAX_ID	Name, address, and Tax Id verified.

### Enumerations (as a child of (business) [nameAddressPhoneAssociation](#)):

Enumeration	Description
NOT_VERIFIED	Supplied information could not be not verified.

Enumeration	Description
WRONG_PHONE	Supplied Phone number is wrong.
NAME_OR_ADDRESS	The name or the address is verified.
BAD_NAME	The name, as submitted, can not be verified.
BAD_ADDRESS	The address, as submitted, can not be verified.
MISSING_ADDRESS	The address information is missing from the submitted data.
NAME_AND_ADDRESS_BAD_PHONE	Name and address verified; bad phone number
NAME_AND_ADDRESS_NO_PHONE	Name and address verified; no phone number submitted
NAME_ADDRESS_PHONE	Name, address, and phone verified.

### Enumerations (as a child of nameAddressSsnAssociation):

Enumeration	Description
NOTHING	Supplied information could not be not verified.
WRONG_SSN	Supplied SSN is wrong.
FIRST_LAST	First and last name verified
FIRST_ADDRESS	First name and address.
FIRST_SSN	First name and SSN verified.
LAST_ADDRESS	Last name and address verified.
ADDRESS_SSN	Address and SSN verified.
LAST_SSN	Last name and SSN verified.
FIRST_LAST_ADDRESS	First name, last name, and address verified.
FIRST_LAST_SSN	First name, last name, and SSN verified.
FIRST_ADDRESS_SSN	First name, address, and SSN verified.
LAST_ADDRESS_SSN	Last name, address, and SSN verified.
FIRST_LAST_ADDRESS_SSN	First name, last name, address, and SSN verified.

**Enumerations (as a child of (principal) nameAddressPhoneAssociation):**

Enumeration	Description
NOTHING	Supplied information could not be not verified.
WRONG_PHONE	Supplied phone number is wrong.
FIRST_LAST	First and last name verified
FIRST_ADDRESS	First name and address.
FIRST_PHONE	First name and phone number verified.
LAST_ADDRESS	Last name and address verified.
ADDRESS_PHONE	Address and phone number verified.
LAST_PHONE	Last name and phone number verified.
FIRST_LAST_ADDRESS	First name, last name, and address verified.
FIRST_LAST_PHONE	First name, last name, and phone number verified.
FIRST_ADDRESS_PHONE	First name, address, and phone number verified.
LAST_ADDRESS_PHONE	Last name, address, and phone number verified.
FIRST_LAST_ADDRESS_PHONE	First name, last name, address, and phone number verified.

**Enumerations (as a child of riskIndicator):**

Enumeration	Description
UNKNOWN	Supplied information could not be not verified.
SSN_DECEASED	The submitted SSN is reported as deceased.
SSN_PRIOR_TO_DOB	The submitted SSN was issued prior the submitted DOB.
SSN_ADDRESS_PHONE_NOT_MATCH	The submitted name and SSN have been verified, but the address and phone do not match.
SSN_INVALID	The submitted SSN is invalid.
PHONE_NUMBER_DISCONNECTED	The submitted phone number may be disconnected.
PHONE_NUMBER_INVALID	The submitted phone number is invalid.
PHONE_NUMBER_PAGER	The submitted phone number is a pager number.
PHONE_NUMBER_MOBILE	The submitted phone number is a mobile number.

Enumeration	Description
ADDRESS_INVALID	The submitted address may be invalid according to postal specifications.
ZIP_BELONGS_POST_OFFICE	The submitted zip code belongs to a post office box.
ADDRESS_INVALID_APARTMENT_DESIGNATION	The submitted address has an invalid apartment designation.
ADDRESS_COMMERCIAL	The submitted address is a transient commercial or institutional address.
PHONE_NUMBER_COMMERCIAL	The submitted phone number matches a transient commercial or institutional address.
PHONE_NUMBER_ZIP_INVALID	The submitted phone number and zip code combination is invalid.
UNABLE_TO_VERIFY_NAS	Unable to verify name, address, and SSN.
UNABLE_TO_VERIFY_ADDRESS	Unable to verify address.
UNABLE_TO_VERIFY_SSN	Unable to verify SSN.
UNABLE_TO_VERIFY_PHONE	Unable to verify the phone number.
UNABLE_TO_VERIFY_DOB	Unable to verify the date of birth.
SSN_MISKEYED	The submitted SSN may have been miskeyed.
ADDRESS_MISKEYED	The submitted address may have been miskeyed.
PHONE_NUMBER_MISKEYED	The submitted phone number may have been miskeyed.
NUMBER_MATCHES_OFAC	The submitted name matches the Office of Foreign Assets Control (OFAC) file.
UNABLE_TO_VERIFY_NAME	Unable to verify name.
SSN_MATCHES_MULTI_NAMES	The SSN is associated with multiple last names.
SSN_RECENTLY_ISSUED	The SSN is recently issued.
ZIP_CORPORATE_MILITARY	The submitted zip code is a corporate-only, military zip code.
DLL_INVALID	The submitted driver's license number is invalid for the submitted state.
NAME_ADDRESS_MATCH_BANKRUPTCY	The submitted name and address match a bankruptcy.
PHONE_AREA_CODE_CHANGING	The submitted phone number area code is changing.
WORK_PHONE_PAGER	The submitted work phone is a pager number.
UNABLE_TO_VERIFY_FIRST_NAME	Unable to verify the first name.



Enumeration	Description
PHONE_ADDRESS_DISTANT	The submitted phone number and address are physically distant (>10 miles).
ADDRESS_MATCHES_PRISON	The submitted address match a prison address.
SSN_LAST_NAME_NO_MATCH	The submitted last name does not associate with the submitted SSN.
SSN_FIRST_NAME_NO_MATCH	The submitted first name does not associate with the submitted SSN.
WORK_HOME_PHONE_DISTANT	The submitted home phone number and work phone number are physically distant (>100 miles)
NAME_ADDRESS_TIN_MISMATCH	The submitted business name and address match a Tax Identification Number (TIN) different than the submitted TIN.
WORK_PHONE_INVALID	The submitted work phone is potentially invalid.
WORK_PHONE_DISCONNECTED	The submitted work phone is potentially disconnected.
WORK_PHONE_MOBILE	The submitted work phone is a mobile number.
ADDRESS_RETURNS_DIFF_PHONE	The submitted address returns a different phone number.
SSN_LNAME_NOT_MATCHED_FNAME_MATCHED	The submitted SSN is associated with the same first name, but a different last name.
PHONE_RESIDENTIAL_LISTING	The submitted phone number is associated with a residential listing.
SINGLE_FAMILY_DWELLING	The submitted business address may be a residential address (single-family dwelling).
SSN_NOT_FOUND	The submitted SSN is not found in the public record.
SSN_BELONGS_TO_DIFF_NAME_ADDRESS	The submitted SSN is associated with a different name and address.
PHONE_BELONGS_TO_DIFF_NAME_ADDRESS	The submitted phone number is associated with a different name and address.
NAME_ADDRESS_UNLISTED	The submitted name and address are associated with an unlisted phone number.
NAME_MISKEYED	The submitted name may have been miskeyed.
NAME_MISSING	The name was missing.
ADDRESS_MISSING	The address was missing.
SSN_MISSING	The SSN was missing.
PHONE_NUMBER_MISSING	The phone number was missing.
DOB_MISSING	The date of birth was missing.

Enumeration	Description
NAME_ADDRESS_RETURN_DIFF_PHONE	The submitted name and address return a different phone number.
DOB_MISKEYED	The submitted date of birthday have been miskeyed.
SSN_NON_US_CITIZEN	The submitted SSN was issued to a non-US citizen.
ALTERNATE_BUSINESS_NAME_FOUND	The submitted business name not found; alternate business name found
DBA_MATCH_PUBLIC_RECORDS	DBA name matched public records
SSN_RECENT	The submitted SSN was issued in the last three years.
SSN_TOO_OLD	The submitted SSN was issued after age five (post-1990).
TIN_NAME_ADDRESS_MISMATCH	The submitted Tax Identification Number (TIN) is associated with a different business name and address.
BUSINESS_NOT_IN_GOOD_STANDING	The submitted business is not in good standing per the Secretary of State.
NAME_ADDRESS_MATCH_JUDGMENT	The submitted name and address match a judgement and/or lien filing.
BUSINESS_INACTIVE	The submitted business is inactive per the Secretary of State.
NO_UPDATE_IN_LAST_THREE_YEARS	No update to the business record in the last three years.
SSN_NOT_PRIMARY	The submitted SSN is not the primary SSN for the submitted identity.
ZIP_CORP_ONLY	The submitted zip code is a corporate only zip code.
ADDRESS_MISMATCH	The is an address mismatch between city/state and zip code.
DL_DIFFERENT	A different driver's license number was found for the submitted applicant.
DL_NOT_FOUND	The submitted driver's license number was not found.
DL_MISKEYED	The submitted driver's license number may have been miskeyed.
UNABLE_TO_VERIFY_DL	Unable to verify the submitted driver's license number.
SSN_INVALID_SSA	The submitted SSN is possibly randomly issued by the SSA, but invalid when first associated with the submitted identity.
SSN_IS_ITIN	The submitted SSN is an Individual Taxpayer Identification Number (ITIN).
SSN_MULTI_IDENTITY	Multiple SSNs reported for applicant

Enumeration	Description
ZIP_MILITARY	The submitted zip code is a military only zip code.
MULTIPLE_SSN_FOUND	Multiple SSNs reported for the applicant.
ADDRESS_DISCREPANCY	Potential address discrepancy - the submitted address may be a previous address.
ADDRESS_PO_BOX	The primary submitted address is a P.O. Box.
SSN_RANDOM_SSA	The submitted SSN is possibly randomly issued by the SSA.
ADDRESS_MISMATCH_SECONDARY	Address mismatch on secondary address range.
NAME_MATCHES_NON_OFAC	The submitted name matches one or more of the non-OFAC global watchlist.
UNABLE_TO_VERIFY_ZIP_CODE	Unable to verify zip code.
IP_ADDRESS_UNKNOWN	The submitted IP Address is unknown.
IP_ADDRESS_DIFFERENT_STATE	The submitted IP Address is assigned to a different State than the Bill-To state.
IP_ADDRESS_DIFFERENT_ZIP	The submitted IP Address is assigned to a different zip code than the Bill-To zip code.
IP_ADDRESS_DIFFERENT_PHONE	The submitted IP Address is assigned to a different area code than the Bill-To phone number.
IP_ADDRESS_DOMAIN_UNKNOWN	The submitted IP Address second level domain is unknown.
IP_ADDRESS_NOT_ASSIGNED_TO_USA	The submitted IP Address is not assigned to the United States.
IP_ADDRESS_NON_ROUTABLE	The submitted IP Address is non-routable over the internet.

