



Transparent Database Engine Template (TDBE)  
Integration Guide Version 1.2  
October 24, 2006



## Table of Contents

<b>1</b>	<b>Introduction to the Transparent Database Engine Template .....</b>	<b>1</b>
<b>2</b>	<b>Use of the Transparent Database Engine Template .....</b>	<b>2</b>
2.1	Basic Example .....	2
2.2	Basic Example without HTML Output.....	3
<b>3</b>	<b>Optional Fields .....</b>	<b>4</b>
3.1	Invoice and Transaction ID Reporting Example .....	4
<b>4</b>	<b>Transaction Types.....</b>	<b>6</b>
4.1	Authorization .....	8
4.2	Auth2Sale.....	8
4.3	Deleting an Authorization.....	9
4.4	Voiding a Transaction.....	9
4.5	Processing A Return/Credit .....	10
4.6	Processing an Internet Check.....	10
4.7	Processing an NCN/Retail Check.....	11
4.8	Card Present/Swiped Transactions .....	13
4.9	Purchase Card Level II Support.....	13
<b>5</b>	<b>Confirmation Script.....</b>	<b>14</b>
<b>6</b>	<b>Security .....</b>	<b>15</b>
<b>Appendix A - HTML Examples.....</b>		<b>1</b>
	HTML Example 2.1 - Basic Example.....	2
	HTML Example 2.2 – Basic Example without HTML Output .....	3
	HTML Example 3.1 – Invoice and Transaction ID Reporting .....	4
	HTML Example 4.1 .....	5
	HTML Example 4.2.....	6
	HTML Example 4.5 .....	7
	HTML Example 4.6.....	8
	HTML Example 5.1 – Confirmation Script Request.....	10



# 1 Introduction to the Transparent Database Engine Template

The **eProcessingNetwork Transparent Database Engine Template (TDBE)** is an Application Programming Interface (API) that provides advanced features for integration with the **eProcessingNetwork Transaction Processing Gateway**. These features require advanced computer skills, including programming skills to implement.

The TDBE is intended for developers that collect and maintain a customer's order information. The TDBE can be utilized from a program or CGI script submitting a CGI form and waiting on the response, allowing merchants to interact with the gateway without customers ever seeing this interaction.

To process transactions with the TDBE, programmers must be able to establish a secure HTTPS session. This usually requires programmers to have installed security libraries on their own server(s) specific to the programming language utilized by each programmer and, while this is beyond the scope of the Technical Support offered by eProcessingNetwork, most languages used offer good documentation on the installation and use of such libraries.

This document contains examples of the interaction that programs or CGI scripts would have with the TDBE, demonstrating how various parameters are used and how these parameters affect the responses.

HTML code for all documented examples is available in the Appendix. Each example utilizes an HTML form to illustrate the interaction that your program or CGI script will have with the gateway. Do not use these HTML forms in your integration with the TDBE.

## 2 Use of the Transparent Database Engine Template

The TDBE is implemented as a POST method CGI function. Do not use the GET method, which exposes sensitive information. Transaction requests to the TDBE should be posted to:

```
https://www.eProcessingNetwork.Com/cgi-bin/tdbe/transact.pl
```

Although the TDBE has many advanced features, the minimum information needed for a TDBE transaction is:

- **ePNAccount Number** - the **eProcessingNetwork** account number that was assigned to the merchant. Do not pass in the Business Name or the Merchant ID assigned by the Acquiring Bank or Merchant Service Provider.
- **CardNo** - the credit card number.
- **ExpMonth** – the credit card expiration month, 01 through 12.
- **ExpYear** - the credit card expiration year. This must be passed in as a 2 digit year, i.e. 05 for 2005.
- **Total** - the amount of the transaction, for example 12.34. Do not include currency symbols or commas.
- **Address** - the street address on file with the cardholder's issuing bank (for AVS).
- **Zip** - the zip code on file with the cardholder's issuing bank (for AVS).
- **Email** - if this is a valid email address, **eProcessingNetwork** will email a transaction summary to the cardholder.
- **CVV2Type** - choices are:
  - 0 – CVV2 should not be used for this transaction
  - 1 – CVV2 should be used for this transaction
  - 2 – the card's CVV2 is illegible
  - 9 – This credit card has no CVV2 imprinted on it
- **CVV2** – the CVV2 value on the card. If you pass in 0 for **CVV2Type**, then pass this as empty.

Responses to requests posted to the TDBE are quoted, comma-delimited strings. The first string always starts with a **Y** (Yes, approved), an **N** (No, declined) or a **U** (No, unable), and is followed by up to 16 characters describing the transaction response. The second string is the Issuing Banks' Address Verification Systems' response to the street address and Zip code supplied. The third string is the Issuing Banks response to the CVV2 information supplied, and is only returned if CVV2 is used for the transaction.

### 2.1 Basic Example

HTML Example 2.1 (see appendix), shown in Figure 2.1, utilizes the minimum requirements as described above. Note the ePNAccount number is 05971. This is an eProcessingNetwork demo or test account. Setting the credit card number field (CardNo) to be the same as the ePNAccount signifies to the gateway that you are running a test transaction.

This example passes in Address and CVV2. Test transactions do not process Address and CVV2. Negative AVS and CVV2 responses will always be given for test transactions. Real transactions do receive appropriate AVS and CVV2 responses.

Shown below is a sample response received when submitting the form.

