



**LINKPOINT COM OBJECT
(LCOM V5.4)**

DRAFT USER MANUAL

Version 1.1

SOFTWARE LICENSING AGREEMENT

LICENSE. LinkPoint International, Inc. (LinkPoint) hereby grants you the right to use all or a portion of the Software accompanying this License, regardless of media, and the related Documentation. This license does not grant you any right to any enhancement or update to the Software and Documentation. You may obtain enhancements and updates, if available, at LinkPoint's then-current terms and conditions.

RESTRICTIONS. Any copy of the Software or derivative works made by you shall be subject to this Agreement. The copy or derivative work shall contain this license as well as related Documentation as well as all notices regarding copyrights, trademarks and other proprietary rights that came with the original Software and Documentation. This Software is protected by the copyright laws of the United States and international copyright treaties. You may not rent or lease the Software. You may not reverse engineer, decompile or disassemble the Software.

TITLE. Title, ownership rights, and intellectual property rights in and to the Software and Documentation and derivative works, if any, shall remain in LinkPoint.

TERMINATION. This Agreement may be terminated immediately by the non-defaulting party if either you or LinkPoint default in its obligations. Upon any termination of this Agreement, you shall immediately discontinue the use of the Software and derivative works. You may also terminate this Agreement at any time by destroying the Software, derivative works, if any, and Documentation and all copies thereof.

INDEMNIFICATION. If you distribute the Software in violation of this Agreement, you agree to indemnify, hold harmless and defend LinkPoint from and against any claims or lawsuits, including attorneys fees that arise or result from the use or distribution of the Software in violation of this Agreement.

EXPORT CONTROLS. You may not download or otherwise export or re-export the Software or any underlying information or technology except in full compliance with all United States and other applicable laws and regulations. In particular, but without limitation, none of the Software or underlying information or technology may be downloaded or otherwise exported or re-exported (i) into (or to a national or resident of) Cuba, Haiti, Iraq, Libya, Yugoslavia, North Korea, Iran, Syria or any other country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Commerce Department's Table of Deny Orders. By downloading or using the Software, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country or on any such list.

GOVERNMENT USE. Use, duplication or disclosure by the Government is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer-Restricted Rights clause at FAR 52.227-19 when applicable, or in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, and in similar clauses in the NASA FAR Supplement. Contractor/manufacturer is LinkPoint International, Inc., 5310 Derry Avenue, Suite W, Agoura Hills, California 91301.

DISCLAIMER OF WARRANTY. You expressly acknowledge and agree that use of the LinkPoint Software is at your sole risk. The Software and related documentation are provided on an AS IS basis, without warranty of any kind, including without limitation the warranties of merchantability, fitness for a particular purpose and non-infringement. The entire risk as to the quality and performance of the Software is borne by you. Should the Software prove defective, you and not LinkPoint, assume the entire cost of any service and repair. This disclaimer of warranty constitutes an essential part of the agreement.

LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL LINKPOINT OR ITS SUPPLIERS OR RESELLERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES. IN NO EVENT WILL LINKPOINT BE LIABLE FOR ANY DAMAGES IN EXCESS OF LINKPOINT'S LIST PRICE FOR A LICENSE TO THE SOFTWARE, EVEN IF LINKPOINT SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. FURTHERMORE, SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION AND EXCLUSION MAY NOT APPLY TO YOU.

COMPLETE AGREEMENT. This Agreement represents the complete and exclusive statement of the agreements concerning this license between the parties and supersedes all prior agreements and representations between them. It may be amended only by a writing executed by both parties. THE USE OF THE SOFTWARE BY YOU IS EXPRESSLY MADE CONDITIONAL ON YOUR ASSENT TO THE TERMS SET FORTH HEREIN, AND LINKPOINT AGREES TO FURNISH THE SOFTWARE AND DOCUMENTATION ONLY UPON THESE TERMS.

CONTROLLING LAW AND SEVERABILITY. If any provision of this Agreement is held to be unenforceable for any reason, such provision shall be reformed only to the extent necessary to make it enforceable, and such decision shall not affect the enforceability (i) of such provision under other circumstances or (ii) of the remaining provisions hereof under all circumstances. Headings shall not be considered in interpreting this Agreement. This Agreement shall be governed by and construed under California law as such law applies to agreements between California residents entered into and to be performed entirely within California, except as governed by Federal law. This Agreement will not be governed by the United Nations Convention of Contracts for the International Sale of Goods, the application of which is hereby expressly excluded.

LinkPoint International, Inc.
5310 Derry Avenue, Suite W
Agoura Hills, California 91301

Table of Contents

ABOUT THIS MANUAL	1
INTRODUCTION	2
METHODS AND FUNCTIONALITY	3
Methods	3
Functionality	8
INSTALLATION.....	9
Pre-Installation System Requirements	9
Installing LCOM.....	9
PRODUCT INTEGRATION.....	10
SAMPLE SCRIPTS.....	11
ACTIVE SERVER PAGE (ASP) SAMPLES	12
Example 1A: API File: api_ordertest.asp.....	12
Example 1B: API File: api_processcard.asp	16
Example 2A: Periodic Billing File: pb_ordertest.asp	25
Example 2B: Periodic Billing File: pb_processcard.asp	28
Example 3A: VirtualCheck File: vc_ordertest.asp	38
Example 3B: VirtualCheck File: vc_processcard.asp	41
SITE SERVER COMMERCE EDITION PAYMENT COMPONENT SAMPLE	51
APPENDIX A	59
Deliverables.....	59
Purchasing Contact Information.....	59
APPENDIX B.....	60
Technical Support.....	60

ABOUT THIS MANUAL

This document was written with the assumption that the reader has a basic administrative knowledge of and permissions for Windows component object model (COM) technology and the NT operating system and environment into which the COM object is being installed.

If this assumption is incorrect, please contact your LinkPoint International representative for further installation assistance.

INTRODUCTION

To support storefront development on various platforms, LinkPoint International has developed a COM object (Windows) Wrapper (LCOM), which provides the same functionality as LinkPoint Select API.

LCOM is an in-process server supporting a single-threaded apartment (STA) model. In addition, LCOM was created as a Release MinDependency build.

With LCOM, transaction processing methods can be invoked from C, C++, Visual Basic, VB Script or any other language that can instantiate a COM object.

LCOM is *not* self-registering: you must run regsvr32.exe against it.

METHODS AND FUNCTIONALITY

Methods

LCOM exposes a single interface called ILinkPointCom that is comprised of the following methods and properties:

```
HRESULT STDMETHODCALLTYPE get_bStat(  
    /* [propget][retval][out] */ BOOL __RPC_FAR *pVal);
```

```
HRESULT STDMETHODCALLTYPE csi_item_adoption(  
    /* [in] */ long item,  
    /* [in] */ long option);
```

```
HRESULT STDMETHODCALLTYPE csi_item_alloc(  
    /* [retval][out] */ long __RPC_FAR *item);
```

```
HRESULT STDMETHODCALLTYPE csi_item_clear(  
    /* [in] */ long item);
```

```
HRESULT STDMETHODCALLTYPE csi_item_drop(  
    /* [in] */ long item);
```

```
HRESULT STDMETHODCALLTYPE csi_item_error(  
    /* [in] */ long item,  
    /* [retval][out] */ long __RPC_FAR *err);
```

```
HRESULT STDMETHODCALLTYPE csi_item_get(  
    /* [in] */ long item,  
    /* [in] */ long item_field,  
    /* [retval][out] */ VARIANT __RPC_FAR *field);
```

```
HRESULT STDMETHODCALLTYPE csi_item_set(  
    /* [in] */ long item,  
    /* [in] */ long item_field,  
    /* [in] */ VARIANT field);
```

```
HRESULT STDMETHODCALLTYPE csi_option_alloc(  
    /* [retval][out] */ long __RPC_FAR *option);
```

```
HRESULT STDMETHODCALLTYPE csi_option_clear(  
    /* [in] */ long option);
```

```
HRESULT STDMETHODCALLTYPE csi_option_drop(  
    /* [in] */ long option);
```

```
HRESULT STDMETHODCALLTYPE csi_option_error(  
    /* [in] */ long option,  
    /* [retval][out] */ long __RPC_FAR *err);
```

```

HRESULT STDMETHODCALLTYPE csi_option_get(
    /* [in] */ long option,
    /* [in] */ long option_field,
    /* [retval][out] */ VARIANT __RPC_FAR *field);

HRESULT STDMETHODCALLTYPE csi_option_set(
    /* [in] */ long option,
    /* [in] */ long option_field,
    /* [in] */ VARIANT field);

HRESULT STDMETHODCALLTYPE csi_order_additem(
    /* [in] */ long order,
    /* [in] */ long item);

HRESULT STDMETHODCALLTYPE csi_order_alloc(
    /* [retval][out] */ long __RPC_FAR *order);

HRESULT STDMETHODCALLTYPE csi_order_clear(
    /* [in] */ long order);

HRESULT STDMETHODCALLTYPE csi_order_drop(
    /* [in] */ long order);

HRESULT STDMETHODCALLTYPE csi_order_get(
    /* [in] */ long order,
    /* [in] */ long order_field,
    /* [retval][out] */ VARIANT __RPC_FAR *field);

HRESULT STDMETHODCALLTYPE csi_order_getesd(
    /* [in] */ long order,
    /* [retval][out] */ VARIANT __RPC_FAR *esd);

HRESULT STDMETHODCALLTYPE csi_order_process(
    /* [in] */ long order);

HRESULT STDMETHODCALLTYPE csi_order_set(
    /* [in] */ long order,
    /* [in] */ long order_field,
    /* [in] */ VARIANT field);

HRESULT STDMETHODCALLTYPE csi_order_setrequest(
    /* [in] */ long order,
    /* [in] */ long req);

HRESULT STDMETHODCALLTYPE csi_req_alloc(
    /* [retval][out] */ long __RPC_FAR *req);

HRESULT STDMETHODCALLTYPE csi_req_clear(
    /* [in] */ long req);

HRESULT STDMETHODCALLTYPE csi_req_drop(
    /* [in] */ long req);

