



# GPS Batch Credit Card SOAP Web Service

Version 2.0

Guardian Payment Systems

---

## Contents

Guardian Payment Systems Batch Credit Card Processing .....	1
Process Details – Authorization/Settle versus Charge .....	1
Technical Details .....	1
Credit Card Authorization Batch .....	2
Submitting an Authorization Batch to the Web Service: SubmitAuthBatch(...) .....	2
Authorization Batch Header Object .....	2
Authorization Batch Request Object .....	2
Authorization Batch Object .....	2
Authorization Object .....	2
Lodging Object .....	3
Authorization Batch Response Object .....	4
Auth Batch Status during Processing: GetCreditCardBatchStatusSummary(...) .....	4
Credit Card Batch Header Object .....	4
Credit Card Batch Status Summary Request Object .....	4
Credit Card Batch Status Summary Response Object .....	4
Credit Card Batch Status Summary Object .....	5
Auth Batch Status after Processing: GetAuthBatchResults(...) .....	5
Credit Card Batch Header Object .....	5
Authorization Batch Result Request Object .....	6
Authorization Batch Result Response Object .....	6
Authorization Results Object .....	6
Credit Card Settle Batch .....	7
Store Authorization Results .....	7

Submitting an Settle Batch to the Web Service: SubmitSettleBatch(...)..... 7

- Credit Card Batch Header Object..... 7
- Settle Batch Request Object ..... 8
- Settle Batch Object ..... 8
- Settle Object..... 8
- Settle Batch Response Object ..... 8

Settle Batch Status during Processing: GetCreditCardBatchStatusSummary(...)..... 9

- Credit Card Batch Header Object..... 9
- Credit Card Batch Status Summary Request Object ..... 9
- Credit Card Batch Status Summary Response Object..... 9
- Credit Card Batch Status Summary Object ..... 10

Settle Batch Status after Processing: GetSettleBatchResults(...) ..... 10

- Credit Card Batch Header Object..... 10
- Settle Batch Result Request Object ..... 10
- Settle Batch Result Response Object ..... 11
- Settle Results Object ..... 11

Credit Card Charge Batch..... 13

Submitting an Charge Batch to the Web Service: SubmitChargeBatch(...)..... 13

- Charge Batch Header Object..... 13
- Charge Batch Request Object ..... 13
- Charge Batch Object ..... 14
- Charge Object..... 14
- Lodging Object ..... 15
- Charge Batch Response Object ..... 15

Charge Batch Status during Processing: GetCreditCardBatchStatusSummary(...)	15
Credit Card Batch Header Object	16
Credit Card Batch Status Summary Request Object	16
Credit Card Batch Status Summary Response Object	16
Credit Card Batch Status Summary Object	16
Charge Batch Status during and after Processing: GetChargeBatchResults(...)	17
Credit Card Batch Header Object	17
Charge Batch Result Request Object	17
Charge Batch Result Response Object	17
Charge Results Object	17
Appendix A - Specific Test Cases	19
Test Data that will pass	20
Specific Return Values	21
Appendix B - Re-Fetching a Specific Response	22
Appendix C – Complete Authorization and Settlement Examples	23
Basic assumptions	23
Examples	24
Generating Web References	24
App.config file	24

## Guardian Payment Systems Batch Credit Card Processing

The Guardian Payment Systems Batch Credit Card Processing system provides an SOAP Web Service for customers to be able to process batches of Credit Card Transactions. Batch processing can take the form of Auth/Settle, Charge, Verify or Refund.

This service provides the following methods:

- Credit Card Charge
- Credit Card Authorization
- Credit Card Settle

This document outlines the process that a customer needs to do to use these services. Additionally, Guardian offers the following services:

- a. XML/POX interface for batch credit card processing, and
- b. SOAP and XML/POX interfaces for real-time credit card processing.

### Process Details – Authorization/Settle versus Charge

If you are asking, which credit card process should I use -- Authorization/Settle or Charge? Charge is the most common form of credit card processing, and it is the preferred form for Internet acceptance.

When would you use the authorization/settle process? Here are some guidelines:

- If the services offered by you or your merchants include a review or editing of the transaction prior to a final processing you would choose Auth/Settle, or
- Another reason to use Auth/Settle is if there is a delay between the time of purchase and deliver of a product. The original auth will remain active for (3) days\*. Once the auth expires the transaction can be sent as another auth/settle or a charge.

Note\*: the length the authorization will stay active varies between processors, but all authorizations expire if a corresponding charge is not sent within the allotted time.

### Technical Details

This service is coded in .Net 3.5 and is exposed as an XML Web Service. The WSDL is:

<https://testsvcs.guardianpayments.com/Processing/CreditCardBatchService.svc?wsdl>

## Credit Card Authorization Batch

Processing the credit card authorization batch causes each transaction in the batch to be authorized. The credit card authorization places a hold on the account's available funds for the authorized amount. Authorization requires a corresponding settle transaction, and if the settle is not issued within three days the authorization expires freeing up the held funds.

Processing batches of credit card authorization transactions follows this basic outline:

1. Submit your batch to the Guardian Web Service, and
2. Monitor until the Guardian Web Service indicates the batch processing is complete.

## Submitting an Authorization Batch to the Web Service: SubmitAuthBatch(...)

Submitting a credit card authorization batch through the SOAP Web service follows this pseudo-syntax:

```
CreditCardAuthBatchResponse = CreditCardBatchWebService.SubmitAuthBatch
(
    CreditCardMessageHeader,
    CreditCardAuthBatchRequest
)
```

### Authorization Batch Header Object

The header object contains information to identify you as a merchant within Guardian's Payment Director Payment gateway. These merchant specific values are provided by Guardian when merchant underwriting is complete.