Note that the response is wrapped with HTML, often useful for debugging. To make it easier to parse this response, pass in another field named "HTML" with the value set to "No", as in HTML Example 2.2.

ePNAccount:	<input type="text" value="05971"/>
CardNo:	<input type="text" value="05971"/>
ExpMonth:	<input type="text" value="12"/>
Exp Year:	<input type="text" value="09"/>
Total:	<input type="text" value="12.34"/>
Address:	<input type="text" value="123 Fake St."/>
Zip:	<input type="text" value="12345"/>
EMail (enter your email address):	<input type="text"/>
CVV2Type:	<input type="text" value="1"/>
CVV2:	<input type="text" value="123"/>
Submit:	<input type="button" value="Submit"/>

Figure 2.1 – Basic Example

```
<html>
<body>
"YAUTH/TKT 031053","Address and Zip Code Do Not Match (N) "
</body>
</html>
```

Figure 2.2 – Response from Basic Example

## 2.2 Basic Example without HTML Output

HTML Example 2.2, shown In Figure 2.3, is the same as HTML example 2.1 with the additional "HTML" field set to "No". The response to Example 2.2, shown below in Figure 2.4, is essentially the same as the response to Example 2.1, except that it is not wrapped in HTML.

ePNAccount:	<input type="text" value="05971"/>
CardNo:	<input type="text" value="05971"/>
ExpMonth:	<input type="text" value="12"/>
Exp Year:	<input type="text" value="09"/>
Total:	<input type="text" value="12.34"/>
Address:	<input type="text" value="123 Fake St."/>
Zip:	<input type="text" value="12345"/>
EMail (enter your email address):	<input type="text"/>
CVV2Type:	<input type="text" value="1"/>
CVV2:	<input type="text" value="123"/>
HTML:	<input type="text" value="No"/>
Submit:	<input type="button" value="Submit"/>

Figure 2.3 – Basic Example Without HTML Output

The first value is the transaction response, the second value is the Issuing Banks' Address Verification Systems' response and the third value is the Issuing Banks's response to the CVV2. Since this is a test transaction, no CVV2 response is returned. CVV2 responses are returned for real transactions.

The transaction response contains two parts, the first character indicates if the transaction was approved and the remaining characters describe the approval or decline.

```
"YAUTH/TKT 031873","Address and Zip Code Do Not Match (N)"
```

Figure 2.4 – Response from Basic Example without HTML Output

### 3 Optional Fields

The following named fields can optionally be passed in, and will be saved by the gateway with each transaction.

FIELD NAME	FIELD DESCRIPTION
Company	Name of Company
FirstName	Person's First Name
LastName	Person's Last Name
Address	Used also for AVS
City	City
State	2 Character State Code
Zip	Used also for AVS
Phone	Phone number
EMail	EMail (note if you pass this in we also send an email to the client)
Inv	Allows setting the Invoice Number (up to 8 digits) as well as reporting (see below)

Table 3.1 – Optional Field Names and Descriptions

By default, the gateway starts each merchants invoice number at 1 and automatically increments that number with each approved transaction.

By passing in a number of up to 8 digits (will be truncated to 8 if longer), the transaction will be submitted to and recorded by the processor with that invoice number. We recommend that invoice numbers passed in start in a high range so that there is no conflict with any transactions entered via the Online Terminal.

Optionally, the field named Inv can be set to the lower-case word “report”, in which case the transaction will be processed using the automatically incremented invoice number, and the response will contain the invoice number used, as well as a Transaction ID.

#### 3.1 Invoice and Transaction ID Reporting Example

HTML Example 3.1 utilizes the named field “Inv” set to “report”, in which case the response will be formatted as shown in Figure 3-2.

ePNAccount:	<input type="text" value="05971"/>
CardNo:	<input type="text" value="05971"/>
ExpMonth:	<input type="text" value="12"/>
ExpYear:	<input type="text" value="09"/>
Total:	<input type="text" value="12.34"/>
Address:	<input type="text" value="123 Fake St"/>
Zip:	<input type="text" value="12345"/>
EMail (enter your email address):	<input type="text"/>
CVV2Type:	<input type="text" value="1"/>
CVV2:	<input type="text" value="123"/>
HTML:	<input type="text" value="No"/>
Inv:	<input type="text" value="report"/>
Submit:	<input type="button" value="Submit"/>

Figure 3.1 – Invoice and Transaction ID Reporting Example  
© 2006 eProcessing Network, LLC

"YAUTH/TKT 031833","Address and Zip Code Do Not Match (N)","","31789","20040831104036-05971-31789"

### Figure 3.2 – Response with Invoice Number and Transaction ID

The first field is the transaction response, followed by the AVS response, followed by the CVV2 response. Next, the invoice number used by the transaction processor and your bank is returned, followed by the Transaction ID assigned by the gateway.

## 4 Transaction Types

By default, the gateway processes TDBE transaction requests as specified in the “Process Transactions As” section of “Processing Controls” in the eProcessingNetwork Merchant Support Center. Options include “Sale”, “Authorization Only” or “Auth Convert”.

- **Sale** – processes the transaction as a Sale transaction. If approved, the transaction is put into the merchants batch and will be settled at the end of the day, at which point the acquiring bank will fund the transaction based on the terms of the merchant agreement. Most (but not all) issuing banks will approve a Sale transaction if there is available credit on a card regardless of whether the Address/Zip passed in match the information they have on file, and regardless of whether the CVV2 passed in is correct or not. The AVS and CVV2 response fields will indicate whether the address and CVV2 information passed in matches what the issuing bank has on file.
- **Authorization Only** – processes the transaction as an Authorization Only transaction. If approved, this indicates that the account has the specified amount available at that time. The gateway saves the transaction in Authorization Only history, but the transaction will not be settled until it is converted to an Offline Sale. Most (but not all) issuing banks will approve an Authorization Only transaction if there is available credit on a card regardless of whether the Address/Zip passed in match the information they have on file, and regardless of whether the CVV2 passed in is correct or not. The AVS and CVV2 response fields will indicate whether the address and CVV2 information passed in matches what the issuing bank has on file.
- **AuthConvert** - processes the transaction as an Authorization Only transaction. If the issuing bank approves the transaction, and the AVS response from the issuing bank is one that the merchant has indicated is acceptable, and the CVV2 response from the issuing bank is one that the merchant has indicated is acceptable, then the transaction is automatically put into the merchant’s batch. The transactions in the merchants batch will be settled at the end of the day, at which point the acquiring bank will fund the transaction based on the terms of the merchant agreement. If either the AVS response or the CVV2 response from the issuing bank are not acceptable to the merchant, the Authorization Only is simply discarded.

