

NETBANX®

NETBANX Recurring Billing API

March 2011

Version 1.0

This manual and accompanying electronic media are proprietary products of Optimal Payments plc. They are to be used only by licensed users of the product.

© 1999–2011 Optimal Payments plc. All rights reserved.

The information within this document is subject to change without notice. The software described in this document is provided under a license agreement, and may be used or copied only in accordance with this agreement. No part of this manual may be reproduced or transferred in any form or by any means without the express written consent of Optimal Payments plc.

All other names, trademarks, and registered trademarks are the property of their respective owners.

Optimal Payments plc makes no warranty, either express or implied, with respect to this product, its merchantability or fitness for a particular purpose, other than as expressly provided in the license agreement of this product. For further information, please contact Optimal Payments plc.

International Head Office

3500 de Maisonneuve W., Suite 700
Montreal, Quebec H3Z 3C1
Canada

Tel.: (514) 380-2700

Fax: (514) 380-2760

Email: info@optimalpayments.com

Technical support: support@optimalpayments.com

Web: www.optimalpayments.com

U.K. Office

Third Floor, Mount Pleasant House
Mount Pleasant
Cambridge CB3 0RN
United Kingdom

Email: info@optimalpayments.co.uk

Technical Support: support@optimalpayments.co.uk

Web: www.optimalpayments.co.uk

U.S. Office

1209 Orange Street
Wilmington, DE 19801

Gatineau Office

75 Promenade du Portage
Gatineau, Quebec J8X 2J9
Canada

1 Recurring Billing API

What are NETBANX Web Services?	1-1
Web Services supported for recurring billing	1-1
System requirements	1-1
Creating recurring billing records	1-2
Recurring debit and credit transactions	1-2
Updating recurring billing records	1-3
Creating billing record segments	1-4
Loading consumer profiles	1-4
Accessing the NETBANX WSDLs and links	1-5
Testing NETBANX Web Services	1-5
Merchant files	1-5
Confirmation file contents	1-6
Using this guide	1-7
Audience	1-7
Functionality	1-7
Symbols	1-8
.NET example	1-8
Building the recurring billing request	1-10
C# CREATION example	1-10
C# UPDATE example	1-13
UploadRequestV1 schema	1-14
UploadRequestV1 elements	1-14
Building the LOAD request	1-21
C# LOAD example	1-21
ConsumerProfileRequestV1 schema	1-22
ConsumerProfileRequestV1 elements	1-23
Processing the response	1-24
UploadResponseV1	1-24
ConsumerProfileResponseV1	1-27

A Using the HTTP Post Method

Introduction	A-1
Creating a recurring billing record	A-1
Updating a recurring billing record	A-2
Loading a consumer profile	A-3
HTML examples	A-4
CREATION post	A-4
LOAD Post	A-6

Responses A-8

B Response Codes

Response Codes B-1
 Action codes B-16
Return Codes B-16

C Geographical Codes

Province codes C-1
State codes C-2
Country codes C-3

Recurring Billing API

What are NETBANX Web Services?

Web Services are a technology that allows applications to communicate with each other in a platform- and programming language-independent manner. A Web Service is a software interface that describes a collection of operations that can be accessed over the network through standardized XML messaging. It uses protocols based on the XML language to describe an operation to execute or data to exchange with another Web Service. Web Services are built on open standards such as SOAP and WSDL.

NETBANX Web Services offer the following benefits:

- Merchants can integrate easily with the Web Service API, using their favourite platform and language.
- Merchants can automate operation and avoid manually keying in information via the NETBANX Web page.
- Merchants can operate independently of changes and updates to the NETBANX Web site.

Web Services supported for recurring billing

NETBANX currently supports the following Web Services for recurring billing:

Table 1-1: Recurring Billing Operations

Operation	Description
CREATION	Allows you to create a recurring billing record for a consumer.
UPDATE	Allows you to modify information in an existing billing record.
LOAD	Allows you to download a consumer profile from NETBANX.

- The CREATION and UPDATE operations require the *UploadRequestV1* document object.
- The CREATION and UPDATE operations return an *UploadResponseV1* document object.
- The LOAD operation requires the *ConsumerProfileRequestV1* document object.
- The LOAD operation returns a *ConsumerProfileResponseV1* document object.

System requirements

The SOAP API has been tested with the following client environments:

Table 1-2: Client Environments

SOAP Client	Programming Environment	Operating Environment
Microsoft .NET 1.1 and 2.0 Framework	Microsoft Visual Studio .NET 2003	Microsoft Windows Server 2003 and Windows XP

Table 1-2: Client Environments

SOAP Client	Programming Environment	Operating Environment
Apache Axis 1.4	Java (J2SE 1.4.X and higher)	Linux and Microsoft Windows XP, Server 2003
Apache Axis2 1.2	Java (J2SE 1.4.X and higher)	Linux and Microsoft Windows XP, Server 2003

For more information:

- J2SE or J2EE 1.3.1 or newer (1.4.X recommended) Sun Microsystems, Inc.
<http://java.sun.com/downloads/index.html>
- Apache Axis 1.4, The Apache Software Foundation
http://www.apache.org/dyn/closer.cgi/ws/axis/1_4
- Apache Axis2, 1.2, The Apache Software Foundation
http://ws.apache.org/axis2/1_2/contents.html
- Microsoft .NET Framework Version 1.1/2.0 Microsoft Corporation
<http://www.microsoft.com/net/default.aspx>



Whatever SOAP client you adopt, it must support document-style messaging.

Creating recurring billing records

The NETBANX recurring billing operations allow you to create and update recurring billing records. To create a recurring billing record, you submit parameters containing information for the following:

- Consumer
- Billing/shipping address
- Contact method
- Payment method (e.g., credit card or Direct Debit)
- Billing schedule

Once you set up your recurring billing records, NETBANX does the rest, processing the recurring transactions for the amount, at the frequency, and over the time period you specify.

Recurring debit and credit transactions

With the NETBANX recurring billing API, you can set up both debit and credit recurring transactions, for both credit card and Direct Debit accounts. For example, you can set up a recurring billing record that deposits \$50 each month into your customer's credit card account. Provide one of the following values for the *transactionMode* element of your request (see Table 1-6: *UploadRequestV1 Elements* on page 1-15 for details):

- CCPURCHASE – debits an amount from a customer's credit card
- CCPAYMENT – credits an amount to a customer's credit card
- DDCHARGE – debits an amount from a customer's bank account
- DDCREDIT – credits an amount to a customer's bank account



The value of the *transactionMode* element must correspond with the *paymentMethod* selected for this record. E.g., if the *paymentMethod* is *Direct Debit*, you must select a *DD* value for *transactionMode*. If *transactionMode* is not defined, then *CCPURCHASE* is the default for a credit card *paymentMethod* and *DDCHARGE* is the default for a *Direct Debit* *paymentMethod*.

See *Building the recurring billing request* on page 1-10 for complete details.

Updating recurring billing records

Once you have created recurring billing records, you can also easily modify them at any time.

- You can change the address or contact information for a consumer.
- You can modify an existing payment method. For example, you can change a credit card expiry date.
- You can change the frequency or amount of a recurring transaction.
- You can change the status of a billing record. For example, if you suspended a billing record because of a chargeback and then later resolved the situation, you can change the status back to *Active* to resume the billing activity.

In order to modify any segment of a billing record, the following elements are required in the request:

- *action* – This must be set to *UPDATE*
- *merchantAccount* – This includes the *accountNum*, *storeID*, and *storePwd* child elements.
- The segment to be updated – For example, if you want to modify the consumer information section of a billing record, you must include *consumerInfo* segment with all its child elements and any new values for those child elements.

Here is an example of a request to update the *consumerInfo* element of a billing record:

```
<merchantAccount>
  <accountNum>1000049569</accountNum>
  <storeID>test</storeID>
  <storePwd>test</storePwd>
</merchantAccount>
<consumerInfo>
  <!-- TO BE UPDATED -->
  <consumerId>7380</consumerId>
  <merchantRefNum>REF-1234</merchantRefNum>
  <firstName>Jane</firstName>
  <lastName>Jones</lastName>
  <title>MS</title>
</consumerInfo>
```

This request would replace the *merchantRefNum*, *firstName*, *lastName*, and *title* elements with the values you included here. If, for example, you wanted to change only the last name, then you would provide a new value for the *lastName* element while providing the original values for the other three elements.



You cannot overwrite the value for any *ID* element (which in this example is the *consumerId*). The *ID* included in any *UPDATE* request serves only to identify the billing record in the request.

Creating billing record segments

You can create some segments of an existing billing record using the UPDATE operation. For example, instead of just updating the information in a billing address, you can create a new billing address altogether. The resulting *billingAddressId* will then be available to use to create a new *paymentMethod* or to update an existing one.

To create a new recurring billing segment using the UPDATE operation, just omit the *id* element for the segment you want to create. You can create the following segments this way:

- *billingAddress*
- *shippingAddress*
- *contactMethod*
- *paymentMethod*
- *billingSchedule*

You cannot overwrite the *merchantAccount* segment.

Here is a partial example of a request to create a new billing address, in which the ID is commented out:

```
...
<billingAddress>
  <!--THE ID IS COMMENTED OUT
  <id> 12233</id>
  -->
  <street>133 Main Street - UPDATED</street>
  <city>Los Angeles</city>
  <state>CA</state>
  <country>US</country>
  <zip>90210</zip>
</billingAddress>
...
```



When you create a new *contactMethod* using the UPDATE operation, the new *contactMethod* becomes the default for the billing record.

Loading consumer profiles

The NETBANX LOAD operation allows you to download a complete consumer profile, returning all the recurring billing record details you provided when you created and/or updated that record. To download a consumer profile, you submit a brief request containing the consumer ID that was created when you initially created the billing record.

- The consumer ID is returned via the API as the *consumerId* in the *UploadResponseV1* response. See Table 1-8: *UploadResponseV1 Elements* on page 1-24 for details.
- The consumer ID can also be found using the Recurring Billing tool in the merchant back office. When you search for billing records by consumer, the first ID column in the Consumer Records results page is the consumer ID.

ID	Ref ID	Consumer	Email	Home Phone	Cell Phone
750	765	Sammi Samuelson	ssamuelson@email.com	333-3333	565-6565

See *Building the LOAD request* on page 1-21 for complete details.

Accessing the NETBANX WSDLs and links

Table 1-3: WSDLs and Links

Link	URL
WSDL	https://webservices.optimalpayments.com/billingWS/BillingService/v1?wsdl
Web Services	https://webservices.optimalpayments.com/billingWS/BillingService/v1
HTTP Post	https://webservices.optimalpayments.com/billingWS/billing/v1/billingUpload

Testing NETBANX Web Services

Once you have configured your Web Services–enabled application, please contact our Technical Support team for instructions on how to test your API calls.

- Email support@optimalpayments.com
- Telephone 1-888-709-8753

Merchant files

NETBANX can provide you with two automated updates associated with recurring billing functions.

- NETBANX can provide you with daily confirmations – via email or SFTP – indicating all recurring billing transactions that were processed for the day. The confirmation summary lists successful and failed transactions, as well as transactions that resulted in an error. In addition, the confirmation email has an attached CSV file containing transaction information. See *Confirmation file contents* on page 1-6 for details.
- NETBANX can notify you – via email or SFTP – of billing records that contain either an expired credit card or a credit card that will expire within the next thirty days. This notification email allows you to contact the consumer affected in order to rectify the problem (e.g., get a new credit card number) before their next recurring billing transaction fails.

In order to configure your merchant account for these merchant confirmation files, please contact Technical Support.

Confirmation file contents

Each record in the confirmation file contains nine fields.



The CSV confirmation file does not contain a header row.

Table 1-4: Confirmation File Contents

Field	Name	Data	Description
1	Consumer ID	String Max = 20	This is a consumer ID assigned by NETBANX. <ul style="list-style-type: none"> In the Recurring Billing API, this is the <i>consumerId</i> returned in the UploadResponseV1. In the Recurring Billing tool in the merchant back office, this is the <i>Consumer ID</i> on the General Information tab.
2	Billing Schedule ID	String Max = 20	This is billing schedule ID assigned by NETBANX. <ul style="list-style-type: none"> In the Recurring Billing API, this is the <i>billingScheduleId</i> returned in the UploadResponseV1. In the Recurring Billing tool in the merchant back office, this is the <i>Billing ID</i> on the Billing Records tab.
3	Confirmation Number	String Max = 20	This is the confirmation number for the individual transaction (e.g., a credit card purchase) that was processed by NETBANX. <ul style="list-style-type: none"> In the Web Services API, this is the <i>confirmationNumber</i> returned in the ccTxnResponseV1 or the ddCheckResponseV1. In the Activity Report in the merchant back office, this is the Confirmation Number on the Transaction Details page.
4	Billing Record Status	Enumeration	This is the status of the consumer's billing record. Possible values are: <ul style="list-style-type: none"> ACTIVE BATCH DISABLED SPD_ERR (suspended due to error) SPD_FLR (suspended due to failure) SPD_MERC (suspended by merchant) SPD_RET (suspended due to a return or chargeback)

Table 1-4: Confirmation File Contents (Continued)

Field	Name	Data	Description
5	Transaction Status	Enumeration	This is the status of the transaction when it was processed. Possible values are: <ul style="list-style-type: none"> • SUCCESS – the transaction was processed successfully • ERROR – the transaction could not be processed (e.g., the card type provided did not match the card number) • FAILED – the transaction was processed but failed (The bank returned a decline response for the transaction)
6	Transaction Date	dateTime	This is the date and time the transaction was processed.
7	Transaction Amount	String Max = 999999999.99	This is the amount of the transaction processed.
8	Transaction Description	String Max = 1024	If the Transaction Status was ERROR, this is text describes why the transaction failed.
9	Billing Schedule Reference ID	String Max = 40	This is billing schedule ID you assigned for your own internal reference purposes. <ul style="list-style-type: none"> • In the Recurring Billing API, this is the <i>merchantRefNum</i> you provided in the UploadRequestV1. • In the Recurring Billing tool in the merchant back office, this is the <i>Billing Reg. ID</i> on the Billing Records tab.

Using this guide

This user guide details major system functions. Each section provides an overview of functions, which are then broken down into procedures with steps to be followed.

Audience

This user's guide is intended for NETBANX merchants using our Web Services API to process transaction requests with NETBANX.



Functionality

This guide may document some features to which you do not have access. Access to such functionality is allotted on a merchant-by-merchant basis. If you have any questions, contact your account manager.

Symbols

This user guide uses the following symbols to bring important items to your attention:

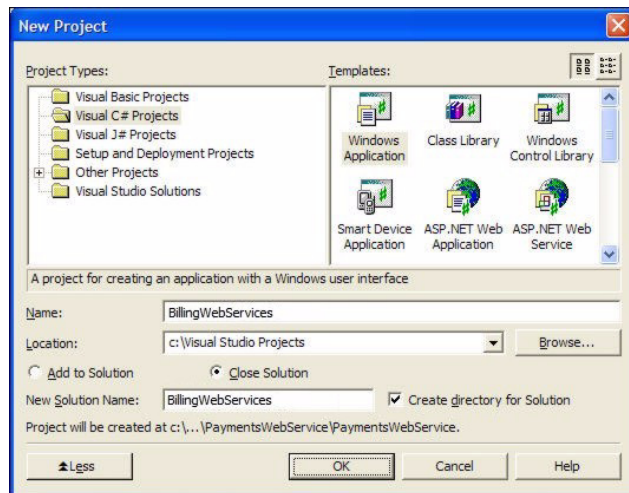
Table 1-5: Symbols

Symbol	Description
	This note icon denotes a hint or tip to help you use the transaction processing application more efficiently.
	This warning icon alerts you about actions you might take that could have important consequences.