## 4.26 companyName

The `companyName` element is an optional child of both the `lienResult` and `bankruptcyData` elements and specifies the name of the company to which the lien or bankruptcy information applies.

**Type** = String; **minLength** = 1; **maxLength** = 120

**Parent Elements:**

[bankruptcyData](#), [lienResult](#)

**Attributes:**

None

**Child Elements:**

None

## 4.27 contactPhone

The `contactPhone` element is an optional child of the `legalEntityCreateRequest`, `legalEntityUpdateRequest`, and the `principal` elements, as well as the `legalEntityRetrievalResponse`. It defines the contact phone number for the parent element.

**Type** = String; **minLength** = 1; **maxLength** = 10

**Parent Elements:**

[legalEntityCreateRequest](#), [principal](#), [legalEntityUpdateRequest](#), [legalEntityRetrievalResponse](#)

**Attributes:**

None

**Child Elements:**

None

## 4.28 countryCode

The `countryCode` element is a required child of the `address` element that defines the country of either the Legal Entity, the principal, or Sub-merchant.

**Type** = String; **minLength** = 2; **maxLength** = 3

### Parent Elements:

[address](#)

### Attributes:

None

### Child Elements:

None

## 4.29 createCredentials

The `createCredentials` element is an optional child of the `submerchantCreateRequest` element and indicates if the system should create login credentials (username, password, and `paypageId`, if necessary) for this Sub-merchant. Submit a value of **true** with each `subMerchantCreateRequest`, only if each Sub-merchant requires individual credentials for transaction submission.

**Type** = Boolean; **Possible Values** = true or false

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.30 credentials

The `credentials` element is an optional child of the `subMerchantCreateResponse` element and contains child elements that define the username and password for use by the sub-merchant, as well as the expiration date of the password.

**Parent Elements:**

[subMerchantCreateResponse](#), [subMerchantRetrievalResponse](#)

**Attributes:**

None

**Child Elements:**

[userName](#), [password](#), [passwordExpirationDate](#)

**Example: credentials**

```
<credentials>
  <userName>UserName</userName>
  <password>Password</password>
  <passwordExpirationDate>YYYY-MM-DDTHH:MM:SS+/-HH:MM</passwordExpirationDate>
</credentials>
```



## 4.31 customerServiceNumber

The `customerServiceNumber` element is a required child of the `subMerchantCreateRequest` element and specifies the customer service phone number of the Sub-merchant. It is also a child of the `subMerchantUpdateRequest` and the `subMerchantRetrievalResponse`.

**Type** = String; **minLength** = 1; **maxLength** = 13

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.32 dateOfBirth

The `dateOfBirth` element is a required child of the `principal` element and specifies the date of birth of the principal.

**Type** = Date; **Format** = yyyy-mm-dd

### Parent Elements:

[principal](#)

### Attributes:

None

### Child Elements:

None

## 4.33 decisionDate

The `decisionDate` element is an optional child of the `legalEntityRetrievalResponse` element. The value represents the date/time a decision was made for a Legal Entity that was in manual review. This element is returned only for Legal Entities that underwent manual review.

**Type** = `dateTime`; **Format** = `YYYY-MM-DDTHH:MM:SS+/-HH:MM`

---

**NOTE:** The `+/-HH:MM` portion of the `dateTime` format represents the offset from UTC time. For example, for Boston the value would be `-05:00` (or `-04:00` depending upon daylight savings time), meaning five hours behind UTC time.

---

### Parent Elements:

[legalEntityRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.34 description

The description element is a child of multiple elements and provides a text description of the accompanying code or score.

**Type** = String; **minLength** = N/A; **maxLength** = varies from 60 to 200 depending upon parent

### Parent Elements:

[overallScore](#), [nameAddressTaxIdAssociation](#), [nameAddressPhoneAssociation](#), [riskIndicator](#), [nameAddressSsnAssociation](#), [businessToPrincipalAssociation](#)

### Attributes:

None

### Child Elements:

None

## 4.35 disable

The `disable` element is an optional child of the `subMerchantUpdateRequest` element and allows you to disable an existing Sub-merchant. A value of **true** indicates that you want to changes the status of the Sub-merchant to disabled. The default value is **false**.

**Type** = Boolean; **Allowed values** = true or false

### Parent Elements:

[subMerchantUpdateRequest](#)

### Attributes:

None

### Child Elements:

None

## 4.36 disabled

The `disabled` element is an optional child of the `subMerchantRetrievalResponse` element and indicates the status of the Sub-merchant. A value of **true** indicates that the Sub-merchant is currently disabled, while a value of **false** indicates that the Sub-merchant is enabled.

**Type** = Boolean; **Allowed values** = true or false

### Parent Elements:

[subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.37 discoverConveyedMid

The `discoverConveyedMid` element is an optional child of the `subMerchantCreateRequest` element and specifies the Discover Merchant Id of the Sub-merchant.

---

**NOTE:** If you have a direct relationship and are funded by Discover for Discover transaction, then you are a Discover conveyed merchant and for proper processing, must supply the merchant ID assigned to you by Discover.

---

**Type** = String; **minLength** = 1; **maxLength** = 15

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.38 dobVerified

The `dobVerified` element is an optional child of the `verificationIndicators` element, when `verificationIndicator` is a descendant of the **principal** element. A value of **true** indicates that the Date of Birth of the principal was verified, while a value of **false** indicates that the DOB could not be verified.

**Type** = Boolean; **Allowed values** = true or false

### Parent Elements:

[verificationIndicators](#)

### Attributes:

None

### Child Elements:

None



## 4.39 doingBusinessAs

The `doingBusinessAs` element is an optional child of the `legalEntityCreateRequest` element and specifies an alternate name of the Legal Entity.

**Type** = String; **minLength** = 1; **maxLength** = 60

**Parent Elements:**

[legalEntityCreateRequest](#), [legalEntityUpdateRequest](#)

**Attributes:**

None

**Child Elements:**

None

## 4.40 driversLicense

The `driversLicense` element is an optional child of the `principal` element and specifies the driver's license number of the principal.

**Type** = String; **minLength** = 1; **maxLength** = 25

### Parent Elements:

[principal](#)

### Attributes:

None

### Child Elements:

None

## 4.41 driversLicenseState

The `driversLicenseState` element is an optional child of the `principal` element and specifies the state that issued the driver's license of the principal.

**Type** = String; **minLength** = 1; **maxLength** = 2

### Parent Elements:

[principal](#)

### Attributes:

None

### Child Elements:

None

## 4.42 eCheck

The `eCheck` element is an optional child of the `subMerchantCreateRequest` and contains child elements that define the `eCheckCompanyName` and `eCheckBillingDescriptor`.

### Parent Elements:

`subMerchantCreateRequest`, `subMerchantUpdateRequest`, `subMerchantRetrievalResponse`

### Attributes:

Attribute Name	Type	Required?	Description
enabled	Boolean	Yes	Defines if eCheck processing is enabled for this merchant. Valid values are either <b>true</b> or <b>false</b> .

### Child Elements:

`eCheckBillingDescriptor`, `eCheckCompanyName`

### Example: eCheck Structure

```
<eCheck enabled="true">
  <eCheckCompanyName>ABA Company</eCheckCompanyName>
  <eCheckBillingDescriptor>9785551111</eCheckBillingDescriptor>
</eCheck>
```

## 4.43 eCheckBillingDescriptor

The `eCheckBillingDescriptor` element is an optional child of the `subMerchantCreateRequest`, and the `subMerchantUpdateRequest` elements. It specifies the Billing Descriptor used for eCheck Transaction. You should specify this element when the Sub-Merchant is enabled for eCheck transaction.

---

**NOTE:** Typically, you use this field to submit the company phone number. You use the `eCheckCompanyName` field to submit the company name, which is the equivalent of a Billing Descriptor on a credit card transaction.

---

**Type** = String; **minLength** = 1; **maxLength** = 10

### Parent Elements:

[eCheck](#)

### Attributes:

None

### Child Elements:

None

## 4.44 eCheckCompanyName

The eCheckCompanyName element is an optional child of the subMerchantCreateRequest and the subMerchantUpdateRequest elements. It specifies the Company Name used for eCheck Transaction. You should specify this element when the Sub-Merchant is enabled for eCheck transaction.

**Type** = String; **minLength** = 1; **maxLength** = 16

### Parent Elements:

[eCheck](#)

### Attributes:

None

### Child Elements:

None

## 4.45 emailAddress

The `emailAddress` element is an optional child of the `principal` element, where it specifies the email address of the principal, and a required child of the `primaryContact` element, where it specifies the email address of the Sub-merchant's primary contact.

**Type** = String; **minLength** = 1; **maxLength** = 100

### Parent Elements:

[principal](#), [primaryContact](#)

### Attributes:

None

### Child Elements:

None

## 4.46 error

The `error` element is a child of the `errors` element and is a text description of the error preventing the processing of the transaction.

**Type** = String; **minLength** = N/A; **maxLength** = 512

### Parent Elements:

[errors](#)

### Attributes:

None

### Child Elements:

None



## 4.47 errorResponse

The `errorResponse` element is the parent element for the XML message returned by the platform for a transaction that the system cannot process due to a validation, authentication, or communication error.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcnpp.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50

### Child Elements:

[transactionId](#), [errors](#)

## 4.48 errors

The `errors` element is a child of the `errorResponse` element and through its child, `error`, provides a text description of the error preventing the processing of the transaction.