The TDBE allows the setting in the “Process Transactions As” section of “Processing Controls” to be overridden on a per-transaction basis by passing in the named field “TranType” set to one of the values shown in Table 4.1.

AuthOnly	Processes the transaction as an authorization only. That is the transaction is not captured (batched). You will later need to perform an “Auth2Sale” transaction to have that transaction captured (batched). You will need to record the “TransID” to later capture the transaction.
Auth2Sale	Capture (batch) an Authorization Only transaction. You will need the “TransID” from the initial authorization to capture.
AuthDel	If you decide not to capture an Authorizaton Only transaction, you have the option to “delete” it from the database. You will need the “TransID” to delete it.
Sale	This is usually the default transaction type, the transaction will be authorized and batched (captured).
AuthConvert	This is a transaction that performs an Authorization Only and then checks the AVS response and the CVV2 response. If the AVS and CVV2 response match one of the values you set in the “Processing Controls” in the Merchant Support Center, then this transaction is approved by us, otherwise we decline the transaction (although an outstanding authorization exists).
Void	This is a special transaction. If you have processed a transaction and decide you no longer want this transaction, you can “void” it. This must be done before the evening “batch close” or “capture” process happens. You will need the “TransID” to perform this transaction.
Return	This processes a “Return” or “Credit” against the cardholder’s card, essentially refunding them money.

Table 4.1 – Transaction Types

## 4.1 Authorization

ePNAccount:	<input type="text" value="04971"/>
CardNo:	<input type="text" value="04971"/>
ExpMonth:	<input type="text" value="12"/>
Exp Year:	<input type="text" value="09"/>
Total:	<input type="text" value="12.34"/>
Address:	<input type="text" value="123 Fake St."/>
Zip:	<input type="text" value="12345"/>
EMail (enter your email address):	<input type="text"/>
CVV2Type:	<input type="text" value="1"/>
CVV2:	<input type="text" value="123"/>
HTML:	<input type="text" value="No"/>
Inv:	<input type="text" value="report"/>
RestrictKey:	<input type="text" value="2D9kQtWZnSLZgvZ"/>
TranType:	<input type="text" value="AuthOnly"/>
Submit:	<input type="button" value="Submit"/>

Figure 4.1 – Authorization Only

HTML Example 4.1 is an example of an Authorization only transaction. The response is:

```
"YAPPROVED 031030","Address and Zip Code Do
Not Match (N)", "", "6044", "20040831143740-
04971-6044"
```

The TransID returned in this case is 20040831143740-04971-6044. The TransID can then be used to execute an Auth2Sale transaction, which will batch this transaction for funding.

## 4.2 Auth2Sale

To convert the previous transaction to a sale: using HTML Example 4.2 (see the appendix).

ePNAccount:	<input type="text" value="04971"/>
TransID:	<input type="text" value="20040831143740-04971"/>
HTML:	<input type="text" value="No"/>
Inv:	<input type="text" value="report"/>
RestrictKey:	<input type="text" value="2D9kQtWZnSLZgvZ"/>
TranType:	<input type="text" value="Auth2Sale"/>
Total:	<input type="text" value="1.00"/>
Submit:	<input type="button" value="Submit"/>

Note that only some information is being sent because the majority of the information is stored with the original transaction. Only the “TransID” is required, but passing in a “Total” of “1.00” prevents receiving an invalid amount error. The actual amount is not relevant, since it uses the amount from the original transaction.

The results are:

```
"YSUCCESSFUL", "", "", "6044", "20040831145904-
04971-6044"
```

That transaction will now be captured. Note the original transaction no longer exists and has been replaced by this new one. Also notice that a new TransID is returned.

### 4.3 Deleting an Authorization

Using HTML Example 4.1 again to get another authorization only transaction, the results are:

"YAPPROVED 031151", "Address and Zip Code Do Not Match (N)", "", "6045", "20040831150318-04971-6045"

ePNAccount:	<input type="text" value="04971"/>
TransID:	<input type="text" value="20040831150318-04971"/>
HTML:	<input type="text" value="No"/>
Inv:	<input type="text" value="report"/>
RestrictKey:	<input type="text" value="2D9kQtWZnSLZgvZ"/>
TranType:	<input type="text" value="AuthDel"/>
Total:	<input type="text" value="1.00"/>
Submit:	<input type="button" value="Submit"/>

Using HTML Example 4.2 again, but changing the TranType to AuthDel:

Results are:

"YSUCCESSFUL", "", "", "6045", "20040831150318-04971-6045"

### 4.4 Voiding a Transaction

ePNAccount:	<input type="text" value="04971"/>
CardNo:	<input type="text" value="04971"/>
ExpMonth:	<input type="text" value="12"/>
Exp Year:	<input type="text" value="09"/>
Total:	<input type="text" value="12.34"/>
Address:	<input type="text" value="123 Fake St."/>
Zip:	<input type="text" value="12345"/>
EMail (enter your email address):	<input type="text"/>
CVV2Type:	<input type="text" value="1"/>
CVV2:	<input type="text" value="123"/>
HTML:	<input type="text" value="No"/>
Inv:	<input type="text" value="report"/>
RestrictKey:	<input type="text" value="2D9kQtWZnSLZgvZ"/>
TranType:	<input type="text" value="Sale"/>
Submit:	<input type="button" value="Submit"/>

Using HTML Example 4.1 again and setting the TranType to Sale to create a transaction that can be voided.