```



```

HRESULT STDMETHODCALLTYPE csi_req_error(
    /* [in] */ long req,
    /* [retval][out] */ long __RPC_FAR *err);

HRESULT STDMETHODCALLTYPE csi_req_get(
    /* [in] */ long req,
    /* [in] */ long req_field,
    /* [retval][out] */ VARIANT __RPC_FAR *field);

HRESULT STDMETHODCALLTYPE csi_req_set(
    /* [in] */ long req,
    /* [in] */ long req_field,
    /* [in] */ VARIANT field);

HRESULT STDMETHODCALLTYPE csi_shipping_alloc(
    /* [retval][out] */ long __RPC_FAR *ship);

HRESULT STDMETHODCALLTYPE csi_shipping_clear(
    /* [in] */ long ship);

HRESULT STDMETHODCALLTYPE csi_shipping_drop(
    /* [in] */ long ship);

HRESULT STDMETHODCALLTYPE csi_shipping_error(
    /* [in] */ long ship,
    /* [retval][out] */ long __RPC_FAR *err);

HRESULT STDMETHODCALLTYPE csi_shipping_get(
    /* [in] */ long ship,
    /* [in] */ long ship_field,
    /* [retval][out] */ VARIANT __RPC_FAR *field);

HRESULT STDMETHODCALLTYPE csi_shipping_process(
    /* [in] */ long ship);

HRESULT STDMETHODCALLTYPE csi_shipping_set(
    /* [in] */ long ship,
    /* [in] */ long ship_field,
    /* [in] */ VARIANT field);

HRESULT STDMETHODCALLTYPE csi_shipping_setrequest(
    /* [in] */ long ship,
    /* [in] */ long req);

HRESULT STDMETHODCALLTYPE csi_tax_alloc(
    /* [retval][out] */ long __RPC_FAR *tax);

HRESULT STDMETHODCALLTYPE csi_tax_clear(
    /* [in] */ long tax);

HRESULT STDMETHODCALLTYPE csi_tax_drop(
    /* [in] */ long tax);

```

```

HRESULT STDMETHODCALLTYPE csi_tax_error(
    /* [in] */ long tax,
    /* [retval][out] */ long __RPC_FAR *err);

HRESULT STDMETHODCALLTYPE csi_tax_get(
    /* [in] */ long tax,
    /* [in] */ long tax_field,
    /* [retval][out] */ VARIANT __RPC_FAR *field);

HRESULT STDMETHODCALLTYPE csi_tax_process(
    /* [in] */ long tax);

HRESULT STDMETHODCALLTYPE csi_tax_set(
    /* [in] */ long tax,
    /* [in] */ long tax_field,
    /* [in] */ VARIANT field);

HRESULT STDMETHODCALLTYPE csi_tax_setrequest(
    /* [in] */ long tax,
    /* [in] */ long req);

HRESULT STDMETHODCALLTYPE csi_util_errorstr(
    /* [in] */ long err,
    /* [retval][out] */ VARIANT __RPC_FAR *str);

HRESULT STDMETHODCALLTYPE csi_util_version(
    /* [retval][out] */ double __RPC_FAR *version);

HRESULT STDMETHODCALLTYPE csi_util_version_string(
    /* [retval][out] */ double __RPC_FAR *version_string);

HRESULT STDMETHODCALLTYPE csi_order_error(
    /* [in] */ long order,
    /* [retval][out] */ long __RPC_FAR *err);

HRESULT STDMETHODCALLTYPE csi_periodic_alloc(
    /* [retval][out] */ long __RPC_FAR *periodic);

HRESULT STDMETHODCALLTYPE csi_periodic_clear(
    /* [in] */ long periodic);

HRESULT STDMETHODCALLTYPE csi_periodic_drop(
    /* [in] */ long periodic);

HRESULT STDMETHODCALLTYPE csi_periodic_get(
    /* [in] */ long periodic,
    /* [in] */ long periodic_field,
    /* [retval][out] */ VARIANT __RPC_FAR *field);

HRESULT STDMETHODCALLTYPE csi_periodic_set(
    /* [in] */ long periodic,
    /* [in] */ long periodic_field,
    /* [in] */ VARIANT field);

HRESULT STDMETHODCALLTYPE csi_pborder_alloc(

```

```

        /* [retval][out] */ long __RPC_FAR *pborder);

HRESULT STDMETHODCALLTYPE csi_pborder_drop(
    /* [in] */ long pborder);

HRESULT STDMETHODCALLTYPE csi_pborder_error(
    /* [in] */ long pborder,
    /* [retval][out] */ long __RPC_FAR *err);

HRESULT STDMETHODCALLTYPE csi_pborder_process(
    /* [in] */ long pborder);

HRESULT STDMETHODCALLTYPE csi_pborder_setorder(
    /* [in] */ long pborder,
    /* [in] */ long order);

HRESULT STDMETHODCALLTYPE csi_pborder_setperiodic(
    /* [in] */ long pborder,
    /* [in] */ long periodic);

HRESULT STDMETHODCALLTYPE csi_pborder_settype(
    /* [in] */ long pborder,
    /* [in] */ long type);

HRESULT STDMETHODCALLTYPE csi_util_pborder_errorstr(
    /* [in] */ long err,
    /* [retval][out] */ VARIANT __RPC_FAR *str);

HRESULT STDMETHODCALLTYPE csi_check_alloc(
    /* [retval][out] */ long __RPC_FAR *check);

HRESULT STDMETHODCALLTYPE csi_check_clear(
    /* [in] */ long check);

HRESULT STDMETHODCALLTYPE csi_check_drop(
    /* [in] */ long check);

HRESULT STDMETHODCALLTYPE csi_check_error(
    /* [in] */ long check,
    /* [retval][out] */ long __RPC_FAR *err);

HRESULT STDMETHODCALLTYPE csi_check_get(
    /* [in] */ long check,
    /* [in] */ long check_field,
    /* [retval][out] */ VARIANT __RPC_FAR *field);

HRESULT STDMETHODCALLTYPE csi_check_set(
    /* [in] */ long check,
    /* [in] */ long check_field,
    /* [in] */ VARIANT field);

HRESULT STDMETHODCALLTYPE csi_checkorder_alloc(
    /* [retval][out] */ long __RPC_FAR *checkorder);

HRESULT STDMETHODCALLTYPE csi_checkorder_drop(

```

```

        /* [in] */ long checkorder);

HRESULT STDMETHODCALLTYPE csi_checkorder_error(
    /* [in] */ long checkorder,
    /* [retval][out] */ long __RPC_FAR *err);

HRESULT STDMETHODCALLTYPE csi_checkorder_process(
    /* [in] */ long checkorder);

HRESULT STDMETHODCALLTYPE csi_checkorder_setcheck(
    /* [in] */ long checkorder,
    /* [in] */ long check);

HRESULT STDMETHODCALLTYPE csi_checkorder_setorder(
    /* [in] */ long checkorder,
    /* [in] */ long order);

HRESULT STDMETHODCALLTYPE csi_checkorder_settype(
    /* [in] */ long checkorder,
    /* [in] */ long type);

HRESULT STDMETHODCALLTYPE csi_util_checkorder_errorstr(
    /* [in] */ long err,
    /* [retval][out] */ VARIANT __RPC_FAR *str);

```

Functionality

Refer to the LinkPoint Select API manual for descriptions of the methods and their arguments.

INSTALLATION

Pre-Installation System Requirements

- Windows NT
- LinkPoint-assigned electronic certificate (.pem file)
- Your merchant account must be provisioned for LinkPoint Select API

Installing LCOM

Installation of LCOM consists of registering the COM object and following the steps below. If you have problems during installation, e-mail LinkPoint International for assistance (see Technical Support).

1. Unzip the LpiCom_5_4.zip file.
2. Copy the LpiCom_5_4.DLL to C:\WINNT\SYSTEM32.
3. From a disk operating system (DOS) window, navigate to C:\WINNT\SYSTEM32.
4. Issue the following command:

```
>regsvr32 LpiCom_5_4.DLL
```
5. Read the ComReadme.txt file for additional information.
6. Follow the necessary instructions to run one of the sample programs to validate the installation and connectivity to the LinkPoint Secure Payment Gateway.

PRODUCT INTEGRATION

To integrate LCOM, you need only to call directly the methods provided by COM object from your application. The *Sample Scripts* section provides several samples that call methods in the LCOM module. You may modify and use these samples as an example to integrate LCOM into your product.

Note: The following are parameters in the sample script that need to be updated for your particular testing and implementation:

- ReqField_Configfile (store name)
- ReqField_Keyfile (absolute path and file name for your .pem file)
- ReqFile_Host (secure host name)
- ReqFile_Port (usually 1139)
- OrderField_Email (test customer e-mail address)

SAMPLE SCRIPTS

To assist in the development of storefronts using LCOM, sample programs have been written in Active Server Page (ASP) showing how to invoke the COM methods. All of the programs mirror the sample *ccapi_ordertest.c* that comes with LinkPoint Select API.