**Parent Elements:**

[errorResponse](#)

**Attributes:**

None

**Child Elements:**

[error](#)

## 4.49 feeProfile

The `feeProfile` element is an optional child of the `subMerchantFunding` element and defines the Fee Profile used for Sub-merchant fund disbursements using Managed Payout.

**Type** = String; **minLength** = 1; **maxLength** = 150

---

---

**NOTE:** You must create the Fee Profiles using the PayFac Portal.

---

---

### Parent Elements:

[subMerchantFunding](#)

### Attributes:

None

### Child Elements:

None

## 4.50 filingDate

The `filingDate` element is an optional child of both the `bankruptcyData` element and the `lienResult` element. It specifies the date of either the most recent lien or most recent bankruptcy filing, depending upon where used.

**Type** = Date; **Format** = yyyy-mm-dd

### Parent Elements:

[bankruptcyData](#), [lienResult](#)

### Attributes:

None

### Child Elements:

None

## 4.51 firstName

The `firstName` element is a required child of both the `principal` element, where it specifies the first name of the principal, and the `primaryContact` element, where it specifies the first name of the Sub-merchant's primary contact.

**Type** = String; **minLength** = 1; **maxLength** = 20

**Parent Elements:**

[principal](#), [primaryContact](#), [legalEntityPrincipalCreateResponse](#)

**Attributes:**

None

**Child Elements:**

None

## 4.52 fraud

The `fraud` element is an optional child of both the `subMerchantCreateRequest` and the `subMerchantUpdateRequest`. It defines whether fraud filters are enabled for the Sub-merchant.

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

Attribute Name	Type	Required?	Description
enabled	Boolean	Yes	Defines if fraud Filters are enabled for this Sub-merchant. Valid values are either <b>true</b> or <b>false</b> .

### Child Elements:

None

## 4.53 fundingSubmerchantId

The `fundingSubmerchantId` element is an optional child of the `subMerchantFunding` element (required if you use Dynamic Payout) and specifies the funding Id used to identify the Sub-merchant in funding instructions. (Also see, [Retrieve Sub-Merchant](#) on page 67)

---

**IMPORTANT:** If you are processing transactions on both the Vantiv Core and Vantiv eCommerce platforms, submit the Sub-merchant Id provided for use on the Vantiv Core platform.

If you are processing transactions only on the Vantiv eCommerce platform, submit a value of `AUTO_GENERATE`. To retrieve the `fundingSubmerchantId` value for use in Funding Instructions, submit a Sub-merchant Retrieval Request. (Please refer to the *Vantiv cnpAPI Reference Guide* for additional information about Funding Instructions.)

---

Type = String; minLength = 1; maxLength = 32

### Parent Elements:

[subMerchantFunding](#)

### Attributes:

None

### Child Elements:

None

## 4.54 hardCodedBillingDescriptor

The `hardCodedBillingDescriptor` element is a required child of the `subMerchantCreateRequest` element and specifies the Billing descriptor that will be hard-coded into the profile of the Sub-merchant. This default value appears on the customer's billing statement unless overridden by the submission of a soft billing descriptor in the Authorization/Sale transaction.

---

---

**IMPORTANT:** If the PayFac is configured with a forced prefix, the `hardCodedBillingDescriptor` value you submit for the Sub-merchant must include the prefix.

---

---

Type = String; minLength = 1; maxLength = 25

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None



## 4.55 hasAcceptedCreditCards

The `hasAcceptedCreditCards` element is a required child of the `legalEntityCreateRequest` element and indicates if this Legal Entity has accepted credit cards in the past.

**Type** = Boolean; **Possible Values** = true or false

### Parent Elements:

[legalEntityCreateRequest](#), [legalEntityRetrievalResponse](#), [legalEntityUpdateRequest](#)

### Attributes:

None

### Child Elements:

None

## 4.56 lastName

The `lastName` element is a required child of both the `principal` element, where it specifies the last name (surname) of the principal, and the `primaryContact` element, where it specifies the last name (surname) of the Sub-merchant's primary contact.

**Type** = String; **minLength** = 1; **maxLength** = 20

**Parent Elements:**

[principal](#), [primaryContact](#), [legalEntityPrincipalCreateResponse](#)

**Attributes:**

None

**Child Elements:**

None

## 4.57 legalEntityAgreement

The `legalEntityAgreement` element is a required child of the `legalEntityAgreementCreateRequest` and contains child elements that provide information about the Legal Entity Agreement. It is also an optional child of the `agreements` element, where it provides information about agreements associated with the designated Legal Entity in a Legal Entity Agreement Retrieval request. You should use this transaction type only if your PayFac Account Manager directs you to do so.

### Parent Elements:

`legalEntityAgreementCreateRequest`, `agreements`

### Attributes:

None

### Child Elements:

Required: `legalEntityAgreementType`, `agreementVersion`, `userFullName`, `userSystemName`, `acceptanceDateTime`

Optional: `userIPAddress`, `manuallyEntered`,

### Example: legalEntityAgreement Structure

```
<legalEntityAgreement>
  <legalEntityAgreementType>MERCHANT_AGREEMENT</legalEntityAgreementType>
  <agreementVersion>Agreement Version Number</agreementVersion>
  <userFullName>Full Name of Signer</userFullName>
  <userSystemName>User Name of Signer of PayFac System</userSystemName>
  <userIPAddress>IP Address of Signer's system</userIPAddress>
  <manuallyEntered>true or false</manuallyEntered>
  <acceptanceDateTime>Date and Time Agreement Signed</acceptanceDateTime>
</legalEntityAgreement>
```

## 4.58 legalEntityAgreementCreateRequest

The `legalEntityAgreementCreateRequest` element is the parent element for the XML message used to create a Legal Entity Agreement. You should use this transaction type only if your PayFac Account Manager directs you to do so.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcnpp.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50

### Child Elements:

Required: [legalEntityAgreement](#)

## 4.59 legalEntityAgreementCreateResponse

The `legalEntityAgreementCreateResponse` element is the parent element for the XML message returned by the platform in response to a successful (or duplicate) `legalEntityAgreementCreateRequest` message.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcn.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50
duplicate	Boolean	No	Appears in the response message only if the <code>legalEntityAgreementCreateRequest</code> was a duplicate request. <b>Allowed values</b> = true or false

### Child Elements:

Required: [transactionId](#)

## 4.60 legalEntityAgreementRetrievalResponse

The `legalEntityAgreementRetrievalResponse` element is the parent element for the XML message returned by the platform in response to a Legal Entity Agreement Retrieval request.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcnpp.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50

### Child Elements:

[legalEntityId](#), [transactionId](#), [agreements](#)

## 4.61 legalEntityAgreementType

The `legalEntityAgreementType` element is a required child of the `legalEntityAgreement` element and defines the type of Legal Entity Agreement. At this time, the only valid value is **MERCHANT\_AGREEMENT**.

**Type** = String (Enum); **minLength** = N/A; **maxLength** = N/A

### Parent Elements:

[legalEntityAgreement](#)

### Attributes:

None

### Child Elements:

None

## 4.62 legalEntityCreateRequest

The `legalEntityCreateRequest` element is the parent element for the XML message used to create a Legal Entity. You must create a Legal Entity prior to adding Sub-merchants controlled by the entity.

---

**NOTE:** When creating a Legal Entity, you must define the controlling Principal of the Legal Entity. Use the `legalEntityPrincipalCreateRequest` (see [Create a Principal](#) on page 52) to add all other Principal with a 10% or greater ownership of the Legal Entity.

---

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcn.com/api/merchant/onboard</code> <b>minLength = N/A maxLength = 50</b>

### Child Elements:

Always Required: [legalEntityName](#), [legalEntityType](#), [taxId](#), [annualCreditCardSalesVolume](#), [hasAcceptedCreditCards](#), [address](#) (some child elements may be optional), [principal](#)

Required/Optional (depending upon if using PayFac Assurance model): [yearsInBusiness](#)

Optional: [doingBusinessAs](#), [contactPhone](#)



## 4.63 legalEntityCreateResponse

The `legalEntityResponse` element is the parent element for the XML message returned by the platform in response to a `legalEntityCreateRequest` message.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcnpp.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50
duplicate	Boolean	No	Appears in the response message only if the <code>legalEntityCreateRequest</code> was a duplicate request. <b>Allowed values</b> = true or false

### Child Elements:

Always Required: [transactionId](#)

Optional: [legalEntityId](#), [pspMerchantId](#), [responseCode](#), [responseDescription](#), [backgroundCheckResults](#), [originalSubMerchant](#)

## 4.64 legalEntityId

The `legalEntityId` element is a required child of the `legalEntityResponse` element and is a system generated identifier for the Legal Entity.

**Type** = String; **minLength** = 1; **maxLength** = 19

**Parent Elements:**

[legalEntityResponse](#), [legalEntityRetrievalResponse](#), [legalEntityResponse](#)

**Attributes:**

None

**Child Elements:**

None

## 4.65 legalEntityName

The `legalEntityName` element is a required child of the `legalEntityCreateRequest` element and defines the name of the Legal Entity controlling the Sub-merchants.

**Type** = String; **minLength** = 1; **maxLength** = 60

**Parent Elements:**

[legalEntityCreateRequest](#), [legalEntityRetrievalResponse](#)

**Attributes:**

None

**Child Elements:**

None

## 4.66 legalEntityOwnershipType

The `legalEntityOwnershipType` element is a required child of the `legalEntityCreateRequest` and the `legalEntityUpdateRequest` where it defines whether the Legal Entity is Public or Private.

**Type** = String (Enum); **Valid Values** = PUBLIC or PRIVATE

---

---

**NOTE:** If the `legalEntityType` is either `TAX_EXEMPT_ORGANIZATION` or `GOVERNMENT_AGENCY`, you must set the `legalEntityOwnershipType` to `PUBLIC`.