Results are:

"YAUTH/TKT 031899", "Address and Zip Code Do Not Match (N)", "", "6046", "20040831150856-04971-6046"

ePNAccount:	<input type="text" value="04971"/>
TransID:	<input type="text" value="0831150856-04971-6046"/>
HTML:	<input type="text" value="No"/>
Inv:	<input type="text" value="report"/>
RestrictKey:	<input type="text" value="2D9kQtWZnSLZgvZ"/>
TranType:	<input type="text" value="Void"/>
Total:	<input type="text" value="1.00"/>
Submit:	<input type="button" value="Submit"/>

Using HTML example 4.2 again and change the TranType to "Void".

The results are:

"YSUCCESSFUL", "", "", "6046", "20040831150856-04971-6046-5"

Note that the TransID has changed it is the original TransID with a "-5" appended to it. This is identifying that this transaction was voided.

## 4.5 Processing A Return/Credit

Using HTML Example 4.5 set the TranType to “Return”.

ePNAccount:	<input type="text" value="04971"/>
CardNo:	<input type="text" value="04971"/>
ExpMonth:	<input type="text" value="12"/>
ExpYear:	<input type="text" value="09"/>
Total:	<input type="text" value="12.34"/>
Address:	<input type="text" value="123 Fake St."/>
Zip:	<input type="text" value="12345"/>
EMail (enter your email address):	<input type="text"/>
CVV2Type:	<input type="text" value="1"/>
CVV2:	<input type="text" value="123"/>
HTML:	<input type="text" value="No"/>
Inv:	<input type="text" value="report"/>
RestrictKey:	<input type="text" value="2D9kQtWZnSLZgvZ"/>
TranType:	<input type="text" value="Return"/>
Submit:	<input type="button" value="Submit"/>

The Results are:

"YSUCCESSFUL", "", "", "6047", "20040831152017-04971-6047"

Note: because of the security implications regarding credits/returns, you must specifically ask the eProcessingNetwork Technical Support people to “enable” the ability to run returns (and voids as well). The RestrictKey, which is described in detail in Section 6, must also be used.

## 4.6 Processing an Internet Check

We allow you to process two types of checks, internet checks where the customer/check writer is unknown to you. We also process NCN/Retail Checks where you are either face to face with the customer or you have an existing relationship with the customer, see the next section.

See HTML Example 4.6 in the appendix. For testing checks we need to switch to another demo account that is set up to accept checks. This account is 060321 password is 060321pw.

For a check we need to send different data, detailed on the next page.

ePNAccount:	<input type="text" value="060321"/>
PaymentType:	<input type="text" value="Check"/>
AccountType:	<input type="text" value="C"/>
AccountClass:	<input type="text" value="Checking"/>
Company:	<input type="text" value="Johns Barbeque"/>
FirstName:	<input type="text" value="John"/>
LastName:	<input type="text" value="Doe"/>
BankName:	<input type="text" value="304th National Bank"/>
Routing:	<input type="text" value="060321"/>
CheckAcct:	<input type="text" value="060321"/>
Check:	<input type="text" value="12345"/>
Address:	<input type="text" value="123 Fake St."/>
City:	<input type="text" value="Houston"/>
State:	<input type="text" value="TX"/>
Zip:	<input type="text" value="77008"/>
Phone:	<input type="text" value="713 880 0327"/>
Total:	<input type="text" value="12.34"/>
EMail (enter your email address):	<input type="text"/>
HTML:	<input type="text" value="No"/>
Inv:	<input type="text" value="report"/>
RestrictKey:	<input type="text" value="2D9kQtWZnSLZgvZ"/>
TranType:	<input type="text" value="Sale"/>
Submit:	<input type="button" value="Submit"/>

PaymentType	For a check pass this in as Check
AccountType	C for Company Check, P for Personal Check
AccountClass	Checking or Savings
Company	Required for a company check.
FirstName	Of the person
LastName	Of the person
BankName	Of the checking or savings account
Routing	Routing number on the check. Use your ePN account number for a test check.
CheckAcct	Account number, use your ePN account number for a test check
Check	Check Number
Address	Address on the check
City	City
State	State (2 characters)
Zip	Zip
Phone	Phone
TranType	MUST be set to Sale

The results are:

```
"YOK 14573-387602", "", "", "12", "20040831155652-060321-12-6"
```

Notice that the check TransID ends in a "-6".

## 4.7 Processing an NCN/Retail Check

We allow you to process retail or face to face checks as well, these are checks processed on the NCN system.