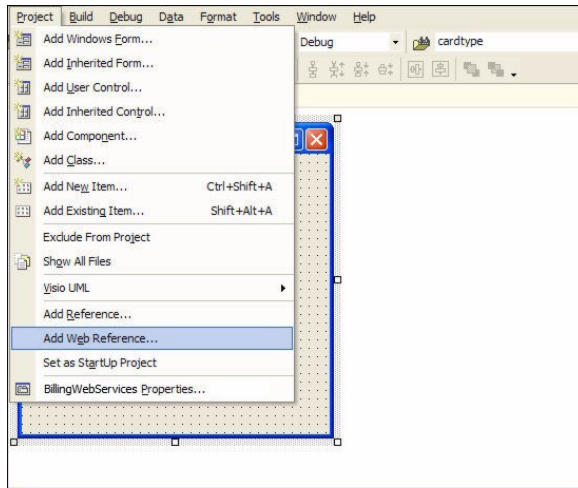
.NET example

To build the .NET example:

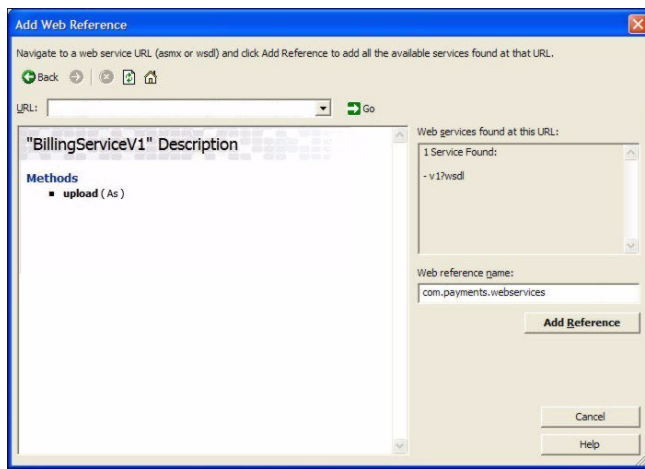
1. Create a new project.



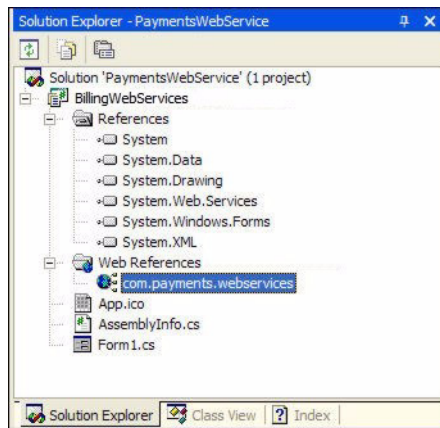
2. Add a Web Reference.



3. Enter the WSDL URL and click the Add Reference button.



The Web client is now built.



4. Build a recurring billing request and process response.
 - See *Building the recurring billing request* on page 1-10 or *Building the LOAD request* on page 1-21.

- See *Processing the response* on page 1-24.

Building the recurring billing request

Recurring billing operations require the *UploadRequestV1* document object. This section describes the structure of an *UploadRequestV1* and how to construct one for both the CREATION and UPDATE operations. See Table 1-6: *UploadRequestV1 Elements* on page 1-15 for details on elements required.



All optional elements that take non-nullable data types (e.g., int or enum) must have their specified attribute set to true when setting values for those elements. For example, see the title element in the example below.

C# CREATION example

The following is a CREATION example in C#.

```
/* Call the Web Service*/

PaymentInstructionV1[] paymentInstruction = new PaymentInstructionV1[1];

paymentInstruction[0] = new PaymentInstructionV1();

paymentInstruction[0].action = ActionReqV1.CREATION;

MerchantAccountV1 merchantAccount = new MerchantAccountV1();
merchantAccount.accountNum = "1000049569";
merchantAccount.storeID = "test";
merchantAccount.storePwd = "test";
paymentInstruction[0].merchantAccount = merchantAccount;

ConsumerInfoV1 consumerInfo = new ConsumerInfoV1();
consumerInfo.merchantRefNum = "REF-1234";
consumerInfo.firstName = "Jane";
consumerInfo.lastName = "Jones";
consumerInfo.title = TitleV1.MS;
consumerInfo.titleSpecified = true;
paymentInstruction[0].consumerInfo = consumerInfo;

AddressV1 billingAddress = new AddressV1();

billingAddress.street = "123 Main Street";
billingAddress.city = "LA";
billingAddress.Item = StateV1.CA;
billingAddress.country = CountryV1.US;
billingAddress.countrySpecified = true;
billingAddress.zip = "90210";
paymentInstruction[0].billingAddress = billingAddress;

AddressV1 shippingAddress = new AddressV1();
shippingAddress.street = "125 Main Street";
shippingAddress.city = "LA";
shippingAddress.Item = StateV1.CA;
shippingAddress.country = CountryV1.US;
shippingAddress.countrySpecified = true;
```

```
shippingAddress.zip = "90210";
paymentInstruction[0].shippingAddress = shippingAddress;

ContactMethodV1[] contactMethods = new ContactMethodV1[2];
contactMethods[0] = new ContactMethodV1();
contactMethods[0].type = ContactTypeV1.PHN_HOME;
contactMethods[0].value = "(111)111-1111";
contactMethods[1] = new ContactMethodV1();
contactMethods[1].type = ContactTypeV1.EMAIL;
contactMethods[1].value = "janejones@emailserver.com";
contactMethods[1] = new ContactMethodV1();
contactMethods[1].type = ContactTypeV1.PHN_CELL;
contactMethods[1].value = "(123)123-1234";
paymentInstruction[0].contactMethods = contactMethods;

PaymentMethodV1 paymentMethod = new PaymentMethodV1();
paymentMethod.merchantRefNum = "6212";
CardV1 card = new CardV1();
card.cardNum = "4111111111111111";
card.cardType = CardTypeV1.VI;
card.cardTypeSpecified = true;
CardExpiryV1 expiry = new CardExpiryV1();
expiry.month = 11;
expiry.year = 2008;
card.cardExpiry = expiry;
paymentMethod.Item = card;
paymentInstruction[0].paymentMethod = paymentMethod;

BillingScheduleV1 billingSchedule = new BillingScheduleV1();
billingSchedule.merchantRefNum = "6211";
billingSchedule.intervalCode = IntervalCodeV1.WEEKLY;
billingSchedule.intervalCodeSpecified = true;
billingSchedule.startDate = new DateTime(2008, 05, 01);
billingSchedule.startDateSpecified = true;
billingSchedule.endDateSpecified = true;
billingSchedule.endDate = new DateTime(2008, 10, 28);
billingSchedule.endDateSpecified = true;
billingSchedule.amount = "14.99";
billingSchedule.lastDayOfTheMonth = BillingScheduleV1LastDayOfTheMonth.Y;
billingSchedule.lastDayOfTheMonthSpecified = true;
billingSchedule.transactionMode = TransactionModeV1.CCPURCHASE;
billingSchedule.transactionModeSpecified = true;
paymentInstruction[0].billingSchedule = billingSchedule;

UploadRequestV1 uploadRequest = new UploadRequestV1();
uploadRequest.paymentInstruction = paymentInstruction;

BillingServiceV1 billingService = new BillingServiceV1();
UploadResponseV1 uploadResponse = null;

uploadResponse = billingService.upload(uploadRequest);

String responseTxt = "";

System.Console.WriteLine("Response received: ");

if (uploadResponse != null)
{
    if (uploadResponse.responseItem != null)
```

```

{
    foreach (ResponseItemV1 responseItem in uploadResponse.responseItem)
    {
        responseTxt += "Code: " + responseItem.code + Environment.NewLine;
        responseTxt += " Request ID: " + responseItem.requestId +
        Environment.NewLine;
        responseTxt += "Confirmation Number: " +
        responseItem.confirmationNumber + Environment.NewLine;
        responseTxt += "Decision: " + responseItem.decision +
        Environment.NewLine;
        responseTxt += "Description: " + responseItem.description +
        Environment.NewLine;

        if (responseItem.detail != null)
        {
            foreach (DetailV1 detail in responseItem.detail)
            {
                responseTxt += detail.tag + ": ";
                responseTxt += detail.value + Environment.NewLine;
            }
        }

        if (responseItem.generatedIds != null)
        {
            GeneratedIdsV1 genId = responseItem.generatedIds[0];
            responseTxt += "consumerId: ";
            responseTxt += genId.consumerId + Environment.NewLine;
            responseTxt += "billingAddressId: ";
            responseTxt += genId.billingAddressId + Environment.NewLine;
            responseTxt += "shippingAddressId: ";
            responseTxt += genId.shippingAddressId + Environment.NewLine;
            responseTxt += "paymentMethodId: ";
            responseTxt += genId.paymentMethodId + Environment.NewLine;
            responseTxt += "billingScheduleId: ";
            responseTxt += genId.billingScheduleId + Environment.NewLine;

            foreach (ContactMethodIdV1 cm in genId.contactMethodIds)
            {
                responseTxt += "contactMethodId (" + cm.type + "): ";
                responseTxt += cm.id + Environment.NewLine;
            }
        }
    }
}

System.Console.WriteLine(responseTxt);

```



For the billingAddress and shippingAddress elements in the example above, use the StateV1 enumeration to specify a province or state and use a string value to specify a region.

C# UPDATE example



All optional elements that take non-nullable data types (e.g., int or enum) must have their specified attribute set to true when setting values for those elements. See the consumerId element in the example below.

The following is an *UPDATE* example in C#, in which the consumer's first name is changed from "Jane" (above) to "Janet".

```

/* Call the Web Service */

PaymentInstructionV1[] paymentInstruction = new PaymentInstructionV1[1];

paymentInstruction[0] = new PaymentInstructionV1();

paymentInstruction[0].action = ActionReqV1.UPDATE;

MerchantAccountV1 merchantAccount = new MerchantAccountV1();
merchantAccount.accountNum = "1000049569";
merchantAccount.storeID = "test";
merchantAccount.storePwd = "test";
paymentInstruction[0].merchantAccount = merchantAccount;

ConsumerInfoV1 consumerInfo = new ConsumerInfoV1();
consumerInfo.consumerId = 7654;
consumerInfo.consumerIdSpecified = true;
consumerInfo.firstName = "Janet";
consumerInfo.lastName = "Jones";
consumerInfo.title = TitleV1.MS;
consumerInfo.titleSpecified = true;
paymentInstruction[0].consumerInfo = consumerInfo;

UploadRequestV1 uploadRequest = new UploadRequestV1();
uploadRequest.paymentInstruction = paymentInstruction;

BillingServiceV1 billingService = new BillingServiceV1();
UploadResponseV1 uploadResponse = null;

uploadResponse = billingService.upload(uploadRequest);

String responseTxt = "";

System.Console.WriteLine("Response received: ");

if (uploadResponse != null)
{
    if (uploadResponse.responseItem != null)
    {
        foreach (ResponseItemV1 responseItem in uploadResponse.responseItem)
        {
            responseTxt += "Code: " + responseItem.code + Environment.NewLine;
            responseTxt += " Request ID: " + responseItem.requestId +
            Environment.NewLine;
            responseTxt += "Confirmation Number: " + responseItem.confirmationNumber +
            Environment.NewLine;
            responseTxt += "Decision: " + responseItem.decision + Environment.NewLine;
            responseTxt += "Description: " + responseItem.description +
            Environment.NewLine;
        }
    }
}

```

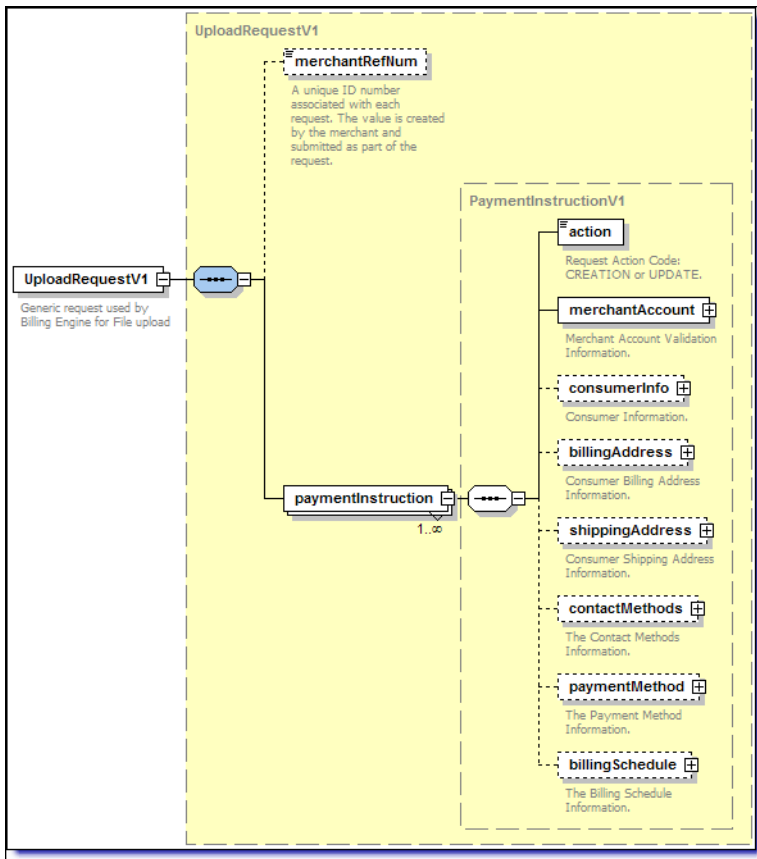
```

if (responseItem.detail != null)
{
    foreach (DetailV1 detail in responseItem.detail)
    {
        responseTxt += detail.tag + ": ";
        responseTxt += detail.value + Environment.NewLine;
    }
}
}
}

System.Console.WriteLine(responseTxt);
    
```

UploadRequestV1 schema

An *UploadRequestV1* document object has the following structure:



UploadRequestV1 elements



Conditional elements in the table below (except for read-only IDs returned by NETBANX) are mandatory when the action element is set to CREATION. When the action element is set to UPDATE, these elements are conditional, depending on what parameters of the billing record you want to update. See Updating recurring billing records on page 1-3 for more information.