Note: Before running any of these samples, you will need to change the store-specific parameters to your store. These parameters are:

- yourstore (your 6-digit store name)
- yourpem (full path and file name for your .pem file)
- securehost (secure host name)
- youraddress (customer e-mail address for receipts)

Note: When passing parameters to the COM object using VB Script variables, you *must* cast them using the CStr(), CInt() and CDbl() functions.

In VB Script, COM/application program interface (API) methods that return a value do so in the method name itself or function return values. Functions that do not return a value still need an IValue for its call. These would be the identifiers flag in the following sample.

In addition, the constants found in the following sample code were derived from the constants defined in the header files included with LinkPoint Select API. The values of these constants must *not* be altered.


```

</tr>
<tr>

<td height="34">Tax: </td>
<td colspan="2" height="34">
  <input type="text" size="20" name="txtTax" value="<%=GetValue("txtTax")%>">
</td>
</tr>
<tr>
<td>Charge Total: </td>
<td colspan="2">
  <input type="text" size="20" name="txtChargeTotal"
value="<%=GetValue("txtChargeTotal")%>">
</td>
</tr>
<tr>
<td>Time: </td>
<td colspan="2">
  <input type="text" size="40" name="txtTime" value="<%=GetValue("txtTime")%>">
</td>
</tr>
<tr>
<td>Ref #: </td>
<td colspan="2">
  <input type="text" size="20" name="txtRef" value="<%=GetValue("txtRef")%>">
</td>
</tr>
<tr>
<td>Approved: </td>
<td colspan="2">
  <input type="text" size="20" name="txtAppr" value="<%=GetValue("txtAppr")%>">
</td>
</tr>
<tr>
<td height="24">Code: </td>
<td height="24" colspan="2">
  <input type="text" size="20" name="txtCode" value="<%=GetValue("txtCode")%>">
</td>
</tr>
<tr>
<td>Error1: </td>
<td colspan="2">
  <input type="text" size="60" name="txtError" value="<%=GetValue("txtError")%>">
</td>
</tr>
<tr>
<td>Error2: </td>
<td colspan="2">
  <input type="text" size="60" name="txtErr" value="<%=GetValue("txtErr")%>">
</td>
</tr>

```

```
<tr>
  <td>Order #: </td>
  <td colspan="2">
    <input type="text" size="40" name="txtOrd" value="<%=GetValue("txtOrd")%>">
  </td>
</tr>
<tr>

  <td>&nbsp;</td>
</tr>
</table>
</form>
</body>
</html>
<%END IF%>
```

Example 1B:

API File: api_processcard.asp

```
<%  
' Created from ccapi_error.h from API 3.8.3  
Const Fail = 0  
Const Succeed = 1  
  
' Created from ccapi_client.h from API 3.8.3  
  
' Request types possible for OrderField_Chargetype  
Const Chargetype_Auth = 0  
Const Chargetype_Sale = 0  
Const Chargetype_Preauth = 1  
Const Chargetype_Postauth = 2  
Const Chargetype_Credit = 3  
Const Chargetype_Voidsale = 9  
Const Chargetype_Error = 0  
  
' Result types possible for OrderField_Result  
Const Result_Live = 0  
Const Result_Good = 1  
Const Result_Duplicate = 2  
Const Result_Decline = 3  
  
' ESD types for ItemField_Esdtype  
Const Esdtype_None = 0  
Const Esdtype_Softgood = 1  
Const Esdtype_Key = 2  
  
' OrderField_t  
Const OrderField_Userid = 1  
Const OrderField_Bcompany = 2  
Const OrderField_Bcountry = 3  
Const OrderField_Bname = 4  
Const OrderField_Baddr1 = 5  
Const OrderField_Baddr2 = 6  
Const OrderField_Bcity = 7  
Const OrderField_Bstate = 8  
Const OrderField_Bzip = 9  
Const OrderField_Sname = 10  
Const OrderField_Saddr1 = 11  
Const OrderField_Saddr2 = 12  
Const OrderField_Scity = 13  
Const OrderField_Sstate = 14  
Const OrderField_Szip = 15  
Const OrderField_Scountry = 16  
Const OrderField_Phone = 17  
Const OrderField_Fax = 18  
Const OrderField_Refer = 19
```

Const OrderField_Shiptype = 20
Const OrderField_Shipping = 21
Const OrderField_Tax = 22
Const OrderField_Subtotal = 23
Const OrderField_Vattax = 24
Const OrderField_Comments = 25
Const OrderField_PONumber = 26
Const OrderField_TaxExempt = 27
Const OrderField_Oid = 0
Const OrderField_MaxItems = 28
Const OrderField_Email = 29
Const OrderField_Cardnumber = 30
Const OrderField_Expmonth = 31
Const OrderField_Expyear = 32
Const OrderField_Chargetype = 33
Const OrderField_Chargetotal = 34
Const OrderField_Referencenumber = 35
Const OrderField_Result = 36
Const OrderField_Addrnum = 37
Const OrderField_Ip = 38
Const OrderField_MercType = 39
Const OrderField_SwitchStartDate = 40
Const OrderField_SwitchIssueNumber = 41

' Responses

Const OrderField_R_Time = 42
Const OrderField_R_Ref = 43
Const OrderField_R_Approved = 44
Const OrderField_R_Code = 45
Const OrderField_R_Ordernum = 46
Const OrderField_R_Error = 47
Const OrderField_R_FraudCode = 48
Const OrderField_R_AuthResponse = 49
Const OrderField_R_Message = 50

' ReqField_t

Const ReqField_Configfile = 0
Const ReqField_Keyfile = 1
Const ReqField_Appname = 2
Const ReqField_Host = 3
Const ReqField_Port = 4

' ItemField_t

Const ItemField_Itemid = 0
Const ItemField_Description = 1
Const ItemField_Price = 2
Const ItemField_Quantity = 3
Const ItemField_Softfile = 4
Const ItemField_Esdtype = 5
Const ItemField_Serial = 6
Const ItemField_MaxOptions = 7

```

' ShippingField_t
Const ShippingField_Country = 0
Const ShippingField_State = 1
Const ShippingField_Total = 2
Const ShippingField_Items = 3
Const ShippingField_Weight = 4
Const ShippingField_Carrier = 5
' Responses
Const ShippingField_R_Total = 6

' TaxField_t
Const TaxField_State = 0
Const TaxField_Zip = 1
Const TaxField_Total = 2
' Responses
Const TaxField_R_Total = 3

' OptionField_t
Const OptionField_Option = 0
Const OptionField_Choice = 1

' PbOrder_t
Const PbOrder_Submit = 0
Const PbOrder_Modify = 1
Const PbOrder_Cancel = 2
Const PbOrder_Process = 3

' PeriodicField_t
Const PeriodicField_Startdate = 0
Const PeriodicField_Periodicity = 1
Const PeriodicField_Installments = 2
Const PeriodicField_Threshold = 3
Const PeriodicField_Comments = 4

' CheckOrder_t
Const CheckOrder_Submit = 0
Const CheckOrder_Void = 1

' CheckField_t;
Const CheckField_AccountType = 0
Const CheckField_TransitRouting = 1
Const CheckField_MICR = 2
Const CheckField_CheckNumber = 3
Const CheckField_Comments = 4

' Opaque contexts
Dim OrderCtx
Dim ItemCtx
Dim OptionCtx
Dim ReqCtx
Dim ShippingCtx
Dim TaxCtx

```

```

Dim ApiDriver

Set txtArray = Server.CreateObject("Scripting.Dictionary")

'-----
' Functions
'-----
Function ProcessCard()

On Error Resume Next

' Local variable declarations
Dim txtVersion
Dim port
Dim price
Dim total
Dim itemtotal
Dim subtotal
Dim taxtotal
Dim esdfiles

Err.Clear
Set ApiDriver = Server.CreateObject("LpiCom_5_4.LinkPointCom")
if (Err.Number <> 0) Then
    Response.Write("Error creating object: " & Err.Description & "<BR>")
    Exit Function
End if

Dim bname
Dim card
Dim baddr1
Dim bstate
Dim bcity
Dim bzip
Dim numShirts
Dim numProgs
Dim shirtprice
Dim progprice

bname = Request.Form("bname")
baddr1 = Request.Form("baddr1")
bstate = Request.Form("bstate")
bcity = Request.Form("bcity")
bzip = Request.Form("bzip")
card = Request.Form("creditcard")
numshirts = Request.Form("numShirts")
numprogs = Request.Form("numProg")
shirtprice = Request.Form("shirtprice")
progprice = Request.Form("progprice")

txtVersion = ApiDriver.csi_util_version()