PaymentType:	Check
NCNMicr:	
NCNRoutingNum:	111111118
NCNAccountNum:	1212121
NCNCheckNum:	1234
NCNTran:	CHECK
NCNDLState:	TX
NCNDLNum:	D24524511
NCNAccountType:	P
NCNOverride:	convert
ePNAccount:	04971
RestrictKey:	
Total:	1.05
Submit	Submit

Here are all the relevant fields that can be used to process an NCN check on. First you set the "PaymentType" to "Check" and this indicates you are going to pay by check. Then you can provide either "NCNMicr" or "NCNRoutingNum/NCNAccountNum/NCNCheckNum". "NCNMicr" would be the MICR code produced by a Check MICR reader, you do not have the MICR then you must provide the "NCNRoutingNum", "NCNAccountNum" and "NCNCheckNum" which is for a keyed check. Then you set the "NCNTran" field to be one of 3 values "CHECK" which is for normal check processing, "VOID" if you are trying to cancel a converted/ACH'ed check or "OVERRIDE" if you get a "Manager Needed" response and want to process the check anyway. Next you must provide "NCNDLState" which is the state for the drivers license it must be a 2 character state abbreviation. If you are not providing the drivers license then enter "XX" here. Next is "NCNDLNum", this must begin with either a "D" for a keyed drivers license, or "G" for a "swiped" drivers license, after the "D" or "G" should be the drivers license data. "NCNOverride" is an optional field it should be either "verify" or "convert", "verify" forces the check to just do a verification (you will still need to deposit the check) or "convert" which forces the check to be converted (or ach'ed). Then use "RestrictKey" and "Total" as you would from the other transaction types. When this returns you will get a response like this:

"YAUTH NUM 741-917","Converted"

If you add "Inv" is "report" you will get:

"YAUTH NUM 742-614","Converted", "", "9318", "20061024141849-04971-9318-6"

The "Converted" value indicates how the check was processed either "Converted" or "Verified". Next is blank, followed by the invoice number (eProcessingNetwork) and the eProcessingNetwork Transaction ID.

## **4.8 Card Present/Swiped Transactions**

The TDBE can be used to process Card Present/Swiped transactions by passing in a variable named Swiped with a value of 1. The TDBE will then recognize the data passed in with the CardNo field as raw MagStripe data, with both Track 1 and Track 2 supported.

## **4.9 Purchase Card Level II Support**

Purchase Card Level II transactions are supported by passing in Tax (e.g. 12.33) and CustCode, Purchase Order specified by the cardholder of up to 17 characters.

## 5 Confirmation Script

Occasionally communication errors happen between server and client programs. A transaction may timeout and not get a response. eProcessingNetwork offers a mechanism to verify whether a transaction has completed.

The URL that should be posted to is:

```
https://www.eprocessingnetwork.com/cgi-bin/tdbe/confirmation.pl
```

Passing in the invoice number from an earlier transaction (developers **must** be using the `Inv=####` field to take advantage of this feature), the confirmation script will return the exact same response that the TDBE returned with the field named `HTML` set to `No` and the field named `Inv` set to `report`. This confirmation script may be used for up to 3 hours after a transaction has taken place.

Using the TDBE to process a Sale transaction and setting the invoice to “1234567”, the returned response is:

```
"YAUTH/TKT 001640","Address and Zip Code Do Not Match(N)",
    "", "1234567", "20040901095723-04971-1234567"
```

Note that the invoice number in the response is the invoice number that is passed in, “1234567”.

The three required field names passed in to the confirmation script are “ePNAccount”, “RestrictKey” and “Inv”, the invoice number of the transaction being tested for.

ePNAcct:	<input type="text" value="04971"/>
Inv:	<input type="text" value="1234567"/>
RestrictKey:	<input type="text" value="2D9kQWZnSLZgvZ"/>
Submit:	<input type="submit" value="Submit"/>

See HTML Example 5.1, Confirmation Script Request, in Appendix A for the HTML code for this example, shown in Figure 5.1.

Figure 5.1 – Confirmation Request

The results are:

```
"YAUTH/TKT 001640","Address and Zip Code do not match","Unknown CVV2
Response", "INV=1234567", "XactID=20040901095723-04971-1234567"
```

The first entry in the response is “Y” for approved, “N” for declined or “U” indicating that the transaction requested could not be found, but the rest are the normal TDBE responses,.

If the `RestrictKey` is not set for the requesting `ePNAccount`, or if the `RestrictKey` passed in is incorrect for the requesting `ePNAccount`, the response will always be:

```
"UService Unavailable"
```

## 6 Security

For this example, account 04971 is used to show the security features in the TDBE. Log into the Merchant Support Center on eProcessingNetwork’s main website. The demo account used for this example is 04971, with username “04971” and password “04971pw” (please do not include the quotes). Then go to the “Processing Controls” section. At the bottom of the page is a section entitled “Advanced”, (shown in Figure 6.1) where the “RestrictKey”, essentially a secret password, can be set. Use the value shown in the Merchant Support Center in your test transaction. The 04971 account is a public account and developers often change the RestrictKey on this account.

**Advanced**

The settings in the **Advanced** section are for advanced developers utilizing the **eProcessingNetwork Transparent Database Engine Template ONLY**. These settings **DO NOT** apply to **ePNs ePN Cart, Order Form, Database Engine Template**, or any other method of processing through ePN.

**Restricted Usage of Transparent Database Engine Template (TDBE)**

If you enable this security method, the TDBE will not process transactions for this ePN account unless the RestrictKey generated below is passed in with the transaction (RestrictKey=). **Please be advised that if you enable this feature, all TDBE transactions will be declined unless the correct RestrictKey is presented.**

Check to use RestrictKey to Restrict TDBE Usage.

