



# **Sage Exchange Desktop v2.0 API Guide**

Sage Payment Solutions

July 2016

© 2016 The Sage Group plc or its licensors. All rights reserved. Sage, Sage logos, and Sage product and service names mentioned herein are the trademarks of The Sage Group plc or its licensors. All other trademarks are the property of their respective owners.

Sage End User License Agreement (EULA): <http://na.sage.com/sage-na/eula>

# Contents

<b>Implementation.....</b>	<b>5</b>
Objects .....	5
CLSID .....	5
PROGID .....	5
COM Objects/Interfaces .....	5
.Net Objects .....	5
IModuleClient members .....	6
GetUniqueld .....	6
GetUniquelds .....	6
GetResponse .....	6
IModuleClient2 members .....	7
SetInstallPromptFlag .....	7
IModuleResponse members .....	7
Methods .....	8
GetStatusCode.....	8
GetStatusDescription .....	8
GetResponseText .....	9
GetResponseLength .....	9
 <b>C++ sample 10</b>	
ModuleClient members .....	13
Methods .....	13
GetUniqueld .....	13
GetUniquelds .....	14
GetResponse .....	14
SetInstallPromptFlag .....	14
ModuleResponse members .....	14
Methods .....	14
GetStatusCode.....	15
GetStatusDescription .....	15
GetResponseText .....	15
GetResponseLength .....	15
 <b>C# sample 16</b>	
 <b>VB 6 sample19</b>	
 <b>Sage Exchange Desktop payment process flow .....</b>	<b>22</b>

Paragraph styles.....	22
-----------------------	----

# Implementation

The Sage Exchange Desktop v2.0 installation path is [Program Files]\Sage Payment Solutions\Sage Exchange Desktop API\clSedApi.dll. Refer to the *Sage Exchange Desktop v2.0 Installation Guide* for additional information.

## Objects

The following objects/interfaces represent implementations of the Sage Exchange Desktop API. The API provides both COM and .Net objects. When using .Net simply add this library as a class library reference to your project/solution.

### CLSID

Item	Registry Key
ModuleClient	CF5D885D-7838-4807-A2BA-BB0D92D9B1EA
ModuleResponse	AF364412-E730-4738-91BB-B73124B96633

### PROGID

Item	Key
ModuleClient	clSedApi.ModuleClient
ModuleResponse	clSedApi.ModuleResponse

### COM Objects/Interfaces

Item	Description
IModuleClient	Provides client-side communication to the Payment Module.
IModuleClient2	This provides additional client-side communication to the Payment Module
IModuleResponse	Represents a Payment Module response and is the interface for accessing response data.

### .Net Objects

Item	Description
ModuleClient	Provides client-side communication to the Payment Module.

Item	Description
ModuleResponse	Represents a Payment Module response and is the interface for accessing response data.

## IModuleClient members

The following table shows the methods.

In C++, this interface inherits from IDispatch

Method	Description
GetUniqueld	Returns a Globally Unique Identifier.
GetUniquelds	Returns an array of Globally Unique Identifiers.
GetResponse	Sends data to the Payment Module and returns the response data.

### GetUniqueld

Gets a Globally Unique Identifier GUID to be assign to a transaction for use later in transaction status queries.

#### C++ Syntax

```
HRESULT GetUniqueId(*BSTR guid);
```

### GetUniquelds

Gets an array of Globally Unique Identifiers GUIDs to be assigned to transactions for use later in transaction status queries.

#### C++ Syntax

```
HRESULT GetUniqueIds(long numberOfIds, SAFEARRAY** guides);
```

### GetResponse

Sends XML encoded string data to the Payment Module for processing and returns a Payment Module Response object. Refer to the *Sage Exchange XML Messaging Guide* for reference of the XML structures and formats. This method will block until completion, timeout, or error.

#### C++ Syntax

```
HRESULT GetResponse(BSTR request, IModuleResponse* response);
```

## IModuleClient2 members

The following table shows the methods.

In C++, this interface inherits from `IDispatch`

Method	Description
<code>SetInstallPromptFlag</code>	This method accepts a boolean value. If it is false, the Sage Exchange Desktop install prompt is suppressed. The default value is true.

### SetInstallPromptFlag

#### C++ Syntax

```
HRESULT SetInstallPromptFlag (bool flag);
```

## IModuleResponse members

The following tables show the methods.