---

---

### Parent Elements:

[legalEntityCreateRequest](#), [legalEntityUpdateRequest](#)

### Attributes:

None

### Child Elements:

None

## 4.67 legalEntityPrincipalCreateRequest

The `legalEntityPrincipalCreateRequest` element is the parent element for the XML message used to create a Legal Entity Principal. You must create a Principal record for each individual holding a 10% or greater stake in the Legal Entity.

---

**NOTE:** When creating a Legal Entity, you must define the controlling Principal of the Legal Entity.

If the `legalEntityType` is `SOLE_PROPRIETORSHIP`, you cannot add or delete Principals.

---

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcn.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50

### Child Elements:

Required: [principal](#)

## 4.68 legalEntityPrincipalCreateResponse

The `legalEntityPrincipalCreateResponse` element is the parent element for the XML message returned by the platform in response to a Legal Entity Principal Create Request.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcnpp.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50
duplicate	String	No	Appears in the response message only if the <code>legalEntityUpdateRequest</code> was a duplicate request. <b>Allowed values</b> = true or false

### Child Elements:

[transactionId](#), [principalId](#), [firstName](#), [lastName](#), [backgroundCheckResults](#)

## 4.69 legalEntityPrincipalDeleteResponse

The legalEntityPrincipalDeleteResponse is the parent element for the XML message returned by the platform in response to a Legal Entity Principal Delete request.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcnpp.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50

### Child Elements:

[transactionId](#), [legalEntityId](#), [principalId](#), [responseDescription](#)

## 4.70 legalEntityResponse

The `legalEntityResponse` element is the parent element for the XML message returned by the platform in response to a Legal Entity Update request.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcn.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50
duplicate	String	No	Appears in the response message only if the <code>legalEntityUpdateRequest</code> was a duplicate request. <b>Allowed values</b> = true or false

### Child Elements:

[transactionId](#), [legalEntityId](#), [responseCode](#), [responseDescription](#), [backgroundCheckResults](#)



## 4.71 legalEntityRetrievalResponse

The `legalEntityRetrievalResponse` element is the parent element for the XML message returned by the platform in response to a Legal Entity retrieval request.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcnpp.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50
overallStatus	String	Yes	Defines the overall status of the Legal Entity. <b>minLength</b> = 1 <b>maxLength</b> = 100

### Child Elements:

`legalEntityName`, `legalEntityType`, `taxId`, `annualCreditCardSalesVolume`,  
`hasAcceptedCreditCards`, `address`, `legalEntityId`, `responseCode`, `responseDescription`,  
`backgroundCheckResults`, `transactionId`, `updateDate`, `decisionDate`, `tinValidationStatus`,  
`sub_merchant_processing_status`

## 4.72 legalEntityType

The `legalEntityType` element is a required child of the `legalEntityCreateRequest` element and defines the business type of the Legal Entity.

**Type** = String (enum); **minLength** = N/A; **maxLength** = N/A

### Parent Elements:

[legalEntityCreateRequest](#), [legalEntityRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

### Enumerations:

Enumeration
INDIVIDUAL_SOLE_PROPRIETORSHIP
CORPORATION
LIMITED_LIABILITY_COMPANY
PARTNERSHIP
LIMITED_PARTNERSHIP
GENERAL_PARTNERSHIP
TAX_EXEMPT_ORGANIZATION
GOVERNMENT_AGENCY

---

**NOTE:** All Legal Entity Type are available for use by Canadian based Legal Entities; however, **LIMITED\_PARTNERSHIP** and **GENERAL\_PARTNERSHIP** will be deprecated in a future release, so you should avoid their use.

---

## 4.73 legalEntityUpdateRequest

The `legalEntityUpdateRequest` element is the parent element for the XML message used to update a Legal Entity. Using this transaction type, you can update the following Legal Entity items:

- `streetAddress1` (entity and/or principal)
- `streetAddress2` (entity and/or principal)
- `city` (entity and/or principal)
- `stateProvince` (entity and/or principal)
- `postalCode` (entity and/or principal)
- `doingBusinessAs`
- `annualCreditCardSalesVolume`
- `hasAcceptedCreditCards`
- `firstName` (principal)
- `lastName` (principal)
- `ssn` (principal)
- `dateOfBirth` (principal)
- `driversLicense` (principal)
- `driversLicenseState` (principal)
- `legalEntityName`
- `legalEntityType`
- `taxId`

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
<code>xmlns</code>	String	Yes	Defines the URI of the schema definition. This is a fixed location and you must specify it as: <code>http://payfac.vantivcn.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50

### Child Elements:

Optional: [address](#), [contactPhone](#), [doingBusinessAs](#), [annualCreditCardSalesVolume](#), [hasAcceptedCreditCards](#), [principal](#), [backgroundCheckFields](#)

## 4.74 lienResult

The `lienResult` element is an optional child of the `backgroundCheckResults` element. It contains child elements that provides information about the lien history of the company from the background checks performed by us.

### Parent Elements:

[backgroundCheckResults](#)

### Attributes:

None

### Child Elements:

[lienType](#), [releasedCount](#), [unreleasedCount](#), [companyName](#), [streetAddress1](#), [streetAddress2](#), [city](#), [state](#), [zip](#), [zip4](#), [filingDate](#)

### Example: lienResult Structure

```
<lienResult>
  <lienType>Subtype of Lien</lienType>
  <releasedCount>2</releasedCount>
  <unreleasedCount>1</unreleasedCount>
  <companyName>Company Name</companyName>
  <streetAddress1>100 Main Street</streetAddress1>
  <streetAddress2>Suite 2</streetAddress2>
  <city>Boston</city>
  <state>MA</state>
  <zip>01150</zip>
  <zip4>2202</zip4>
  <filingDate>2011-05-13</filingDate>
</lienResult>
```

## 4.75 lienType

The `lienType` element is an optional child of the `lienResult` element and specifies the sub-type of lien discovered during the background check process.

**Type** = String; **minLength** = 1; **maxLength** = 50

### Parent Elements:

[lienResult](#)

### Attributes:

None

### Child Elements:

None

## 4.76 manuallyEntered

The manuallyEntered element is a required child of the legalEntityAgreement element and defines whether information about the party signing the Legal Entity Agreement was entered manually or electronically. When this element is set to **false**, you must include a value for the userIPAddress element. When this element is set to **true**, the userIPAddress element is optional.

**Type** = boolean; **Allowed Values** = true or false

### Parent Elements:

[legalEntityAgreement](#)

### Attributes:

None

### Child Elements:

None

## 4.77 maxTransactionAmount

The `maxTransactionAmount` element is a required child of the `subMerchantCreateRequest` element and specifies the maximum transaction amount (in the purchase currency) allowed on a single transaction submitted by the Sub-merchant. Supply the value in dollars without a decimal point. For example, a value of 50000 signifies \$50,000.

**Type** = Long; **minLength** = N/A; **maxLength** = 12

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.78 merchantCategoryCode

The merchantCategoryCode element is a required child of the subMerchantCreateRequest element and specifies the Merchant Category Code of the Sub-merchant.

**Type** = String; **minLength** = 1; **maxLength** = 4

**Parent Elements:**

[subMerchantCreateRequest](#), [subMerchantRetrievalResponse](#)

**Attributes:**

None

**Child Elements:**

None



## 4.79 merchantIdentString

The `merchantIdentString` element is a required child of the `subMerchantCreateResponse` element and is a system generated merchant identifier required to process transactions for this Sub-merchant. It is also returned as a child of both the `originalSubMerchant` and `subMerchantRetrievalResponse` elements.

---

**NOTE:** This value is the unique string used to identify the merchant when submitting payment transactions to our system. In the `cnpAPI` used for payment transactions, it is the `merchantId` attribute used in the `batchRequest`, `batchResponse`, and `cnpOnlineRequest`.

---

Type = String; minLength = 1; maxLength = 50

### Parent Elements:

[subMerchantCreateResponse](#), [originalSubMerchant](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.80 merchantName

The merchantName element is a required child of the subMerchantCreateRequest element and specifies the name of the Sub-merchant.

**Type** = String; **minLength** = 1; **maxLength** = 50

**Parent Elements:**

[subMerchantCreateRequest](#), [subMerchantRetrievalResponse](#)

**Attributes:**

None

**Child Elements:**

None

## 4.81 nameAddressPhoneAssociation

The nameAddressPhoneAssociation element is an optional child of the verificationResult element and contains child elements providing the results of background checks performed on the Legal Entity that indicate how tightly coupled the name is to the phone number and address provided.

### Parent Elements:

[verificationResult](#)

### Attributes:

None

### Child Elements:

[code](#), [description](#)

### Example: nameAddressPhoneAssociation Structure

```
<nameAddressPhoneAssociation>
  <code>Name_Address_Phone_info</code>
  <description>Text Description of Code</description>
</nameAddressPhoneAssociation>
```

## 4.82 nameAddressSsnAssociation

The nameaddressSsnAssociation element is an optional child of the verificationResult element and contains child elements providing the results of background checks performed on the Legal Entity that indicate how tightly coupled the name is to the SSN and address provided.

### Parent Elements:

[verificationResult](#)

### Attributes:

None

### Child Elements:

[code](#), [description](#)

### Example: nameAddressTaxIdAssociation Structure

```
<nameAddressSsnAssociation>
  <code>Name_Address_SSN_info</code>
  <description>Text Description of Code</description>
</nameAddressSsnAssociation>
```

## 4.83 nameAddressTaxIdAssociation

The `nameaddressTaxIdAssociation` element is an optional child of the `verificationResult` element and contains child elements providing the results of background checks performed on the Legal Entity that indicate how tightly coupled the name is to the Tax Id and address provided.

### Parent Elements:

[verificationResult](#)

### Attributes:

None

### Child Elements:

[code](#), [description](#)

### Example: nameAddressTaxIdAssociation Structure

```
<nameAddressTaxIdAssociation>
  <code>Name_Address_TIN_info</code>
  <description>Text Description of Code</description>
</nameAddressTaxIdAssociation>
```

## 4.84 nameVerified

The `nameVerified` element is an optional child of the `verificationIndicators` element. A value of **true** indicates that the name of the principal or business was verified, while a value of **false** indicates that the name could not be verified.