Figure 6.1 – “Advanced” section of “Processing Controls” showing RestrictKey setting.

ePNAccount:	<input style="width: 100%;" type="text" value="04971"/>
CardNo:	<input style="width: 100%;" type="text" value="04971"/>
ExpMonth:	<input style="width: 100%;" type="text" value="12"/>
Exp Year:	<input style="width: 100%;" type="text" value="09"/>
Total:	<input style="width: 100%;" type="text" value="12.34"/>
Address:	<input style="width: 100%;" type="text" value="123 Fake St."/>
Zip:	<input style="width: 100%;" type="text" value="12345"/>
E-Mail (enter your email address):	<input style="width: 100%;" type="text"/>
CVV2Type:	<input style="width: 100%;" type="text" value="1"/>
CVV2:	<input style="width: 100%;" type="text" value="123"/>
HTML:	<input style="width: 100%;" type="text" value="No"/>
Inv:	<input style="width: 100%;" type="text" value="report"/>
RestrictKey:	<input style="width: 100%;" type="text" value="2D9kQtWZnSLZgvZ"/>
Submit:	<input type="button" value="Submit"/>

In this example, the “RestrictKey” that was passed in is the “RestrictKey” set in the Advanced section of Processing Controls.

Figure 6.2 – Transaction utilizing RestrictKey security.

If the “RestrictKey” feature is turned on in the “Processing Controls”, and an incorrect RestrictKey is passed in, the response will be as follows:

ePNAccount:	<input type="text" value="04971"/>
CardNo:	<input type="text" value="04971"/>
ExpMonth:	<input type="text" value="12"/>
Exp Year:	<input type="text" value="09"/>
Total:	<input type="text" value="12.34"/>
Address:	<input type="text" value="123 Fake St"/>
Zip:	<input type="text" value="12345"/>
EMail (enter your email address):	<input type="text"/>
CVV2Type:	<input type="text" value="1"/>
CVV2:	<input type="text" value="123"/>
HTML:	<input type="text" value="No"/>
Inv:	<input type="text" value="report"/>
RestrictKey:	<input type="text" value="WrongKey"/>
Submit:	<input type="button" value="Submit"/>

The response will always be:

"UService Unavailable"

No significant information is given because this is assumed to be an attempt to commit fraud.

# Appendix A - HTML Examples

## HTML Example 2.1 - Basic Example

```
<HTML>
<BODY>
<form action="https://www.eprocessingnetwork.com/cgi-bin/tdbe/transact.pl" method=post>
<table>
  <TR>
    <TD>ePNAccount:</TD>
    <TD><input type=text name="ePNAccount" value="05971"></TD>
  </TR>
  <TR>
    <TD>CardNo:</TD>
    <TD><input type=text name="CardNo" value="05971"></TD>
  </TR>
  <TR>
    <TD>ExpMonth:</TD>
    <TD><input type=text name="ExpMonth" value="12"></TD>
  </TR>
  <TR>
    <TD>ExpYear:</TD>
    <TD><input type=text name="ExpYear" value="09"></TD>
  </TR>
  <TR>
    <TD>Total:</TD>
    <TD><input type=text name="Total" value="12.34"></TD>
  </TR>
  <TR>
    <TD>Address:</TD>
    <TD><input type=text name="Address" value="123 Fake St."></TD>
  </TR>
  <TR>
    <TD>Zip:</TD>
    <TD><input type=text name="Zip" value="12345"></TD>
  </TR>
  <TR>
    <TD>EMail (enter your email address):</TD>
    <TD><input type=text name="EMail" value=""></TD>
  </TR>
  <TR>
    <TD>CVV2Type:</TD>
    <TD><input type=text name="CVV2Type" value="1"></TD>
  </TR>
  <TR>
    <TD>CVV2:</TD>
    <TD><input type=text name="CVV2Type" value="123"></TD>
  </TR>
  <TR>
    <TD>Submit:</TD>
    <TD><input type=submit name="submit" value="Submit"></TD>
  </TR>
</table>
</form>
</BODY>
</HTML>
```

## HTML Example 2.2 – Basic Example without HTML Output

```
<HTML>
<BODY>
<form action="https://www.eprocessingnetwork.com/cgi-bin/tdbe/transact.pl" method=post>
<table>
<TR>
  <TD>ePNAccount:</TD>
  <TD><input type=text name="ePNAccount" value="05971"></TD>
</TR>
<TR>
  <TD>CardNo:</TD>
  <TD><input type=text name="CardNo" value="05971"></TD>
</TR>
<TR>
  <TD>ExpMonth:</TD>
  <TD><input type=text name="ExpMonth" value="12"></TD>
</TR>
<TR>
  <TD>ExpYear:</TD>
  <TD><input type=text name="ExpYear" value="09"></TD>
</TR>
<TR>
  <TD>Total:</TD>
  <TD><input type=text name="Total" value="12.34"></TD>
</TR>
<TR>
  <TD>Address:</TD>
  <TD><input type=text name="Address" value="123 Fake St."></TD>
</TR>
<TR>
  <TD>Zip:</TD>
  <TD><input type=text name="Zip" value="12345"></TD>
</TR>
<TR>
  <TD>EMail (enter your email address):</TD>
  <TD><input type=text name="EMail" value=""></TD>
</TR>
<TR>
  <TD>CVV2Type:</TD>
  <TD><input type=text name="CVV2Type" value="1"></TD>
</TR>
<TR>
  <TD>CVV2:</TD>
  <TD><input type=text name="CVV2Type" value="123"></TD>
</TR>
<TR>
  <TD>HTML:</TD>
  <TD><input type=text name="HTML" value="No"></TD>
</TR>
<TR>
  <TD>Submit:</TD>
  <TD><input type=submit name="submit" value="Submit"></TD>
</TR>
</table>
</form>
</BODY>
</HTML>
```