### Authorization Batch Request Object

The CreditCardAuthBatchRequest object will contain the specific information on the batch and the credit card transactions that are being authorized. The CreditCardAuthBatchRequest object is defined as:

Field Name	Type	Description
CreditCardAuthBatch	CreditCardAuthBatch	The transactions being authorized.

### Authorization Batch Object

The Guardian CreditCardAuthBatch object is defined as follows:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Required. Your system's reference number to identify this batch.
CreditCardAuths	CreditCardAuth[]	Required. An array of objects, each one referring to a single authorization request. NOTE: In Visual Studio, this can be a List if defined as such in the "Dictionary collection type" of the Service Reference Configuration.
TransactionCount	int	Required. Count of all transactions in the array
TotalAmount	decimal	Required. Sum of the amount of each transaction in the array

### Authorization Object

The Guardian CreditCardAuth object is defined as follows:

GPS Batch Credit Card SOAP Web Service

---

Field Name	Type	Description
ClientTransactionIdentifier	string(50)	Required. Your system's reference number to identify this transaction.
CardNumber	String	Required. Credit Card number. Not required if using an account already stored in the Guardian Virtual Account Vault.
VirtualCreditCardAccountIdentifier	String	Optional. When the card number is already stored in the Guardian Vault, this value replaces the CardNumber field.
ExpirationDate	Date	Required. Valid Formats include any valid date, including "MM-01-YYYY", "01-MM-YYYY", "Month 01, Year", etc.
CardholderName	String	Required. Name on card.
BillingAddress1	String	Optional.
BillingAddress2	String	Optional.
BillingPostalCode	String	Optional.
BillingCountry	String	Optional.
CardSecurityCode	string	Optional. CVV code from the back of the card <b>Note: Industry regulations strictly prohibit retention or storage of this field.</b>
Amount	Decimal	Required.
CashierId	String	Optional. A courtesy field provided to you to identify the source of this transaction.
Track	String	Optional. Track data read from the swipe of the card <b>Note: Industry regulations strictly prohibit retention or storage of this field.</b>
OriginType	Enum	Optional. Values are (Internet, Phone, Mail or POS)
LocationId	String	Optional. A courtesy field to identify the location this transaction originated from
IndustryType	Enum	Optional. Values are (Ecommerce, Retail, Moto, Lodging)
LodgingDetail	Lodging	Optional.
ClientData1	String(50)	Optional. A value from your system that will show up in Guardian Reporting. Examples include Customer Id, Donor Id, Order Description, etc.
ClientData2	String(50)	Optional. A value from your system that will show up in Guardian Reporting. Examples include Customer Id, Donor Id, Order Description, etc.
ClientData3	String(50)	Optional. A value from your system that will show up in Guardian Reporting. Examples include Customer Id, Donor Id, Order Description, etc.

**Lodging Object**

The Lodging object is defined as:

Field Name	Type	Description
FolioNumber	String	Optional. Folio number
CheckInDate	Date	Optional. Any valid date format
CheckOutDate	Date	Optional. Any valid date format

### Authorization Batch Response Object

The Guardian CreditCardAuthBatchResponse object is defined as:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Echo of your system's batch Id that was supplied in the batch object.
BatchIdentifier	String(50)	Guardian's id for this batch.

### Auth Batch Status during Processing: GetCreditCardBatchStatusSummary(...)

While the batch is processing, a call to the GetCreditCardBatchStatusSummary method can retrieve intermediate results. This is entirely optional, and if you have large batches, you may wish to invoke a timer to inquire on the batch status to run every minute. This gives you counts of the ongoing progress of this batch. When the batch is complete, the BatchStatus that is returned from this call will be "Complete". Additionally you receive an array of all current transaction status and their counts. So, if all transactions show Invalid, you will know that you have a pretty serious error in your original transaction data.

Getting the current status of a credit card authorization batch through the SOAP Web service follows this pseudo-syntax:

```
CreditCardBatchStatusSummaryResponse =
    CreditCardBatchWebService.GetCreditCardBatchStatusSummary
    (
        CreditCardMessageHeader,
        CreditCardBatchStatusSummaryRequest
    )
```

### Credit Card Batch Header Object

As was the case when submitting the batch, the header object for getting batch status contains information to identify you as a merchant within Guardian's Payment Director Payment gateway, and examples for creating and populating the header can be found in the Appendices.

### Credit Card Batch Status Summary Request Object

The CreditCardBatchStatusSummaryRequest object will contain the identifying information on the batch and is defined as:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Optional. Echo of your system's batch Id that was supplied in the batch object.
BatchIdentifier	String(50)	Optional. Guardian's id for this batch.

Either the ClientBatchIdentifier or the BatchIdentifier must be provided.

### Credit Card Batch Status Summary Response Object

The CreditCardBatchStatusSummaryResponse object is defined as follows:

Field Name	Type	Description
------------	------	-------------

Field Name	Type	Description
ClientBatchIdentifier	String	Your system's reference number to identify this batch.
BatchStatus	String	Current status of the batch: <ul style="list-style-type: none"> <li>• Received: batch has been received, waiting for validation</li> <li>• Validating: batch is being validated</li> <li>• Validated: batch has been validated, is waiting for processing</li> <li>• Processing: batch is being processed</li> <li>• Complete: batch has been processed.</li> </ul>
CreditCardStatusSummaryList	CreditCardStatusSummary[]	Array of CreditCardStatusSummary objects

### Credit Card Batch Status Summary Object

The CreditCardStatusSummary object is defined as follows:

Field Name	Type	Description
Status	String	Transaction status: <ul style="list-style-type: none"> <li>• Received: the transaction has been received, is waiting of processing</li> <li>• Invalid: Guardian marked the transaction invalid (e.g. invalid routing number)</li> <li>• Processing: transaction is processing</li> <li>• Failed: transaction has completed resulting in an error</li> <li>• Complete: transaction processing has finished normally</li> </ul>
Count	Int	Count of transactions in the given status.