In C++, this interface inherits from `IDispatch`

## Methods

Method	Description
GetStatusCode	Returns the communication status code: 0 = Success 1 = Invalid Request Data 2 = Communication Startup Error 3 = Communication Error 4 = Module Configuration Not Found 5 = Communication Recovery Error 6 = User Canceled 7 = Timeout 8 = Internet Connection Error 9 = Unknown Error 10 = Remote Data Access Error 11 = Update In Progress
GetStatusDescription	Returns the communication status description
GetResponseText	Returns the XML encoded response string
GetResponseLength	Returns the length of the XML encoded response string

### GetStatusCode

Returns the communication status code. A value of zero means success, a non-zero value indicates an error was encountered communicating with the Payment Module.

#### C++ Syntax

```
HRESULT GetStatusCode(long* statusCode );
```

### GetStatusDescription

Returns the communication status textual description.

#### C++ Syntax

```
HRESULT GetStatusDescription(BSTR* statusDescription );
```



## GetResponseText

Returns the XML encoded response string data from the Payment Module.

### C++ Syntax

```
HRESULT GetResponseText (BSTR* responseText );
```

## GetResponseLength

Returns the string length of the XML encoded response.

### C++ Syntax

```
HRESULT GetResponseLength (long* responseLength );
```

**Note:** Send an “ACK” string to the Sage Exchange Desktop and it will return an “ACK”. This is useful in determining if the Sage Exchange Desktop is running.

# C++ sample

```
#import "clSedApi.tlb" no_namespace, raw_interfaces_only

void Process()
{
    CoInitialize(0);

    HRESULT hr;
    IModuleClientPtr pMC( uuidof(ModuleClient));
    IModuleResponsePtr pMR;

    IModuleClient2Ptr pMC2;
    pMC2 = pMC;
    pMC2->SetInstallPromptFlag(false);

    _bstr_t xmlReq;
    BSTR bstrXmlResp = NULL;
    BSTR bstrUti = NULL;
    long statusCode = 0;

    hr = pMC->GetUniqueID(&bstrUti);
    if( FAILED(hr) )
    {
        // handle COM error
    }

    if( bstrUti )
        ::SysFreeString(bstrUti);

    xmlReq = "<?xml version=\"1.0\" encoding=\"utf-16\"?>";
    xmlReq += "<Request_v1 xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" "
    xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\">";
    xmlReq += "    <Application>";
    xmlReq += "        <ApplicationID>DEMO</ApplicationID>";
    xmlReq += "        <LanguageID>EN</LanguageID>";
    xmlReq += "    </Application>";
    xmlReq += "    <Payments>";
    xmlReq += "        <PaymentType>";
    xmlReq += "            <Merchant>";
```

```

xmlReq += "        <MerchantID>999999999997</MerchantID>";
xmlReq += "        <MerchantKey>K3QD6YWyhFD</MerchantKey>";
xmlReq += "    </Merchant>";
xmlReq += "    <TransactionBase>";
xmlReq += "        <TransactionID>";
xmlReq += bstrUti;
xmlReq += "</TransactionID>";
xmlReq += "        <TransactionType>11</TransactionType>";
xmlReq += "        <Reference1>PO 123456</Reference1>";
xmlReq += "        <Amount>19.99</Amount>";
xmlReq += "    </TransactionBase>";
xmlReq += "    <Customer>";
xmlReq += "        <Name>";
xmlReq += "            <FirstName>John</FirstName>";
xmlReq += "            <MI>A</MI>";
xmlReq += "            <LastName>Doe</LastName>";
xmlReq += "        </Name>";
xmlReq += "        <Address>";
xmlReq += "            <AddressLine1>12345 Street</AddressLine1>";
xmlReq += "            <AddressLine2>Apt #2</AddressLine2>";
xmlReq += "            <City>Some City</City>";
xmlReq += "            <State>Some State</State>";
xmlReq += "            <ZipCode>12345</ZipCode>";
xmlReq += "            <Country>Some Country</Country>";
xmlReq += "            <EmailAddress>john.doe@domain.com</EmailAddress>";
xmlReq += "            <Telephone>1234567891</Telephone>";
xmlReq += "            <Fax>1234567890</Fax>";
xmlReq += "        </Address>";
xmlReq += "        <Company>";
xmlReq += "            <Name>John's Company</Name>";
xmlReq += "            <Address>";
xmlReq += "                <AddressLine1>12345 Street</AddressLine1>";
xmlReq += "                <AddressLine2>Apt #2</AddressLine2>";
xmlReq += "                <City>Some City</City>";
xmlReq += "                <State>Some State</State>";
xmlReq += "                <ZipCode>12345</ZipCode>";
xmlReq += "                <Country>Some Country</Country>";
xmlReq += "                <EmailAddress>john.doe@domain.com</EmailAddress>";
xmlReq += "                <Telephone>1234567891</Telephone>";
xmlReq += "                <Fax>1234567890</Fax>";
xmlReq += "            </Address>";
xmlReq += "        </Company>";
xmlReq += "    </Customer>";
xmlReq += "    <ShippingRecipient>";
xmlReq += "        <Name>";