The *UploadRequestV1* document object may contain the following elements:

Table 1-6: UploadRequestV1 Elements

Element	Child Elements	Required	Type	Description
merchantRefNum		Optional	String Max = 40	This is a unique ID number you can create for each request.
paymentInstruction		Required		
	action	Required	Enumeration	This is the action to perform. Possible values are: <ul style="list-style-type: none"> • CREATION – creates a new billing record • UPDATE – updates an existing record
	merchantAccount	Required		
		Child Elements of merchant Account		
	accountNum	Required	String Max = 10	This is the merchant account number.
	storeID	Required	String Max = 80	This is the store identifier, used to authenticate the request. It is defined by NETBANX and provided to the merchant as part of the integration process.
	storePwd	Required	String Max = 20	This is the store password, used to authenticate the request. It is defined by NETBANX and provided to the merchant as part of the integration process.
	consumerInfo	Conditional		
		Child Elements of consumerInfo		
	consumerId	Conditional	Int	This is a read-only consumer ID assigned by NETBANX.
	merchantRefNum	Conditional	String Max= 40	This is a consumerInfo consumer ID you can set for your own internal reference purposes.
	firstName	Conditional	String Max= 40	This is the first name of the consumer.
	lastName	Conditional	String Max= 40	This is the last name of the consumer.
	title	Conditional	Enumeration	This is the title of the consumer. Possible values are: <ul style="list-style-type: none"> • MR • MS
	userName	Optional	String Max = 40	This is the user name the consumer uses to log in to an account at your site.
	password	Optional	String Max = 40	This is the password the consumer uses to log in to an account at your site.
	customerIp	Optional	Int Max = 15	This is the consumer's IP address from which the consumer signed up at your site. Must be in standard IP address format, e.g., 127.0.0.1

Table 1-6: UploadRequestV1 Elements (Continued)

Element	Child Elements	Required	Type	Description
	billingAddress	Conditional		
		Child Elements of billingAddress		
	id	Conditional	Int	This is a read-only billing address ID assigned by NETBANX.
	street	Conditional	String Max = 50	This is the street and number for the billing address.
	street2	Optional	String Max = 50	This is further information for the billing address (e.g., apartment #).
	city	Conditional	String Max = 40	This is the city for the billing address.
	region/state	Conditional	If state, then enumeration If region, then string Max = 40	This is the state/province/region of the consumer. Provide <i>state</i> if within U.S./Canada. Provide <i>region</i> if outside of U.S./Canada. See Appendix C: <i>Geographical Codes</i> for correct codes to use.
	country	Conditional	Enumeration	This is the country of the billing address. See <i>Country codes</i> on page C-3 for the correct codes to use.
	zip	Conditional	String Max = 40	This is the ZIP code of the billing address if in the U.S.; otherwise, this is the postal code.
	shippingAddress	Conditional		<i>shippingAddress</i> takes the same elements as <i>billingAddress</i> .
	contactMethod	Optional		
		Child Elements of contactMethod		
	id	Optional	Int	This is a read-only contact ID assigned by NETBANX.
	type	Optional	Enumeration	This is the contact method type. Possible values are: <ul style="list-style-type: none"> • PHN_HOME • PHN_CELL • EMAIL
	value	Optional	String Max = 100	This is the contact method value.
	paymentMethod	Conditional		
		Child Elements of paymentMethod		
	id	Conditional	Int	This is a read-only payment method ID assigned by NETBANX.
	merchantRefNum	Conditional	String Max = 40	This is a paymentMethod consumer ID you can set for your own internal reference purposes.

Table 1-6: UploadRequestV1 Elements (Continued)

Element	Child Elements	Required	Type	Description
	ccHolderName	Conditional	String Max = 100	This is the name of the card holder.
	card	Conditional		
			Child Elements of card	
	cardNum	Conditional	String Min = 8 Max = 20	This is the credit card number.
	cardExpiry	Conditional		
	month	Conditional	String = 2	This is the month the credit card expires.
	year	Conditional	String = 4	This is the year the credit card expires.
	cardType	Conditional	Enumeration	This is the credit card brand. Possible values are: <ul style="list-style-type: none"> • AM = American Express • CB = Carte Blanche • DC = Diners Club • DI = Discover • JC = JCB • LA = Laser • MC = MasterCard • MD = Maestro • N = Novus • SO = Solo • SW = Switch • VD = Visa Debit • VE = Visa Electron • VI = Visa
	issueNum	Conditional	Integer Max = 2	The 1- or 2-digit number located on the front of the card, following the card number. NOTE: The <i>issueNum</i> element can be used only when the <i>cardType</i> is MD (Maestro), SO (Solo), or SW (Switch).
	check	Conditional		
			Child Elements of check	
	accountType	Conditional	Enumeration	This is the type of checking account used for the transaction. Possible values are: <ul style="list-style-type: none"> • PC (Personal Checking) • PS (Personal Savings) • PL (Personal Loan) • BC (Business Checking) • BS (Business Savings) • BL (Business Loan)
	bankName	Conditional	String Max = 40	This is the name of the customer's bank, to which this transaction is posted.
	checkNum	Conditional	Long Max = 8	This is the check serial number, provided at the time of the transaction request. NETBANX will automatically increment this number with each recurring payment.

Table 1-6: UploadRequestV1 Elements (Continued)

Element	Child Elements	Required	Type	Description
	accountNum	Conditional	String Max = 17	This is the customer's bank account number.
	routingNum	Conditional	String Min = 6 Max = 9	For USD accounts, this is the 9-digit routing number of the customer's bank. For British pound accounts, this is the 6-digit sort code of the customer's bank. For Canadian dollar accounts, this is a combination of the 3-digit institution ID and the 5-digit transit number of the customer's bank branch. Do not include spaces or dashes.
	billingAddressId	Conditional	Int	This is a read-only billing address ID assigned by NETBANX.
	shippingAddressId	Conditional	Int	This is a read-only shipping address ID assigned by NETBANX.
	billingSchedule	Conditional		
		Child Elements of billingSchedule		
	id	Conditional	Int	This is a read-only billing schedule ID assigned by NETBANX.
	paymentMethodId	Conditional	Int	This is a read-only payment method ID assigned by NETBANX.
	transactionMode	Optional	Enumeration	This value determines whether the recurring transaction is a credit or a debit to your consumer. Possible values are: <ul style="list-style-type: none"> CCPURCHASE – the amount is debited from the consumer's credit card CCPAYMENT – the amount is credited to the consumer's credit card DDCHARGE – the amount is debited from the consumer's bank account DDCREDIT – the amount is credited to the consumer's bank account. NOTE: The value of the transactionMode element must correspond with the paymentMethod selected for this record. E.g., if the paymentMethod is Direct Debit, you must select a DD value here. If transactionMode is not defined, then CCPURCHASE is the default for a credit card paymentMethod and DDCHARGE is the default for a Direct Debit paymentMethod.
	merchantRefNum	Conditional	String Max= 40	This is a billingSchedule consumer ID you set for your own internal reference purposes.

Table 1-6: UploadRequestV1 Elements (Continued)

Element	Child Elements	Required	Type	Description
	intervalCode	Conditional	Enumeration	<p>This specifies how often the recurring billing transaction is made. Possible values are:</p> <ul style="list-style-type: none"> • 1STAND15TH – Transaction is processed on the first and fifteenth of each month. • ANNUALLY – Transaction is processed once each year. • BIWEEKLY – Transaction is processed every two weeks. • DAYS – Transaction is processed at the interval of days you specify with the <i>intervalValue</i> element. • MONTHLY – Transaction is processed once each month. • QUARTERLY – Transaction is processed once every three months. • SANNUALLY – Transaction is processed every six months. • WEEKLY – Transaction is processed once a week.
	intervalValue	Conditional	String Max = 165	<p>Provide only if the <i>intervalCode</i> element is set to DAYS. The <i>intervalValue</i> is the number of days between payments. For example, if <i>intervalValue</i> is set to 10, the recurring billing transaction would be processed every 10 days.</p>
	startDate	Conditional	date	<p>This is the date on which the recurring billing transaction will begin. The resulting behaviour of the recurring transaction depends on the Interval you have set:</p> <ul style="list-style-type: none"> • 1STAND15TH – Transaction will be made on this date and then subsequently on the 1st and 15th of each month. • ANNUALLY – Transaction will be made this day each year. • BIWEEKLY – Transaction will be made this day of the week every two weeks. • DAYS – Transaction will be made this day, and then on every <i>nth</i> day, depending on the value set for the <i>intervalValue</i> element. • MONTHLY – Transaction will be made this day each month (e.g. the 15th). If the <i>lastDayOfTheMonth</i> element is set to Y, the first recurring transaction is made on the day of the month you specify, but all subsequent monthly transactions will be made on the last day of each month. • QUARTERLY – Transaction will be made on this day every 3 months. • SANNUALLY – Transaction will be made on this day twice a year (e.g., if you select January 1, the second payment will be made on July 1). • WEEKLY – Transaction will be made this day of the week every week.

Table 1-6: UploadRequestV1 Elements (Continued)

Element	Child Elements	Required	Type	Description
	nextBillingDate	Conditional	date	For the <i>update</i> operation, this sets a new billing date for the billing record. For example, you can use this element to change the monthly date a recurring charge is made or to skip one or more scheduled charges before resuming.
	endDate	Conditional	date	This is the date on which the recurring payment will end. After the last payment date defined here, the billing record is automatically disabled. If no end date is specified, the recurring payment continues indefinitely.
	serviceName	Optional	String Max = 240	This designates the type of billing record (e.g., monthly subscription).
	statusCode	Conditional	Enumeration	This is the status of the billing record. Possible values are: <ul style="list-style-type: none"> ACTIVE DISABLED SPD_MERC (suspended by merchant)
	amount	Conditional	String Max= 999999999.99	This is the amount of the recurring billing transaction.
	lastDayOfTheMonth	Conditional	Enumeration	If the <i>intervalCode</i> element is set to MONTHLY, you can set this element to Y (yes). If you do so, the first recurring payment is made on the day of the month you specified, but all subsequent monthly payments will be made on the last day of each month. Possible values are: <ul style="list-style-type: none"> Y N
	siteUrl	Optional	String Max = 255	This is the URL of the site at which your consumer joined as a member.
	keywords	Optional		
		Child Element of keywords		
	keyword	Conditional	String Max = 255	This is a tag you can use for reporting purposes in the NETBANX merchant back office. For example, you can use this as a tag to identify the transaction or the product purchased at your site.
	welcomeEmail	Optional	Enumeration	This indicates whether to send a welcome email to the consumer once that consumer's recurring billing record has been set up for your site. Possible values are: <ul style="list-style-type: none"> Y N

Building the LOAD request

The LOAD operation requires the *ConsumerProfileRequestV1* document object. This section describes the structure of an *ConsumerProfileRequestV1* and how to construct one. See Table 1-7: *ConsumerProfileRequestV1 Elements* on page 1-23 for details on elements required.

C# LOAD example

The following is a *LOAD* example in C#, for which the consumer profile will be downloaded for consumer ID 12715.

```

/* Call the Web Service*/

MerchantAccountV1 merchantAccount = new MerchantAccountV1();
merchantAccount.accountNum = "1000049569";
merchantAccount.storeID = "test";
merchantAccount.storePwd = "test";

ConsumerProfileRequestV1 loadRequest = new ConsumerProfileRequestV1();
long [] consumerIds = new long[1];
consumerIds[0] = 12715;
loadRequest.consumerId = consumerIds;
loadRequest.merchantRefNum = "REF-LOAD-001";
loadRequest.merchantAccount = merchantAccount;

BillingServiceV1 billingService = new BillingServiceV1();
ConsumerProfileResponseV1 loadResponse = billingService.load(loadRequest);
String responseTxt = "";

System.Console.WriteLine("Response received: ");

if (loadResponse != null)
{
    if (loadResponse.consumerProfile != null)
    {
        foreach (ConsumerProfileV1 profileItem in loadResponse.consumerProfile)
        {
            responseTxt += "Code: " + profileItem.code + Environment.NewLine;
            responseTxt += "Decision: " + profileItem.decision + Environment.NewLine;
            responseTxt += "Description: " + profileItem.description
                + Environment.NewLine;
            responseTxt += "ConsumerId: " + profileItem.consumerId
                + Environment.NewLine;
            responseTxt += "Merchant Ref Number: "
                + profileItem.consumerInfo.merchantRefNum
                + Environment.NewLine;
            responseTxt += "Title: " + profileItem.consumerInfo.title
                + Environment.NewLine;
            responseTxt += "Name: " + profileItem.consumerInfo.firstName
                + " " + profileItem.consumerInfo.lastName
                + Environment.NewLine;

            if (profileItem.billingAddresses != null )
            {
                for (int i = 0; i < profileItem.billingAddresses.Length; i++)
                {
                    responseTxt += "BillingAddress id: " +
                        profileItem.billingAddresses[i].id + Environment.NewLine;
                }
            }
        }
    }
}

```

```
    }
  }

  if (profileItem.shippingAddresses != null)
  {
    for (int i = 0; i < profileItem.shippingAddresses.Length; i++)
    {
      responseTxt += "ShippingAddress id: "
      + profileItem.shippingAddresses[i].id + Environment.NewLine;
      responseTxt += "street: " + profileItem.shippingAddresses[i].street
      + Environment.NewLine;
      responseTxt += "street2: "
      + profileItem.shippingAddresses[i].street2 + Environment.NewLine;
      responseTxt += "city: " + profileItem.shippingAddresses[i].city
      + Environment.NewLine;
      responseTxt += "region/province: "
      + profileItem.shippingAddresses[i].Item + Environment.NewLine;
      responseTxt += "country: "
      + profileItem.shippingAddresses[i].country + Environment.NewLine;
      responseTxt += "zip/postal code: "
      + profileItem.shippingAddresses[i].zip + Environment.NewLine;
    }
  }

  if (profileItem.paymentMethods != null)
  {
    for (int i = 0; i < profileItem.paymentMethods.Length; i++)
    {
      responseTxt += "PaymentMethod id: " +
      profileItem.paymentMethods[i].id + Environment.NewLine;
    }
  }

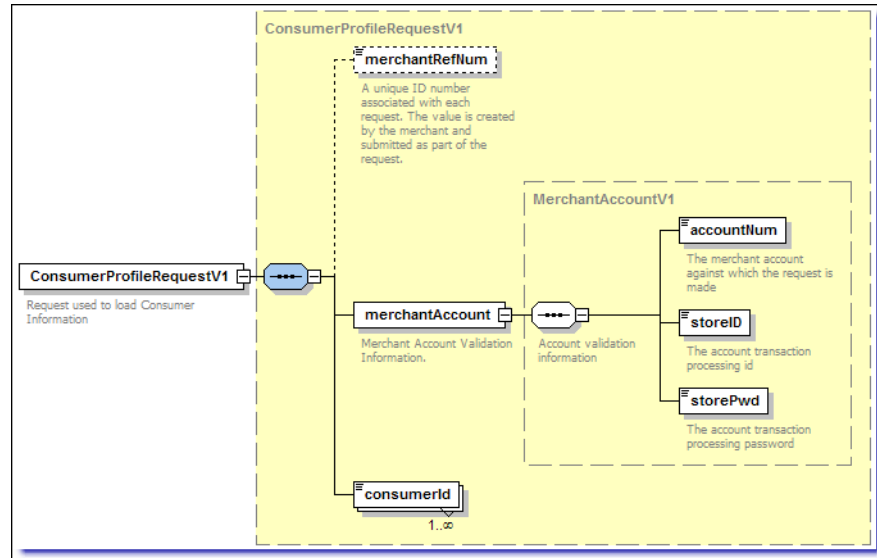
  if (profileItem.billingSchedules != null)
  {
    for (int i = 0; i < profileItem.billingSchedules.Length; i++)
    {
      responseTxt += "BillSchedule id: " +
      profileItem.billingSchedules[i].id + Environment.NewLine;
    }
  }
}

}

System.Console.WriteLine(responseTxt);
```

ConsumerProfileRequestV1 schema

A *ConsumerProfileRequestV1* document object has the following structure:



ConsumerProfileRequestV1 elements

The *ConsumerProfileRequestV1* document object contains the following elements:

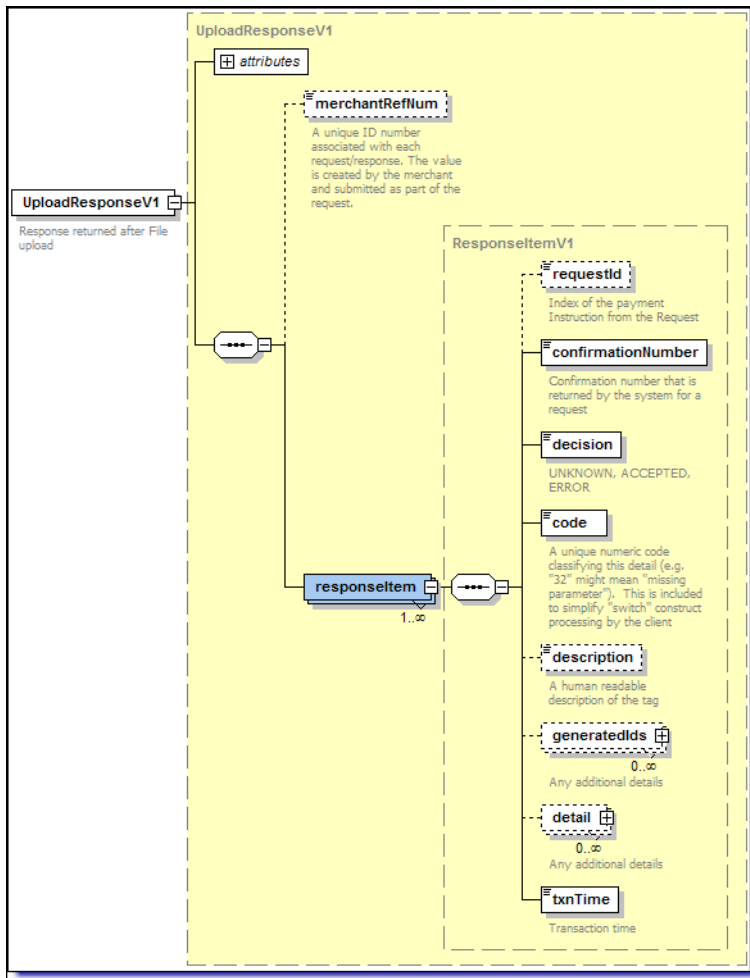
Table 1-7: ConsumerProfileRequestV1 Elements

Element	Child Elements	Required	Type	Description
merchantRefNum		Optional	String Max = 40	This is a unique ID number you can create for each request.
merchantAccount	accountNum	Required	String Max = 10	This is the merchant account number.
	storeID	Required	String Max = 80	This is the store identifier, used to authenticate the request. It is defined by NETBANX and provided to the merchant as part of the integration process.
	storePwd	Required	String Max = 20	This is the store password, used to authenticate the request. It is defined by NETBANX and provided to the merchant as part of the integration process.
consumerId		Required	Int	This is a read-only consumer ID assigned by NETBANX. This defines the consumer for which you want to download a profile. See <i>Loading consumer profiles</i> on page 1-4 for details on where to find the <i>consumerId</i> . NOTE: You can include only one <i>consumerId</i> with each API request.

Processing the response

UploadResponseV1

An *UploadResponseV1* has the following structure:



The following elements are relevant for an *UploadResponseV1*:

Table 1-8: UploadResponseV1 Elements

Element	Child Element	Required	Type	Description
attributes		Not Applicable		This element is used only for Recurring Billing files uploaded through the Batch Upload tool in the NETBANX merchant back office. The following values may be returned: <ul style="list-style-type: none"> acceptedCount – the number of records processed acceptedTotal – the value of the records processed errorCount – the number of records that failed errorTotal – the value of the records that failed

Table 1-8: UploadResponseV1 Elements (Continued)

Element	Child Element	Required	Type	Description
merchantRefNum		Optional	String	This is the merchantRefNum of the original request, if you provided one.
responseItem		Required		
	requestId	Required	Int	This is the transaction ID number returned by NETBANX.
	confirmationNumber	Required	String Max = 20	This is the confirmation number returned by NETBANX.
	decision	Required	Enumeration	This is the status of the request. One of the following is returned: <ul style="list-style-type: none"> Accepted – the request was processed. Error – the request was attempted, but failed for some reason. Unknown – the response could not be processed properly and you should contact Technical Support.
	code	Required	Int	This is a numeric code that categorizes the response. See Appendix B: <i>Response Codes</i> .
	description	Optional	String Max = 1024	This is a human readable description of the code element.
	generateIds	Optional		
	consumerId	Optional	String Max = 20	If a consumer was created with the request that was processed, this is the unique consumer ID.
	billingAddressId	Optional	String Max = 20	If a billing address was created with the request that was processed, this is the unique billing address ID.
	shippingAddressId	Optional	String Max = 20	If a shipping address was created with the request that was processed, this is the unique shipping address ID.
	contactMethodId	Optional		
	type	Optional	Enumeration	This is the contact method type. Possible values are: <ul style="list-style-type: none"> PHN_HOME PHN_CELL EMAIL
	id	Optional	String Max = 20	If a contact method was created with the request that was processed, this is the contact method ID.
	paymentMethodId	Optional	String Max = 20	If a payment method was created with the request that was processed, this is the payment method ID.
	billingScheduleId	Optional	String Max = 20	If a billing schedule was created with the request that was processed, this is the billing schedule ID.
	detail	Optional		
	tag	Optional	String Max = 30	This is the classification of the detail element.
	value	Optional	String Max = 1024	This is the description of the detail.

Table 1-8: UploadResponseV1 Elements (Continued)

Element	Child Element	Required	Type	Description
	txnTime	Required	dateTime	This is the date and time the request was processed.

To process the response:

1. Get the response details by accessing the elements in the response object.

```
uploadResponse = billingService.upload(uploadRequest);

String responseTxt = "";

System.Console.WriteLine("Response received: ");

if (uploadResponse != null)
{
    if (uploadResponse.responseItem != null)
    {
        foreach (ResponseItemV1 responseItem in uploadResponse.responseItem)
        {
            responseTxt += "Code: " + responseItem.code + Environment.NewLine;
            responseTxt += " Request ID: " + responseItem.requestId +
            Environment.NewLine;
            responseTxt += "Confirmation Number: " +
            responseItem.confirmationNumber + Environment.NewLine;
            responseTxt += "Decision: " + responseItem.decision +
            Environment.NewLine;
            responseTxt += "Description: " + responseItem.description +
            Environment.NewLine;

            if (responseItem.detail != null)
            {
                foreach (DetailV1 detail in responseItem.detail)
                {
                    responseTxt += detail.tag + ": ";
                    responseTxt += detail.value + Environment.NewLine;
                }
            }

            if (responseItem.generatedIds != null)
            {
                GeneratedIdsV1 genId = responseItem.generatedIds[0];
                responseTxt += "consumerId: ";
                responseTxt += genId.consumerId + Environment.NewLine;
                responseTxt += "billingAddressId: ";
                responseTxt += genId.billingAddressId + Environment.NewLine;
                responseTxt += "shippingAddressId: ";
                responseTxt += genId.shippingAddressId + Environment.NewLine;
                responseTxt += "paymentMethodId: ";
                responseTxt += genId.paymentMethodId + Environment.NewLine;
                responseTxt += "billingScheduleId: ";
                responseTxt += genId.billingScheduleId + Environment.NewLine;

                foreach (ContactMethodIdV1 cm in genId.contactMethodIds)
                {
                    responseTxt += "contactMethodId (" + cm.type + "): ";
                    responseTxt += cm.id + Environment.NewLine;
                }
            }
        }
    }
}
```

```

    }
  }
}

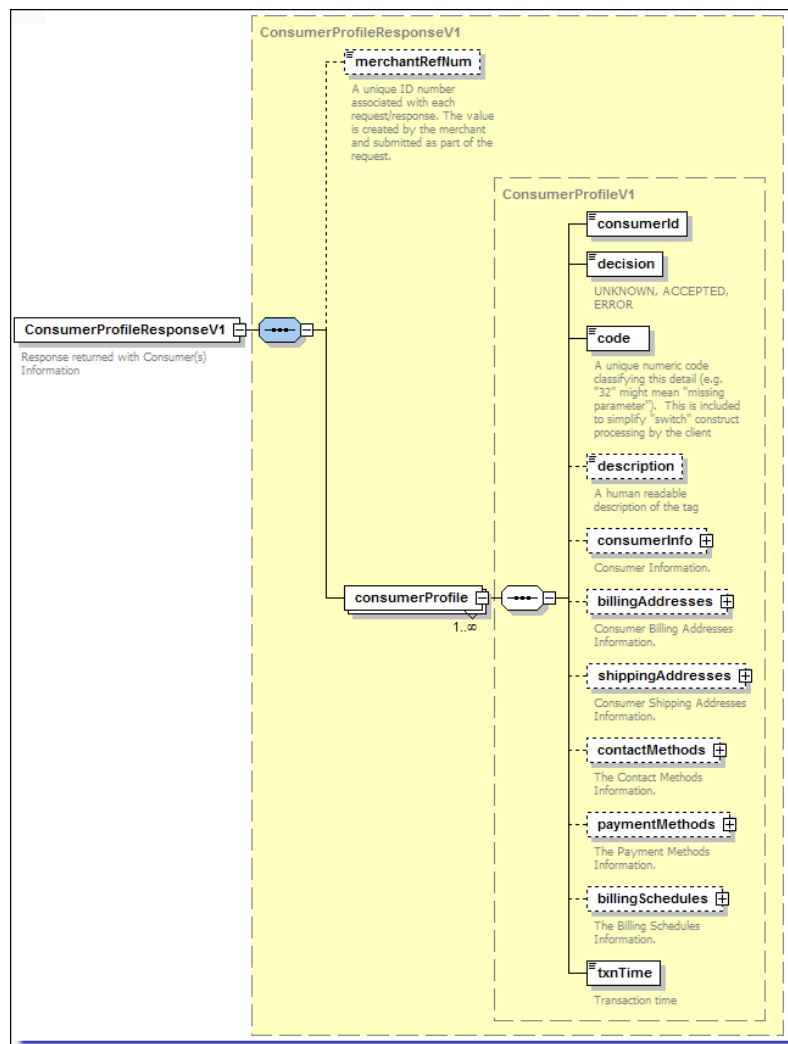
System.Console.WriteLine(responseTxt);

```

2. Process based on the *decision* element. You would insert handling code appropriate to your application. You can also look at the *code* element to provide more fine-grained control for your application. See Appendix B: *Response Codes* for more details.

ConsumerProfileResponseV1

A *ConsumerProfileResponseV1* has the following structure:



The following elements are relevant for an *ConsumerProfileResponseV1*:

Table 1-9: ConsumerProfileResponseV1 Elements

Element	Child Element	Required	Type	Description
merchantRefNum		Optional	String	This is the merchantRefNum of the original <i>ConsumerProfileRequestV1</i> request, if you provided one.
consumerProfile		Required		
	consumerId	Required	Int	This is the consumer ID number returned by NETBANX when the billing record was created.
	decision	Required	Enumeration	This is the status of the request. One of the following is returned: <ul style="list-style-type: none"> Accepted – the request was processed. Error – the request was attempted, but failed for some reason. Unknown – the response could not be process properly and you should contact Technical Support.
	code	Required	Int	This is a numeric code that categorizes the response. See Appendix B: <i>Response Codes</i> .
	description	Optional	String Max = 1024	This is a human readable description of the <i>code</i> element.
	txnTime	Required	dateTime	This is the date and time the request was processed.



See Table 1-6: *UploadRequestV1 Elements* on page 1-15 for a description of the rest of the elements returned in the *ConsumerProfileResponseV1*.

To process the response:

1. Get the response details by accessing the elements in the response object.

```

ConsumerProfileResponseV1 loadResponse = billingService.load(loadRequest);
String responseTxt = "";

System.Console.WriteLine("Response received: ");

if (loadResponse != null)
{
    if (loadResponse.consumerProfile != null)
    {
        foreach (ConsumerProfileV1 profileItem in loadResponse.consumerProfile)
        {
            responseTxt += "Code: " + profileItem.code + Environment.NewLine;
            responseTxt += "Decision: " + profileItem.decision + Environment.NewLine;
            responseTxt += "Description: " + profileItem.description
                + Environment.NewLine;
            responseTxt += "ConsumerId: " + profileItem.consumerId
                + Environment.NewLine;
            responseTxt += "Merchant Ref Number: "
                + profileItem.consumerInfo.merchantRefNum
                + Environment.NewLine;
            responseTxt += "Title: " + profileItem.consumerInfo.title
                + Environment.NewLine;
            responseTxt += "Name: " + profileItem.consumerInfo.firstName
                + " " + profileItem.consumerInfo.lastName
                + Environment.NewLine;
        }
    }
}

```

```

if( profileItem.billingAddresses != null )
{
    for (int i = 0; i < profileItem.billingAddresses.Length; i++)
    {
        responseTxt += "BillingAddress id: " +
            profileItem.billingAddresses[i].id + Environment.NewLine;
    }
}

if (profileItem.shippingAddresses != null)
{
    for (int i = 0; i < profileItem.shippingAddresses.Length; i++)
    {
        responseTxt += "ShippingAddress id: "
            + profileItem.shippingAddresses[i].id + Environment.NewLine;
        responseTxt += "street: " + profileItem.shippingAddresses[i].street
            + Environment.NewLine;
        responseTxt += "street2: "
            + profileItem.shippingAddresses[i].street2 + Environment.NewLine;
        responseTxt += "city: " + profileItem.shippingAddresses[i].city
            + Environment.NewLine;
        responseTxt += "region/province: "
            + profileItem.shippingAddresses[i].Item + Environment.NewLine;
        responseTxt += "country: "
            + profileItem.shippingAddresses[i].country + Environment.NewLine;
        responseTxt += "zip/postal code: "
            + profileItem.shippingAddresses[i].zip + Environment.NewLine;
    }
}

if (profileItem.paymentMethods != null)
{
    for (int i = 0; i < profileItem.paymentMethods.Length; i++)
    {
        responseTxt += "PaymentMethod id: " +
            profileItem.paymentMethods[i].id + Environment.NewLine;
    }
}

if (profileItem.billingSchedules != null)
{
    for (int i = 0; i < profileItem.billingSchedules.Length; i++)
    {
        responseTxt += "BillSchedule id: " +
            profileItem.billingSchedules[i].id + Environment.NewLine;
    }
}
}
}

}

```

```
System.Console.WriteLine(responseTxt);
```

2. Process based on the *decision* element. You would insert handling code appropriate to your application. You can also look at the *code* element to provide more fine-grained control for your application. See Appendix B: *Response Codes* for more details.

Using the HTTP Post Method

Introduction

In addition to a standard Web Service–based call, you can also use the HTTP Post method to post recurring billing requests to NETBANX.

Note that the URLs in the examples below point to the NETBANX Production environment. In order to test your integration, please contact Technical Support to obtain Test URLs and Test merchant account parameters.

- Email support@optimalpayments.com
- Telephone 1-888-709-8753

Creating a recurring billing record

The following is an example of creating a recurring billing record. For complete details on creating a recurring billing record, see *Creating recurring billing records* on page 1-2.