### Auth Batch Status after Processing: GetAuthBatchResults(...)

Once the batch processing is complete at Guardian Payment System's Payment Director payment gateway, you can download the results of the specific batch. The amount of time required to process the batch varies based on batch size, so if you attempt to retrieve the results too soon, you will receive an exception stating the batch is currently being processed.

Getting the current status or the final results of a credit card authorization batch through the SOAP Web service follows this pseudo-syntax:

```
CreditCardAuthBatchResultResponse = CreditCardBatchWebService.GetAuthBatchResults
(
    CreditCardMessageHeader,
    CreditCardAuthBatchResultRequest
)
```

### Credit Card Batch Header Object

As was the case when submitting the batch, the header object for getting batch status contains information to identify you as a merchant within Guardian's Payment Director Payment gateway, and examples for creating and populating the header can be found in the Appendices.

### Authorization Batch Result Request Object

The CreditCardAuthBatchResultRequest object will contain the identifying information on the batch and is defined as:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Optional. Echo of your system's batch Id that was supplied in the batch object.
BatchIdentifier	String(50)	Optional. Guardian's id for this batch.

Either the ClientBatchIdentifier or the BatchIdentifier must be provided.

### Authorization Batch Result Response Object

The CreditCardAuthBatchResultResponse object is defined as follows:

Field Name	Type	Description
ClientBatchIdentifier	String	Your system's reference number to identify this batch.
BatchIdentifier	String	Guardian's Batch Identifier
CreditCardAuthResults	CreditCardAuthResult[]	Array of CreditCardAuthResult objects – 1 for each transaction submitted for authorization

### Authorization Results Object

The CreditCardAuthResult object contains the result of every transaction in the batch and is defined as:

Field Name	Type	Description
ClientTransactionIdentifier	String	Your system's reference number to identify this transaction.
TransactionIdentifier	String	Guardian's transaction Identifier
ResponseCode	String	Response code from the Credit Card processor
ResponseMessage	String	Descriptive message of the response from the processor
TransactionDate	DateTime	The UTC time stamp of the transaction from Guardian's perspective.
ProcessorDate	Date	Time stamp of the transaction from the processor's perspective.
AVSResponseCode	String	Processor's AVS response code.
AVSResponseMessage	String	Processor's descriptive AVS response message.
CVVResponseCode	String	Processor's CVV response code.
CVVResponseMessage	String	Processor's descriptive CVV response message.
AuthorizationCode	String	The Card Processor Authorization Number; your system will store this value. It is used for settlement and tracking of the transaction.
Status	String	Transaction status: <ul style="list-style-type: none"> <li>• Received: transaction has been received, waiting for processing</li> <li>• Invalid: Guardian marked the transaction invalid (e.g. invalid card number)</li> <li>• Processing: transaction is processing</li> <li>• Failed: transaction has completed resulting in an error</li> <li>• Complete: transaction processing has finished normally</li> <li>• Canceled: At your request, Guardian canceled the batch containing this transaction</li> </ul>

Field Name	Type	Description
		<ul style="list-style-type: none"><li>• Hold: At your request, Guardian has put the batch containing this transaction on hold</li></ul>

## Credit Card Settle Batch

An authorization of a credit card merely reserves, or holds, funds in the account, and it is the settle transaction that actually commits funds that have been held from the authorization. If you choose to authorize transactions and then wish to settle them at a later time, the implementation is quite different than a credit card charge transaction, in which the authorization and settlement happen in the same transaction. This is far simpler to deal with and in most cases will handle your needs. If however, you wish to settle at the end of day, or after your goods or services have been delivered, you must follow the following process.

You must create a settlement request similar to the authorization request discussed above. The difference with this is that the batch request contains a list of individual authorized transactions that you wish to settle.

There is no direct mapping, and no requirement for a mapping, between a batch of transactions to be authorized and a batch of transactions to be settled. The only requirement for a settle batch is that the transactions have already been authorized, and the settle amount is less than or equal to the authorized amount.

## Store Authorization Results

For settlement, you must store each transaction's authorization code returned by the Web Service `GetAuthBatchResults` method and the amount of the original transaction.

## Submitting an Settle Batch to the Web Service: `SubmitSettleBatch(...)`

Submitting a credit card authorization batch through the SOAP Web service follows this pseudo-syntax:

```
CreditCardSettleBatchResponse = CreditCardBatchWebService.SubmitSettleBatch
(
    CreditCardMessageHeader,
    CreditCardSettleBatchRequest
)
```

## Credit Card Batch Header Object

As was the case when submitting the authorization batch, the header object for submitting the settle batch status contains information to identify you as a merchant within Guardian's Payment Director Payment gateway, and examples for creating and populating the header can be found in the Appendices.

### Settle Batch Request Object

The CreditCardSettleBatchRequest object will contain the specific information on the batch and the credit card transactions that are being settled. The CreditCardsettleBatchRequest object is defined as:

Field Name	Type	Description
CreditCardSettleBatch	CreditCardSettleBatch	The batch being settled.