```

```

xmlReq += "        <FirstName>John</FirstName>";
xmlReq += "        <MI>A</MI>";
xmlReq += "        <LastName>Doe</LastName>";
xmlReq += "    </Name>";
xmlReq += "    <Address>";
xmlReq += "        <AddressLine1>12345 Street</AddressLine1>";
xmlReq += "        <AddressLine2>Apt #2</AddressLine2>";
xmlReq += "        <City>Some City</City>";
xmlReq += "        <State>Some State</State>";
xmlReq += "        <ZipCode>12345</ZipCode>";
xmlReq += "        <Country>Some Country</Country>";
xmlReq += "        <EmailAddress>john.doe@domain.com</EmailAddress>";
xmlReq += "        <Telephone>1234567891</Telephone>";
xmlReq += "        <Fax>1234567890</Fax>";
xmlReq += "    </Address>";
xmlReq += "    <Company>";
xmlReq += "        <Name>John's Company</Name>";
xmlReq += "        <Address>";
xmlReq += "            <AddressLine1>12345 Street</AddressLine1>";
xmlReq += "            <AddressLine2>Apt #2</AddressLine2>";
xmlReq += "            <City>Some City</City>";
xmlReq += "            <State>Some State</State>";
xmlReq += "            <ZipCode>12345</ZipCode>";
xmlReq += "            <Country>Some Country</Country>";
xmlReq += "            <EmailAddress>john.doe@domain.com</EmailAddress>";
xmlReq += "            <Telephone>1234567891</Telephone>";
xmlReq += "            <Fax>1234567890</Fax>";
xmlReq += "        </Address>";
xmlReq += "    </Company>";
xmlReq += "</ShippingRecipient>";
xmlReq += "</PaymentType>";
xmlReq += "</Payments>";
xmlReq += "</Request v1>";

if( bstrUti )
    ::SysFreeString(bstrUti);

hr = pMC->GetResponse(xmlReq, &pMR.GetInterfacePtr());
if( FAILED(hr) )
{
    // handle COM error
}

hr = pMR->GetStatusCode(&statusCode);
if( FAILED(hr) )

```

```

{
    // handle COM error
}

if( statusCode != 0 )
{
    // a non zero value means the Payment Module encountered an error processing requested
data
    // a transaction status query shouldbe use to determine the transaction status
}

hr = pMR->GetResponseText(&bstrXmlResp);
if( FAILED(hr) )
{
    // handle COM error
}
if( bstrXmlResp )
    ::SysFreeString(bstrXmlResp);

pMC = NULL;
pMR = NULL;
::CoUninitialize();
}

```

**Note:** XML documents should be created using a proper DOM to observe proper XML encoding.

## ModuleClient members

The following tables show the methods.

### Methods

Method	Description
GetUniqueld	Returns a Globally Unique Identifier
GetUniquelds	Returns an array of Globally Unique Identifiers
GetResponse	Sends data to the Payment Module and returns the response data
SetInstallPromptFlag	Use this method to suppress the SED install prompt

### GetUniqueld

Gets a Globally Unique Identifier GUID to be assign to a transaction for use later in transaction status queries.

#### C# Syntax

```
string GetUniqueId();
```

### GetUniqueIds

Gets an array of Globally Unique Identifiers GUIDs to be assigned to transactions for use later in transaction status queries.

#### C# Syntax

```
string [] GetUniqueIds(int numberOfIds);
```

### GetResponse

Sends XML encoded string data to the Payment Module for processing and returns a Payment Module Response object. This method will block until completion, timeout, or error.

#### C++ Syntax

```
ModuleResponse GetResponse(string request);
```

### SetInstallPromptFlag

Accepts a boolean value. Suppresses the SED install screen if the boolean is false.