## HTML Example 3.1 – Invoice and Transaction ID Reporting

```
<HTML><BODY>
<form action="https://www.eprocessingnetwork.com/cgi-bin/tdbe/transact.pl" method=post>
<table>
  <TR>
    <TD>ePNAccount:</TD>
    <TD><input type=text name="ePNAccount" value="05971"></TD>
  </TR>
  <TR>
    <TD>CardNo:</TD>
    <TD><input type=text name="CardNo" value="05971"></TD>
  </TR>
  <TR>
    <TD>ExpMonth:</TD>
    <TD><input type=text name="ExpMonth" value="12"></TD>
  </TR>
  <TR>
    <TD>ExpYear:</TD>
    <TD><input type=text name="ExpYear" value="09"></TD>
  </TR>
  <TR>
    <TD>Total:</TD>
    <TD><input type=text name="Total" value="12.34"></TD>
  </TR>
  <TR>
    <TD>Address:</TD>
    <TD><input type=text name="Address" value="123 Fake St."></TD>
  </TR>
  <TR>
    <TD>Zip:</TD>
    <TD><input type=text name="Zip" value="12345"></TD>
  </TR>
  <TR>
    <TD>EMail (enter your email address):</TD>
    <TD><input type=text name="EMail" value=""></TD>
  </TR>
  <TR>
    <TD>CVV2Type:</TD>
    <TD><input type=text name="CVV2Type" value="1"></TD>
  </TR>
  <TR>
    <TD>CVV2:</TD>
    <TD><input type=text name="CVV2Type" value="123"></TD>
  </TR>
  <TR>
    <TD>HTML:</TD>
    <TD><input type=text name="HTML" value="No"></TD>
  </TR>
  <TR>
    <TD>Inv:</TD>
    <TD><input type=text name="Inv" value="report"></TD>
  </TR>
  <TR>
    <TD>Submit:</TD>
    <TD><input type=submit name="submit" value="Submit"></TD>
  </TR>
</table>
</form>
</BODY></HTML>
```

## HTML Example 4.1

```
<HTML><BODY>
<form action="https://www.eprocessingnetwork.com/cgi-bin/tdbe/transact.pl"
method=post>
<table>
  <TR>
    <TD>ePNAccount:</TD>
    <TD><input type=text name="ePNAccount" value="04971"></TD>
  </TR>
  <TR>
    <TD>CardNo:</TD>
    <TD><input type=text name="CardNo" value="04971"></TD>
  </TR>
  <TR>
    <TD>ExpMonth:</TD>
    <TD><input type=text name="ExpMonth" value="12"></TD>
  </TR>
  <TR>
    <TD>ExpYear:</TD>
    <TD><input type=text name="ExpYear" value="09"></TD>
  </TR>
  <TR>
    <TD>Total:</TD>
    <TD><input type=text name="Total" value="12.34"></TD>
  </TR>
  <TR>
    <TD>Address:</TD>
    <TD><input type=text name="Address" value="123 Fake St."></TD>
  </TR>
  <TR>
    <TD>Zip:</TD>
    <TD><input type=text name="Zip" value="12345"></TD>
  </TR>
  <TR>
    <TD>EMail (enter your email address):</TD>
    <TD><input type=text name="EMail" value=""></TD>
  </TR>
  <TR>
    <TD>CVV2Type:</TD>
    <TD><input type=text name="CVV2Type" value="1"></TD>
  </TR>
  <TR>
    <TD>CVV2:</TD>
    <TD><input type=text name="CVV2Type" value="123"></TD>
  </TR>
  <TR>
    <TD>HTML:</TD>
    <TD><input type=text name="HTML" value="No"></TD>
  </TR>
  <TR>
    <TD>Inv:</TD>
    <TD><input type=text name="Inv" value="report"></TD>
  </TR>
  <TR>
    <TD>RestrictKey:</TD>
    <TD><input type=text name="RestrictKey" value="2D9kQtWZnSLZgvZ"></TD>
  </TR>
  <TR>
    <TD>Submit:</TD>
    <TD><input type=submit name="submit" value="Submit"></TD>
  </TR>
</table>
</form>
</BODY></HTML>
```

## HTML Example 4.2

```
<HTML>
<BODY>
<form action="https://www.eprocessingnetwork.com/cgi-bin/tdbe/transact.pl"
      method=post>
<table>
<TR>
      <TD>ePNAccount:</TD>
      <TD><input type=text name="ePNAccount" value="04971"></TD>
</TR>
<TR>
      <TD>TransID:</TD>
      <TD><input type=text name="TransID" value=""></TD>
</TR>
<TR>
      <TD>HTML:</TD>
      <TD><input type=text name="HTML" value="No"></TD>
</TR>
<TR>
      <TD>Inv:</TD>
      <TD><input type=text name="Inv" value="report"></TD>
</TR>
<TR>
      <TD>RestrictKey:</TD>
      <TD><input type=text name="RestrictKey" value="2D9kQtWZnSLZgvZ"></TD>
</TR>
<TR>
      <TD>TranType:</TD>
      <TD><input type=text name="TranType" value="Auth2Sale"></TD>
</TR>
<TR>
      <TD>Total:</TD>
      <TD><input type=text name="Total" value="1.00"></TD>
</TR>
<TR>
      <TD>Submit:</TD>
      <TD><input type=submit name="submit" value="Submit"></TD>
</TR>
</table>
</form>
</BODY>
</HTML>
```