To create a recurring billing record via HTTP Post:

1. Create a transaction request like the following example:

```
<UploadRequestV1 xmlns="http://www.optimalpayments.com/billing/xmlschema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.optimalpayments.com/billing/xmlschema/v1">
  <paymentInstruction>
    <action>CREATION</action>
    <merchantAccount>
      <accountNum>1000049569</accountNum>
      <storeID>test</storeID>
      <storePwd>test</storePwd>
    </merchantAccount>
    <consumerInfo>
      <merchantRefNum>REF-1234</merchantRefNum>
      <firstName>Jane</firstName>
      <lastName>Jones</lastName>
      <title>MS</title>
    </consumerInfo>
    <billingAddress>
      <street>123 Main Street</street>
      <city>LA</city>
      <state>CA</state>
      <country>US</country>
      <zip>90210</zip>
    </billingAddress>
    <shippingAddress>
      <street>125 Main Street</street>
      <city>LA</city>
      <state>CA</state>
      <country>US</country>
  </paymentInstruction>
</UploadRequestV1>
```

```

    <zip>90210</zip>
  </shippingAddress>
  <contactMethods>
    <contactMethod>
      <type>PHN_HOME</type>
      <value>(111) 111-1111</value>
    </contactMethod>
    <contactMethod>
      <type>EMAIL</type>
      <value>janejones@emailserver.com</value>
    </contactMethod>
    <contactMethod>
      <type>PHN_CELL</type>
      <value>(123) 123-1234</value>
    </contactMethod>
  </contactMethods>
  <paymentMethod>
    <merchantRefNum>6212</merchantRefNum>
    <card>
      <cardNum>4342111111111111</cardNum>
      <cardExpiry>
        <month>12</month>
        <year>2009</year>
      </cardExpiry>
      <cardType>VI</cardType>
    </card>
  </paymentMethod>
  <billingSchedule>
    <transactionMode>DDCHARGE</transactionMode>
    <merchantRefNum>6211</merchantRefNum>
    <intervalCode>WEEKLY</intervalCode>
    <startDate>2007-05-01</startDate>
    <endDate>2007-10-28</endDate>
    <amount>14.99</amount>
    <lastDayOfTheMonth>Y</lastDayOfTheMonth>
  </billingSchedule>
</paymentInstruction>
</UploadRequestV1>

```



See Table 1-6: UploadRequestV1 Elements on page 1-15 for a list and description of parameters to include in this request.

2. Include this transaction request in an HTTP Post (see *HTML examples* on page A-4).
3. Send the HTTP Post to the following URL:

<https://webservices.optimalpayments.com/billingWS/billing/v1/billingUpload>

Updating a recurring billing record

The following is an example of updating the payment method in an existing recurring billing record. For complete details on updating a recurring billing record, see *Updating recurring billing records* on page 1-3.

To update a recurring billing record via HTTP Post:**1. Create a transaction request like the following example:**

```
<UploadRequestV1 xmlns="http://www.optimalpayments.com/billing/xmlschema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.optimalpayments.com/billing/xmlschema/v1">
  <paymentInstruction>
    <action>UPDATE</action>
    <merchantAccount>
      <accountNum>1000049569</accountNum>
      <storeID>YourID</storeID>
      <storePwd>YourPwd</storePwd>
    </merchantAccount>
    <consumerInfo>
      <consumerId>14958</consumerId>
    </consumerInfo>
    <paymentMethod>
      <!-- TO BE UPDATED -->
      <id>14984</id>
      <merchantRefNum>6212</merchantRefNum>
      <card>
        <cardNum>4342111111112222</cardNum>
        <cardExpiry>
          <month>12</month>
          <year>2009</year>
        </cardExpiry>
        <cardType>VI</cardType>
      </card>
    </paymentMethod>
  </paymentInstruction>
</UploadRequestV1>
```



See Table 1-6: UploadRequestV1 Elements on page 1-15 for a list and description of parameters to include in this request.

2. Include this transaction request in an HTTP Post (see *CREATION post* on page A-4).
3. Send the HTTP Post to the following URL:

<https://webservices.optimalpayments.com/billingWS/billing/v1/billingUpload>

Loading a consumer profile

The following is an example of loading a consumer profile from NETBANX. For complete details on loading a consumer profile, see *Loading consumer profiles* on page 1-4.

To load a consumer profile via HTTP Post:**1. Create a transaction request like the following example:**

```
<ConsumerProfileRequestV1
xmlns="http://www.optimalpayments.com/billing/xmlschema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.optimalpayments.com/billing/xmlschema/v1">
  <merchantAccount>
    <accountNum>1000010000</accountNum>
    <storeID>YourID</storeID>
```

```

    <storePwd>YourPassword</storePwd>
  </merchantAccount>
  <consumerId>1018</consumerId>
</ConsumerProfileRequestV1>

```



See Table 1-7: ConsumerProfileRequestV1 Elements on page 1-23 for a list and description of parameters to include in this request.

2. Include this transaction request in an HTTP Post (see *LOAD Post* on page A-6).
3. Send the HTTP Post to the following URL:

<https://webservices.optimalpayments.com/billingWS/billing/v1/consumerUpload>

HTML examples

CREATION post

This example shows a form version for posting a request to create a recurring billing record.

```

<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
  <title>Create Recurring Billing Record </title>
  <script language=javascript type="text/javascript">
    function updateUrl() {
      document.actionForm.action =
document.actionForm.urls.options[document.actionForm.urls.selectedIndex].value;
    }
    function doSubmit() {
      document.actionForm.action =
document.actionForm.urls.options[document.actionForm.urls.selectedIndex].value;
      document.actionForm.submit();
    }
  </script>
</head>

<body>
  <form name="actionForm" action="" enctype="multipart/form-data" method="post">
    <p>
      Select Environment:
      <br>
      <select name="urls" onchange="updateUrl()">
        <option
value="https://webservices.optimalpayments.com/billingWS/billing/v1/billingUpload"
selected>NETBANX</option>

        </select>
      </p>
      <p>
        XML Document to Create Recurring Billing Record:<br>
        <textarea name="txnRequest" cols=50 rows=20>
<?xml version="1.0" encoding="UTF-8"?>
<UploadRequestV1 xmlns="http://www.optimalpayments.com/billing/xmlschema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.optimalpayments.com/billing/xmlschema/v1">

```

```

<paymentInstruction>
  <action>CREATION</action>
  <merchantAccount>
    <accountNum>1000049569</accountNum>
    <storeID>test</storeID>
    <storePwd>test</storePwd>
  </merchantAccount>
  <consumerInfo>
    <merchantRefNum>REF-1234</merchantRefNum>
    <firstName>Jane</firstName>
    <lastName>Jones</lastName>
    <title>MS</title>
  </consumerInfo>
  <billingAddress>
    <street>123 Main Street</street>
    <city>LA</city>
    <state>CA</state>
    <country>US</country>
    <zip>90210</zip>
  </billingAddress>
  <shippingAddress>
    <street>125 Main Street</street>
    <city>LA</city>
    <state>CA</state>
    <country>US</country>
    <zip>90210</zip>
  </shippingAddress>
  <contactMethods>
    <contactMethod>
      <type>PHN_HOME</type>
      <value>(111) 111-1111</value>
    </contactMethod>
    <contactMethod>
      <type>EMAIL</type>
      <value>janejones@emailserver.com</value>
    </contactMethod>
    <contactMethod>
      <type>PHN_CELL</type>
      <value>(123) 123-1234</value>
    </contactMethod>
  </contactMethods>
  <paymentMethod>
    <merchantRefNum>6212</merchantRefNum>
    <card>
      <cardNum>4342111111111111</cardNum>
      <cardExpiry>
        <month>12</month>
        <year>2009</year>
      </cardExpiry>
      <cardType>VI</cardType>
    </card>
  </paymentMethod>
  <billingSchedule>
    <transactionMode>DDCHARGE</transactionMode>
    <merchantRefNum>6211</merchantRefNum>
    <intervalCode>WEEKLY</intervalCode>
    <startDate>2007-05-01</startDate>
    <endDate>2007-10-28</endDate>
    <amount>14.99</amount>
  </billingSchedule>

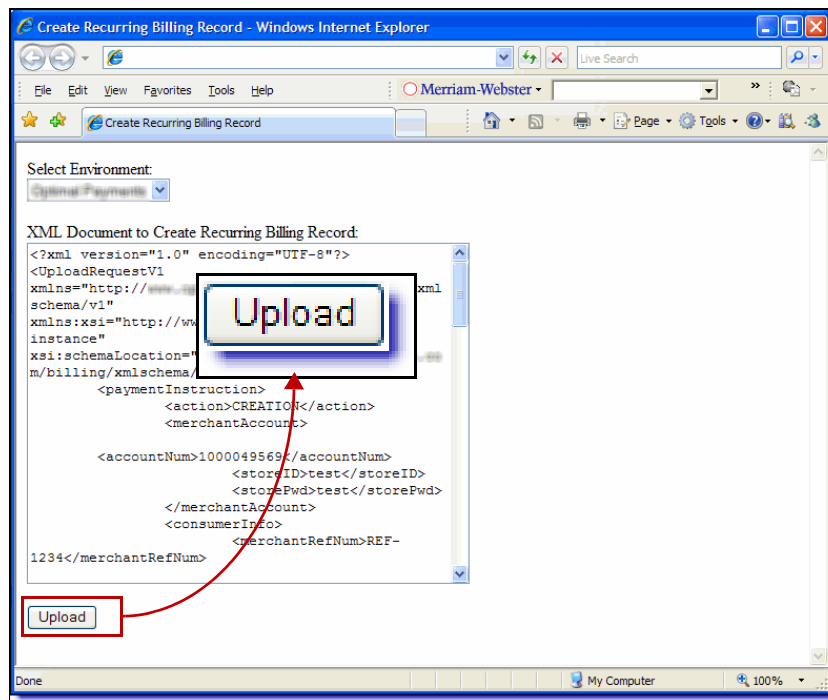
```

```

        <lastDayOfMonth>Y</lastDayOfMonth>
    </billingSchedule>
</paymentInstruction>
</UploadRequestV1>
</textarea>
</p>
<p>
    <input type="submit" value="Upload" onclick="doSubmit()">
</p>
</form>
</body>
</html>

```

Viewed with a browser, the code would look something like this:



Clicking the Upload button sends the recurring billing record creation request via HTTP Post to NETBANX.



The maximum file size supported is 100K. If your HTTP Post exceeds 100K it will fail.

LOAD Post

This example shows a form version for posting a request to download a consumer profile.

```

<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
    <title>Download Consumer Profile</title>
    <script language=javascript type="text/javascript">
        function updateUrl() {
            document.actionForm.action =
document.actionForm.urls.options[document.actionForm.urls.selectedIndex].value;

```

```

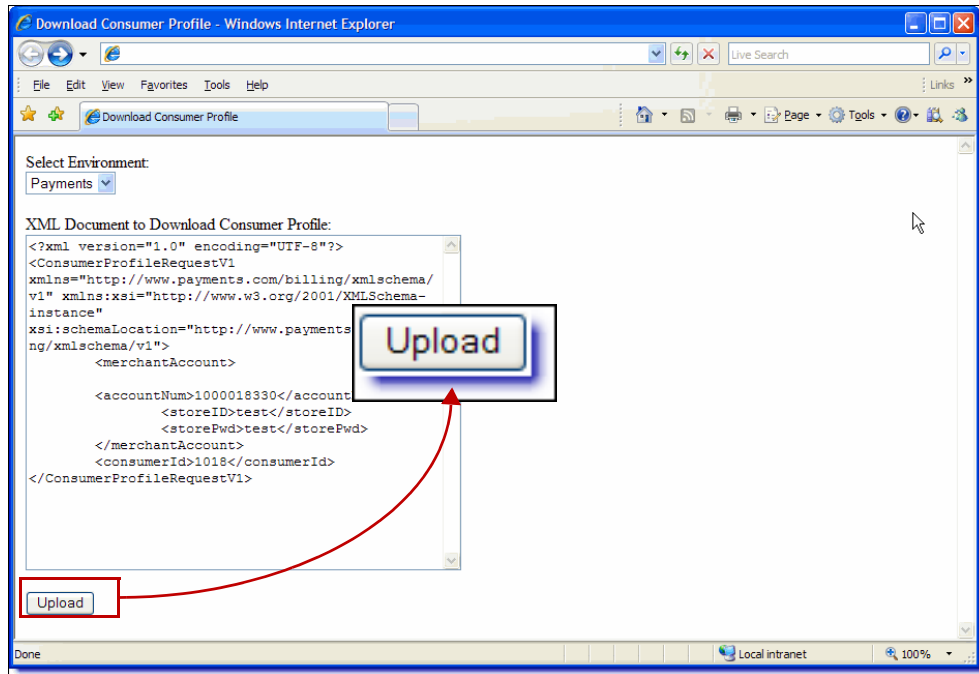
    }
    function doSubmit() {
        document.actionForm.action =
document.actionForm.urls.options[document.actionForm.urls.selectedIndex].value;
        document.actionForm.submit();
    }
</script>
</head>

<body>
  <form name="actionForm" action="" enctype="multipart/form-data" method="post">
    <p>
      Select Environment:
      <br>
      <select name="urls" onchange="updateUrl()">
        <option
value="https://webservices.optimalpayments.com/billingWS/billing/v1/consumerUpload"
selected>NETBANX</option>

        </select>
      </p>
      <p>
        XML Document to Download Consumer Profile:<br>
        <textarea name="txnRequest" cols=50 rows=20>
<?xml version="1.0" encoding="UTF-8"?>
<ConsumerProfileRequestV1
xmlns="http://www.optimalpayments.com/billing/xmlschema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.optimalpayments.com/billing/xmlschema/v1">
  <merchantAccount>
    <accountNum>1000010000</accountNum>
    <storeID>YourID</storeID>
    <storePwd>YourPassword</storePwd>
  </merchantAccount>
  <consumerId>1018</consumerId>
</ConsumerProfileRequestV1>
  </textarea>
      </p>
      <p>
        <input type="submit" value="Upload" onclick="doSubmit()">
      </p>
    </form>
  </body>
</html>

```

Viewed with a browser, the code would look something like this:



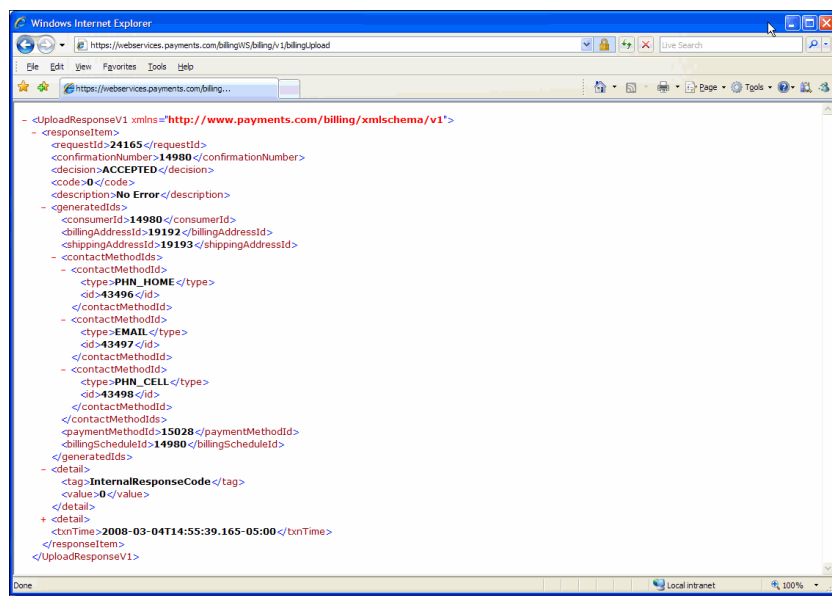
Clicking the Upload button sends the recurring billing consumer profile LOAD request via HTTP Post to NETBANX.