#### C++ Syntax

```
HRESULT SetInstallPromptFlag(bool flag);
```

## ModuleResponse members

The following tables show the methods.

### Methods

Method	Description
GetStatusCode	Returns the communication status code .
GetStatusDescription	Returns the communication status description.
GetResponseText	Returns the XML encoded response string.
GetResponseLength	Returns the length of the XML encoded response string.

## GetStatusCode

Returns the communication status code. A value of zero means success, a non-zero value indicates an error was encountered communicating with the Payment Module.

### C# Syntax

```
int GetStatusCode();
```

## GetStatusDescription

Returns the communication error description.

### C# Syntax

```
string GetStatusDescription();
```

## GetResponseText

Returns the XML encoded response string data from the Payment Module.

### C# Syntax

```
string GetResponseText();
```

## GetResponseLength

Returns the string length of the XML encoded response.

### C# Syntax

```
int GetResponseLength();
```

# C# sample

```
using clSedApi;

void Process()
{
    string xmlResp = null;
    long retVal = 0;
    ModuleClient mc = new ModuleClient();
    mc.SetInstallPromptFlag(false);
    IModuleResponse mr = null;
    string xmlReq = "";

    string uti = mc.GetUniqueId();

    xmlReq = "<?xml version=\"1.0\" encoding=\"utf-16\"?>";
    xmlReq += "<Request_v1 xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"";
    xmlReq += "<Application>";
    xmlReq += "    <ApplicationID>DEMO</ApplicationID>";
    xmlReq += "    <LanguageID>EN</LanguageID>";
    xmlReq += "</Application>";
    xmlReq += "<Payments>";
    xmlReq += "    <PaymentType>";
    xmlReq += "        <Merchant>";
    xmlReq += "            <MerchantID>999999999997</MerchantID>";
    xmlReq += "            <MerchantKey>K3QD6YWyhFD</MerchantKey>";
    xmlReq += "        </Merchant>";
    xmlReq += "        <TransactionBase>";
    xmlReq += "            <TransactionID>";
    xmlReq += uti;
    xmlReq += "</TransactionID>";
    xmlReq += "        <TransactionType>11</TransactionType>";
    xmlReq += "        <Reference1>PO 123456</Reference1>";
    xmlReq += "        <Amount>19.99</Amount>";
    xmlReq += "</TransactionBase>";
    xmlReq += "    <Customer>";
    xmlReq += "        <Name>";
    xmlReq += "            <FirstName>John</FirstName>";
    xmlReq += "        <MI>A</MI>";
```



```

xmlReq += "        <LastName>Doe</LastName>";
xmlReq += "    </Name>";
xmlReq += "    <Address>";
xmlReq += "        <AddressLine1>12345 Street</AddressLine1>";
xmlReq += "        <AddressLine2>Apt #2</AddressLine2>";
xmlReq += "        <City>Some City</City>";
xmlReq += "        <State>Some State</State>";
xmlReq += "        <ZipCode>12345</ZipCode>";
xmlReq += "        <Country>Some Country</Country>";
xmlReq += "        <EmailAddress>john.doe@domain.com</EmailAddress>";
xmlReq += "        <Telephone>1234567891</Telephone>";
xmlReq += "        <Fax>1234567890</Fax>";
xmlReq += "    </Address>";
xmlReq += "    <Company>";
xmlReq += "        <Name>John's Company</Name>";
xmlReq += "        <Address>";
xmlReq += "            <AddressLine1>12345 Street</AddressLine1>";
xmlReq += "            <AddressLine2>Apt #2</AddressLine2>";
xmlReq += "            <City>Some City</City>";
xmlReq += "            <State>Some State</State>";
xmlReq += "            <ZipCode>12345</ZipCode>";
xmlReq += "            <Country>Some Country</Country>";
xmlReq += "            <EmailAddress>john.doe@domain.com</EmailAddress>";
xmlReq += "            <Telephone>1234567891</Telephone>";
xmlReq += "            <Fax>1234567890</Fax>";
xmlReq += "        </Address>";
xmlReq += "    </Company>";
xmlReq += "</Customer>";
xmlReq += "<ShippingRecipient>";
xmlReq += "    <Name>";
xmlReq += "        <FirstName>John</FirstName>";
xmlReq += "        <MI>A</MI>";
xmlReq += "        <LastName>Doe</LastName>";
xmlReq += "    </Name>";
xmlReq += "    <Address>";
xmlReq += "        <AddressLine1>12345 Street</AddressLine1>";
xmlReq += "        <AddressLine2>Apt #2</AddressLine2>";
xmlReq += "        <City>Some City</City>";
xmlReq += "        <State>Some State</State>";
xmlReq += "        <ZipCode>12345</ZipCode>";
xmlReq += "        <Country>Some Country</Country>";
xmlReq += "        <EmailAddress>john.doe@domain.com</EmailAddress>";
xmlReq += "        <Telephone>1234567891</Telephone>";
xmlReq += "        <Fax>1234567890</Fax>";
xmlReq += "    </Address>";