```

```

txtArray.Add "txtVersion", txtVersion

OrderCtx =   ApiDriver.csi_order_alloc()
ItemCtx =   ApiDriver.csi_item_alloc()
OptionCtx =  ApiDriver.csi_option_alloc()
ReqCtx =   ApiDriver.csi_req_alloc()
ShippingCtx = ApiDriver.csi_shipping_alloc()
TaxCtx =   ApiDriver.csi_tax_alloc()

Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Configfile, "yourstore")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Keyfile, "yourpem")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Host, "securehost")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Port, 1139)

port = ApiDriver.csi_req_get(ReqCtx, ReqField_Port)
'Response.Write("Port = " & port & "<BR>")

Flag = ApiDriver.csi_order_setrequest(OrderCtx, ReqCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_order_setrequest: " &
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Flag = ApiDriver.csi_tax_setrequest(TaxCtx, ReqCtx)
Flag = ApiDriver.csi_shipping_setrequest(ShippingCtx, ReqCtx)

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Userid, "nobody")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bname, CStr(bname))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bcompany, "ClearCommerce")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Baddr1, CStr(baddr1))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bcity, CStr(bstate))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bstate, CStr(bcity))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bzip, CStr(bzip))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bcountry, "US")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Sname, "John Q. Public")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Saddr1, "11500 Metric Blvd. Suite 300")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Scity, "Denver")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Sstate, "CO")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Szip, "80219")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Scountry, "US")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Phone, "512-832-0132")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Fax, "512-832-8901")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Comments, "No comments today")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Cardnumber, CStr(card))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_ChargeType, Chargetype_Auth)
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Expmonth, "12")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Expyear, "01")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Email, "youremail")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Result, Result_Good)

```



```

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Addrnum, "111")

price = CDbI(shirtprice)
itemtotal = numshirts * price

Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Itemid, "T-shirt 001")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Description, "T-shirt of ClearCommerce Logo")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Price, price)
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Quantity, CInt(numshirts))

Flag = ApiDriver.csi_option_set(OptionCtx, OptionField_Option, "Color")
Flag = ApiDriver.csi_option_set(OptionCtx, OptionField_Choice, "Red")

Flag = ApiDriver.csi_item_addoption(ItemCtx, OptionCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = txtError & "csi_item_addoption: " &
    ApiDriver.csi_util_errorstr(ApiDriver.csi_item_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Flag = ApiDriver.csi_option_drop(OptionCtx)

Flag = ApiDriver.csi_order_additem(OrderCtx, ItemCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = txtError & "csi_order_additem: " &
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Flag = ApiDriver.csi_item_clear(ItemCtx)

price = CDbI(progprice)
itemtotal = itemtotal + (numprogs * price)

Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Itemid, "Game 033")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Description, "Blast Em Game Software")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Softfile, "file.zip")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Serial, "")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Esdtype, Esdtype_Softgood)
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Price, price)
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Quantity, CInt(numprogs))

Flag = ApiDriver.csi_order_additem(OrderCtx, ItemCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = txtError & "csi_order_additem: " +
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing

```

```

'Exit Function
End If

Flag = ApiDriver.csi_item_drop(ItemCtx)

subtotal = itemtotal
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Subtotal, subtotal)

Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Country, "US")
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_State, "CO")
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Total, subtotal)
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Items, 2)
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Weight, 1.00)
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Carrier, 1)

Flag = ApiDriver.csi_shipping_process(ShippingCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_shipping_process: " +
    ApiDriver.csi_util_errorstr(ApiDriver.csi_shipping_error(ShippingCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

shiptotal = ApiDriver.csi_shipping_get(ShippingCtx, ShippingField_R_Total)
txtShipTotal = shiptotal

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Shipping, shiptotal)
Flag = ApiDriver.csi_shipping_drop(ShippingCtx)

price = shiptotal + subtotal

Flag = ApiDriver.csi_tax_set(TaxCtx, TaxField_State, "CO")
Flag = ApiDriver.csi_tax_set(TaxCtx, TaxField_Zip, "80219")
Flag = ApiDriver.csi_tax_set(TaxCtx, TaxField_Total, price)

Flag = ApiDriver.csi_tax_process(TaxCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_tax_process: " + ApiDriver.csi_util_errorstr(ApiDriver.csi_tax_error(TaxCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

taxtotal = ApiDriver.csi_tax_get(TaxCtx, TaxField_R_Total)
txtTax = taxtotal

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Tax, taxtotal)
Flag = ApiDriver.csi_tax_drop(TaxCtx)

```

```

total = subtotal + taxtotal + shiptotal

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Chargetotal, total)
txtChargeTotal = total

Flag = ApiDriver.csi_order_process(OrderCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_order_process: " +
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
Else
    txtTime = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Time)
    txtRef = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Ref)
    txtAppr = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Approved)
    txtCode = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Code)
    txtErr = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Error)
    txtOrd = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Ordernum)
End If

esdfiles = ApiDriver.csi_order_getesd(OrderCtx)
nLBound = LBound(esdfiles)
nUBound = UBound(esdfiles)

If ApiDriver.bStat <> Succeed Then
    txtError = "csi_order_process: " +
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
Else
    For i = nLBound to nUBound
        file = esdfiles(i)
        If (file <> "") AND (file <> "Could not create URL.") Then
            Response.Write("ESD(" & i & ") = " & file & "<BR>")
        End If
    Next
End If

' When debugging the COM object comment out the next line!
'Flag = ApiDriver.csi_order_drop(OrderCtx)
Flag = ApiDriver.csi_req_drop(ReqCtx)

```

```

txtArray.Add "txtError",txtError
txtArray.Add "txtTime",txtTime
txtArray.Add "txtRef",txtRef
txtArray.Add "txtAppr",txtAppr
txtArray.Add "txtCode",txtCode
txtArray.Add "txtErr",txtErr

```

```
txtArray.Add "txtOrd",txtOrd
txtArray.Add "txtTax",txtTax
txtArray.Add "txtChargeTotal",txtChargeTotal
txtArray.Add "txtShipTotal",txtShipTotal
Set ApiDriver = nothing
```

End Function

```
Function GetValue(txtField)
    if txtArray.Exists(txtField) then
        GetValue = txtArray.Item(txtField)
    end if
End Function
%>
```


Example 2B:

Periodic Billing File: pb_processcard.asp

```
<%  
' Created from ccapi_error.h from API 3.8.3  
Const Fail = 0  
Const Succeed = 1  
  
' Created from ccapi_client.h from API 3.8.3  
  
'Request types possible for OrderField_Chargetype  
Const Chargetype_Auth = 0  
Const Chargetype_Sale = 0  
Const Chargetype_Preauth = 1  
Const Chargetype_Postauth = 2  
Const Chargetype_Credit = 3  
Const Chargetype_Voidsale = 9  
Const Chargetype_Error = 0  
  
'Result types possible for OrderField_Result  
Const Result_Live = 0  
Const Result_Good = 1  
Const Result_Duplicate = 2  
Const Result_Decline = 3  
  
'ESD types for ItemField_Esdtype  
Const Esdtype_None = 0  
Const Esdtype_Softgood = 1  
Const Esdtype_Key = 2  
  
' OrderField_t  
Const OrderField_Oid = 0  
Const OrderField_Userid = 1  
Const OrderField_Bcompany = 2  
Const OrderField_Bcountry = 3  
Const OrderField_Bname = 4  
Const OrderField_Baddr1 = 5  
Const OrderField_Baddr2 = 6  
Const OrderField_Bcity = 7  
Const OrderField_Bstate = 8  
Const OrderField_Bzip = 9  
Const OrderField_Sname = 10  
Const OrderField_Saddr1 = 11  
Const OrderField_Saddr2 = 12  
Const OrderField_Scity = 13  
Const OrderField_Sstate = 14  
Const OrderField_Szip = 15  
Const OrderField_Scountry = 16  
Const OrderField_Phone = 17  
Const OrderField_Fax = 18
```


Const OrderField_Refer = 19
Const OrderField_Shiptype = 20
Const OrderField_Shipping = 21
Const OrderField_Tax = 22
Const OrderField_Subtotal = 23
Const OrderField_Vattax = 24
Const OrderField_Comments = 25
Const OrderField_PONumber = 26
Const OrderField_TaxExempt = 27

Const OrderField_MaxItems = 28
Const OrderField_Email = 29
Const OrderField_Cardnumber = 30
Const OrderField_Expmonth = 31
Const OrderField_Expyear = 32
Const OrderField_Chargetype = 33
Const OrderField_Chargetotal = 34
Const OrderField_Referencenumber = 35
Const OrderField_Result = 36
Const OrderField_Addrnum = 37
Const OrderField_Ip = 38
Const OrderField_MercType = 39
Const OrderField_SwitchStartDate = 40
Const OrderField_SwitchIssueNumber = 41

' Responses

Const OrderField_R_Time = 42
Const OrderField_R_Ref = 43
Const OrderField_R_Approved = 44
Const OrderField_R_Code = 45
Const OrderField_R_Ordernum = 46
Const OrderField_R_Error = 47
Const OrderField_R_FraudCode = 48
Const OrderField_R_AuthResponse = 49
Const OrderField_R_Message = 50

' ReqField_t

Const ReqField_Configfile = 0
Const ReqField_Keyfile = 1
Const ReqField_Appname = 2
Const ReqField_Host = 3
Const ReqField_Port = 4

' ItemField_t

Const ItemField_Itemid = 0
Const ItemField_Description = 1
Const ItemField_Price = 2
Const ItemField_Quantity = 3
Const ItemField_Softfile = 4
Const ItemField_Esdtype = 5
Const ItemField_Serial = 6
Const ItemField_MaxOptions = 7

```

' ShippingField_t
Const ShippingField_Country = 0
Const ShippingField_State = 1
Const ShippingField_Total = 2
Const ShippingField_Items = 3
Const ShippingField_Weight = 4
Const ShippingField_Carrier = 5
' Responses
Const ShippingField_R_Total = 6

' TaxField_t
Const TaxField_State = 0
Const TaxField_Zip = 1
Const TaxField_Total = 2
' Responses
Const TaxField_R_Total = 3

' OptionField_t
Const OptionField_Option = 0
Const OptionField_Choice = 1

' PbOrder_t
Const PbOrder_Submit = 0
Const PbOrder_Modify = 1
Const PbOrder_Cancel = 2
Const PbOrder_Process = 3

' PeriodicField_t
Const PeriodicField_Startdate = 0
Const PeriodicField_Periodicity = 1
Const PeriodicField_Installments = 2
Const PeriodicField_Threshold = 3
Const PeriodicField_Comments = 4

' CheckOrder_t
Const CheckOrder_Submit = 0
Const CheckOrder_Void = 1

' CheckField_t;
Const CheckField_AccountType = 0
Const CheckField_TransitRouting = 1
Const CheckField_MICR = 2
Const CheckField_CheckNumber = 3
Const CheckField_Comments = 4

' Opaque contexts
Dim OrderCtx
Dim ItemCtx
Dim OptionCtx
Dim ReqCtx
Dim ShippingCtx