The maximum file size supported is 100K. If your HTTP Post exceeds 100K it will fail.

Responses

NETBANX returns a response like the following for a successful CREATION or UPDATE request.



NETBANX returns a response like the following for a successful LOAD request.

```

<?xml version='1.0' encoding='UTF-8'>
<ConsumerProfileResponseV1 xmlns='http://www.payments.com/billing/xmlschema/v1'>
<consumerProfile>
<consumerId>1018</consumerId>
<decision>ACCEPTED</decision>
<code>0</code>
<description>No Error</description>
<consumerInfo>
<billingAddresses>
<shippingAddresses>
<contactMethods>
<contactMethod>
<id>2121</id>
<type>EMAIL</type>
<value>janejones@emailserver.com</value>
</contactMethod>
</contactMethods>
<paymentMethods>
<billingSchedules>
<txnTime>2008-04-08T15:30:32.078-04:00</txnTime>
</consumerProfile>
</ConsumerProfileResponseV1>

```

NETBANX returns a response like the following for a failed recurring billing request.

```

<?xml version='1.0' encoding='UTF-8'>
<UploadResponseV1 xmlns='http://www.payments.com/billing/xmlschema/v1'>
<responseItem>
<requestId>24166</requestId>
<confirmationNumber>1</confirmationNumber>
<decision>ERROR</decision>
<code>8000</code>
<description>Unable to perform operation. Try again or contact Technical Support.</description>
<detail>
<tag>InternalResponseCode</tag>
<value>8000</value>
</detail>
<detail>
<tag>InternalResponseDescription</tag>
<value>Unable to perform operation. Try again or contact Technical Support.</value>
</detail>
</responseItem>
</UploadResponseV1>

```



For an explanation of the elements in the response, see Processing the response on page 1-24.
For an explanation of error codes, see Appendix B: Response Codes.

Response Codes

Response Codes

The following table describes the response codes that could be returned by NETBANX Web Services.