```

```

xmlReq += "        <Company>";
xmlReq += "        <Name>John's Company</Name>";
xmlReq += "        <Address>";
xmlReq += "            <AddressLine1>12345 Street</AddressLine1>";
xmlReq += "            <AddressLine2>Apt #2</AddressLine2>";
xmlReq += "            <City>Some City</City>";
xmlReq += "            <State>Some State</State>";
xmlReq += "            <ZipCode>12345</ZipCode>";
xmlReq += "            <Country>Some Country</Country>";
xmlReq += "            <EmailAddress>john.doe@domain.com</EmailAddress>";
xmlReq += "            <Telephone>1234567891</Telephone>";
xmlReq += "            <Fax>1234567890</Fax>";
xmlReq += "        </Address>";
xmlReq += "    </Company>";
xmlReq += "    </ShippingRecipient>";
xmlReq += "    </PaymentType>";
xmlReq += "    </Payments>";
xmlReq += "</Request v1>";

mr = mc.GetResponse(xmlReq);

retVal = mr.GetStatusCode();
if( retVal != 0 )
{
    // handle Payment Module communication error
    // a transaction status query should be used to determine the status of the transaction
}

xmlResp = mr.GetResponseText();
}

```

**Note:** XML documents should be created using a proper DOM to observe proper XML encoding.

## VB 6 sample

```
sub Process()  
Dim xmlResp as string  
Dim retVal as integer  
Dim mc as Object  
Dim mr as Object  
Dim xmlReq as string  
Dim uti as string  
  
Mc = CreateObject("clSedApi.ModuleClient")  
mc.SetInstallPromptFlag(false)  
uti = mc.GetUniqueId()  
  
    xmlReq = "<?xml version=\"1.0\" encoding=\"utf-16\"?>"  
    xmlReq = xmlReq & "<Request_v1 xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"  
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\">"  
    xmlReq = xmlReq & "    <Application>"  
    xmlReq = xmlReq & "        <ApplicationID>DEMO</ApplicationID>"  
    xmlReq = xmlReq & "        <LanguageID>EN</LanguageID>"  
    xmlReq = xmlReq & "    </Application>"  
    xmlReq = xmlReq & "    <Payments>"  
    xmlReq = xmlReq & "        <PaymentType>"  
    xmlReq = xmlReq & "            <Merchant>"  
    xmlReq = xmlReq & "                <MerchantID>999999999997</MerchantID>"  
    xmlReq = xmlReq & "                <MerchantKey>K3QD6YWyhfD</MerchantKey>"  
    xmlReq = xmlReq & "            </Merchant>"  
    xmlReq = xmlReq & "        <TransactionBase>"  
    xmlReq = xmlReq & "            <TransactionID>"  
    xmlReq = xmlReq & uti  
    xmlReq = xmlReq & "</TransactionID>"  
    xmlReq = xmlReq & "        <TransactionType>l1</TransactionType>"  
    xmlReq = xmlReq & "        <Reference1>PO 123456</Reference1>"  
    xmlReq = xmlReq & "        <Amount>19.99</Amount>"  
    xmlReq = xmlReq & "    </TransactionBase>"  
    xmlReq = xmlReq & "    <Customer>"  
    xmlReq = xmlReq & "        <Name>"  
    xmlReq = xmlReq & "            <FirstName>John</FirstName>"  
    xmlReq = xmlReq & "            <MI>A</MI>"  
    xmlReq = xmlReq & "            <LastName>Doe</LastName>"  
    xmlReq = xmlReq & "        </Name>"  
    xmlReq = xmlReq & "        <Address>"
```

```

xmlReq = xmlReq & "          <AddressLine1>12345 Street</AddressLine1>"
xmlReq = xmlReq & "          <AddressLine2>Apt #2</AddressLine2>"
xmlReq = xmlReq & "          <City>Some City</City>"
xmlReq = xmlReq & "          <State>Some State</State>"
xmlReq = xmlReq & "          <ZipCode>12345</ZipCode>"