**Type** = Boolean; **Allowed values** = true or false

### Parent Elements:

[verificationIndicators](#)

### Attributes:

None

### Child Elements:

None

## 4.85 originalLegalEntityId

The `originalLegalEntityId` element is an optional child of the `legalEntityCreateResponse`, returned when the `legalEntityCreateRequest` is a duplicate. The value indicates the Legal Entity ID of the original Legal Entity.

**Type** = String; **minLength** = 1; **maxLength** = 19

### Parent Elements:

[legalEntityCreateResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.86 originalLegalEntityStatus

The `originalLegalEntityStatus` element is an optional child of the `legalEntityCreateResponse`, returned when the `legalEntityCreateRequest` is a duplicate. The value indicates the status of the Legal Entity.

**Type** = String; **minLength** = 1; **maxLength** = 100

### Parent Elements:

[legalEntityCreateResponse](#)

### Attributes:

None

### Child Elements:

None



## 4.87 originalSubMerchant

The `originalSubMerchant` element is optional child of the `subMerchantCreateResponse` message. This element and its children are returned only if the system has determined the `subMerchantCreateRequest` is a duplicate and provides the information from the creation of the original Sub-merchant.

### Parent Elements:

[subMerchantCreateResponse](#)

### Attributes:

None

### Child Elements:

[address](#), [disabled](#), [bankAccountNumber](#), [bankRoutingNumber](#), [customerServiceNumber](#), [amexMid](#), [discoverConveyedMid](#), [hardCodedBillingDescriptor](#), [maxTransactionAmount](#), [merchantCategoryCode](#), [merchantName](#), [url](#), [subMerchantId](#), [transactionId](#), [merchantIdentString](#)

## 4.88 overallScore

The `overallScore` element is an optional child of the `verificationResult` element and contains child elements providing the results of background checks performed on the Legal Entity.

### Parent Elements:

[verificationResult](#)

### Attributes:

None

### Child Elements:

[score](#), [description](#)

### Example: overallScore Structure

```
<overallScore>
  <score>Overall results for Business</score>
  <description>Text Description of Score</description>
</overallScore>
```

## 4.89 password

The `password` element is a required child of the `credentials` element and specifies the password portion of the authentication credentials used to submit transaction to the production environment.

**Type** = String; **minLength** = 1; **maxLength** = 72

### Parent Elements:

[credentials](#)

### Attributes:

None

### Child Elements:

None

## 4.90 passwordExpirationDate

The `passwordExpirationDate` is an optional of the `credentials` element that specifies the expiration date of the provided password.

**Type** = `dateTime`; **Format** = `YYYY-MM-DDTHH:MM:SS+/-HH:MM`

---

<b>NOTE:</b>	The <code>+/-HH:MM</code> portion of the <code>dateTime</code> format represents the offset from UTC time. For example, for Boston the value would be <code>-05:00</code> , meaning five hours behind UTC time.
--------------	---

---

### Parent Elements:

[credentials](#)

### Attributes:

None

### Child Elements:

None

## 4.91 paypageCredential

The `paypageCredential` element is a required child of the `paypageCredentials` element and contains child elements that define the `paypageId` and associated `userName` used to submit card info via the PayPage application.

### Parent Elements:

[paypageCredentials](#)

### Attributes:

None

### Child Elements (required):

[userName](#), [paypageId](#)

### Example: paypageCredential Structure

```
<paypageCredentials>
  <userName>UserName</userName>
  <paypageId>Asd23thI974Jpk32</paypageId>
</paypageCredentials>
```

## 4.92 paypageCredentials

The `paypageCredentials` element is an optional child of the `subMerchantCreateResponse` and `subMerchantRetrievalResponse` elements and contains child elements that define the `paypageId` and associated `userName` used to submit card info via the PayPage application.

**Parent Elements:**

[subMerchantCreateResponse](#), [subMerchantRetrievalResponse](#)

**Attributes:**

None

**Child Elements (required):**

[paypageCredential](#)

## 4.93 paypageId

The `paypageId` element is an optional child of the `paypageCredential` element and specifies the PayPage identifier used to submit PayPage transactions to the production environment. The system returns a `paypageId` only if the Sub-merchant uses the Vault/PayPage feature and you set the `createCredentials` element to true in the create request.

**Type** = String; **minLength** = 1; **maxLength** = 50

### Parent Elements:

[paypageCredential](#)

### Attributes:

None

### Child Elements:

None

## 4.94 phone

The `phone` element is a required child of the `primaryContact` element. It defines the phone number for the Sub-merchant's primary contact.

**Type** = String; **minLength** = 1; **maxLength** = 13

### Parent Elements:

[primaryContact](#)

### Attributes:

None

### Child Elements:

None



## 4.95 phoneVerified

The `phoneVerified` element is an optional child of the `verificationIndicators` element. A value of **true** indicates that the phone number (of the business) was verified, while a value of **false** indicates that the phone number could not be verified.

**Type** = Boolean; **Allowed values** = true or false

### Parent Elements:

[verificationIndicators](#)

### Attributes:

None

### Child Elements:

None

## 4.96 postalCode

Depending upon where used, this is either the postal code of the Legal Entity controlling the Sub-merchant or of the principal, or of the Sub-merchant. This `postalCode` element is always required for a Legal Entity and for a principal if the Legal Entity is a Sole Proprietorship (`legalEntityType`).

**Type** = String; **minLength** = 1; **maxLength** = 7

### Parent Elements:

[address](#)

### Attributes:

None

### Child Elements:

None

## 4.97 primaryContact

The `primaryContact` element is an optional child of both the `subMerchantCreateRequest` and the `subMerchantUpdateRequest` transactions. The children of this element specify information about the primary contact of the Sub-merchant.

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements (all required):

[firstName](#), [lastName](#), [phone](#), [emailAddress](#)

### Example: primaryContact

```
<primaryContact>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <emailAddress>John.Doe@company.com</emailAddress>
  <phone>9785552222</phone>
</primaryContact>
```

## 4.98 principal

The `principal` element has multiple uses in the schema. In two instances it is the parent for the child elements used to provide information about the Legal Entity principal owners/officers. Its structure changes based upon whether it is part of a `legalEntityCreateRequest`, `legalEntityUpdateRequest`, `legalEntityPrincipalCreateRequest`, or a `legalEntityResponse`. It is also a child of the `backgroundCheckResults` element.

### Parent Elements:

[legalEntityCreateRequest](#), [legalEntityCreateResponse](#), [legalEntityUpdateRequest](#), [backgroundCheckResults](#), [legalEntityPrincipalCreateRequest](#),

### Attributes:

None

### Child Elements (when used in `legalEntityCreateRequest` or `legalEntityPrincipalCreateRequest`):

---

<b>NOTE:</b>	Although part of the schema structure, the <code>principalId</code> element is not used in a <code>legalEntityCreateRequest</code> message.
	Several of the elements listed below are optional in the schema, but now required per regulations.

---

Required: [firstName](#), [lastName](#), [ssn](#), [address](#), [dateOfBirth](#), [stakePercent](#)

Optional: [emailAddress](#), [contactPhone](#), [driversLicense](#), [driversLicenseState](#), [title](#)

### Example: `principal` Structure when used in `legalEntityCreateRequest` and `legalEntityPrincipalCreateRequest`

```
<principal>
  <title>Chief Financial Officer</title>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <emailAddress>jdoe@mail.net</emailAddress>
  <ssn>012345678</ssn>
  <contactPhone>9785551234</contactPhone>
  <dateOfBirth>1958-12-12</dateOfBirth>
  <driversLicense>S123456789</driversLicense>
  <driversLicenseState>MA</driversLicenseState>
  <address>
```

```
<streetAddress1>100 Main Street</streetAddress1>
<streetAddress2>Suite 400</streetAddress2>
<city>Lowell</city>
<stateProvince>MA</stateProvince>
<postalCode>01852</postalCode>
<countryCode>USA</countryCode>
</address>
<stakePercent>25%</stakePercent>
</principal>
```

**Child Elements (when used in legalEntityCreateResponse):**

[principalId](#), [firstName](#), [lastName](#), [responseCode](#), [responseDescription](#), [backgroundCheckResults](#)

**Example: principal Structure when used in a legalEntityCreateResponse**

```
<principal>
  <principalId>1234567890</principalId>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <backgroundCheckResults>
    <business>
      <verificationResult>
        <overallScore>
          <score>50</score>
          <description>Business verified on multiple sources - no contradictory
findings</description>
        </overallScore>
      </verificationResult>
    </business>
    <nameAddressTaxIdAssociation/>
    <nameAddressPhoneAssociation/>
    <verificationIndicators>
      <nameVerified>false</nameVerified>
      <addressVerified>false</addressVerified>
      <cityVerified>false</cityVerified>
      <stateVerified>false</stateVerified>
      <zipVerified>false</zipVerified>
      <phoneVerified>false</phoneVerified>
      <taxIdVerified>false</taxIdVerified>
    </verificationIndicators>
  </backgroundCheckResults>
</principal>
```

```
</verificationIndicators>
<riskIndicators/>
</verificationResult>
</business>
<principal>
  <verificationResult>
    <overallScore>
      <score>50</score>
      <description>Full name, address, phone , SSN verified</description>
    </overallScore>
    <nameAddressSsnAssociation/>
    <nameAddressPhoneAssociation/>
    <verificationIndicators>
      <nameVerified>false</nameVerified>
      <addressVerified>false</addressVerified>
      <phoneVerified>false</phoneVerified>
      <ssnVerified>false</ssnVerified>
      <dobVerified>false</dobVerified>
    </verificationIndicators>
    <riskIndicators/>
  </verificationResult>
</principal>
<businessToPrincipalAssociation>
  <score>50</score>
  <description>Association found between the person and the company in the
Company Contacts data</description>
</businessToPrincipalAssociation>
  <bankruptcyData>
    <bankruptcyType>btype</bankruptcyType>
    <bankruptcyCount>3</bankruptcyCount>
    <companyName>testCompanyName</companyName>
    <streetAddress1>testStreetAddress1</streetAddress1>
    <streetAddress2>testStreetAddress2</streetAddress2>
    <city>testCity</city>
    <state>ST</state>
```

```

    <zip>bZip5</zip>
    <zip4>Zip4</zip4>
    <filingDate>1999-11-05</filingDate>
  </bankruptcyData>
  <lienResult>
    <lienType>ltype</lienType>
    <releasedCount>3</releasedCount>
    <companyName>testLienCompanyName</companyName>
    <streetAddress1>testLienStreetAddress1</streetAddress1>
    <streetAddress2>testLienStreetAddress2</streetAddress2>
    <city>testLienCity</city>
    <state>ST</state>
    <zip>bZip5</zip>
    <zip4>Zip4</zip4>
    <filingDate>1999-11-05</filingDate>
  </lienResult>
</backgroundCheckResults>
</principal>