### Settle Batch Object

The Guardian CreditCardSettleBatch object is defined as follows:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Required. Your system's reference number to identify this batch.
CreditCardSettles	CreditCardSettle[]	Required. An array of objects, each one referring to a single settle request. NOTE: In Visual Studio, this can be a List if defined as such in the "Dictionary collection type" of the Service Reference Configuration.
TransactionCount	int	Required. Count of all transactions in the array
TotalAmount	decimal	Required. Sum of the amount of each transaction in the array

### Settle Object

The Guardian CreditCardSettle object is defined as follows:

Field Name	Type	Description
ClientTransactionIdentifier	string(50)	Required. Your system's reference number to identify this transaction.
AuthTransactionIdentifier	String(50)	Required. Guardian's transaction identifier returned in the authorization.
Amount	Decimal	Required. Amount to be settled; this amount must be less than or equal to the authorized amount.
AuthorizationCode	String	Required. Authorization code returned in the authorization.
ClientData1	String(50)	Optional. A value from your system that will show up in Guardian Reporting.
ClientData2	String(50)	Optional. A value from your system that will show up in Guardian Reporting.
ClientData3	String(50)	Optional. A value from your system that will show up in Guardian Reporting.

### Settle Batch Response Object

The Guardian CreditCardSettleBatchResponse object is defined as:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Echo of your system's batch Id that was supplied in the batch object.
BatchIdentifier	String(50)	Guardian's id for this settle batch.

## Settle Batch Status during Processing: GetCreditCardBatchStatusSummary(...)

While the batch is processing, a call to the GetCreditCardBatchStatusSummary method can retrieve intermediate results. This is entirely optional, and if you have large batches, you may wish to invoke a timer to inquire on the batch status to run every minute. This gives you counts of the ongoing progress of this batch. When the batch is complete, the BatchStatus that is returned from this call will be “Complete”. Additionally you receive an array of all current transaction status and their counts. So, if all transactions show Invalid, you will know that you have a pretty serious error in your original transaction data.

Getting the current status of a credit card authorization batch through the SOAP Web service follows this pseudo-syntax:

```
CreditCardBatchStatusSummaryResponse =
    CreditCardBatchWebService.GetCreditCardBatchStatusSummary
    (
        CreditCardMessageHeader,
        CreditCardBatchStatusSummaryRequest
    )
```

### Credit Card Batch Header Object

As was the case when submitting the batch, the header object for getting batch status contains information to identify you as a merchant within Guardian’s Payment Director Payment gateway, and examples for creating and populating the header can be found in the Appendices.

### Credit Card Batch Status Summary Request Object

The CreditCardBatchStatusSummaryRequest object will contain the identifying information on the batch and is defined as:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Optional. Echo of your system’s batch Id that was supplied in the batch object.
BatchIdentifier	String(50)	Optional. Guardian’s id for this batch.

Either the ClientBatchIdentifier or the BatchIdentifier must be provided.

### Credit Card Batch Status Summary Response Object

The CreditCardBatchStatusSummaryResponse object is defined as follows:

Field Name	Type	Description
ClientBatchIdentifier	String	Your system’s reference number to identify this batch.
BatchStatus	String	Current status of the batch: <ul style="list-style-type: none"> <li>• Received: batch has been received, waiting for validation</li> <li>• Validating: batch is being validated</li> <li>• Validated: batch has been validated, is waiting for processing</li> <li>• Processing: batch is being processed</li> <li>• Complete: batch has been processed.</li> </ul>

Field Name	Type	Description
CreditCardStatusSummaryList	CreditCardStatusSummary[]	Array of CreditCardStatusSummary objects

### Credit Card Batch Status Summary Object

The CreditCardStatusSummary object is defined as follows:

Field Name	Type	Description
Status	String	Transaction status: <ul style="list-style-type: none"> <li>• Received: the transaction has been received, is waiting of processing</li> <li>• Invalid: Guardian marked the transaction invalid (e.g. invalid routing number)</li> <li>• Processing: transaction is processing</li> <li>• Failed: transaction has completed resulting in an error</li> <li>• Complete: transaction processing has finished normally</li> </ul>
Count	Int	Count of transactions in the given status.

### Settle Batch Status after Processing: GetSettleBatchResults(...)

Once the settlement has completed processing at the Guardian Payment Systems host, you can download the results of the specific batch. The amount of time required to process the batch varies based on batch size, so if you attempt to retrieve the results too soon, you will receive an exception stating that the batch is currently being processed.

Getting the current status or the final results of a credit card settle batch through the SOAP Web service follows this pseudo-syntax:

```
CreditCardSettleBatchResultResponse = CreditCardBatchWebService.GetSettleBatchResults
(
    CreditCardMessageHeader,
    CreditCardSettleBatchResultRequest
)
```

### Credit Card Batch Header Object

As was the case when submitting the batch, the header object for getting batch status contains information to identify you as a merchant within Guardian’s Payment Director Payment gateway, and examples for creating and populating the header can be found in the Appendices.

### Settle Batch Result Request Object

The Guardian CreditCardSettleBatchResultRequest object is defined as:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Echo of your system’s batch Id that was supplied in the batch object.
BatchIdentifier	String(50)	Guardian’s id for this settle batch.

Either the ClientBatchIdentifier or the BatchIdentifier must be provided.

### Settle Batch Result Response Object

The CreditCardSettleBatchResultResponse object is defined as follows:

Field Name	Type	Description
ClientBatchIdentifier	String	Your system's reference number to identify this batch.
BatchIdentifier	String	Guardian's Batch Identifier
CreditCardSettleResults	CreditCardSettleResult[]	Array of CreditCardSettleResult objects – 1 for each transaction submitted for settlement.