Table B-1: Response Codes

Response Code	Action	Description
1000	R	An internal error occurred. Please retry the transaction.
1001	R	An error occurred with the external processing gateway. Please retry the transaction.
1002	R	An internal error occurred. Please retry the transaction.
1003	D	An error occurred with the external processing gateway. Do not retry the transaction. Contact Technical Support for more information.
1004	M	Your account is not enabled for this transaction type. Please verify your parameters and retry the transaction.
1006	D	An error occurred with the external processing gateway. Do not retry the transaction. Contact Technical Support for more information.
1007	R	An internal error occurred. Please retry the transaction.
1008	D	An internal error occurred. Do not retry the transaction. Contact Technical Support for more information.
1018	D	The external processing gateway has reported the transaction is unauthorized. Do not retry the transaction. Contact Technical Support for more information.
1028	M	The external processing gateway has reported invalid data. Please verify your parameters and retry the transaction.
1043	D	The external processing gateway has reported the account type is invalid. Do not retry the transaction. Contact Technical Support for more information.
1060	D	The external processing gateway has reported a limit has been exceeded. Do not retry the transaction.
1078	R	The external processing gateway has reported a system error. Please retry the transaction.
1087	D	The external processing gateway has rejected the transaction. Do not retry the transaction. Contact Technical Support for more information.
2000	D	The echeck status cannot be found.
2001	C	You submitted a request containing a check number already used within the last 24 hours. Please use another Check Number value and retry the transaction.
2002	M	The echeck request cannot be found. Please verify your parameters and retry.
2003	M	The payment type you provided conflicts with the bank account type you provided. Please verify your parameters and retry the transaction.
2004	D	The echeck transaction cannot be found.
2005	M	An internal error occurred. Please verify your parameters and retry the transaction.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
2006	C	You have submitted a Decline transaction in response to a Settlement attempt.
2007	M	The payment type included with your request cannot be used in Credit transaction mode. Please verify your parameters and retry the request.
2008	C	You have submitted an invalid routing number. Please verify your parameter and retry the transaction.
2009	C	You have submitted an invalid bank account number. Please verify your parameter and retry the transaction.
2010	C	You have submitted an invalid check number. Please verify your parameter and retry the transaction.
2100	D	You have attempted to refund a Charge transaction that is not in a completed state. Do not retry the transaction.
2101	R	The requested refund amount exceeds the remaining Charge amount. Please verify your parameters and retry the transaction.
2102	R	You have submitted an invalid bank country in your request. Please verify your parameter and retry the transaction.
2500	D	A suitable merchant account could not be found for this request. Do not retry the transaction. Contact Technical Support for more information.
2501	D	An internal error occurred. Do not retry the transaction. Contact Technical Support for more information.
2502	M	The authentication failed for your transaction request. Either your credentials were entered incorrectly or your request was tampered with. Please verify your request and retry the transaction.
2503	M	You submitted an invalid shop ID. It must be a numeric value.
2504	M	You submitted a shop ID for which a shop does not exist. Please verify your parameters and retry the transaction.
2505	M	No merchant account could be found for the currency you submitted. Please verify your parameters and retry the transaction.
2506	M	More than one merchant account was found for the currency you submitted. Please provide a merchant account number and retry the request.
2507	M	The merchant account submitted does not correspond to the shop ID you submitted. Please verify your parameters and retry the transaction.
2508	M	You submitted a request that is missing the <i>encodedMessage</i> parameter. Please verify your parameters and retry the transaction.
2509	M	You submitted a request that has an invalid <i>encodedMessage</i> parameter. The <i>encodedMessage</i> could not be Base64 decoded. Please verify your parameters and retry the request.
2510	M	You submitted an invalid XML request. Please verify your request and retry the transaction.
2511	R	An internal error occurred. Please retry the transaction. If the error persists, contact Technical Support.
2512	M	You submitted a request that is missing the <i>signature</i> parameter. Please verify your parameters and retry the transaction.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
2513	M	You submitted a request that is missing the Shop ID parameter. Please verify your parameters and retry the transaction.
2514	D	An internal error occurred. Your merchant credentials could not be found. Do not retry the transaction. Contact Technical Support for more information.
2515	R	An internal error occurred. Please retry the transaction.
2516	D	An internal error occurred. Do not retry the transaction. Please contact Technical Support.
2517	D	An internal error occurred. Do not retry the transaction. Please contact Technical Support.
2518	R	An internal error occurred. Please retry the transaction. If the error persists, contact Technical Support.
2519	D	You submitted a request for a shop that is disabled. Do not retry the transaction.
2520	M	The <i>cartItem</i> and <i>feeItem</i> amounts do not equal the <i>totalAmount</i> . Please verify your parameters and retry the transaction.
2521	D	An internal error occurred. Your shop is not configured properly. Do not retry the transaction. Contact Technical Support for more information.
2523	R	An internal error occurred. Please retry the transaction. If the error persists, contact Technical Support.
2524	D	An internal error occurred. The customer profile is not configured properly. Do not retry the transaction. Contact Technical Support for more information.
2525	M	The customer profile associated with the token provided could not be located. Please verify your parameters and retry the transaction.
2526	R	An internal profile management error occurred and your request could not be processed. Please retry the transaction. If the error persists, contact Technical Support.
2527	M	You submitted a request with an unsupported locale. Please verify your parameters and retry the transaction.
2528	M	You submitted a request with a merchant customer ID that is not associated with the customer token provided. Please verify your parameters and retry the transaction.
2529	R	An internal error occurred. Please retry the transaction.
2530	D	An internal error occurred. Do not retry the transaction. Contact Technical Support for more information.
2531	M	A customer profile with the same merchant customer ID already exists. Please verify your parameters and retry the transaction.
2532	D	Your shop is not configured to process any payments. Do not retry the transaction. Contact Technical Support for more information.
2533	D	Your shop is not configured to process the selected payment method. Do not retry the transaction. Contact Technical Support for more information.
2534	M	The customer profile associated with the token you provided is not active. Please verify your parameters and ensure the customer status is active, and retry the transaction.
2535	M	The customer profile associated with the token you provided has a bank account that is not validated. Please verify your parameters or have your customer validate their account, and retry the transaction.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
2536	D	The transaction was declined because the maximum number of attempts has been reached. Do not retry the transaction.
2537	D	The bank account validation failed because the maximum number of attempts has been reached. Do not retry the transaction.
2538	D	The customer entered an invalid micro deposit amount.
2539	R	An internal error occurred. Please retry the transaction. If the error persists, contact Technical Support.
2540	M	The merchant reference number already exists. Please verify your parameters and retry the transaction.
2541	R	An internal error occurred when sending the micro deposit transaction. Please retry the transaction. If the error persists, contact Technical Support.
2542	D	The customer profile validation failed the AVS check.
2543	M	The customer has a status that cannot be validated. Please reset the customer status before retrying the transaction.
2544	D	The customer profile does not have the credit card information required for validation.
2445	D	The customer tried to register an EFT bank account that is already used by another customer. Do not retry the transaction.
2546	D	The customer tried to register a credit card that is already used by another customer. Do not retry the transaction.
2547	M	The customer profile associated with the token you provided does not have a credit card. Please verify your parameters, and retry the transaction.
2548	M	The customer profile used for this transaction has not been created. Please verify your parameters and retry the transaction.
2701	None	The Equifax validation was accepted with the condition that the customer's request should be verified.
2702	R	An internal error has occurred while processing an Equifax transaction. Retry the transaction. If the error persists, contact Technical Support.
2703	R	Some or all of the Equifax databases are down. Retry the transaction. If the error persists, contact Technical Support.
2704	D	The Equifax validation was rejected. Do not retry the transaction.
2705	R	The Equifax validation was rejected and should be treated manually. For example, the merchant can contact the customer directly.
2706	R	The Equifax validation was rejected due to invalid input. Please verify your parameters and retry the transaction.
2707	R	The Equifax validation was rejected due to an invalid address. Please verify your parameters and retry the transaction.
2708	R	An internal error occurred. The Equifax response codes are missing. Please verify your parameters and retry the transaction.
2709	D	The bank rejected the transaction. It could not process the request.
3001	C	You submitted an unsupported card type with your request. Please verify this parameter and retry the transaction.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
3002	C	You submitted an invalid card number or brand or combination of card number and brand with your request. Please verify these parameters and retry the transaction.
3003	C	You submitted an incorrect value for the <i>cvdIndicator</i> parameter with your request. Please verify this parameter and retry the transaction.
3004	C	You have requested an AVS check. Please ensure that the zip/postal code is provided.
3005	C	You submitted an incorrect value for the <i>cvd</i> parameter with your request. Please verify this parameter and retry the transaction.
3006	C	You submitted an expired credit card number with your request. Please verify this parameter and retry the request.
3007	D	Your request has failed the AVS check. Note that the amount has still been reserved on the customer's card and will be released in 3-5 business days. Please ensure the billing address is accurate before retrying the transaction.
3008	D	You submitted a card type for which the merchant account is not configured.
3009	D	Your request has been declined by the issuing bank.
3012	D	Your request has been declined by the issuing bank because the credit card expiry date submitted is invalid.
3013	D	Your request has been declined by the issuing bank due to problems with the credit card account.
3014	D	Your request has been declined – the issuing bank has returned an unknown response. Contact the cardholder's credit card company for further investigation
3015	D	The bank has requested that you process the transaction manually by calling the cardholder's credit card company.
3016	D	The bank has requested that you retrieve the card from the cardholder – it may be a lost or stolen card.
3017	C	You submitted an invalid credit card number with your request. Please verify this parameter and retry the transaction.
3018	R	The bank has requested that you retry the transaction.
3019	C	Your request has failed the CVD check. Please note that the amount has still been reserved on the customer's card and will be released in 3–5 business days. Please ensure the CVD value is accurate before retrying the transaction.
3020	R	The bank has requested that you retry the transaction.
3021	M	The confirmation number included in this request could not be found. Please verify this parameter and retry the transaction.
3022	D	The card has been declined due to insufficient funds.
3023	D	Your request has been declined by the issuing bank due to its proprietary card activity regulations.
3024	D	Your request has been declined because the issuing bank does not permit the transaction for this card.
3025	R	The external processing gateway has reported invalid data. Please verify your parameters and retry the transaction.
3026	D	The external processing gateway has reported the account type is invalid. Do not retry the transaction. Contact Technical Support for more information.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
3027	D	The external processing gateway has reported a limit has been exceeded. Do not retry the transaction.
3028	R	The external processing gateway has reported a system error. Please retry the transaction.
3029	D	The external processing gateway has rejected the transaction. Do not retry the transaction. Contact Technical Support for more information.
3030	D	The external processing gateway has reported the transaction is unauthorized. Do not retry the transaction. Contact Technical Support for more information.
3200	M	You have submitted an invalidly formatted authorization ID for this settlement. Please verify this parameter and retry the transaction.
3201	M	The authorization ID included in this settlement request could not be found. Please verify this parameter and retry the transaction.
3202	D	You have exceeded the maximum number of settlements allowed. Contact Technical Support for more information.
3203	M	The authorization is either fully settled or cancelled.
3204	M	The requested settlement amount exceeds the remaining authorization amount.
3205	M	The authorization you are attempting to settle has expired.
3206	D	The external processing gateway has rejected the transaction. Do not retry the transaction. Contact Technical Support for more information.
3402	M	The requested credit amount exceeds the remaining settlement amount.
3403	M	You have already processed the maximum number of credits allowed for this settlement.
3404	M	The settlement has already been fully credited.
3405	M	The settlement you are attempting to credit has expired.
3406	M	The settlement you are attempting to credit has not been batched yet. There are no settled funds available to credit.
3407	M	The settlement referred to by the transaction response ID you provided cannot be found. Please verify this parameter and retry the transaction.
3408	M	You have submitted an invalidly formatted response ID for the original purchase or settlement. Please verify this parameter and retry the transaction.
3409	M	The authorization ID included in this credit request could not be found. Please verify this parameter and retry the transaction.
3412	M	The Credit transaction you attempted was not permitted because your merchant account is in overdraft.
3413	M	The requested Credit amount exceeds the permissible Visa credit ratio. Please verify this parameter and retry the transaction.
3500	M	The confirmation number included in this request could not be found. Please verify this parameter and retry the transaction.
3501	M	The requested authorization reversal amount exceeds the remaining authorization amount.
3502	D	The authorization has already been settled. You cannot process an authorization reversal transaction against an authorization that has been settled.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
3503	M	The authorization reversal transaction is not supported for the card type used for the authorization you are attempting to reverse.
3504	M	The external processing gateway for which your merchant account is configured does not support partial authorization reversals. Ensure that the amount you are trying to reverse is identical to the amount in the original authorization and retry the transaction.
3601	D	The 3D Secure authentication of this cardholder by the card issuer failed.
3602	M	The confirmation number included in the 3D Secure authentication request could not be found. The confirmation number must be the one returned by the payment processor in response to the original authorization or purchase.
3603	M	You submitted a request that is not available for 3D Secure authentication.
3604	D	The original authorization or purchase request has expired. The 3D Secure authentication request must be completed within 60 minutes of the original authorization or purchase.
3605	D	The 3D Secure authentication service was not available for the card used in this transaction.
3701	M	The confirmation number included in this request could not be found. Please verify this parameter and retry the transaction.
3702	D	You cannot cancel this payment transaction. It is not in a state that can be cancelled – for example, it may already have been completed and therefore not be in a pending state.
3800	D	The transaction was declined by the authentication gateway. Do not retry the transaction.
3801	D	The transaction was declined by the payment gateway. Do not retry the transaction.
3802	R	The transaction was not completed successfully. Please retry the transaction.
3803	D	The transaction attempt failed. Do not retry the transaction. Please contact Technical Support for more information.
4001	D	The card number or email address associated with this transaction is in our negative database.
4002	D	The transaction was declined by our Risk Management department.
5000	M	Your merchant account authentication failed. Either your store ID/password are invalid or the IP address from which you are sending the transaction has not been authorized. Please verify these parameters and retry the transaction. Please contact Technical Support if the error persists.
5001	M	You submitted an invalid currency code with your request. Please verify this parameter and retry the transaction.
5002	M	You submitted an invalid payment method with your request. Please verify this parameter and retry the transaction.
5003	C	You submitted an invalid amount with your request. Please verify this parameter and retry the transaction.
5004	M	You submitted an invalid account type with your request. Please verify this parameter and retry the transaction.
5005	M	You submitted an invalid operation type with your request. Please verify this parameter and retry the transaction.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
5006	M	You submitted an invalid personal ID type with your request. Please verify this parameter and retry the transaction.
5007	M	You submitted an invalid product type with your request. Please verify this parameter and retry the transaction.
5008	M	You submitted an invalid carrier with your request. Please verify this parameter and retry the transaction.
5009	M	You submitted an invalid ship method with your request. Please verify this parameter and retry the transaction.
5010	M	You submitted an invalid ID country with your request. Please verify this parameter and retry the transaction.
5011	M	You submitted an invalid order date and time with your request. Please verify this parameter and retry the transaction.
5012	M	You submitted an invalid ship country parameter with your request. Please verify this parameter and retry the transaction.
5013	M	You submitted an invalid ship state parameter with your request. Please verify this parameter and retry the transaction.
5014	M	You submitted an invalid transaction type with your request. Please verify this parameter and retry the transaction.
5016	M	Either you submitted an invalid merchant account or the merchant account could not be found. Please verify this parameter and retry the transaction. Please contact Technical Support if the error persists.
5017	D	The merchant account submitted with your request is not enabled. Do not retry the transaction. Contact Technical Support for more information.
5018	M	You submitted a request that is missing a mandatory field. Please verify your parameters and retry the transaction.
5019	D	There is no processor set up for the merchant account submitted with your request. Do not retry the transaction. Contact Technical Support for more information.
5020	M	The currency type included with your request does not match the currency type of your merchant account. Please verify your parameters and retry the request.
5021	D	Your transaction request has been declined. Please verify the transaction details before you attempt this transaction again. Should you require more information, please contact Technical Support.
5022	M	You submitted a request that is missing search criteria. Please verify your parameters and retry the transaction.
5023	M	You submitted an invalid XML request. Please verify your request and retry the transaction.
5024	M	You submitted an invalid ID expiry date for this request. Please verify this parameter and retry the transaction.
5025	M	The search criteria you submitted is currently not supported. Please verify your search criteria and retry.
5026	M	The Web Services API does not currently support credit card transactions. Please verify your parameters and retry your transaction.
5027	M	You submitted an invalid risk service name with your transaction request. Please verify your parameters and retry your transaction.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
5028	M	You submitted a batch file with the same file name, file size, and number of entries within the last 24 hours.
5029	M	You submitted an improperly formatted batch file.
5030	M	You submitted a batch file that has exceeded the maximum number of rows allowed.
5031	M	The transaction you have submitted has already been processed.
5032	C	You submitted an invalid city with your request. Please verify this parameter and retry the transaction.
5033	C	You submitted an invalid country with your request. Please verify this parameter and retry the transaction.
5034	C	You submitted an invalid email address with your request. Please verify this parameter and retry the transaction.
5035	C	You submitted an invalid name with your request. Please verify this parameter and retry the transaction.
5036	C	You submitted an invalid phone number with your request. Please verify this parameter and retry the transaction.
5037	C	You submitted an invalid zip/postal code with your request. Please verify this parameter and retry the transaction.
5038	C	You submitted an invalid state/province with your request. Please verify this parameter and retry the transaction.
5039	C	You submitted an invalid street with your request. Please verify this parameter and retry the transaction.
5040	D	Your merchant account is not configured for the transaction you attempted. Please contact Technical Support for more information.
5041	M	You submitted an invalid <i>merchantData</i> parameter. Please verify this parameter and retry the transaction.
5042	M	The merchant reference number is missing or invalid or it exceeds the maximum permissible length. Please verify this parameter and retry the transaction.
5043	M	You submitted an invalid account open date with your request. Please verify this parameter and retry the transaction.
5044	M	You submitted an invalid customer ID with your request. Please verify this parameter and retry the transaction.
5045	M	You submitted an invalid customer IP address with your request. Please verify this parameter and retry the transaction.
5046	M	You submitted an invalid merchant SIC code with your request. Please verify this parameter and retry the transaction.
5047	M	You submitted an invalid value for the previous customer parameter with your request. Please verify this parameter and retry the transaction.
5048	M	You submitted an invalid product code with your request. Please verify this parameter and retry the transaction.
5049	M	You submitted an invalid value for user data with your request. Please verify this parameter and retry the transaction.
5050	D	An error occurred with your merchant account configuration. Please contact Technical Support.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
5051	C	You have submitted an invalid confirmation number. Please verify your parameter and retry.
5052	C	You have submitted an invalid Zip/Postal code. Please verify your parameter and retry the transaction.
5053	C	You have submitted invalid personal ID information. Please verify your parameters and retry the transaction.
5054	M	You cannot submit recurring billing elements with this operation type. Please verify that your request includes the correct operation type and retry the transaction.
5055	M	You submitted an invalid ECI value. Please verify this parameter and retry the transaction.
5056	M	An ECI value must be provided for 3D Secure authentication. Please verify this parameter and retry the transaction.
5057	M	You submitted a credit card brand that does not support 3D Secure authentication. Please verify this parameter and retry the transaction.
5058	M	A CAVV value must be provided for 3D Secure authentication. Please verify this parameter and retry the transaction.
5059	M	You submitted an invalid value for the currency request service.
5060	M	You submitted an invalid <i>recurringIndicator</i> parameter with your request. Please verify this parameter and retry the transaction.
5061	M	A security check failed while processing this transaction. Please retry the transaction. If the error persists, contact Technical Support.
5062	M	You have submitted invalid characters as part of your request. Please verify that all parameters contain only characters that can be URL encoded, and retry the transaction.
5063	M	You have submitted an invalid <i>KEYWORD</i> with your request. Please verify this parameter and retry the transaction."
7000	D	The address you are trying to create already exists.
7001	M	Unable to create address. Try again or contact Technical Support.
7002	M	Unable to update address. Try again or contact Technical Support.
7003	M	Unable to locate the specified address.
7004	M	Unable to load address. Try again or contact Technical Support.
7005	M	You have provided an invalid address type.
7006	M	Unable to locate addresses for consumer.
7007	M	Unable to load addresses. Try again or contact Technical Support.
7010	D	The consumer you are trying to create already exists.
7011	M	Unable to create consumer. Try again or contact Technical Support.
7012	M	Unable to update consumer. Try again or contact Technical Support.
7013	M	Unable to locate the specified consumer.
7014	M	Unable to load consumer. Try again or contact Technical Support.
7015	M	You have specified an invalid consumer creation type.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
7016	M	You have specified an invalid consumer title.
7020	D	The contact method you are trying to create already exists.
7021	M	Unable to create contact method. Try again or contact Technical Support.
7022	M	Unable to update contact method. Try again or contact Technical Support.
7023	M	Unable to locate the specified contact method.
7024	M	Unable to load contact method. Try again or contact Technical Support.
7025	M	You have provided an invalid contact method type.
7026	M	Unable to locate contact methods for consumer.
7027	M	Unable to load contact methods. Try again or contact Technical Support.
7040	D	The payment method you are trying to create already exists.
7041	M	Unable to create payment method. Try again or contact Technical Support.
7042	M	Unable to update payment method. Try again or contact Technical Support.
7043	M	Unable to locate the specified payment method.
7044	M	Unable to load payment method. Try again or contact Technical Support.
7045	M	You have provided an invalid payment method type.
7046	M	Unable to locate payment methods for consumer.
7047	M	Unable to load payment methods. Try again or contact Technical Support.
7050	D	The billing schedule you are trying to create already exists.
7051	M	Unable to create billing schedule. Try again or contact Technical Support.
7052	M	Unable to update billing schedule. Try again or contact Technical Support.
7053	M	Unable to locate the specified billing schedule.
7054	M	Unable to load billing schedule. Try again or contact Technical Support.
7055	M	You have provided an invalid payment interval type.
7056	M	Unable to locate billing schedules for consumer.
7057	M	Unable to load billing schedules. Try again or contact Technical Support.
7058	M	Billing amount must be greater than zero.
7059	M	Unable to update billing schedule status. Try again or contact Technical Support.
7070	M	Unable to execute the search operation. Try again or contact Technical Support.
7080	M	Unable to update records. Try again or contact Technical Support.
7090	D	The consumer status you are trying to create already exists.
7091	M	Unable to create consumer status. Try again or contact Technical Support.
7092	M	Unable to update consumer status. Try again or contact Technical Support.
7093	M	Unable to locate the specified consumer status.
7094	M	Unable to load consumer status. Try again or contact Technical Support.
7095	M	Unable to locate the merchant legal entity.
7096	M	Unable to retrieve merchant legal entity. Try again or contact Technical Support.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
7097	D	An internal error occurred. Response Detail could not be generated.
7098	M	You have provided an invalid credit card number.
8000	M	Unable to perform operation. Try again or contact Technical Support.
8001	M	You have submitted an invalid request. Please verify your parameters and retry.
8002	M	Your request is missing the <i>consumerInfo</i> element.
8003	M	Your request is missing a value for the <i>firstName</i> element.
8004	M	Your request is missing a value for the <i>lastName</i> element.
8005	M	Your request is missing the <i>billingAddress/shippingAddress</i> element.
8006	M	Your request is missing <i>billingAddress/shippingAddress</i> information. There is no value for the <i>street</i> element.
8007	M	Your request is missing <i>billingAddress/shippingAddress</i> information. There is no value for the <i>city</i> element.
8008	M	Your request is missing <i>billingAddress/shippingAddress</i> information. There is no value for the <i>zip</i> element.
8009	M	Your request is missing <i>billingAddress/shippingAddress</i> information. There is no value for the <i>country</i> element.
8010	M	Your request is missing <i>billingAddress/shippingAddress</i> information. There is no value for the <i>state</i> or <i>region</i> element.
8011	M	Your request is missing the <i>contactMethod</i> element.
8012	M	Your request is missing a home telephone number or a cell phone number or an email address for the <i>type</i> element.
8013	M	Your request is missing the <i>paymentMethod</i> element.
8014	M	Your request is missing a value for the <i>id</i> child element of the <i>billingAddress</i> element.
8015	M	Your request is missing either the <i>card</i> or <i>check</i> child element of the <i>paymentMethod</i> element.
8016	M	Your request is missing the <i>card</i> element.
8017	M	Your request is missing a value for the <i>cardNum</i> element.
8018	M	Your request is missing a value for the <i>cardType</i> element.
8019	M	Your request is missing the <i>cardExpiry</i> element.
8020	M	Your request is missing a value for the <i>month</i> element.
8021	M	Your request is missing a value for the <i>year</i> element.
8022	M	Your request is missing the <i>check</i> element.
8023	M	Your request is missing a value for the <i>accountNum</i> element.
8024	M	Your request is missing a value for the <i>accountType</i> element.
8025	M	Your request is missing a value for the <i>routingNum</i> element.
8026	M	Your request is missing a value for the <i>bankName</i> element.
8027	M	Your request is missing a value for the <i>checkNum</i> element.
8028	M	Your request is missing the <i>billingSchedule</i> element.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
8029	M	Your request is missing a value for the <i>intervalCode</i> element.
8030	M	Your request is missing a value for the <i>nextBillingDate</i> element.
8031	M	Your request is missing a value for the <i>startDate</i> element.
8032	M	Your request is missing a value for the <i>amount</i> element.
8033	M	Your request is missing the <i>merchantAccount</i> element.
8034	M	Your request is missing a value for the <i>accountNum</i> element.
8035	M	Your request is missing a value for the <i>storeID</i> element.
8036	M	Your request is missing a value for the <i>storePwd</i> element.
8037	M	The credentials supplied with the <i>merchantAccount</i> element could not be validated.
8038	M	An internal error occurred. Please retry the request.
8039	M	Your request is missing a value for the <i>consumerId</i> element.
8040	M	Your request is missing a value for the <i>title</i> element.
8041	M	Your request is missing a value for the <i>merchantRefNum</i> child element of the <i>consumerInfo</i> element.
8042	M	Your request is missing a value for the <i>street2</i> element.
8043	M	You have submitted an invalid <i>address id</i> element. It does not match the <i>consumerId</i> submitted with the request.
8044	M	You have submitted an invalid <i>contactMethod id</i> element. It does not match the <i>consumerId</i> submitted with the request.
8045	M	Your request is missing a value for the <i>paymentMethodId</i> element.
8046	M	Your request is missing a value for the <i>ccHolderName</i> element.
8047	M	Your request is missing a value for the <i>merchantRefNum</i> child element of the <i>paymentMethod</i> element.
8048	M	Your request is missing a value for the <i>id</i> child element of the <i>billingSchedule</i> element.
8049	M	Your request is missing a value for the <i>serviceName</i> element.
8050	M	Your request is missing a value for the <i>statusCode</i> element.
8051	M	Your request is missing a value for the <i>lastDayOfTheMonth</i> element.
8052	M	Your request is missing a value for the <i>merchantRefNum</i> child element of the <i>billingSchedule</i> element.
8053	M	Your request contains two <i>consumerId</i> elements that don't match.
8054	M	You attempted an unsupported operation.
8055	M	You have submitted an invalid <i>endDate</i> element.
8056	M	You have submitted an invalid <i>issueNumber</i> element.
8057	M	You have submitted an invalid <i>statusCode</i> element.
8058	M	You have submitted an invalid <i>intervalCode</i> element.
8059	M	You have submitted an invalid <i>title</i> element.
8060	M	You have submitted an invalid <i>cardType</i> element.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
8061	M	You have submitted an invalid <i>accountType</i> element.
8062	M	You have submitted an invalid <i>state</i> element.
8063	M	You have submitted an invalid <i>country</i> element.
8064	M	You have submitted an invalid <i>startDate</i> element.
8065	M	You have submitted an invalid <i>endDate</i> element.
8066	M	You have submitted an invalid <i>nextBillingDate</i> element.
8067	M	The <i>endDate</i> cannot be before the <i>startDate</i> .
8068	M	The <i>nextBillingDate</i> cannot be after the <i>endDate</i> .
8069	M	The <i>nextBillingDate</i> cannot be before the <i>startDate</i> .
8070	M	You have submitted an invalid <i>cardExpiry</i> element.
8071	M	You have submitted an invalid paymentMethod <i>billingAddressId</i> . Either it does not match the <i>consumerId</i> submitted with the request or it is the wrong address type.
8072	M	You have submitted an invalid paymentMethod <i>shippingAddressId</i> . Either it does not match the <i>consumerId</i> submitted with the request or it is the wrong address type.
8073	M	You have submitted an invalid contactMethod <i>type</i> element.
8074	M	You have submitted an invalid paymentMethod <i>id</i> element. It does not match the <i>consumerId</i> submitted with the request.
8075	M	You have submitted an invalid billingSchedule <i>id</i> element. It does not match the <i>consumerId</i> submitted with the request.
8076	M	You have submitted an invalid <i>consumerId</i> element. The consumer specified is not registered with this merchant.
8077	M	You have submitted an invalid billingSchedule <i>paymentMethodId</i> element. The payment method identified is not registered with this consumer.
8078	D	You have submitted a duplicate request within the last 24 hours.
8079	D	You have submitted a file that has already been processed within the last 24 hours.
8080	M	You have submitted an invalid record type. It is missing the <i>recurringBillingDetails</i> element.
8081	D	You have submitted an empty file.
8082	M	You have submitted an invalid request.
8083	M	Your request is missing a value for the <i>cvv</i> element.
8084	M	You have submitted an invalid <i>cvv</i> element.
8085	M	Your request is missing a value for the Merchant Transaction ID element.
8086	M	Your request is missing a value for the Transaction ID element.
8087	M	You have submitted an invalid <i>accountNum</i> element.
8088	M	You have submitted a record with an invalid number of fields.
8089	D	You have submitted a file that exceeds the file size limit.
8090	M	You have submitted an invalid Transaction Date element.
8091	M	You have submitted an invalid ID Expiration element.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
8092	M	You have submitted an unsupported transaction type for the Transaction Code element.
8093	M	You have submitted an invalid <i>billingSchedule</i> element. Your request is either missing a <i>paymentMethodId</i> or missing a new <i>paymentMethod</i> element.
8094	M	You have submitted an invalid <i>paymentMethod</i> element. Your request is either missing a <i>billingAddressId</i> or missing a new <i>billingAddress</i> element.
8095	M	You have submitted a file that exceeds the limit of 1000 records.
8096	M	You have submitted an invalid <i>transactionMode</i> element. Please verify this parameter and retry the transaction.
8100	M	Your request is missing a value for the <i>aggregateAccountNum</i> element.
8101	M	You have submitted an invalid <i>aggregateAccountNum</i> element.
8102	M	You have submitted an invalid <i>Version</i> element.
8103	M	You have submitted an invalid <i>storeID</i> element.
8104	M	You have submitted an invalid <i>storePwd</i> element.
8105	M	You have submitted an <i>accountNum</i> element that is not associated with the <i>aggregateAccountNum</i> element.
8106	M	You have submitted an invalid <i>aggregateAccountNum</i> element. No aggregated accounts were found.
8107	M	Your request is missing a value for the <i>intervalValue</i> element.
8108	M	You have submitted an invalid <i>intervalValue</i> element.
8109	M	The <i>intervalValue</i> element cannot be provided with the defined <i>intervalCode</i> .
8110	M	The value for the <i>intervalValue</i> element must be between 1 and 365.
8112	M	You have submitted an invalid <i>merchantAccount</i> element. The merchant account number provided is not set up for Recurring Billing.
8119	M	Your request is missing a value for the <i>userName</i> element.
8120	M	Your request is missing a value for the <i>password</i> element.
8121	M	Your request is missing a value for the <i>customerIp</i> element.
8122	M	Your request is missing a value for the <i>siteUrl</i> element.
8123	M	Your request is missing a value for the <i>keyword</i> element.
8124	M	Your request is missing a value for the <i>welcomeEmail</i> element.
9000	D	Your transaction request could not be processed because your merchant account has no available child accounts for load balancing.
9001	D	Your transaction request could not be processed because your merchant account has no available child accounts for tier routing.
9002	D	Your transaction request could not be processed because your merchant account has no available child accounts for sticky routing.
9003	D	Your transaction request could not be processed because your merchant account has no available child accounts for bin country routing.
9004	D	Your transaction request could not be processed because your merchant account has no available child accounts for brand routing.