```

```

Dim TaxCtx
Dim ApiDriver

Set txtArray = Server.CreateObject("Scripting.Dictionary")

'-----
' Functions
'-----
Function ProcessCard()

On Error Resume Next

' Local variable declarations
Dim txtVersion
Dim port
Dim price
Dim total
Dim itemtotal
Dim subtotal
Dim taxtotal
Dim esdfiles
Dim Periodicity
Dim Installments
Dim Threshold
Dim pbtype

Err.Clear
Set ApiDriver = Server.CreateObject("LpiCom_5_4.LinkPointCom")
if (Err.Number <> 0) Then
    Response.Write("Error creating object: " & Err.Description & "<BR>")
    Exit Function
End if

Dim bname
Dim card
Dim numShirts
Dim numProgs

bname = Request.Form("bname")
card = Request.Form("creditcard")
numshirts = Request.Form("numShirts")
numprogs = Request.Form("numProg")

Periodicity = Request.Form("periodicity")
Installments = Request.Form("installments")
Threshold = Request.Form("threshold")

txtVersion = ApiDriver.csi_util_version()
txtArray.Add "txtVersion", txtVersion

```

```

OrderCtx =    ApiDriver.csi_order_alloc()
ItemCtx =    ApiDriver.csi_item_alloc()
OptionCtx =  ApiDriver.csi_option_alloc()
ReqCtx =    ApiDriver.csi_req_alloc()
ShippingCtx = ApiDriver.csi_shipping_alloc()
TaxCtx =    ApiDriver.csi_tax_alloc()

PeriodicCtx = ApiDriver.csi_periodic_alloc()
PborderCtx = ApiDriver.csi_pborder_alloc()

Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Configfile, "yourstore")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Keyfile, "yourpem")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Host, "securehost")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Port, 1139)

port = ApiDriver.csi_req_get(ReqCtx, ReqField_Port)
'Response.Write("Port = " & port & "<BR>")

Flag = ApiDriver.csi_order_setrequest(OrderCtx, ReqCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_order_setrequest: " &
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Flag = ApiDriver.csi_periodic_set(PeriodicCtx, PeriodicField_Startdate, "20001207")
Flag = ApiDriver.csi_periodic_set(PeriodicCtx, PeriodicField_Periodicity, CStr(Periodicity))
Flag = ApiDriver.csi_periodic_set(PeriodicCtx, PeriodicField_Installments, CInt(Installments))
Flag = ApiDriver.csi_periodic_set(PeriodicCtx, PeriodicField_Threshold, CInt(Threshold))
Flag = ApiDriver.csi_periodic_set(PeriodicCtx, PeriodicField_Comments, "This is pbill test")

Flag = ApiDriver.csi_tax_setrequest(TaxCtx, ReqCtx)
Flag = ApiDriver.csi_shipping_setrequest(ShippingCtx, ReqCtx)

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Userid, "nobody")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bname, CStr(bname))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bcompany, "ClearCommerce")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Baddr1, "11500 Metric Blvd. Suite 300")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bcity, "Denver")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bstate, "CO")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bzip, "80219")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bcountry, "US")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Sname, "John Q. Public")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Saddr1, "11500 Metric Blvd. Suite 300")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Scity, "Denver")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Sstate, "CO")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Szip, "80219")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Scountry, "US")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Phone, "512-832-0132")

```

```

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Fax, "512-832-8901")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Comments, "No comments today")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Cardnumber, CStr(card))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_ChargeType, Chargetype_Auth)
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Expmonth, "12")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Expyear, "01")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Email, "youremail")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Result, Result_Good)
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Addrnum, "111")

price = 17.95
itemtotal = numshirts * price

Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Itemid, "T-shirt 001")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Description, "T-shirt of ClearCommerce Logo")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Price, price)
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Quantity, CInt(numshirts))

Flag = ApiDriver.csi_option_set(OptionCtx, OptionField_Option, "Color")
Flag = ApiDriver.csi_option_set(OptionCtx, OptionField_Choice, "Red")

Flag = ApiDriver.csi_item_adoption(ItemCtx, OptionCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = txtError & "csi_item_adoption: " &
    ApiDriver.csi_util_errorstr(ApiDriver.csi_item_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Flag = ApiDriver.csi_option_drop(OptionCtx)

Flag = ApiDriver.csi_order_additem(OrderCtx, ItemCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = txtError & "csi_order_additem: " &
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))

txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Flag = ApiDriver.csi_item_clear(ItemCtx)

price = 14.50
itemtotal = itemtotal + (numprogs * price)

Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Itemid, "Game 033")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Description, "Blast Em Game Software")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Softfile, "file.zip")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Serial, "")

```

```
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Esdtype, Esdtype_Softgood)
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Price, price)
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Quantity, CInt(numprogs))
```

```
Flag = ApiDriver.csi_order_additem(OrderCtx, ItemCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = txtError & "csi_order_additem: " +
ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If
```

```
Flag = ApiDriver.csi_item_drop(ItemCtx)
```

```
subtotal = itemtotal
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Subtotal, subtotal)
```

```
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Country, "US")
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_State, "CO")
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Total, subtotal)
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Items, 2)
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Weight, 1.00)
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Carrier, 1)
```

```
Flag = ApiDriver.csi_shipping_process(ShippingCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_shipping_process: " +
ApiDriver.csi_util_errorstr(ApiDriver.csi_shipping_error(ShippingCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If
```

```
shiptotal = ApiDriver.csi_shipping_get(ShippingCtx, ShippingField_R_Total)
txtShipTotal = shiptotal
```

```
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Shipping, shiptotal)
Flag = ApiDriver.csi_shipping_drop(ShippingCtx)
```

```
price = shiptotal + subtotal
```

```
Flag = ApiDriver.csi_tax_set(TaxCtx, TaxField_State, "CO")
Flag = ApiDriver.csi_tax_set(TaxCtx, TaxField_Zip, "80219")
Flag = ApiDriver.csi_tax_set(TaxCtx, TaxField_Total, price)
```

```
Flag = ApiDriver.csi_tax_process(TaxCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_tax_process: " + ApiDriver.csi_util_errorstr(ApiDriver.csi_tax_error(TaxCtx))
    txtArray.Add "txtError",txtError
```

```

    'Set ApiDriver = nothing
    'Exit Function
End If

taxtotal = ApiDriver.csi_tax_get(TaxCtx, TaxField_R_Total)
txtTax = taxtotal

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Tax, taxtotal)
Flag = ApiDriver.csi_tax_drop(TaxCtx)

total = subtotal + taxtotal + shiptotal

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Chargetotal, total)
txtChargeTotal = total

Flag = ApiDriver.csi_pborder_setorder(PborderCtx, OrderCtx)
If ApiDriver.bStat <> Succeed Then
    txtError = "csi_pborder_setorder: " +
    ApiDriver.csi_util_pborder_errorstr(ApiDriver.csi_pborder_error(PborderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Flag = ApiDriver.csi_pborder_setperiodic(PborderCtx, PeriodicCtx)
If ApiDriver.bStat <> Succeed Then
    txtError = "csi_pborder_setperiodic: " +
    ApiDriver.csi_util_pborder_errorstr(ApiDriver.csi_pborder_error(PborderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

pbtype = PbOrder_Submit

'pbtype = PbOrder_Cancel
'pbtype = PbOrder_Modify
'pbtype = PbOrder_Process

Flag = ApiDriver.csi_pborder_settype(PborderCtx, CInt(pbtype))
If ApiDriver.bStat <> Succeed Then
    txtError = "csi_pborder_settype: " +
    ApiDriver.csi_util_pborder_errorstr(ApiDriver.csi_pborder_error(PborderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Flag = ApiDriver.csi_pborder_process(PborderCtx)
if ApiDriver.bStat <> Succeed Then

```

```

    txtError = "csi_pborder_process: " +
    ApiDriver.csi_util_pborder_errorstr(ApiDriver.csi_pborder_error(PborderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
Else
    txtTime = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Time)
    txtRef = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Ref)
    txtAppr = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Approved)
    txtCode = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Code)
    txtErr = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Error)
    txtOrd = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Ordernum)
End If

esdfiles = ApiDriver.csi_order_getesd(OrderCtx)
nLBound = LBound(esdfiles)
nUBound = UBound(esdfiles)

If ApiDriver.bStat <> Succeed Then
    txtError = "csi_order_process: " +
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
Else
    For i = nLBound to nUBound
        file = esdfiles(i)
        If (file <> "") AND (file <> "Could not create URL.") Then
            Response.Write("ESD(" & i & ") = " & file & "<BR>")
        End If
    Next
End If

' When debugging the COM object comment out the next line!
'Flag = ApiDriver.csi_order_drop(OrderCtx)
Flag = ApiDriver.csi_req_drop(ReqCtx)

txtArray.Add "txtError",txtError
txtArray.Add "txtTime",txtTime
txtArray.Add "txtRef",txtRef
txtArray.Add "txtAppr",txtAppr
txtArray.Add "txtCode",txtCode
txtArray.Add "txtErr",txtErr
txtArray.Add "txtOrd",txtOrd
txtArray.Add "txtTax",txtTax
txtArray.Add "txtChargeTotal",txtChargeTotal
txtArray.Add "txtShipTotal",txtShipTotal
Set ApiDriver = nothing

End Function

Function GetValue(txtField)