### Settle Results Object

The CreditCardSettleResult object contains the result of every transaction in the batch and is defined as:

Field Name	Type	Description
ClientTransactionIdentifier	String	Your system's reference number to identify this transaction.
TransactionIdentifier	String	This is the Guardian Transaction Identifier for the settlement transaction. This must be preserved in case you need to issue a refund against this transaction at a later date.
ResponseCode	String	Response code from the Credit Card processor
ResponseMessage	String	Descriptive message of the response from the processor
TransactionDate	DateTime	The UTC time stamp of the transaction from Guardian's perspective.
ProcessorDate	Date	Time stamp of the transaction from the processor's perspective.
AuthorizationCode		This is the AUTH code from the settlement.
AuthClientTransactionIdentifier		The client transaction identifier that you sent in on the original authorization request. This allows you to easily tie the entire transaction together.
AuthTransactionIdentifier		The Guardian Payment Systems transaction identifier for the original authorization that you are settling.
Status	String	Transaction status: <ul style="list-style-type: none"> <li>• Received: transaction has been received, waiting for processing</li> <li>• Invalid: Guardian marked the transaction invalid (e.g. invalid card number)</li> <li>• Processing: transaction is processing</li> <li>• Failed: transaction has completed resulting in an error</li> <li>• Complete: transaction processing has finished normally</li> <li>• Canceled: At your request, Guardian canceled the batch containing this transaction</li> </ul> Hold: At your request, Guardian has put the batch containing this transaction on hold

The contents of the CreditCardSettleResults could be used in the following manner:

```
//
// Build a message to show the results of the batch - you would
// actually process your results at this time.
//
StringBuilder msg = new StringBuilder(
    string.Format("Results of Settlement{0}", Environment.NewLine));
msg.Append(string.Format("Your Settlement Batch ID :{0}{1}",
```

## GPS Batch Credit Card SOAP Web Service

---

```
        response.ClientBatchIdentifier,
        Environment.NewLine);
msg.Append(string.Format("Guardian Settlement Batch ID :{0}{1}",
    response.BatchIdentifier,
    Environment.NewLine));
foreach (CCBatch.CreditCardSettleResult result
    in response.CreditCardSettleResults)
{
    result.
    msg.Append(string.Format("====={0}",
        Environment.NewLine));
    msg.Append(string.Format("Settle Response Code: {0}: {1}{2}",
        result.ResponseCode,
        result.ResponseMessage,
        Environment.NewLine));
    msg.Append(string.Format("Your Settlement TX-ID: {0}{1}",
        result.ClientTransactionIdentifier,
        Environment.NewLine));
    msg.Append(string.Format("Your Authorization TX-ID: {0}{1}",
        result.AuthClientTransactionIdentifier,
        Environment.NewLine));
    msg.Append(string.Format("Settle Auth Code: {0}{1}",
        result.AuthorizationCode,
        Environment.NewLine));
}
MessageBox.Show(msg.ToString());
```

## Credit Card Charge Batch

A Credit Card Charge is an authorization and settlement in a single transaction and issuing charge transactions provides a process that is simpler than the authorization/settle process. The customer's account is immediately charged for the full amount of the transaction.

Submitting a batch of charge transactions follows the same process as submitting an authorization batch. The following table maps the authorization objects to the charge objects:

Credit Card Authorization Objects	Credit Card Charge Objects
CreditCardMessageHeader	CreditCardMessageHeader
CreditCardAuthBatchRequest	CreditCardChargeBatchRequest
CreditCardAuthBatch	CreditCardChargeBatch
CreditCardAuth	CreditCardCharge
CreditCardAuthBatchResponse	CreditCardChargeBatchResponse
CreditCardAuthBatchResultRequest	CreditCardChargeBatchResultRequest
CreditCardAuthBatchResultResponse	CreditCardChargeBatchResultResponse
CreditCardAuthResult	CreditCardChargeResult

Processing batches of credit card charge transactions follows this basic outline:

1. Submit your batch to the Guardian Web Service, and
2. Monitor until the Guardian Web Service indicates the batch processing is complete.

### Submitting a Charge Batch to the Web Service: SubmitChargeBatch(...)

Submitting a credit card charge batch through the SOAP Web service follows this pseudo-syntax:

```
CreditCardChargeBatchResponse = CreditCardBatchWebService.SubmitChargeBatch
(
    CreditCardMessageHeader,
    CreditCardChargeBatchRequest
)
```

#### Charge Batch Header Object

The header object contains information to identify you as a merchant within Guardian's Payment Director Payment gateway. These merchant specific values are provided by Guardian when merchant underwriting is complete.

#### Charge Batch Request Object

The CreditCardChargeBatchRequest object will contain the specific information on the batch and the credit card transactions that are being charged. The CreditCardChargeBatchRequest object is defined as:

## GPS Batch Credit Card SOAP Web Service

---

Field Name	Type	Description
CreditCardChargeBatch	CreditCardChargeBatch	The batch containing the transactions being charged.

### Charge Batch Object

The Guardian CreditCardChargeBatch object is defined as follows:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Required. Your system's reference number to identify this batch.
CreditCardCharges	CreditCardCharge[]	Required. An array of objects, each one referring to a single charge request. NOTE: In Visual Studio, this can be a List if defined as such in the "Dictionary collection type" of the Service Reference Configuration.
TransactionCount	int	Required. Count of all transactions in the array
TotalAmount	decimal	Required. Sum of the amount of each transaction in the array

### Charge Object

The Guardian CreditCardCharge object is defined as follows:

Field Name	Type	Description
ClientTransactionIdentifier	string(50)	Required. Your system's reference number to identify this transaction.
CardNumber	String	Required. Credit Card number. Not required if using an account already stored in the Guardian Virtual Account Vault.
VirtualCreditCardAccountIdentifier	String	Optional. When the card number is already stored in the Guardian Vault, this value replaces the CardNumber field.
ExpirationDate	Date	Required. Valid Formats include any valid date, including "MM-01-YYYY", "01-MM-YYYY", "Month 01, Year", etc.
CardholderName	String	Required. Name on card.
BillingAddress1	String	Optional.
BillingAddress2	String	Optional.
BillingPostalCode	String	Optional.
BillingCountry	String	Optional.
CardSecurityCode	string	Optional. CVV code from the back of the card <b>Note: Industry regulations strictly prohibit retention or storage of this field.</b>
Amount	Decimal	Required.
CashierId	String	Optional. A courtesy field provided to you to identify the source of this transaction.
Track	String	Optional. Track data read from the swipe of the card <b>Note: Industry regulations strictly prohibit retention or storage of this field.</b>
OriginType	Enum	Optional. Values are (Internet, Phone, Mail or POS)
LocationId	String	Optional. A courtesy field to identify the location this transaction originated from
IndustryType	Enum	Optional. Values are (Ecommerce, Retail, Moto, Lodging)
LodgingDetail	Lodging	Optional.
ClientData1	String(50)	Optional. A value from your system that will show up in

Field Name	Type	Description
		Guardian Reporting. Examples include Customer Id, Donor Id, Order Description, etc.
ClientData2	String(50)	Optional. A value from your system that will show up in Guardian Reporting. Examples include Customer Id, Donor Id, Order Description, etc.
ClientData3	String(50)	Optional. A value from your system that will show up in Guardian Reporting. Examples include Customer Id, Donor Id, Order Description, etc.

### Lodging Object

The Lodging object is defined as:

Field Name	Type	Description
FolioNumber	String	Optional. Folio number
CheckInDate	Date	Optional. Any valid date format
CheckOutDate	Date	Optional. Any valid date format

### Charge Batch Response Object

The Guardian CreditCardChargeBatchResponse object is defined as:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Echo of your system's batch Id that was supplied in the batch object.
BatchIdentifier	String(50)	Guardian's id for this batch.

### Charge Batch Status during Processing: GetCreditCardBatchStatusSummary(...)

While the batch is processing, a call to the GetCreditCardBatchStatusSummary method can retrieve intermediate results. This is entirely optional, and if you have large batches, you may wish to invoke a timer to inquire on the batch status to run every minute. This gives you counts of the ongoing progress of this batch. When the batch is complete, the BatchStatus that is returned from this call will be "Complete". Additionally you receive an array of all current transaction status and their counts. So, if all transactions show Invalid, you will know that you have a pretty serious error in your original transaction data.

Getting the current status of a credit card authorization batch through the SOAP Web service follows this pseudo-syntax:

```
CreditCardBatchStatusSummaryResponse =
    CreditCardBatchWebService.GetCreditCardBatchStatusSummary
    (
        CreditCardMessageHeader,
        CreditCardBatchStatusSummaryRequest
    )
```

### Credit Card Batch Header Object

As was the case when submitting the batch, the header object for getting batch status contains information to identify you as a merchant within Guardian’s Payment Director Payment gateway, and examples for creating and populating the header can be found in the Appendices.

### Credit Card Batch Status Summary Request Object

The CreditCardBatchStatusSummaryRequest object will contain the identifying information on the batch and is defined as:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Optional. Echo of your system’s batch Id that was supplied in the batch object.
BatchIdentifier	String(50)	Optional. Guardian’s id for this batch.

Either the ClientBatchIdentifier or the BatchIdentifier must be provided.

### Credit Card Batch Status Summary Response Object

The CreditCardBatchStatusSummaryResponse object is defined as follows:

Field Name	Type	Description
ClientBatchIdentifier	String	Your system’s reference number to identify this batch.
BatchStatus	String	Current status of the batch: <ul style="list-style-type: none"> <li>• Received: batch has been received, waiting for validation</li> <li>• Validating: batch is being validated</li> <li>• Validated: batch has been validated, is waiting for processing</li> <li>• Processing: batch is being processed</li> <li>• Complete: batch has been processed.</li> </ul>
CreditCardStatusSummaryList	CreditCardStatusSummary[]	Array of CreditCardStatusSummary objects – 1 for each transaction submitted for authorization

### Credit Card Batch Status Summary Object

The CreditCardStatusSummary object is defined as follows:

Field Name	Type	Description
Status	String	Transaction status: <ul style="list-style-type: none"> <li>• Received: the transaction has been received, is waiting of processing</li> <li>• Invalid: Guardian marked the transaction invalid (e.g. invalid routing number)</li> <li>• Processing: transaction is processing</li> <li>• Failed: transaction has completed resulting in an error</li> <li>• Complete: transaction processing has finished normally</li> </ul>
Count	Int	Count of transactions in the given status.

### Charge Batch Status during and after Processing: GetChargeBatchResults(...)

Once the batch processing is complete at Guardian Payment System’s Payment Director payment gateway, you can download the results of the specific batch. The amount of time required to process the batch varies based on batch size, so if you attempt to retrieve the results too soon, you will receive an exception stating that the batch is currently being processed.

Getting the current status or the final results of a credit card charge batch through the SOAP Web service follows this pseudo-syntax:

```
CreditCardChargeBatchResultResponse = CreditCardBatchWebService.GetChargeBatchResults
(
    CreditCardMessageHeader,
    CreditCardChargeBatchResultRequest
)
```

#### Credit Card Batch Header Object

As was the case when submitting the batch, the header object for getting batch status contains information to identify you as a merchant within Guardian’s Payment Director Payment gateway, and examples for creating and populating the header can be found in the Appendices.