Table B-1: Response Codes (Continued)

Response Code	Action	Description
9005	D	Your transaction request could not be processed because your merchant account has no available child accounts for keyword routing.

Action codes

The transaction processor returns an action along with each response code. The meanings for the action codes are as follows:

- C = Consumer Parameter Error. The consumer has provided incorrect information. Ask the customer to correct the information.
- D = Do Not Retry
- M = Merchant Parameter Error. Your application has provided incorrect information. Verify your information.
- R = Retry

Return Codes

These are the codes used by the ACH or EFT system for identifying the various reasons a check or payment is returned. These codes are generated by customers' banks (RDFI) to return items.

Table B-2: Return Codes

Code	Description
R01	Insufficient Funds
R02	Account Closed
R03	No Account/Unable to Locate Account
R04	Invalid Account Number
R06	Returned per ODFI's Request
R07	Authorization Revoked by Customer *
R08	Payment Stopped or Stop Payment
R09	Uncollected Funds
R10	Customer Advises Not Authorized *
R11	Check Truncation Entry Return
R12	Branch Sold to Another RDFI
R13	RDFI Not Qualified to Participate
R14	Representative Payee Deceased or Unable to Continue in that Capacity
R15	Beneficiary or Account Holder Deceased
R16	Account Frozen
R17	File Record Edit Criteria
R18	Improper Effective Entry Date

Table B-2: Return Codes (Continued)

Code	Description
R19	Amount Field Error
R20	Non-Transaction Account
R21	Invalid Company Identification
R22	Invalid Individual ID Number
R23	Credit Entry Refused by Receiver
R24	Duplicate Entry
R25	Addenda Error
R26	Mandatory Field Error
R27	Trace Number Error
R28	Routing Number Check Digit Error
R29	Corporate Customer Advises Not Authorized
R30	RDFI Not Participant in CK Truncation Program
R31	Permissible Return Entry
R32	RDFI Non-Settlement
R33	Return of XCX Entry
R34	Limited Participation DFI
R35	Return of Improper Debit Entry
R36	Return of Improper Credit Entry
R51	Item is Ineligible, Notice Not Provided, Signature Not Genuine, or Item Altered
R52	Stop Payment on Item
900	Validation Rejection
901	Insufficient Funds ("RA" direct withdrawals only)
902	Cannot Locate Account
903	Stopped/Recalled/Revoked Payment
904	Post-Dated or Date Expired
905	Account Closed
906	Account Transferred
907	No Chequing Privileges
908	Funds Not Free
910	Drawee/Payee Deceased
911	Blocked Account
912	Erroneous or Invalid Account Number
914	Erroneous Drawee/Payee Name
915	Refused by Drawee/Payee
916	Exceeds Funds Transfer Cap
917	Funds Transfer Not Accepted

Table B-2: Return Codes (Continued)

Code	Description
990	Default by a Financial Institution
998	No Return Agreement

*An initially cleared item can be returned as “Unauthorized”, “Authorized”, or “Authorization Revoked” for up to 60 days following the date of Presentment.

Geographical Codes

Province codes

Table C-1: Province Codes

Province	Code
Alberta	AB
British Columbia	BC
Manitoba	MB
New Brunswick	NB
Newfoundland	NL
Nova Scotia	NS
Northwest Territories	NT
Nunavut	NU
Ontario	ON
Prince Edward Island	PE
Quebec	QC
Saskatchewan	SK
Yukon	YT

State codes

Table C-2: State Codes

State	Code	State	Code
Alabama	AL	Alaska	AK
Arizona	AZ	Arkansas	AR
California	CA	Colorado	CO
Connecticut	CT	Delaware	DE
District of Columbia	DC	Florida	FL
Georgia	GA	Hawaii	HI
Idaho	ID	Illinois	IL
Indiana	IN	Iowa	IA
Kansas	KS	Kentucky	KY
Louisiana	LA	Maine	ME
Maryland	MD	Massachusetts	MA
Michigan	MI	Minnesota	MN
Mississippi	MS	Missouri	MO
Montana	MT	Nebraska	NE
Nevada	NV	New Hampshire	NH
New Jersey	NJ	New Mexico	NM
New York	NY	North Carolina	NC
North Dakota	ND	Ohio	OH
Oklahoma	OK	Oregon	OR
Pennsylvania	PA	Rhode Island	RI
South Carolina	SC	South Dakota	SD
Tennessee	TN	Texas	TX
Utah	UT	Vermont	VT
Virginia	VA	Washington	WA
West Virginia	WV	Wisconsin	WI
Wyoming	WY	United States Federal	US
International	IT	Puerto Rico	PR
U.S. Virgin Islands	VI	Northern Mariana Is.	MP
Guam	GU	American Samoa	AS
Palau	PW	Armed Forces Americas	AA
Armed Forces Europe	AE	Armed Forces Pacific	AP

Country codes

Table C-3: Country Codes

Country	Code	Country	Code
Afghanistan	AF	Åland Islands	AX
Albania	AL	Algeria	DZ
American Samoa	AS	Andorra	AD
Angola	AO	Anguilla	AI
Antarctica	AQ	Antigua and Barbuda	AG
Argentina	AR	Armenia	AM
Aruba	AW	Australia	AU
Austria	AT	Azerbaijan	AZ
Bahamas	BS	Bahrain	BH
Bangladesh	BD	Barbados	BB
Belarus	BY	Belgium	BE
Belize	BZ	Benin	BJ
Bermuda	BM	Bhutan	BT
Bolivia	BO	Bosnia and Herzegovina	BA
Botswana	BW	Bouvet Island	BV
Brazil	BR	British Indian Ocean Territory	IO
Brunei Darussalam	BN	Bulgaria	BG
Burkina Faso	BF	Burundi	BI
Cambodia	KH	Cameroon	CM
Canada	CA	Cape Verde	CV
Cayman Islands	KY	Central African Republic	CF
Chad	TD	Chile	CL
China	CN	Christmas Island	CX
Cocos (Keeling) Islands	CC	Colombia	CO
Comoros	KM	Congo	CG
Congo, Democratic Republic of	CD	Cook Islands	CK
Costa Rica	CR	Côte D'Ivoire	CI
Croatia	HR	Cuba	CU
Cyprus	CY	Czech Republic	CZ
Denmark	DK	Djibouti	DJ
Dominica	DM	Dominican Republic	DO
Ecuador	EC	Egypt	EG

Table C-3: Country Codes (Continued)

Country	Code	Country	Code
El Salvador	SV	Equatorial Guinea	GQ
Eritrea	ER	Estonia	EE
Ethiopia	ET	Falkland Islands	FK
Faroe Islands	FO	Fiji	FJ
Finland	FI	France	FR
Metropolitan France	FX	French Guiana	GF
French Polynesia	PF	French Southern Territories	TF
Gabon	GA	Gambia	GM
Georgia	GE	Germany	DE
Ghana	GH	Gibraltar	GI
Greece	GR	Greenland	GL
Grenada	GD	Guadeloupe	GP
Guam	GU	Guatemala	GT
Guernsey	GG	Guinea	GN
Guinea-Bissau	GW	Guyana	GY
Haiti	HT	Heard and McDonald Islands	HM
Honduras	HN	Hong Kong	HK
Hungary	HU	Iceland	IS
India	IN	Indonesia	ID
Iran (Islamic Republic of)	IR	Iraq	IQ
Ireland	IE	Isle of Man	IM
Israel	IL	Italy	IT
Jamaica	JM	Japan	JP
Jersey	JE	Jordan	JO
Kazakhstan	KZ	Kenya	KE
Kiribati	KI	Korea, Democratic People's Republic	KP
Korea, Republic of	KR	Kuwait	KW
Kyrgyzstan	KG	Lao People's Democratic Republic	LA
Latvia	LV	Lebanon	LB
Lesotho	LS	Liberia	LR
Libyan Arab Jamahiriya	LY	Liechtenstein	LI
Lithuania	LT	Luxembourg	LU
Macau	MO	Macedonia	MK
Madagascar	MG	Malawi	MW

Table C-3: Country Codes (Continued)

Country	Code	Country	Code
Malaysia	MY	Maldives	MV
Mali	ML	Malta	MT
Marshall Islands	MH	Martinique	MQ
Mauritania	MR	Mauritius	MU
Mayotte	YT	Mexico	MX
Micronesia, Federated States of	FM	Moldova, Republic of	MD
Monaco	MC	Mongolia	MN
Montenegro	ME	Montserrat	MS
Morocco	MA	Mozambique	MZ
Myanmar	MM	Namibia	NA
Nauru	NR	Nepal	NP
The Netherlands	NL	Netherlands Antilles	AN
New Caledonia	NC	New Zealand	NZ
Nicaragua	NI	Niger	NE
Nigeria	NG	Niue	NU
Norfolk Island	NF	Northern Mariana Islands	MP
Norway	NO	Oman	OM
Pakistan	PK	Palau	PW
Palestinian Territory, Occupied	PS	Panama	PA
Papua New Guinea	PG	Paraguay	PY
Peru	PE	Philippines	PH
Pitcairn	PN	Poland	PL
Portugal	PT	Puerto Rico	PR
Qatar	QA	Reunion	RE
Romania	RO	Russian Federation	RU
Rwanda	RW	Saint Barthélemy	BL
Saint Helena	SH	Saint Kitts and Nevis	KN
Saint Lucia	LC	Saint Vincent and the Grenadines	VC
Samoa	WS	San Marino	SM
Sao Tome and Principe	ST	Saudi Arabia	SA
Senegal	SN	Serbia	RS
Seychelles	SC	Sierra Leone	SL
Singapore	SG	Slovakia (Slovak Republic)	SK
Slovenia	SI	Solomon Islands	SB

Table C-3: Country Codes (Continued)

Country	Code	Country	Code
Somalia	SO	South Africa	ZA
South Georgia and the South Sandwich Islands	GS	Spain	ES
Sri Lanka	LK	St. Pierre and Miquelon	PM
Sudan	SD	Suriname	SR
Svalbard and Jan Mayen Islands	SJ	Swaziland	SZ
Sweden	SE	Switzerland	CH
Syrian Arab Republic	SY	Taiwan	TW
Tajikistan	TJ	Tanzania, United Republic of	TZ
Timor-Leste	TL	Thailand	TH
Togo	TG	Tokelau	TK
Tonga	TO	Trinidad and Tobago	TT
Tunisia	TN	Turkey	TR
Turkmenistan	TM	Turks and Caicos Islands	TC
Tuvalu	TV	Uganda	UG
Ukraine	UA	United Arab Emirates	AE
United Kingdom	GB	United States	US
United States Minor Outlying Islands	UM	Uruguay	UY
Uzbekistan	UZ	Vanuatu	VU
Vatican City State (Holy See)	VA	Venezuela	VE
Vietnam	VN	Virgin Islands (British)	VG
Virgin Islands (U.S.)	VI	Wallis and Futuna Islands	WF
Western Sahara	EH	Yemen	YE
Zambia	ZM	Zimbabwe	ZW