



GPS Real-Time Credit Card SOAP Web Service

Version 2.0

Guardian Payment Systems

Contents

Guardian Payment Systems Credit Card Processing.....	1
Process Details – Authorization/Settle versus Charge.....	1
More Process Details - Real-Time Authorization versus Batch Authorization.....	1
Technical Details	2
Real-time Credit Card Authorization.....	3
Submit an Authorization Transaction to the Web Service: Authorize(...).....	3
Authorization Batch Header Object.....	3
Authorization Request Object.....	3
Authorization Object.....	3
Lodging Object	4
Authorization Response Object	4
Authorization Results Object	4
Re-fetching Authorization Results: GetAuthResult(...)	5
Authorization Batch Header Object.....	6
Result Request Object.....	6
Authorization Response Object	6
Authorization Results Object	6
Real-Time Credit Card Charge.....	6
Authorization Batch Header Object.....	6
Charge Request Object	6
Charge Object.....	7
Lodging Object	7
Charge Response Object	8

Charge Results Object 8

Re-fetching Charge Results: GetChargeResult(...) 8

Charge Batch Header Object 9

Result Request Object 9

Charge Response Object 9

Charge Results Object 9

Real-time Credit Card Refund 9

Submit a Refund Transaction to the Web Service: Refund(...) 9

Refund Batch Header Object 10

Refund Request Object 10

Refund Object 10

Refund Response Object 10

Refund Results Object 10

Re-fetching Refund Results: GetRefundResult(...) 11

Refund Batch Header Object 11

Result Request Object 12

Refund Response Object 12

Refund Results Object 12

Appendix A - Specific Test Cases 13

Test Data that will pass 14

Specific Return Values 15

Appendix B – Complete Authorization and Settlement Examples 16

Basic assumptions 16

Examples 17

Generating Web References 17

App.config file 17

Appendix C – Example Code 18

 Create the Web Service Objects 18

CreditCardService webSvc = new CreditCardService(); 18

webSvc.Url = "https://theservername/Processing/CreditCardService.svc";
..... 18

// 18

// If you are using ASYNC processing, hook the following events..... 18

// 18

webSvc.AuthorizeCompleted += new
AuthorizeCompletedEventHandler(webSvc_AuthorizeCompleted); 18

webSvc.RefundCompleted += new
RefundCompletedEventHandler(webSvc_RefundCompleted); 18

// 18

// Provide the token and activation code you received from Guardian
client services..... 18

// 18

CreditCardMessageHeader soapHdr = new CreditCardMessageHeader(); 18

soapHdr.EndpointToken = "your specific credential token";
soapHdr.ActivationCode = "your specific activation code";
webSvc.Header = soapHdr; 18

 GetCreditCardBatchStatus 18

Guardian Payment Systems Credit Card Processing

The Guardian Payment Systems Real-Time Credit Card Processing system provides a SOAP Web Service for customers to be able to process individual Credit Card Transactions in “real-time” – receiving instant response/status for each transaction submitted.

This service provides the following methods:

- Credit Card Authorization
- Credit Card Charge
- Credit Card Refund

This document outlines the process that a customer needs to do to use these services. Additionally, a batch Settlement service is available. This document also outlines the process for using this service.

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.

More Process Details - Real-Time Authorization versus Batch Authorization

Guardian Payment Systems provides interfaces for both one-time Authorizations and batch Authorizations. Your choice to implement one or the other methods of authorizing Credit Card transactions will be based on how your system acquires the transactions.

However, batch authorization is the preferred method, and one advantage of batch-oriented processing is that Guardian provides reporting at the batch level. Please refer to the Guardian Payment Systems Credit Card Batch spec for more details.

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/CreditCardService.svc?wsdl>

Real-time Credit Card Authorization

A Credit Card Authorization places a hold on funds in the customer’s account. The funds in the customer’s account are not removed based on an authorization. An authorization requires a corresponding settlement transactions before funds are moved. Without the settlement transaction, the authorization expires and the funds held in the customer’s account are released.

If you choose to utilize the Guardian Real-Time Credit Card Authorization interface, you will also need to get the Guardian Batch Credit Card Interface spec, because credit card settlement is only available through the batch interface.

Remember that *you must store the authorization result* in order to settle the transaction at a later time.

Submit an Authorization Transaction to the Web Service: Authorize(...)

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

```
CreditCardAuthResponse = CreditCardBatchWebService.Authorize
(
    CreditCardMessageHeader,
    CreditCardAuthRequest
)
```

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 Request Object

The CreditCardAuthRequest object will contain the specific information on the credit card transaction. The CreditCardAuthRequest object is defined as:

Field Name	Type	Description
CreditCardAuth	CreditCardAuth	The transaction being authorized.

Authorization Object

The Guardian CreditCardAuth 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.