#### Charge Batch Result Request Object

The CreditCardChargeBatchResultRequest object will contain the identifying information on the batch and is defined as:

Field Name	Type	Description
ClientBatchIdentifier	string(50)	Optional. Echo of your system’s batch Id that was supplied in the batch object.
BatchIdentifier	String(50)	Optional. Guardian’s id for this batch.

Either the ClientBatchIdentifier or the BatchIdentifier must be provided.

#### Charge Batch Result Response Object

The CreditCardChargeBatchResultResponse object is defined as follows:

Field Name	Type	Description
ClientBatchIdentifier	String	Your system’s reference number to identify this batch.
BatchIdentifier	String	Guardian’s Batch Identifier
CreditCardChargeResults	CreditCardChargeResult[]	Array of CreditCardChargeResult objects – 1 for each transaction submitted.

#### Charge Results Object

The CreditCardChargeResult object contains the result of every transaction in the batch and is defined as:

Field Name	Type	Description
ClientTransactionIdent	String	Your system’s reference number to identify this transaction.

## GPS Batch Credit Card SOAP Web Service

---

Field Name	Type	Description
Identifier		
TransactionIdentifier	String	Guardian's transaction Identifier
ResponseCode	String	Response code from the Credit Card processor
ResponseMessage	String	Descriptive message of the response from the processor
TransactionDate	DateTime	The UTC time stamp of the transaction from Guardian's perspective.
ProcessorDate	Date	Time stamp of the transaction from the processor's perspective.
AVSResponseCode	String	Processor's AVS response code.
AVSResponseMessage	String	Processor's descriptive AVS response message.
CVVResponseCode	String	Processor's CVV response code.
CVVResponseMessage	String	Processor's descriptive CVV response message.
AuthorizationCode	String	The Card Processor Authorization Number; your system will store this value. It is used for tracking the transaction.
Status	String	Transaction status: <ul style="list-style-type: none"> <li>• Received: transaction has been received, waiting for processing</li> <li>• Invalid: Guardian marked the transaction invalid (e.g. invalid card number)</li> <li>• Processing: transaction is processing</li> <li>• Failed: transaction has completed resulting in an error</li> <li>• Complete: transaction processing has finished normally</li> <li>• Canceled: At your request, Guardian canceled the batch containing this transaction</li> <li>• Hold: At your request, Guardian has put the batch containing this transaction on hold</li> </ul>

## Appendix A - Specific Test Cases

For your testing, you may wish to generate specific return codes so that you can ensure that your application handles them properly.

This can easily be accomplished for CreditCardAuthorization by manipulating the “cents” portion of the amount. The following table shows what results you can expect based on the amount field:

Amount	RC	Message
13.00	0	Approved
13.01	1	Duplicate Approved (Transaction previously Approved)
13.02	2	Partially Approved
13.03	3	Invalid Merchant
13.04	4	Declined
13.05	5	Do not honor
13.06	6	Error
13.07	7	Pick up card - special
13.12	12	Invalid Transaction
13.13	13	Invalid Amount
13.14	14	Invalid Card
13.15	15	Invalid issuer
13.19	19	Re-enter transaction
13.21	21	No Action Taken
13.28	28	File is temporarily unavailable
13.33	33	Expired Card
13.39	39	No credit account Visa ePay
13.41	41	Lost card
13.43	43	Stolen card
13.51	51	Insufficient Funds
13.52	52	No checking account
13.53	53	No savings account
13.55	55	Invalid PIN
13.57	57	Trans not permitted
13.58	58	Transaction not allowed at terminal
13.59	59	Restricted
13.61	61	Amount limit exceeded
13.62	62	Restricted card
13.63	63	Security violation
13.65	65	Activity count limit exceeded

Amount	RC	Message
13.75	75	PIN tries exceeded
13.76	76	Reversal: Unable to locate previous message
13.77	77	Previous message located for a repeat or reversal
13.78	78	Duplicate
13.79	79	Already reversed
13.80	80	Invalid date
13.81	81	PIN crypto error
13.82	82	Incorrect CVV
13.83	83	Unable to verify PIN
13.85	85	No reason to decline a request ...
13.91	91	Try Again- Communications Error
13.92	92	Communications Failure
13.93	93	Law violation
13.94	94	Duplicate transaction
13.96	96	System error
13.99	199	An internal exception has occurred.

Additionally, if you wish to test how you handle CVV and AVS responses, for any amount other than 13.00 – 13.99 if you pass in the first digit of the cents as an odd digit the test server will return a failure for the CVV code. If you pass in the second digit of the cents as an odd digit, the AVS will send back a failure code. See the table below.

Amount	CVV	AVS
1.00	Success	Success
1.01	Success	Fail
1.10	Fail	Success
1.11	Fail	Fail

### Test Data that will pass

The following test data will pass the various tests:

CardNumber            4000300020001000  
 CardHolderName      Mr John Smith  
 BillingAddress1       3361 Boyington Dr Suite 180  
 BillingPostalCode     75006  
 CardSecurityCode    123  
 ExpirationDate       Any valid future date

### Specific Return Values

The following values may be returned for AVS and CVV checks.

The AVS Response Codes vary based on the Credit Card Provider. The following table shows what is returned:

Description	Visa	Master Card	AMEX	Discover	Diners Club
Address matches, Postal Code does not	A	A	A	Y	A
Neither Address nor Postal Code matches	N	N	N	N	N
Address Could not be Verified	U	U	U	W	U
Address and Postal Code match	X,Y	X,Y	Y	X,A	X,Y
Postal Code Matches, Address does not	W,Z	W,Z	Z	T,Z	W,Z
System Problem – could not verify	R	R	R	U	R

The CVV Response Codes that you may get back include the following:

CVV Code	Description
M	CVV Code matches the account number
N	CVV Code does not match the account number
P	CVV Code could not be verified
U	Card issuer does not support CVV Code Checking

## Appendix B - Re-Fetching a Specific Response

In case you need to re-fetch the response for a transaction, you can invoke either of the following methods. This may be needed if you received a 193 error and wish to try again, or you genuinely need to re-fetch the results. To do this, you must create a `CreditCardResultRequest` and supply either the Guardian transaction identifier, or your Client TransactionIdentifier and then call the appropriate method to retrieve the requested result.