```

**Example:** principal Structure when used in legalEntityUpdateRequest

```

<principal>
  <principalId>4</principalId>
  <title>CF0</title>
  <emailAddress>jdoe@mail.net</emailAddress>
  <contactPhone>9785551234</contactPhone>
  <address>
    <streetAddress1>100 Main Street</streetAddress1>
    <streetAddress2>Suite 400</streetAddress2>
    <city>Lowell</city>
    <stateProvince>MA</stateProvince>
    <postalCode>01852</postalCode>
    <countryCode>USA</countryCode>
  </address>
  <stakePercent>33</stakePercent>
  <backgrgroundCheckFields>

```

```
<firstName>John</firstName>
<lastName>Doe</lastName>
<ssn>012345678</ssn>
<dateOfBirth>1958-12-12</dateOfBirth>
<driversLicense>S123456789</driversLicense>
<driversLicenseState>MA</driversLicenseState>
<backgroundCheckFields>
</principal>
```



## 4.99 principalId

The `principalId` element is a required child of the `principal` element, when used in a `legalEntityCreateResponse` message. It is a system generated identifier for the principal/owner.

---

**NOTE:** Although part of the schema structure, this element is not used in a `legalEntityCreateRequest` message.

---

**Type** = Integer; **minLength** = N/A; **maxLength** = 10

**Parent Elements:**

`principal`, `legalEntityPrincipalCreateResponse`

**Attributes:**

None

**Child Elements:**

None

## 4.100 pspMerchantId

The `pspMerchantId` element is a required child of the `subMerchantCreateRequest` element and is the PayFac supplied identifier string for the Sub-merchant within the PayFac's systems.

**Type** = String; **minLength** = 1; **maxLength** = 32

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantRetrievalResponse](#), [subMerchantUpdateRequest](#)

### Attributes:

None

### Child Elements:

None

## 4.101 purchaseCurrency

The `purchaseCurrency` element is an optional child of the `subMerchantCreateRequest` element and defines the purchase currency of transactions from this Sub-merchant. If this element is not included in the create request, the purchase currency value defaults to the value defined in the Sub-merchant template used to provision other Sub-merchants associated with this Legal Entity. When included in the create request, the `purchaseCurrency` element is also returned in the Sub-merchant retrieval request and the Sub-merchant create response if the create request was a duplicate.

**Type** = String; **minLength** = 3; **maxLength** = 3

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantRetrievalResponse](#), [subMerchantCreateResponse](#) (if duplicate), [subMerchantUpdateRequest](#)

### Attributes:

None

### Child Elements:

None

## 4.102 releasedCount

The `releasedCount` element is an optional child of the `lienResult` element and specifies the count of released lien records found.

**Type** = int; **totalDigits** = N/A

### Parent Elements:

[lienResult](#)

### Attributes:

None

### Child Elements:

None

## 4.103 response

The response element is the parent of the XML message acknowledging the receipt of a Legal Entity Update (legalEntityUpdateRequest), Sub-merchant update (subMerchantUpdateRequest), or Disable Sub-merchant message.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and must be specified as: <code>http://payfac.vantivcn.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50

### Child Elements:

[transactionId](#)

## 4.104 responseCode

The `responseCode` element is an optional child of the `legalEntityResponse` message. The code indicate either success or the reason for the failure of the request. Possible values are listed in the table below.

**Type** = String; **minLength** = N/A; **maxLength** = 2

**Parent Elements:**

[legalEntityResponse](#), [legalEntityRetrievalResponse](#)

**Attributes:**

None

**Child Elements:**

None

**Possible Values:**

Code	responseDescription
10	Approved
20	Manual Review
30	Retry
35	Manual Review - Duplicate
36	Duplicate
40	Declined
98	Manual Review - End Point Down
99	Manual Review - Background Check Error

---

**NOTE:** If you receive a `responseCode` 30 - Retry, the decision notes will contain additional details about the information that you should provide/correct on the retry.

---

## 4.105 responseDescription

The `responseDescription` element is an optional child of the `legalEntityResponse` message and is a text description of the `responseCode`.

**Type** = String; **minLength** = 1; **maxLength** = 100

### Parent Elements:

[legalEntityResponse](#), [legalEntityRetrievalResponse](#), [legalEntityResponse](#)

### Attributes:

None

### Child Elements:

None

### Possible Values:

responseDescription	Code
Approved	10
Manual Review	20
Retry	30
Manual Review - Duplicate	35
Duplicate	36
Declined	40
Manual Review - Background Check Error	99

---

**NOTE:** If you receive a `responseCode` 30 - Retry, the decision notes will contain additional details about the information that you should provide/correct on the retry.

---

## 4.106 riskIndicator

The `verificationResult` element is an optional child of the `business` and `principal` elements and contains child elements providing the results of background checks performed on the Legal Entity.

### Parent Elements:

[riskIndicators](#)

### Attributes:

None

### Child Elements:

[code](#), [description](#)

### Example: riskIndicator Structure

```
<riskIndicator>
  <code>Risk Indicator Info</code>
  <description>Text Description of Code</description>
</riskIndicator>
```



## 4.107 riskIndicators

The `riskIndicators` element is an optional child of the `verificationResult` element and contains child elements providing the results of background checks performed on the Legal Entity.

### Parent Elements:

[verificationResult](#)

### Attributes:

None

### Child Elements:

[riskIndicator](#)

### Example: riskIndicators Structure

```
<riskIndicators>
  <riskIndicator>
    <code>Risk Indicator Info</code>
    <description>Text Description of Code</description>
  </riskIndicator>
</riskIndicators>
```

## 4.108 score

Depending upon where used, this element defines overall score assigned based on the background checks of either the business, the principal, or the business-to-principal association.

**Type** = String (Enum); **minLength** = N/A; **maxLength** = N/A

### Parent Elements:

[overallScore](#), [businessToPrincipalAssociation](#)

### Attributes:

None

### Child Elements:

None

### Enumerations: (Higher number is better.)

Enums	Definition when Descendant of business	Definition when Descendant of principal	Definition when Descendant of businessToPrincipalAssociation
0	Nothing found to confirm existence of business.	Nothing verified - ID elements not found or are associated with a different person.	No information found to link Principal to Business
10	One or more of the following: <ul style="list-style-type: none"> <li>Significant contradictory findings</li> <li>OFAC match exists</li> <li>Input address and phone are invalid</li> </ul>	One or more of the following: <ul style="list-style-type: none"> <li>Critical ID elements not verified or are associated with a different person</li> <li>OFAC matches</li> <li>deceased/invalid SSNs</li> <li>SSN issued prior to DOB</li> <li>records with invalid addresses and phones</li> </ul>	Principal's verified name partially matches input Business name.
20	Partial verification of the input data; the business may exist, but cannot be positively confirmed.	Minimal verifications; critical ID elements not verified or are associated with a different person	Principal's verified address matches input Business address.

<b>Enums</b>	<b>Definition when Descendant of business</b>	<b>Definition when Descendant of principal</b>	<b>Definition when Descendant of businessToPrincipalAs sociation</b>
30	Business identity is confirmed, but failure to verify all identity elements.	Several ID elements verified; SSN or address verification failures exist.	Principal's verified address matches input Business address and Principal's verified name partially matches input Business name.
40	Business identity is confirmed at the input address	Last name, address, and SSN verified; first name or phone verification failures.	Principal linked to Business by SSN /Tax Id match.
50	Business is verified using multiple sources. no contradictory findings.	Full name, address, phone, and SSN verified.	Principal is likely an officer or employee of Business confirmed by public records.

## 4.109 settlementCurrency

The `settlementCurrency` element is an required child of the `subMerchantCreateRequest` element and defines the settlement currency of transactions from this Sub-merchant.

**Type** = String; **minLength** = 3; **maxLength** = 3

**Parent Elements:**

[subMerchantCreateRequest](#), [subMerchantRetrievalResponse](#)

**Attributes:**

None

**Child Elements:**

None

## 4.110 ssn

The `ssn` element is a child of the `principal` element and specifies the Social Security Number of the principal. If the `entityType` is `INDIVIDUAL_SOLE_PROPRIETORSHIP`, you must submit the principal's SSN. For other Legal Entity types, you can substitute the Tax ID for the SSN in this element.

**Type** = String; **minLength** = 9; **maxLength** = 25

### Parent Elements:

[principal](#)

### Attributes:

None

### Child Elements:

None

## 4.111 ssnVerified

The `ssnVerified` element is an optional child of the `verificationIndicators` element, when `verificationIndicator` is a descendant of the `principal` element. A value of **true** indicates that the SSN of the principal was verified, while a value of **false** indicates that the SSN could not be verified.

**Type** = Boolean; **Allowed values** = true or false

### Parent Elements:

[verificationIndicators](#)

### Attributes:

None

### Child Elements:

None

## 4.112 stakePercent

The `stakePercent` element is a required child of the `principal` element and defines the percentage of the Legal Entity owned by the principal.