Field Name	Type	Description
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 Note: Industry regulations strictly prohibit retention or storage of this field.
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 Note: Industry regulations strictly prohibit retention or storage of this field.
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 Response Object

The Guardian CreditCardAuthResponse object is defined as:

Field Name	Type	Description
CreditCardAuthResult	CreditCard AuthResult	The results of the authorization

Authorization Results Object

The CreditCardAuthResult object contains the result the transaction 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"> • Received: transaction has been received, waiting for processing • Invalid: Guardian marked the transaction invalid (e.g. invalid card number) • Processing: transaction is processing • Failed: transaction has completed resulting in an error • Complete: transaction processing has finished normally • Canceled: At your request, Guardian canceled the batch containing this transaction • Hold: At your request, Guardian has put the batch containing this transaction on hold

Re-fetching Authorization Results: GetAuthResult(...)

Authorization results can be pulled anytime after the Authorize call completes. GetAuthResult returns the same authorization information as the original Authorize call; the GetAuthResult follows this basic syntax:

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

```
CreditCardAuthResponse = CreditCardBatchWebService.GetAuthResult
(
    CreditCardMessageHeader,
    CreditCardResultRequest
)
```

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).

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.

Result Request Object

The CreditCardResultRequest object identifies the transaction you are retrieving results for:

Field Name	Type	Description
ClientTransactionIdentifier	String(50)	Your system’s reference number to identify this transaction.
TransactionIdentifier	String(50)	Guardian’s reference number to identify this transaction.

Either the ClientTransactionIdentifier or the TransactionIdentifier must be provided.

Authorization Response Object

The Guardian CreditCardAuthResponse object is defined above.

Authorization Results Object

The CreditCardAuthResult object is defined above.

Real-Time Credit Card Charge

A Credit Card Charge is an authorization and settlement in a single transaction. The customer’s account is immediately charged for the full amount of the transaction.

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

```
CreditCardChargeResponse = CreditCardBatchWebService.Charge
(
    CreditCardMessageHeader,
    CreditCardChargeRequest
)
```

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.

Charge Request Object

The CreditCardChargeRequest object will contain the specific information on the credit card transaction and is defined as:

Field Name	Type	Description
CreditCardCharge	CreditCardCharge	The transaction.

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 Note: Industry regulations strictly prohibit retention or storage of this field.
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 Note: Industry regulations strictly prohibit retention or storage of this field.
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

Charge Response Object

The Guardian CreditCardChargeResponse object is defined as:

Field Name	Type	Description
CreditCardChargeResult	CreditCardChargeResult	The results of the authorization

Charge Results Object

The CreditCardChargeResult object contains the result of the transaction 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"> • Received: transaction has been received, waiting for processing • Invalid: Guardian marked the transaction invalid (e.g. invalid card number) • Processing: transaction is processing • Failed: transaction has completed resulting in an error • Complete: transaction processing has finished normally • Canceled: At your request, Guardian canceled the batch containing this transaction • Hold: At your request, Guardian has put the batch containing this transaction on hold

Re-fetching Charge Results: GetChargeResult(...)

Charge results can be pulled anytime after the Charge call completes. GetChargeResult returns the same information as the original Charge call; the GetChargeResult follows this basic syntax:

Submitting the request for credit card charge results through the SOAP Web service follows this pseudo-syntax:

```
CreditCardChargeResponse = CreditCardBatchWebService.GetChargeResult
(
```

```
CreditCardMessageHeader,  
CreditCardResultRequest
```

)
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).

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.

Result Request Object

The CreditCardResultRequest object identifies the transaction you are retrieving results for:

Field Name	Type	Description
ClientTransactionIdentifier	String(50)	Your system’s reference number to identify this transaction.
TransactionIdentifier	String(50)	Guardian’s reference number to identify this transaction.

Either the ClientTransactionIdentifier or the TransactionIdentifier must be provided.

Charge Response Object

The Guardian CreditCardChargeResponse object is defined above.

Charge Results Object

The CreditCardChargeResult object is defined above.

Real-time Credit Card Refund

A Credit Card Refund is a full or partial refund against a single charge or settlement transaction. The customer’s account is immediately refunded for the desired amount of the transaction. Note, you are prevented from refunding more than the “current transaction balance”. For example, if the original charge was for \$100.00, you can send a single refund for up to \$100.00, or you can send several refunds as long as the running total does not exceed the \$100.00 original amount, so you could send in a refund for \$35.00 and another for \$60.00 and then finally one for \$5.00.

Submit a Refund Transaction to the Web Service: Refund(...)

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

```
CreditCardRefundResponse = CreditCardBatchWebService.Refund  
(  
    CreditCardMessageHeader,
```

CreditCardRefundRequest

)

Refund 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.

Refund Request Object

The CreditCardRefundRequest object will contain the specific information on the credit card transaction and is defined as:

Field Name	Type	Description
CreditCardRefund	CreditCardRefund	The transaction.