## HTML Example 4.5

```
<HTML><BODY>
<form action="https://www.eprocessingnetwork.com/cgi-bin/tdbe/transact.pl" method=post>
<table>
  <TR>
    <TD>ePNAccount:</TD>
    <TD><input type=text name="ePNAccount" value="04971"></TD>
  </TR>
  <TR>
    <TD>CardNo:</TD>
    <TD><input type=text name="CardNo" value="04971"></TD>
  </TR>
  <TR>
    <TD>ExpMonth:</TD>
    <TD><input type=text name="ExpMonth" value="12"></TD>
  </TR>
  <TR>
    <TD>ExpYear:</TD>
    <TD><input type=text name="ExpYear" value="09"></TD>
  </TR>
  <TR>
    <TD>Total:</TD>
    <TD><input type=text name="Total" value="12.34"></TD>
  </TR>
  <TR>
    <TD>Address:</TD>
    <TD><input type=text name="Address" value="123 Fake St."></TD>
  </TR>
  <TR>
    <TD>Zip:</TD>
    <TD><input type=text name="Zip" value="12345"></TD>
  </TR>
  <TR>
    <TD>EMail (enter your email address):</TD>
    <TD><input type=text name="EMail" value=""></TD>
  </TR>
  <TR>
    <TD>CVV2Type:</TD>
    <TD><input type=text name="CVV2Type" value="1"></TD>
  </TR>
  <TR>
    <TD>CVV2:</TD>
    <TD><input type=text name="CVV2Type" value="123"></TD>
  </TR>
  <TR>
    <TD>HTML:</TD>
    <TD><input type=text name="HTML" value="No"></TD>
  </TR>
  <TR>
    <TD>Inv:</TD>
    <TD><input type=text name="Inv" value="report"></TD>
  </TR>
  <TR>
    <TD>RestrictKey:</TD>
    <TD><input type=text name="RestrictKey" value="2D9kQtWZnSLZgvZ"></TD>
  </TR>
  <TR>
    <TD>TranType:</TD>
    <TD><input type=text name="TranType" value="AuthOnly"></TD>
  </TR>
  <TR>
    <TD>Submit:</TD>
    <TD><input type=submit name="submit" value="Submit"></TD>
  </TR>
</table></form></BODY></HTML>
```

## HTML Example 4.6

```
<HTML>
<BODY>
<form action="https://www.eprocessingnetwork.com/cgi-bin/tdbe/transact.pl"
method=post>
<table>
<TR>
<TD>ePNAccount:</TD>
<TD><input type=text name="ePNAccount" value="060321"></TD>
</TR>
<TR>
<TD>PaymentType:</TD>
<TD><input type=text name="PaymentType" value="Check"></TD>
</TR>
<TR>
<TD>AccountType:</TD>
<TD><input type=text name="AccountType" value="C"></TD>
</TR>
<TR>
<TD>AccountClass:</TD>
<TD><input type=text name="AccountClass" value="Checking"></TD>
</TR>
<TR>
<TD>Company:</TD>
<TD><input type=text name="Company" value="Johns Barbeque"></TD>
</TR>
<TR>
<TD>FirstName:</TD>
<TD><input type=text name="FirstName" value="John"></TD>
</TR>
<TR>
<TD>LastName:</TD>
<TD><input type=text name="LastName" value="Doe"></TD>
</TR>
<TR>
<TD>BankName:</TD>
<TD><input type=text name="BankName" value="304th National Bank"></TD>
</TR>
<TR>
<TD>Routing:</TD>
<TD><input type=text name="Routing" value="060321"></TD>
</TR>
<TR>
<TD>CheckAcct:</TD>
<TD><input type=text name="CheckAcct" value="060321"></TD>
</TR>
<TR>
<TD>Check:</TD>
<TD><input type=text name="Check" value="12345"></TD>
</TR>
<TR>
<TD>Address:</TD>
<TD><input type=text name="Address" value="123 Fake St."></TD>
</TR>
<TR>
<TD>City:</TD>
<TD><input type=text name="City" value="Houston"></TD>
</TR>
```

```

<TR>
  <TD>State:</TD>
  <TD><input type=text name="State" value="TX"></TD>
</TR>
<TR>
  <TD>Zip:</TD>
  <TD><input type=text name="Zip" value="77008"></TD>
</TR>
<TR>
  <TD>Phone:</TD>
  <TD><input type=text name="Phone" value="713 880 0327"></TD>
</TR>
<TR>
  <TD>Total:</TD>
  <TD><input type=text name="Total" value="12.34"></TD>
</TR>
<TR>
  <TD>EMail (enter your email address):</TD>
  <TD><input type=text name="EMail" value=""></TD>
</TR>
<TR>
  <TD>HTML:</TD>
  <TD><input type=text name="HTML" value="No"></TD>
</TR>
<TR>
  <TD>Inv:</TD>
  <TD><input type=text name="Inv" value="report"></TD>
</TR>
<TR>
  <TD>RestrictKey:</TD>
  <TD><input type=text name="RestrictKey" value="2D9kQtWZnSLZgvZ"></TD>
</TR>
<TR>
  <TD>TranType:</TD>
  <TD><input type=text name="TranType" value="Sale"></TD>
</TR>
<TR>
  <TD>Submit:</TD>
  <TD><input type=submit name="submit" value="Submit"></TD>
</TR>
</table>
</form>
</BODY>
</HTML>

```

## HTML Example 5.1 – Confirmation Script Request

```
<HTML><BODY>
<form action="https://www.eprocessingnetwork.com/cgi-bin/tdbe/confirmation.pl"
method=post>
<table>
  <TR>
    <TD>ePNAcct:</TD>
    <TD><input type=text name="ePNAcct" value="04971"></TD>
  </TR>
  <TR>
    <TD>Inv:</TD>
    <TD><input type=text name="Inv" value="report"></TD>
  </TR>
  <TR>
    <TD>RestrictKey:</TD>
    <TD><input type=text name="RestrictKey" value="2D9kQtWZnSLZgvZ"></TD>
  </TR>
  <TR>
    <TD>Submit:</TD>
    <TD><input type=submit name="submit" value="Submit"></TD>
  </TR>
</table>
</form>
</BODY>
</HTML>
```