```



```
if txtArray.Exists(txtField) then
    GetValue = txtArray.Item(txtField)
end if
End Function
%>
```


Example 3B:

VirtualCheck File: vc_processcard.asp

```
<%  
' Created from ccapi_error.h from API 3.8.3  
Const Fail = 0  
Const Succeed = 1  
  
' Created from ccapi_client.h from API 3.8.3  
  
'Request types possible for OrderField_Chargetype  
Const Chargetype_Auth = 0  
Const Chargetype_Sale = 0  
Const Chargetype_Preauth = 1  
Const Chargetype_Postauth = 2  
Const Chargetype_Credit = 3  
Const Chargetype_Voidsale = 9  
Const Chargetype_Error = 0  
  
'Result types possible for OrderField_Result  
Const Result_Live = 0  
Const Result_Good = 1  
Const Result_Duplicate = 2  
Const Result_Decline = 3  
  
'ESD types for ItemField_Esdtype  
Const Esdtype_None = 0  
Const Esdtype_Softgood = 1  
Const Esdtype_Key = 2  
  
' OrderField_t  
Const OrderField_Oid = 0  
Const OrderField_Userid = 1  
Const OrderField_Bcompany = 2  
Const OrderField_Bcountry = 3  
Const OrderField_Bname = 4  
Const OrderField_Baddr1 = 5  
Const OrderField_Baddr2 = 6  
Const OrderField_Bcity = 7  
Const OrderField_Bstate = 8  
Const OrderField_Bzip = 9  
Const OrderField_Sname = 10  
Const OrderField_Saddr1 = 11  
Const OrderField_Saddr2 = 12  
Const OrderField_Scity = 13  
Const OrderField_Sstate = 14  
Const OrderField_Szip = 15  
Const OrderField_Scountry = 16  
Const OrderField_Phone = 17  
Const OrderField_Fax = 18
```

Const OrderField_Refer = 19
Const OrderField_Shiptype = 20
Const OrderField_Shipping = 21
Const OrderField_Tax = 22
Const OrderField_Subtotal = 23
Const OrderField_Vattax = 24
Const OrderField_Comments = 25
Const OrderField_PONumber = 26
Const OrderField_TaxExempt = 27
Const OrderField_MaxItems = 28
Const OrderField_Email = 29
Const OrderField_Cardnumber = 30
Const OrderField_Expmonth = 31
Const OrderField_Expyear = 32
Const OrderField_Chargetype = 33
Const OrderField_Chargetotal = 34
Const OrderField_Referencenumber = 35
Const OrderField_Result = 36
Const OrderField_Addrnum = 37
Const OrderField_Ip = 38
Const OrderField_MercType = 39
Const OrderField_SwitchStartDate = 40
Const OrderField_SwitchIssueNumber = 41

' Responses

Const OrderField_R_Time = 42
Const OrderField_R_Ref = 43
Const OrderField_R_Approved = 44
Const OrderField_R_Code = 45
Const OrderField_R_Ordernum = 46
Const OrderField_R_Error = 47
Const OrderField_R_FraudCode = 48
Const OrderField_R_AuthResponse = 49
Const OrderField_R_Message = 50

' ReqField_t

Const ReqField_Configfile = 0
Const ReqField_Keyfile = 1
Const ReqField_Appname = 2
Const ReqField_Host = 3
Const ReqField_Port = 4

' ItemField_t

Const ItemField_Itemid = 0
Const ItemField_Description = 1
Const ItemField_Price = 2
Const ItemField_Quantity = 3
Const ItemField_Softfile = 4
Const ItemField_Esdtype = 5
Const ItemField_Serial = 6
Const ItemField_MaxOptions = 7

```

' ShippingField_t
Const ShippingField_Country = 0
Const ShippingField_State = 1
Const ShippingField_Total = 2
Const ShippingField_Items = 3
Const ShippingField_Weight = 4
Const ShippingField_Carrier = 5

' Responses
Const ShippingField_R_Total = 6

' TaxField_t
Const TaxField_State = 0
Const TaxField_Zip = 1
Const TaxField_Total = 2
' Responses
Const TaxField_R_Total = 3

' OptionField_t
Const OptionField_Option = 0
Const OptionField_Choice = 1

' PbOrder_t
Const PbOrder_Submit = 0
Const PbOrder_Modify = 1
Const PbOrder_Cancel = 2
Const PbOrder_Process = 3

' PeriodicField_t
Const PeriodicField_Startdate = 0
Const PeriodicField_Periodicity = 1
Const PeriodicField_Installments = 2
Const PeriodicField_Threshold = 3
Const PeriodicField_Comments = 4

' CheckOrder_t
Const CheckOrder_Submit = 0
Const CheckOrder_Void = 1

' CheckField_t;
Const CheckField_AccountType = 0
Const CheckField_TransitRouting = 1
Const CheckField_MICR = 2
Const CheckField_CheckNumber = 3
Const CheckField_Comments = 4

' Opaque contexts
Dim OrderCtx
Dim ItemCtx
Dim OptionCtx
Dim ReqCtx
Dim ShippingCtx

```

```

Dim TaxCtx
Dim CheckCtx
Dim CheckorderCtx
Dim ApiDriver

Set txtArray = Server.CreateObject("Scripting.Dictionary")

'-----
' Functions
'-----
Function ProcessCard()

On Error Resume Next

' Local variable declarations
Dim txtVersion
Dim port
Dim price
Dim total
Dim itemtotal
Dim subtotal
Dim taxtotal
Dim esdfiles
Dim CheckorderType

Err.Clear
Set ApiDriver = Server.CreateObject("LpiCom_5_4.LinkPointCom")
if (Err.Number <> 0) Then
    Response.Write("Error creating object: " & Err.Description & "<BR>")
    Exit Function
End if

Dim bname
Dim numShirts
Dim numProgs
Dim accounttype
Dim checkno
Dim route
Dim micr

bname = Request.Form("bname")
numshirts = Request.Form("numShirts")
numprogs = Request.Form("numProg")

accounttype = Request.Form("accounttype")
checkno = Request.Form("checkno")
route = Request.Form("route")
micr = Request.Form("micr")

txtVersion = ApiDriver.csi_util_version()
txtArray.Add "txtVersion", txtVersion

```



```

OrderCtx =    ApiDriver.csi_order_alloc()
ItemCtx =    ApiDriver.csi_item_alloc()
OptionCtx =  ApiDriver.csi_option_alloc()
ReqCtx =     ApiDriver.csi_req_alloc()
ShippingCtx = ApiDriver.csi_shipping_alloc()
TaxCtx =     ApiDriver.csi_tax_alloc()

CheckCtx = ApiDriver.csi_check_alloc()
CheckorderCtx = ApiDriver.csi_checkorder_alloc()

Flag = ApiDriver.csi_check_set(CheckCtx, CheckField_AccountType, CStr(accounttype))
Flag = ApiDriver.csi_check_set(CheckCtx, CheckField_TransitRouting, CStr(route))
Flag = ApiDriver.csi_check_set(CheckCtx, CheckField_MICR, CStr(micr))
Flag = ApiDriver.csi_check_set(CheckCtx, CheckField_CheckNumber, CInt(checkno))

Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Configfile, "yourstore")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Keyfile, "yourpem")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Host, "securehost")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Port, 1139)

port = ApiDriver.csi_req_get(ReqCtx, ReqField_Port)
'Response.Write("Port = " & port & "<BR>")

Flag = ApiDriver.csi_order_setrequest(OrderCtx, ReqCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_order_setrequest: " &
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Installments=12
Threshold=3
Flag = ApiDriver.csi_periodic_set(PeriodicCtx, PeriodicField_Startdate, "20001207")
Flag = ApiDriver.csi_periodic_set(PeriodicCtx, PeriodicField_Periodicity, "m1")
Flag = ApiDriver.csi_periodic_set(PeriodicCtx, PeriodicField_Installments, CInt(Installments))
Flag = ApiDriver.csi_periodic_set(PeriodicCtx, PeriodicField_Threshold, CInt(Threshold))
Flag = ApiDriver.csi_periodic_set(PeriodicCtx, PeriodicField_Comments, "This is pbill test")

Flag = ApiDriver.csi_tax_setrequest(TaxCtx, ReqCtx)
Flag = ApiDriver.csi_shipping_setrequest(ShippingCtx, ReqCtx)

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Userid, "nobody")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bname, CStr(bname))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bcompany, "ClearCommerce")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Baddr1, "11500 Metric Blvd. Suite 300")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bcity, "Denver")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bstate, "CO")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bzip, "80219")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bcountry, "US")

```

```

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Sname, "John Q. Public")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Saddr1, "11500 Metric Blvd. Suite 300")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Scity, "Denver")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Sstate, "CO")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Szip, "80219")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Scountry, "US")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Phone, "512-832-0132")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Fax, "512-832-8901")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Comments, "No comments today")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_ChargeType, Chargetype_Auth)
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Email, "youremail")
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Result, Result_Good)
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Addrnum, "111")