**Type** = Integer; **Minimum value** = 0; **Maximum value** = 100

### Parent Elements:

[principal](#)

### Attributes:

None

### Child Elements:

None

## 4.113 state

The state element is an optional child of both the `bankruptcyData` and `lienResults` elements, where it specifies the state of the company with bankruptcy or lien information.

**Type** = String; **minLength** = N/A; **maxLength** = 2

### Parent Elements:

[bankruptcyData](#), [lienResult](#)

### Attributes:

None

### Child Elements:

None



## 4.114 stateVerified

The `cityVerified` element is an optional child of the `verificationIndicators` element, when `verificationIndicators` is a descendant of the `business` element. A value of **true** indicates that the city (of the business) was verified, while a value of **false** indicates that the city could not be verified.

**Type** = Boolean; **Allowed values** = true or false

### Parent Elements:

[verificationIndicators](#)

### Attributes:

None

### Child Elements:

None

## 4.115 stateProvince

Depending upon where used, this is either the code representing the state of the business entity controlling the Sub-merchant or of the principal. This `stateProvince` element is always required for a Legal Entity and for a principal if the Legal Entity is a Sole Proprietorship (`legalEntityType`). If the Legal Entity is not a Sole Proprietorship, this element is optional.

**Type** = String; **minLength** = 1; **maxLength** = 2

### Parent Elements:

[address](#)

### Attributes:

None

### Child Elements:

None

## 4.116 streetAddress1, streetAddress2

The elements `streetAddress1` and `streetAddress2` define the address information. The `streetAddress1` element is always required for a Legal Entity. It is also required for the principal if the Legal Entity is a Sole Proprietorship (`legalEntityType`). The `streetAddress2` element is always optional.

**Type** = String; **minLength** = 1; **maxLength** = 60

### Parent Elements:

[address](#), [bankruptcyData](#), [lienResult](#)

### Attributes:

None

### Child Elements:

None

## 4.117 subMerchantCreateRequest

The `subMerchantCreateRequest` element is the parent element for the XML message used to create a Sub-merchant associated with a Legal Entity. You must create a Legal Entity prior to adding Sub-merchants controlled by the entity.

---

**NOTE:** After the creation of a Sub-merchant, wait a minimum of two minutes before attempting to process transactions for the Sub-merchant. This is the minimum amount of time required for information about the newly created Sub-merchant to propagate through our system. Attempts to process transactions for a new Sub-merchant sooner than two minutes will result in system errors.

---

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and must be specified as: <code>http://payfac.vantivcn.com/api/merchant/onboard</code> <b>minLength = N/A maxLength = 50</b>

### Child Elements:

Required: [merchantName](#), [customerServiceNumber](#), [hardCodedBillingDescriptor](#), [maxTransactionAmount](#), [merchantCategoryCode](#), [bankRoutingNumber](#), [bankAccountNumber](#), [pspMerchantId](#), [address](#) (some child elements may be optional), [settlementCurrency](#)

Required/Optional (depending upon if using PayFac Assurance model): [url](#)

Optional: [address](#), [amexMid](#), [discoverConveyedMid](#), [purchaseCurrency](#), [primaryContact](#), [createCredentials](#), [eCheck](#), [fraud](#), [subMerchantFunding](#)

## 4.118 subMerchantCreateResponse

The subMerchantResponse element is the parent element for the XML message returned by the platform in response to a subMerchantCreateRequest message.

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and must be specified as: <code>http://payfac.vantivcn.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50
duplicate	Boolean	No	Appears in the response message only if the submerchantCreateRequest was a duplicate request. <b>Allowed values</b> = true or false

### Child Elements:

Required: [transactionId](#), [merchantIdentString](#)

Optional: [subMerchantId](#), [originalSubMerchant](#)

## 4.119 subMerchantFunding

The subMerchantFunding element is an optional child of the subMerchantCreateRequest, the subMerchantUpdateRequest elements, as well as the subMerchantRetrievalResponse. Through its child elements, it defines either the Fee Profile applied to the funding of the Sub-merchant, when the Managed Payout funding feature is enabled, or the fundingSubMerchantId, when the Dynamic Payout feature is enabled.

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

Attribute Name	Type	Required?	Description
enabled	Boolean	Yes	Set to true and include the fundingSubmerchantId element when using Dynamic Payout. Set to true and include the feeProfile element when using Managed Payout for the Sub-merchant. If neither condition applies (i.e., you are funding directly), set to false. <b>Allowed values</b> = true or false

### Child Elements (optional):

[feeProfile](#) (required for Managed Payout), [fundingSubmerchantId](#) (required for Dynamic Payout)

### Example: subMerchantFunding Structure for Managed Payout

```
<subMerchantFunding enabled="true">  
  <feeProfile>Name of Fee Profile</feeProfile>  
</subMerchantFunding>
```

### Example: subMerchantFunding Structure for Dynamic Payout

```
<subMerchantFunding enabled="true">  
  <fundingSubmerchantId>Unique Funding Id</fundingSubmerchantId>  
</subMerchantFunding>
```

## 4.120 subMerchantId

The `subMerchantId` element is an optional child of the `subMerchantCreateResponse` element and is a system generated identifier for this Sub-merchant within the legal entity. You use this value as a parameter in the calls when updating, retrieving, and disabling a Sub-merchant.

---

**NOTE:** This value is unique within the Legal Entity, but not across Legal Entities.

---

**Type** = String; **minLength** = N/A; **maxLength** = 19

**Parent Elements:**

[subMerchantCreateResponse](#), [subMerchantRetrievalResponse](#)

**Attributes:**

None

**Child Elements:**

None

## 4.121 sub\_merchant\_processing\_status

The sub\_merchant\_processing\_status element is a required child of the legalEntityRetrievalResponse and indicates if you can create sub-merchants associated with the Legal Entity, based upon the state of the Legal Entity. A value of true indicates that you can create sub-merchants.

**Type** = Boolean; **Allowed values** = true or false.

### Parent Elements:

[legalEntityRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None



## 4.122 subMerchantRetrievalResponse

The subMerchantRetrievalResponse element is the parent element for the XML message returned by the platform in response to a Sub-merchant Retrieval Request.

---

**NOTE:** If you are processing transactions only on the Vantiv eCommerce platform and using Dynamic Payout, after creating a Sub-merchant using a value of AUTO\_GENERATE for the fundingSubmerchantId, retrieve the assigned fundingSubmerchantId value by submitting a Sub-merchant Retrieval Request.

---

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
xmlns	String	Yes	Defines the URI of the schema definition. This is a fixed location and must be specified as: <code>http://payfac.vantivcn.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50

### Child Elements:

merchantName, amexMid, discoverConveyedMid, url, customerServiceNumber, hardCodedBillingDescriptor, maxTransactionAmount, purchaseCurrency, merchantCategoryCode, bankRoutingNumber, bankAccountNumber, pspMerchantId, fraud, address, primaryContact, createCredentials, eCheck, subMerchantFunding, settlementCurrency, subMerchantId, amexSellerId, disabled, transactionId, merchantIdentString, credentials, paypageCredentials, updateDate

## 4.123 subMerchantUpdateRequest

The `subMerchantUpdateRequest` element is the parent element for the XML message used to update a Sub-merchant associated with a Legal Entity. You can update the following items:

- `url`
- `customerServiceNumber`
- `hardCodedBillingDescriptor`
- `maxTransactionAmount`
- `bankRoutingNumber`
- `bankAccountNumber`
- `eCheckBillingDescriptor`
- `streetAddress1`
- `streetAddress2`
- `city`
- `stateProvince`
- `postalCode`
- `primaryContact`
- `eCheckCompanyName`

### Parent Elements:

None

### Attributes:

Attribute Name	Type	Required?	Description
<code>xmlns</code>	String	Yes	Defines the URI of the schema definition. This is a fixed location and must be specified as: <code>http://payfac.vantivcn.com/api/merchant/onboard</code> <b>minLength</b> = N/A <b>maxLength</b> = 50

### Child Elements:

`amexMid`, `discoverConveyedMid`, `url`, `customerServiceNumber`, `hardCodedBillingDescriptor`, `maxTransactionAmount`, `bankRoutingNumber`, `bankAccountNumber`, `address`, `primaryContact`, `disable`, `fraud`, `eCheck`, `subMerchantFunding`

## 4.124 taxAuthority

The `taxAuthority` element is an optional child of both the `submerchantCreateRequest` and the `submerchantUpdateRequest` elements; however, you must specify a value if the MCC of the sub-merchant is 9311 (Tax Payments - Government Agencies). The element specifies the Tax Authority (taxing body) for which the sub-merchant collects tax payments.

**Type** = String; **minLength** = N/A; **maxLength** = N/A

---

**NOTE:** Although **maxLength** is not defined in the schema, the maximum length of the **taxAuthority + Billing Descriptor prefix** cannot exceed 20 characters. Please ask your Implementation consultant for additional information.

---

### Parent Elements:

`subMerchantCreateRequest`, `subMerchantUpdateRequest`

### Attributes:

None

### Child Elements:

None

## 4.125 taxAuthorityState

The `taxAuthorityState` element is an optional child of both the `submerchantCreateRequest` and the `submerchantUpdateRequest` elements; however, you must specify a value if the MCC of the sub-merchant is 9311 (Tax Payments - Government Agencies). The element specifies the state for which the sub-merchant collects tax payments.

**Type** = String; **minLength** = 1; **maxLength** = 2

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#)

### Attributes:

None

### Child Elements:

None

## 4.126 taxId

The `taxId` element is a required child of the `legalEntityCreateRequest` element and defines the Tax ID (or EIN if applicable) of the Legal Entity controlling the Sub-merchants. If the `entityType` is `INDIVIDUAL_SOLE_PROPRIETORSHIP` and a Tax ID is not available, you can include the principal's SSN in this element.

**Type** = String; **minLength** = 9; **maxLength** = 25

### Parent Elements:

[legalEntityCreateRequest](#), [legalEntityRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.127 taxIdVerified

The `taxIdVerified` element is an optional child of the `verificationIndicators` element, when `verificationIndicators` is a descendant of the `business` element. A value of **true** indicates that the Tax Id (of the business) was verified, while a value of **false** indicates that the Tax Id could not be verified.