xmlReq = xmlReq & "          <Country>Some Country</Country>"
xmlReq = xmlReq & "          <EmailAddress>john.doe@domain.com</EmailAddress>"
xmlReq = xmlReq & "          <Telephone>1234567891</Telephone>"
xmlReq = xmlReq & "          <Fax>1234567890</Fax>"
xmlReq = xmlReq & "        </Address>"
xmlReq = xmlReq & "      <Company>"
xmlReq = xmlReq & "        <Name>John's Company</Name>"
xmlReq = xmlReq & "        <Address>"
xmlReq = xmlReq & "          <AddressLine1>12345 Street</AddressLine1>"
xmlReq = xmlReq & "          <AddressLine2>Apt #2</AddressLine2>"
xmlReq = xmlReq & "          <City>Some City</City>"
xmlReq = xmlReq & "          <State>Some State</State>"
xmlReq = xmlReq & "          <ZipCode>12345</ZipCode>"
xmlReq = xmlReq & "          <Country>Some Country</Country>"
xmlReq = xmlReq & "          <EmailAddress>john.doe@domain.com</EmailAddress>"
xmlReq = xmlReq & "          <Telephone>1234567891</Telephone>"
xmlReq = xmlReq & "          <Fax>1234567890</Fax>"
xmlReq = xmlReq & "        </Address>"
xmlReq = xmlReq & "      </Company>"
xmlReq = xmlReq & "    </Customer>"
xmlReq = xmlReq & "  <ShippingRecipient>"
xmlReq = xmlReq & "    <Name>"
xmlReq = xmlReq & "      <FirstName>John</FirstName>"
xmlReq = xmlReq & "      <MI>A</MI>"
xmlReq = xmlReq & "      <LastName>Doe</LastName>"
xmlReq = xmlReq & "    </Name>"
xmlReq = xmlReq & "    <Address>"
xmlReq = xmlReq & "      <AddressLine1>12345 Street</AddressLine1>"
xmlReq = xmlReq & "      <AddressLine2>Apt #2</AddressLine2>"
xmlReq = xmlReq & "      <City>Some City</City>"
xmlReq = xmlReq & "      <State>Some State</State>"
xmlReq = xmlReq & "      <ZipCode>12345</ZipCode>"
xmlReq = xmlReq & "      <Country>Some Country</Country>"
xmlReq = xmlReq & "      <EmailAddress>john.doe@domain.com</EmailAddress>"
xmlReq = xmlReq & "      <Telephone>1234567891</Telephone>"
xmlReq = xmlReq & "      <Fax>1234567890</Fax>"
xmlReq = xmlReq & "    </Address>"
xmlReq = xmlReq & "  <Company>"
xmlReq = xmlReq & "    <Name>John's Company</Name>"
xmlReq = xmlReq & "    <Address>"

```

```

xmlReq = xmlReq & "          <AddressLine1>12345 Street</AddressLine1>"
xmlReq = xmlReq & "          <AddressLine2>Apt #2</AddressLine2>"
xmlReq = xmlReq & "          <City>Some City</City>"
xmlReq = xmlReq & "          <State>Some State</State>"
xmlReq = xmlReq & "          <ZipCode>12345</ZipCode>"
xmlReq = xmlReq & "          <Country>Some Country</Country>"
xmlReq = xmlReq & "          <EmailAddress>john.doe@domain.com</EmailAddress>"
xmlReq = xmlReq & "          <Telephone>1234567891</Telephone>"
xmlReq = xmlReq & "          <Fax>1234567890</Fax>"
xmlReq = xmlReq & "          </Address>"
xmlReq = xmlReq & "        </Company>"
xmlReq = xmlReq & "      </ShippingRecipient>"
xmlReq = xmlReq & "    </PaymentType>"
xmlReq = xmlReq & "  </Payments>"
xmlReq = xmlReq & "</Request v1>"

mr = mc.GetResponse(xmlReq)

retVal = mr.GetStatusCode()
If retVal <> 0 Then

  ' handle Payment Module communication error
  ' a transaction status query should be used to determine the status of the transaction
End If

xmlResp = mr.GetResponseText()
end sub

```

**Note:** XML documents should be created using a proper DOM to observe proper XML encoding.

# Sage Exchange Desktop payment process flow