```

```

price = 17.95
itemtotal = numshirts * price

```

```

Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Itemid, "T-shirt 001")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Description, "T-shirt of ClearCommerce Logo")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Price, price)
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Quantity, CInt(numshirts))

```

```

Flag = ApiDriver.csi_option_set(OptionCtx, OptionField_Option, "Color")
Flag = ApiDriver.csi_option_set(OptionCtx, OptionField_Choice, "Red")

```

```

Flag = ApiDriver.csi_item_adoption(ItemCtx, OptionCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = txtError & "csi_item_adoption: " &
    ApiDriver.csi_util_errorstr(ApiDriver.csi_item_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

```

```

Flag = ApiDriver.csi_option_drop(OptionCtx)

```

```

Flag = ApiDriver.csi_order_additem(OrderCtx, ItemCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = txtError & "csi_order_additem: " &
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

```

```

Flag = ApiDriver.csi_item_clear(ItemCtx)

```

```

price = 14.50
itemtotal = itemtotal + (numprogs * price)

```

```

Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Itemid, "Game 033")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Description, "Blast Em Game Software")

```

```

Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Softfile, "file.zip")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Serial, "")
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Esdtype, Esdtype_Softgood)
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Price, price)
Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Quantity, CInt(numprogs))

Flag = ApiDriver.csi_order_additem(OrderCtx, ItemCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = txtError & "csi_order_additem: " +
ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Flag = ApiDriver.csi_item_drop(ItemCtx)

subtotal = itemtotal
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Subtotal, subtotal)

Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Country, "US")
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_State, "CO")
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Total, subtotal)
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Items, 2)
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Weight, 1.00)
Flag = ApiDriver.csi_shipping_set(ShippingCtx, ShippingField_Carrier, 1)

Flag = ApiDriver.csi_shipping_process(ShippingCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_shipping_process: " +
ApiDriver.csi_util_errorstr(ApiDriver.csi_shipping_error(ShippingCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

shiptotal = ApiDriver.csi_shipping_get(ShippingCtx, ShippingField_R_Total)
txtShipTotal = shiptotal

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Shipping, shiptotal)
Flag = ApiDriver.csi_shipping_drop(ShippingCtx)

price = shiptotal + subtotal

Flag = ApiDriver.csi_tax_set(TaxCtx, TaxField_State, "CO")
Flag = ApiDriver.csi_tax_set(TaxCtx, TaxField_Zip, "80219")
Flag = ApiDriver.csi_tax_set(TaxCtx, TaxField_Total, price)

Flag = ApiDriver.csi_tax_process(TaxCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_tax_process: " + ApiDriver.csi_util_errorstr(ApiDriver.csi_tax_error(TaxCtx))
    txtArray.Add "txtError",txtError

```

```

    'Set ApiDriver = nothing
    'Exit Function
End If

taxtotal = ApiDriver.csi_tax_get(TaxCtx, TaxField_R_Total)
txtTax = taxtotal

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Tax, taxtotal)
Flag = ApiDriver.csi_tax_drop(TaxCtx)

total = subtotal + taxtotal + shiptotal

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Chargetotal, total)
txtChargeTotal = total

Flag = ApiDriver.csi_checkorder_setorder(CheckorderCtx, OrderCtx)
If ApiDriver.bStat <> Succeed Then
    txtError = "csi_checkorder_setorder: " +
    ApiDriver.csi_util_checkorder_errorstr(ApiDriver.csi_checkorder_error(CheckorderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Flag = ApiDriver.csi_checkorder_setcheck(CheckorderCtx, CheckCtx)
If ApiDriver.bStat <> Succeed Then
    txtError = "csi_checkorder_setcheck: " +
    ApiDriver.csi_util_checkorder_errorstr(ApiDriver.csi_checkorder_error(CheckorderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

CheckorderType = CheckOrder_Submit
'CheckorderType = CheckOrder_Void

Flag = ApiDriver.csi_checkorder_settype(CheckorderCtx, CInt(CheckorderType))
If ApiDriver.bStat <> Succeed Then
    txtError = "csi_checkorder_settype: " +
    ApiDriver.csi_util_checkorder_errorstr(ApiDriver.csi_checkorder_error(CheckorderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
End If

Flag = ApiDriver.csi_checkorder_process(CheckorderCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_checkorder_process: " +
    ApiDriver.csi_util_checkorder_errorstr(ApiDriver.csi_checkorder_error(CheckorderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function

```

```

Else
    txtTime = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Time)
    txtRef = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Ref)
    txtAppr = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Approved)
    txtCode = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Code)

    txtErr = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Error)
    txtOrd = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Ordernum)
End If

esdfiles = ApiDriver.csi_order_getesd(OrderCtx)
nLBound = LBound(esdfiles)
nUBound = UBound(esdfiles)

If ApiDriver.bStat <> Succeed Then
    txtError = "csi_order_process: " +
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    txtArray.Add "txtError",txtError
    'Set ApiDriver = nothing
    'Exit Function
Else
    For i = nLBound to nUBound
        file = esdfiles(i)
        If (file <> "") AND (file <> "Could not create URL.") Then
            Response.Write("ESD(" & i & ") = " & file & "<BR>")
        End If
    Next
End If

' When debugging the COM object comment out the next line!
' Flag = ApiDriver.csi_order_drop(OrderCtx)
Flag = ApiDriver.csi_req_drop(ReqCtx)

txtArray.Add "txtError",txtError
txtArray.Add "txtTime",txtTime
txtArray.Add "txtRef",txtRef
txtArray.Add "txtAppr",txtAppr
txtArray.Add "txtCode",txtCode
txtArray.Add "txtErr",txtErr
txtArray.Add "txtOrd",txtOrd
txtArray.Add "txtTax",txtTax
txtArray.Add "txtChargeTotal",txtChargeTotal
txtArray.Add "txtShipTotal",txtShipTotal
Set ApiDriver = nothing

End Function

Function GetValue(txtField)
    if txtArray.Exists(txtField) then
        GetValue = txtArray.Item(txtField)
    end if
End Function

```

End Function
%>

SITE SERVER COMMERCE EDITION PAYMENT COMPONENT SAMPLE

To use this sample, the VB Script will have to be edited in the area of line-item processing due to the fact that line items, including attributes, are particular to different stores. This VB Script file will then have to be added to the order payment pipeline (OPP) as a scriptor component in the payment section of the order pipeline.

```
*****
```

```
' ProcessCard Pipeline Script
' This script processes credit cards from a
' Site Server Commerce Edition order
' processing pipeline
*****
```

```
Sub MSCSOpen(config)
```

```
    ' Initialization code goes here
    ' Not used in this script...no startup code
```

```
End Sub
```

```
Sub MSCSClose()
```

```
    ' Clean up code goes here
    ' Not used in this script...no cleanup code
```

```
End Sub
```

```
Function MSCSExecute(config, orderform, context, flags)
```

```
*****
```

```
' This function is called by the Purchase pipeline,
' which passes the following objects:
'
'     config:      a dictionary containing custom parameters
'     orderform:  the order form object
'     context:    a dictionary containing references to
'                 objects and data that the site stores in
'                 the Application object
'     flags:      Reserved. This parameter should be zero
*****
```

```
On Error Resume Next
```

```
' Created from ccapi_error.h from API 3.8
Const Fail = 0
Const Succeed = 1
```

```
'Request types possible for OrderField_Chargetype
Const Chargetype_Auth = 0
```

Const Chargetype_Sale = 0
Const Chargetype_Preauth = 1
Const Chargetype_Postauth = 2
Const Chargetype_Credit = 3
Const Chargetype_Error = 0

'Result types possible for OrderField_Result
Const Result_Live = 0
Const Result_Good = 1
Const Result_Duplicate = 2
Const Result_Decline = 3

'ESD types for ItemField_Esdtype
Const Esdtype_None = 0
Const Esdtype_Softgood = 1
Const Esdtype_Key = 2

' OrderField_t
Const OrderField_Userid = 1
Const OrderField_Bcompany = 2
Const OrderField_Bcountry = 3
Const OrderField_Bname = 4
Const OrderField_Baddr1 = 5
Const OrderField_Baddr2 = 6
Const OrderField_Bcity = 7
Const OrderField_Bstate = 8
Const OrderField_Bzip = 9
Const OrderField_Sname = 10
Const OrderField_Saddr1 = 11
Const OrderField_Saddr2 = 12
Const OrderField_Scity = 13
Const OrderField_Sstate = 14
Const OrderField_Szip = 15
Const OrderField_Scountry = 16
Const OrderField_Phone = 17
Const OrderField_Fax = 18
Const OrderField_Refer = 19
Const OrderField_Shiptype = 20
Const OrderField_Shipping = 21
Const OrderField_Tax = 22
Const OrderField_Subtotal = 23
Const OrderField_Vattax = 24
Const OrderField_Comments = 25
Const OrderField_PONumber = 26
Const OrderField_TaxExempt = 27
Const OrderField_Oid = 0
Const OrderField_MaxItems = 28
Const OrderField_Email = 29
Const OrderField_Cardnumber = 30
Const OrderField_Expmonth = 31
Const OrderField_Expyear = 32
Const OrderField_Chargetype = 33

Const OrderField_Chargetotal = 34
Const OrderField_Referencenumber = 35
Const OrderField_Result = 36
Const OrderField_Addrnum = 37
Const OrderField_Ip = 38
Const OrderField_MercType = 39
Const OrderField_SwitchStartDate = 40
Const OrderField_SwitchIssueNumber = 41