**Type** = Boolean; **Allowed values** = true or false

### Parent Elements:

[verificationIndicators](#)

### Attributes:

None

### Child Elements:

None

## 4.128 tinValidationStatus

The `tinValidationStatus` element is an optional child of the `legalEntityRetrievalResponse` element and specifies the status of the Tax Id Number validation operation. Currently, possible values are: Approved, Failed, Pending, and Not Required.

**Type** = String; **minLength** = 1; **maxLength** = 20

### Parent Elements:

[legalEntityRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.129 title

The `title` element is an optional child of the `principal` element that specifies the title of the principal. It is, however, required when your Organization is configured (in your Merchant Profile) to use the PayFac Assurance model. In this case, failure to include the element will result in a validation error.

---

<b>NOTE:</b>	<b>Although the title element is optional, you must include a value if you support American Express Opt Blue transactions.</b>
--------------	--

---

**Type** = String; **minLength** = N/A; **maxLength** = 60

### Parent Elements:

[principal](#)

### Attributes:

None

### Child Elements:

None



## 4.130 transactionId

The `transactionId` element is a required child of the `legalEntityResponse` element and is a system generated identifier for the submitted request.

**Type** = String; **minLength** = 1; **maxLength** = 19

### Parent Elements:

[legalEntityAgreementCreateResponse](#), [legalEntityResponse](#), [legalEntityRetrievalResponse](#), [subMerchantCreateResponse](#), [response](#), [errorResponse](#), [subMerchantRetrievalResponse](#), [approvedMccResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.131 unreleasedCount

The `unreleasedCount` element is an optional child of the `lienResult` element and specifies the count of unreleased lien records found.

**Type** = int; **totalDigits** = N/A

### Parent Elements:

[lienResult](#)

### Attributes:

None

### Child Elements:

None

## 4.132 updateDate

The updateDate element is an optional child of both the legalEntityRetrievalResponse and the subMerchantRetrievalResponse element. The value represents the last date/time an update was performed to the Legal Entity/Sub-Merchant.

**Type** = dateTime; **Format** = YYYY-MM-DDTHH:MM:SS+/-HH:MM

---

---

**NOTE:** The +/-HH:MM portion of the dateTime format represents the offset from UTC time. For example, for Boston the value would be -05:00, meaning five hours behind UTC time.

---

---

### Parent Elements:

[legalEntityRetrievalResponse](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.133 url

The `url` element is a required child of the `subMerchantCreateRequest` element and specifies the URL of the Sub-merchant.

**Type** = String; **minLength** = 1; **maxLength** = 120

### Parent Elements:

[subMerchantCreateRequest](#), [subMerchantUpdateRequest](#), [subMerchantRetrievalResponse](#)

### Attributes:

None

### Child Elements:

None

## 4.134 userFullName

The `userFullName` element is a required child of the `legalEntityAgreement` element and defines name of the party signing the Legal Entity Agreement.

**Type** = String; **minLength** = 1; **maxLength** = 50

### Parent Elements:

[legalEntityAgreement](#)

### Attributes:

None

### Child Elements:

None

## 4.135 userIPAddress

The `userIPAddress` element is a child of the `legalEntityAgreement` element and defines the IP Address of the party signing the Legal Entity Agreement. This element is required when the `manuallyEntered` element is set to **false**. It is not required when the `manuallyEntered` element is set to **true**. The only allowed characters are lower-case a-z, upper-case A-Z, numbers 0-9, colon (:), and a period (.).

**Type** = String; **minLength** = 1; **maxLength** = 40

### Parent Elements:

[legalEntityAgreement](#)

### Attributes:

None

### Child Elements:

None

## 4.136 userName

The `username` element is a required child of the `credentials` element and specifies the user name portion of the authentication credentials used to submit transaction to the production environment.

**Type** = String; **minLength** = 1; **maxLength** = 72

---

---

**NOTE:** Although the V4 schema specifies this element as `username`, the actual element returned in the response file is `userName`.

---

---

### Parent Elements:

`credentials`, `paypageCredential`

### Attributes:

None

### Child Elements:

None

## 4.137 userSystemName

The `userSystemName` element is a required child of the `legalEntityAgreement` element and defines name the party signing the Legal Entity Agreement uses on the PayFac system.

**Type** = String; **minLength** = 1; **maxLength** = 50

### Parent Elements:

[legalEntityAgreement](#)

### Attributes:

None

### Child Elements:

None



## 4.138 verificationIndicators

The `verificationIndicators` element is an optional child of the `verificationResult` element and contains child elements that indicated whether or not certain data points about the Legal Entity were verified during the background checks.

### Parent Elements:

[verificationResult](#)

### Attributes:

None

### Child Elements:

Common: [nameVerified](#), [addressVerified](#), [phoneVerified](#)

Descendant of bussiness: [cityVerified](#), [stateVerified](#), [zipVerified](#), [taxIdVerified](#)

Descendant of principal: [ssnVerified](#), [dobVerified](#)

### Example: verificationIndicators Structure (when descendant of bussiness)

```
<verificationIndicators>
  <nameVerified>true or false</nameVerified>
  <addressVerified>true or false</addressVerified>
  <cityVerified>true or false</cityVerified>
  <stateVerified>true or false</stateVerified>
  <zipVerified>true or false</zipVerified>
  <phoneVerified>true or false</phoneVerified>
  <taxIdVerified>true or false</taxIdVerified>
</verificationIndicators>
```

### Example: verificationIndicators Structure (when descendant of principal)

```
<verificationIndicators>
  <nameVerified>true or false</nameVerified>
  <addressVerified>true or false</addressVerified>
  <phoneVerified>true or false</phoneVerified>
  <ssnVerified>true or false</ssnVerified>
  <dobVerified>true or false</dobVerified>
```

</verificationIndicators>

## 4.139 verificationResult

The `verificationResult` element is an optional child of the `business` and `principal` elements and contains child elements providing the results of background checks performed on the Legal Entity.

### Parent Elements:

[business](#), [principal](#)

### Attributes:

None

### Child Elements:

Common: [overallScore](#), [nameAddressPhoneAssociation](#), [verificationIndicators](#), [riskIndicators](#)

Descendant of `business`: [nameAddressTaxIdAssociation](#)

Descendant of `principal`: [nameAddressSsnAssociation](#)

### Example: verificationResult Structure (as child of business)

```
<verificationResult>
  <overallScore>
    <score>Overall results for Business</score>
    <description>Text Description of Score</description>
  </overallScore>
  <nameAddressTaxIdAssociation>
    <code>Name_Address_TIN_info</code>
    <description>Text Description of Code</description>
  </nameAddressTaxIdAssociation>
  <nameAddressPhoneAssociation>
    <code>Name_Address_Phone_info</code>
    <description>Text Description of Code</description>
  </nameAddressPhoneAssociation>
  <verificationIndicators>
    <nameVerified>true or false</nameVerified>
    <addressVerified>true or false</addressVerified>
    <cityVerified>true or false</cityVerified>
    <zipVerified>true or false</zipVerified>
  </verificationIndicators>
</verificationResult>
```

```
<phoneVerified>true or false</phoneVerified>
<taxIdVerified>true or false</taxIdVerified>
</verificationIndicators>
<riskIndicators>
  <riskIndicator>
    <code>Risk Indicator Info</code>
    <description>Text Description of Code</description>
  </riskIndicator>
</riskIndicators>
</verificationResult>
```

**Example: verificationResult Structure (as child of principal)**

```
<verificationResult>
  <overallScore>
    <score>Overall Results for Principal</score>
    <description>Text Description of Score</description>
  </overallScore>
  <nameAddressSsnAssociation>
    <code>Name_Address_SSN_info</code>
    <description>Text Description of Code</description>
  </nameAddressSsnAssociation>
  <nameAddressPhoneAssociation>
    <code>Name_Address_Phone_info</code>
    <description>Text Description of Code</description>
  </nameAddressPhoneAssociation>
  <verificationIndicators>
    <nameVerified>true or false</nameVerified>
    <addressVerified>true or false</addressVerified>
    <phoneVerified>true or false</phoneVerified>
    <ssnVerified>true or false</ssnVerified>
    <dobVerified>true or false</dobVerified>
  </verificationIndicators>
  <riskIndicators>
    <riskIndicator>
      <code>Risk Indicator results</code>
```

```
        <description>Text Description of Code</description>
      </riskIndicator>
    </riskIndicators>
  </verificationResult>
```

## 4.140 yearsInBusiness

The `yearsInBusiness` element is an optional child of the `legalEntityCreateRequest` element. It is, however, required when your Organization is configured (in your Merchant Profile) to use the PayFac Assurance model. In this case, failure to include this element will result in a validation error.

**Type** = Integer; **minLength** = N/A; **maxLength** = 3

### Parent Elements:

[legalEntityCreateRequest](#), [legalEntityUpdateRequest](#)

### Attributes:

None

### Child Elements:

None

## 4.141 zip

The `zip` element defines the company postal code in both the `bankruptcyData` and `lienResults` elements. The element is optional.

**Type** = String; **minLength** = N/A; **maxLength** = 5

### Parent Elements:

[bankruptcyData](#), [lienResult](#)

### Attributes:

None

### Child Elements:

None

## 4.142 zip4

The `zip4` element defines the extended +4 portion of the company postal code in both the `bankruptcyData` and `lienResults` elements. The element is optional.

**Type** = String; **minLength** = N/A; **maxLength** = 4

### Parent Elements:

[bankruptcyData](#), [lienResult](#)

### Attributes:

None

### Child Elements:

None



## 4.143 zipVerified

The `zipVerified` element is an optional child of the `verificationIndicators` element, when `verificationIndicators` is a descendant of the `business` element. A value of **true** indicates that the zip code (of the business) was verified, while a value of **false** indicates that the zip code could not be verified.

**Type** = Boolean; **Allowed values** = true or false

### Parent Elements:

[verificationIndicators](#)

### Attributes:

None

### Child Elements:

None