Example:

```
// Create a single request
CreditCardResultRequest request = new CreditCardResultRequest();
// Populate it with either your transaction ID, or the Guardian ID
request.ClientTransactionIdentifier = "your tx id";
//                               or
request.TransactionIdentifier = "Guardian tx id";
// Call the web service to obtain the result
CreditCardAuthResponse response = webSvc.GetAuthResult(request);
CreditCardAuthResult result = response.CreditCardAuthResult;
// Look at the individual fields in the result
string authCode = result.AuthorizationCode;
string responseCode = result.ResponseCode;
string responseMsg = result.ResponseMessage;
```

*GetAuthResult* , *GetVerifyResult*, *GetChargeResult*, or *GetRefundResult* allow you to re-fetch the results. You must supply the *TransactionIdentifier* or the *ClientTransactionIdentifier* from the original transaction.

If the original transaction timed out with a 193 response code, Guardian re-submits the request to the provider and returns the new results. If the original transaction was successful, the previously fetched results are returned to you.

You can retry this transaction at any time, but if the failure is due to a connection down to the processor, you may wish to wait a period of time (30-45 seconds).

## Appendix C – Complete Authorization and Settlement Examples

This appendix pulls it all together for an Authorization and Settlement scenario. Keep in mind that doing CreditCardCharge calls accomplishes the authorization and settlement in a single transaction, but if your business needs call for two distinct actions, this example shows how that is done.

### Basic assumptions

- This assumes that you are using .Net C# or VB.NET (but this can be done with any environment that interacts with standard XML Web Services).
- This assumes that you have some system in place for gathering the details of the payments you wish to authorize and settle.
- This assumes that you have a SECURE and PERSISTANT way to store the results of your authorizations so that you can retrieve that information later for your settlement.

### *Step 1 – Gather payment information*

Using your systems, gather the payment information. This may be from your web site, your accounting system, or any other means that you have of gathering payment information.

### *Step 2 – Authorize the payment(s)*

Make web service calls to the Guardian real time web service to obtain an authorization code for each payment. This may be done in real time from your web site as you gather the payment details.

### *Step 3 – Create a Settlement batch*

At some point in time, when you are ready to settle the authorized transactions, you must create a settlement batch. This batch relies on some key information from the authorization results, so be sure that you have saved these results. Authorizations expire in seven days, so you should attempt to settle the transaction in a timely manner, or you will have to authorize it again.

### *Step 4 – Submit the Settlement batch*

Once you have created your settlement batch, you must submit it to Guardian Payment Systems for settlement of the funds. At this time, Guardian sends the settlement requests on to the various card issuer banks. As soon as this batch is sent, you will receive back a unique token that identifies this submitted batch. You must retain this token so that you can subsequently fetch the results once they have completed processing.

### *Step 5 – Fetch the batch status summary*

Once the settlement batch has been submitted, you can retrieve a summary of the current status of the processing results. You must have the unique token from Step 4 that was returned when the settlement batch was sent in.

### *Step 6 – Fetch the results of the Settlement batch*

Once the settlement batch is completed, you can retrieve the results to ensure that each transaction successfully completed the settlement. You must have the unique token from Step 4 that was returned when the settlement batch was sent in

### **Examples**

The Guardian Payment Systems website ([www.guardianpayments.com](http://www.guardianpayments.com)) has a link that allows you to download complete code examples in both C# and VB.NET for three scenarios:

1. Real Time Authorization and later Batch Settlement (described above)
2. Batch Authorization and later Batch Settlement (similar to above, but with a batch authorization rather than real time)
3. Real Time Charge (a single server call to authorize and settle a transaction with a single round-trip)

These heavily commented complete examples will assist you in incorporating the Guardian Payment Systems Credit Card Processing into your applications.

### **Generating Web References**

The Web Reference that you need to point to is:

<https://testsvcs.guardianpayments.com/Processing/CreditCardService.svc>

for the real time service and

<https://testsvcs.guardianpayments.com/Processing/CreditCardBatchService.svc>

for the batch service. Note that these are two different services.

### **App.config file**

.Net conveniently stores configuration information in an app.config file. This file is used in development and it is renamed to myprogram.exe.config when you compile. This is the file that these samples use to store basic configuration information. In these examples the app.config file contains specific credentials that are needed to test or use the RTLPS Credit Card Services.

The appSettings section contains the following:

```
<appSettings>
  <add key="EndpointToken" value="merchant-token-assigned-by-Guardian"/>
  <add key="ActivationCode" value="activation-code-assigned-by-Guardian"/>
  <add key="Guardian_BaseURL"
    value="https://testsvcs.guardianpayments.com/Processing/" />
</appSettings>
```

The EndpointToken and ActivationCode are assigned by the Guardian Payment Systems support personnel and are unique to your site. The Guardian\_BaseURL should start out as listed above, and once you are certified for production, it can be changed to the production URL.

The applicationSettings section contains some auto-generated values that are obtained from Visual Studio when the Web Service proxy was generated. They are NOT used by the application, the Guardian\_BaseURL is what is actually used to “find” the web service. You do not have to follow this method, but it does make it easier to switch environments from TEST to PRODUCTION.