Refund Object

The Guardian CreditCardRefund object is defined as follows:

Field Name	Type	Description
ClientTransactionIdentifier	string(50)	Required. Your system’s reference number to identify this transaction.
OriginalTransactionIdentifier	String(50)	Required. The Guardian transaction identifier you are requesting the refund for.
RefundAmount	Decimal	The amount of this refund. Must be less than the original transaction amount. Remember that multiple refunds can be issued against the original transaction, and the sum of all refunds must remain less than the original transaction amount.
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.

Refund Response Object

The Guardian CreditCardRefundResponse object is defined as:

Field Name	Type	Description
CreditCardRefundResult	CreditCardRefund Result	The results of the transaction.

Refund Results Object

The CreditCardRefundResult object contains the result the transaction 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.
AuthorizationCode	String	The Card Processor Authorization Number; your system will store this value. It is used for settlement and tracking of the transaction.
OriginalTransactionIdentifier	String	Echo of the Guardian transaction identifier for the original transaction being refunded.
Status	String	Transaction status: <ul style="list-style-type: none"> • Received: transaction has been received, waiting for processing • Invalid: Guardian marked the transaction invalid (e.g. invalid card number) • Processing: transaction is processing • Failed: transaction has completed resulting in an error • Complete: transaction processing has finished normally • Canceled: At your request, Guardian canceled the batch containing this transaction • Hold: At your request, Guardian has put the batch containing this transaction on hold

Re-fetching Refund Results: GetRefundResult(...)

Refund results can be pulled any time after the Refund call completes. GetRefundResult returns the same information as the original Refund call; the GetRefundResult follows this basic syntax:

Submitting a credit card refund result request through the SOAP Web service follows this pseudo-syntax:

```
CreditCardRefundResponse = CreditCardBatchWebService.GetRefundResult
(
    CreditCardMessageHeader,
    CreditCardResultRequest
)
```

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).

Refund 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.

Result Request Object

The CreditCardResultRequest object identifies the transaction you are retrieving results for:

Field Name	Type	Description
ClientTransactionIdentifier	String(50)	Your system's reference number to identify this transaction.
TransactionIdentifier	String(50)	Guardian's reference number to identify this transaction.

Either the ClientTransactionIdentifier or the TransactionIdentifier must be provided.

Refund Response Object

The Guardian CreditCardRefundResponse object is defined above.

Refund Results Object

The CreditCardRefundResult object is defined above.

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 on Web Service Authorization and Charge methods 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 – 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) 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.

Appendix C – Example Code

Create the Web Service Objects

A code snippet follows:

```
CreditCardService webSvc = new CreditCardService();
webSvc.Url = "https://theservername/Processing/CreditCardService.svc";
//
// If you are using ASYNC processing, hook the following events
//
webSvc.AuthorizeCompleted += new
    AuthorizeCompletedEventHandler(webSvc_AuthorizeCompleted);
webSvc.ChargeCompleted += new
    ChargeCompletedEventHandler(webSvc_ChargeCompleted);
webSvc.RefundCompleted += new
    RefundCompletedEventHandler(webSvc_RefundCompleted);
//
// Provide the token and activation code you received from Guardian client services
//
CreditCardMessageHeader soapHdr = new CreditCardMessageHeader();
soapHdr.EndpointToken = "your specific credential token";
soapHdr.ActivationCode = "your specific activation code";
webSvc.Header = soapHdr;
```

GetCreditCardBatchStatus

While the batch is processing, a call to the GetCreditCardBatchStatus 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.

```
CreditCardBatchStatusSummaryRequest request = new CreditCardBatchStatusSummaryRequest();
//
// If we request status on a batch not yet fully received, skip it.
// The BatchIdentifier is returned from the SubmitSettleBatch call
//
if (BatchIdentifier.Equals(Guid.Empty))
    return;
request.BatchIdentifier = BatchIdentifier.ToString();
```

```
// -- or - (you can send either your batch identifier, or the Guardian identifier
request.ClientBatchIdentifier = ClientBatchIdentifier;
//
// Call the Guardian Server. BatchStatus.ToLower() == "complete" when this is finished
//
webSvc.GetCreditCardBatchStatus(request);
StringBuilder msg = new StringBuilder();
msg.Append(string.Format("Your Batch Id: {0} / Status {1}{2}",
    response.ClientBatchIdentifier,
    response.BatchStatus,
    Environment.NewLine));
foreach (CCBatch.CreditCardStatusSummary status in response.CreditCardStatusSummaryList)
{
    msg.Append(string.Format("Status / Count: {0} / {1}{2}",
        status.Status,
        status.Count,
        Environment.NewLine));
}
MessageBox.Show(msg.ToString(),
    "Status Summary",
    MessageBoxButtons.OK,
    MessageBoxIcon.Information);
```