' Responses

Const OrderField_R_Time = 42
Const OrderField_R_Ref = 43
Const OrderField_R_Approved = 44
Const OrderField_R_Code = 45
Const OrderField_R_Ordernum = 46
Const OrderField_R_Error = 47
Const OrderField_R_FraudCode = 48
Const OrderField_R_AuthResponse = 49
Const OrderField_R_Message = 50

' ReqField_t

Const ReqField_Configfile = 0
Const ReqField_Keyfile = 1
Const ReqField_Appname = 2
Const ReqField_Host = 3
Const ReqField_Port = 4

' ItemField_t

Const ItemField_Itemid = 0
Const ItemField_Description = 1
Const ItemField_Price = 2
Const ItemField_Quantity = 3
Const ItemField_Softfile = 4
Const ItemField_Esdtype = 5
Const ItemField_Serial = 6
Const ItemField_MaxOptions = 7

' ShippingField_t

Const ShippingField_Country = 0
Const ShippingField_State = 1
Const ShippingField_Total = 2
Const ShippingField_Items = 3
Const ShippingField_Weight = 4
Const ShippingField_Carrier = 5

' Responses

Const ShippingField_R_Total = 6

' TaxField_t

Const TaxField_State = 0
Const TaxField_Zip = 1
Const TaxField_Total = 2

' Responses

```

Const TaxField_R_Total = 3

' OptionField_t
Const OptionField_Option = 0
Const OptionField_Choice = 1

' Opaque contexts
Dim OrderCtx
Dim ItemCtx
Dim OptionCtx
Dim ReqCtx
Dim ApiDriver

' Local variable declarations
Dim price
Dim total
Dim itemprice
Dim subtotal
Dim taxtotal
Dim expyear

Dim txtVersion
Dim txtTime
Dim txtRef
Dim txtAppr
Dim txtCode
Dim txtErr
Dim txtError
Dim txtOrd

txtAppr = ""
subtotal = 0.0

*****
' Instantiate the LinkPoint COM object
*****
Set ApiDriver = CreateObject("LpiCom_5_4.LinkPointCom")

txtVersion = ApiDriver.csi_util_version()

OrderCtx =   ApiDriver.csi_order_alloc()
ItemCtx =   ApiDriver.csi_item_alloc()
OptionCtx = ApiDriver.csi_option_alloc()
ReqCtx =   ApiDriver.csi_req_alloc()

*****
' Set merchant specific parameters
*****
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Configfile, "yourstore")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Keyfile, "yourpem")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Host, "securehost")
Flag = ApiDriver.csi_req_set(ReqCtx, ReqField_Port, 1139)

```

```

Flag = ApiDriver.csi_order_setrequest(OrderCtx, ReqCtx)
if ApiDriver.bStat <> Succeed Then
    txtError = "csi_order_setrequest: " &
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    orderform.[_purchase_errors].Add(txtError)
End If

*****
' Get Billing and Shipping info
*****
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Userid, CStr(orderform.shopper_ID))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bname, CStr(orderform.bill_to_name))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Baddr1, CStr(orderform.bill_to_street))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bcity, CStr(orderform.bill_to_city))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bstate, CStr(orderform.bill_to_state))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bzip, CStr(orderform.bill_to_zip))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Bcountry, CStr(orderform.bill_to_country))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Sname, CStr(orderform.ship_to_name))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Saddr1, CStr(orderform.ship_to_street))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Scity, CStr(orderform.ship_to_city))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Sstate, CStr(orderform.ship_to_state))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Szip, CStr(orderform.ship_to_zip))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Scountry, CStr(orderform.ship_to_country))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Phone, CStr(orderform.ship_to_phone))

'Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Comments, "No comments today")

*****
' Set the ChargeType to Auth/Sale
*****
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_ChargeType, Chargetype_Auth)

*****
' Get Credit Card info
*****
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Cardnumber, CStr(orderform.[_cc_number]))
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Expmonth, CStr(orderform.[_cc_expmonth]))

' Extract a 2 digit year
expyear = orderform.[_cc_expyear]
expyear = Mid(expyear, 2)
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Expyear, CStr(expyear))

*****
' Get the Customer e-mail address for receipt
*****
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Email, CStr(orderform.[_shopper_email]))

*****
' For testing set for Result_Good. To go live, set to Result_Live
*****

```

```

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Result, Result_Good)

*****
' Process each item in the orderform.items list. This will need to be tailored for each
' application since the handling of different types of items is different for each store.
*****
For each item in orderform.items

    Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Itemid, CStr(item.SKU))
    Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Description, CStr(item.[_product_name]))

    itemprice = item.[_iadjust_currentprice] / 100.0
    Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Price, CDbI(itemprice))
    Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Quantity, CInt(item.Quantity))

    *****
    ' For T-shirts, get the attributes size and color. Else it's a program. Setup
    ' for a ESD SoftGood download.
    *****
    If item.SKU <> "555" Then
        Flag = ApiDriver.csi_option_set(OptionCtx, OptionField_Option, "Color")
        Flag = ApiDriver.csi_option_set(OptionCtx, OptionField_Choice, CStr(item.attr_value2))

        Flag = ApiDriver.csi_item_addoption(ItemCtx, OptionCtx)
        If ApiDriver.bStat <> Succeed Then
            txtError = txtError & "csi_item_addoption: " &
ApiDriver.csi_util_errorstr(ApiDriver.csi_item_error(OrderCtx))
            orderform.[_purchase_errors].Add(txtError)
        End If

        Flag = ApiDriver.csi_option_set(OptionCtx, OptionField_Option, "Size")
        Flag = ApiDriver.csi_option_set(OptionCtx, OptionField_Choice, CStr(item.attr_value1))

        Flag = ApiDriver.csi_item_addoption(ItemCtx, OptionCtx)
        If ApiDriver.bStat <> Succeed Then
            txtError = txtError & "csi_item_addoption: " &
ApiDriver.csi_util_errorstr(ApiDriver.csi_item_error(OrderCtx))
            orderform.[_purchase_errors].Add(txtError)
        End If
    Else
        Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Softfile, CStr(item.[_product_name]))
        Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Serial, "")
        Flag = ApiDriver.csi_item_set(ItemCtx, ItemField_Esdtype, Esdtype_Softgood)
    End If

    Flag = ApiDriver.csi_order_additem(OrderCtx, ItemCtx)
    if ApiDriver.bStat <> Succeed Then
        txtError = txtError & "csi_order_additem: " &
ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
        orderform.[_purchase_errors].Add(txtError)
    End If

```

```

Flag = ApiDriver.csi_item_clear(ItemCtx)

*****
' Accumulate the subtotal for all items
*****
subtotal = subtotal + (item.Quantity * itemprice)

Next

Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Subtotal, CDbI(subtotal))

*****
' Get Shipping and Handling
*****
fltShip = orderform.[_shipping_total] + orderform.[_handling_total]
fltShip = fltShip / 100.00
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Shipping, CDbI(fltShip))

*****
' Get Tax
*****
fltTax = orderform.[_tax_total]
fltTax = fltTax / 100.00
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Tax, CDbI(fltTax))

*****
' Calculate the ChargeTotal
*****
total = subtotal + fltTax + fltShip
Flag = ApiDriver.csi_order_set(OrderCtx, OrderField_Chargetotal, CDbI(total))

*****
' Finally, process the order through the CSI Gateway
*****
Flag = ApiDriver.csi_order_process(OrderCtx)
If ApiDriver.bStat <> Succeed Then
    txtError = "csi_order_process: " +
    ApiDriver.csi_util_errorstr(ApiDriver.csi_order_error(OrderCtx))
    orderform.[_purchase_errors].Add(txtError)
Else
    txtTime = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Time)
    txtRef = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Ref)
    txtAppr = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Approved)
    txtCode = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Code)
    txtErr = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Error)
    txtOrd = ApiDriver.csi_order_get(OrderCtx, OrderField_R_Ordernum)
End If

Flag = ApiDriver.csi_item_drop(ItemCtx)
Flag = ApiDriver.csi_option_drop(OptionCtx)
Flag = ApiDriver.csi_order_drop(OrderCtx)
Flag = ApiDriver.csi_req_drop(ReqCtx)

```

Set ApiDriver = nothing

```
*****  
' Return the appropriate status  
*****  
If txtAppr = "APPROVED" Then  
    MSCSExecute = 1  
    orderform.[_payment_auth_code] = txtAppr  
Else  
    MSCSExecute = 2  
    orderform.[_purchase_errors].Add("Error: " & txtErr)  
    orderform.[_payment_auth_code] = "DENIED"  
End If  
End Function
```

APPENDIX A

Deliverables

After your purchase of the LCOM wrapper, a file containing all deliverables will be sent to you by e-mail.

LCOM is in the following format:

- Windows NT platforms a Zip file will be delivered

The content of the LCOM zip file is:

1. LinkPoint COM Object

- LpiCom_5_4.dll

2. Readme File

- ComReadme.txt

3. ASP Samples

- global.asa
- api_ordertest.asp
- api_processcard.asp
- pb_ordertest.asp
- pb_processcard.asp
- vc_ordertest.asp
- vc_processcard.asp

4. Site Server Commerce Edition Payment Component Sample

- ProcessCard.vbs

5. LinkPoint COM Object User Manual (this document)

Purchasing Contact Information

Telephone: (800) 456-5989, extension 2966

E-mail: lcomsupport@linkpoint.com

APPENDIX B

Technical Support

E-mail: lcomsupport@linkpoint.com