E Stream Software

Application COM Interface

Overview

This document will give you a guideline on how to use and call E Stream Software Application COM Interface. Some examples are provided to help users gain understanding on how to interact with E Stream Software Application via COM Interface. Examples are written in JavaScript, other programming languages can be chosen to call E Stream Software Application COM Interface.

List of Application Objects in E Stream Software Application

1. Application Title
- name of the software application.

2. Application Build No
- show the build number of E Stream software application that currently running in your system.

3. Application Release Date
- show the release date of E Stream software application that currently running in your system.

4. Modules
- show the list of available modules in your current E Stream Software application.

5. Data Views Objects

- list all the available fields for each business objects.

6. Business Objects

- Collection of business object. It provides a gateway to access the business entity in your software application.

7. Access Right Options

- List of available access right options in E Stream software application.

8. Actions

- List of available actions in E Stream software application.

Application Objects - Application Title, Application Build Number, Application Release Date

---

Before users call COM interface, few steps must be taken :-

Step 1 : E Stream Application Software (SQL Financial Accounting) must be installed and run.
Step 2 : Users have to write some code to create COM server. [**Show Code**]

1. Application Title [**Show Code**]
- Retrieve the name of the E Stream Software Application.

2. Application Build Number [**Show Code**]
- Retrieve the build number of E Stream Software Application.

3. Application Release Date [**Show Code**]
- Retrieve release date of E Stream Software Application.

Example:

Application Objects - Modules, Access Right Options, Actions

Before users call COM interface, few steps must be taken :-

Step 1 : E Stream Application Software (SQL Financial Accounting) must be installed and run.
Step 2 : Users have to write some code to create COM server. [**Show Code**]

1. Modules [**Show Code**]
- Retrieve the list of available modules in E Stream Software Application.

2. Access Right Options [**Show Code**]
- Retrieve the list of available access right options in E Stream Software Application.

3. Actions [**Show Code**]
- Retrieve the list of available actions in E Stream Software Application.

Example:

Modules

Application Objects - DataViews

Before users call COM interface, few steps must be taken :-

Step 1 : E Stream Application Software (SQL Financial Accounting) must be installed and run.
Step 2 : Users have to write some code to create COM server. [**Show Code**]

1. DataViews [**Show Code**]
- Retrieve the list of DataViews and Data Accesses under each DataViews.

Example:

Click here to show a list of data views.

DataView List	Data Accesses	Fields

Application Objects - Business Objects

Before users call COM interface, few steps must be taken :-

Step 1 : E Stream Application Software (SQL Financial Accounting) must be installed and run.
Step 2 : Users have to write some code to create COM server. [**Show Code**]

1. Business Objects [**Show Code**]
- Collection of business object.
- A business object provides a gateway to access the a business entity in application.
- Examples of business entities are Invoice, Customer, Agent, Stock Item.
- All business objects possess same methods and properties for uniform access.

Example:

Click here to show a list of business objects.

Business Objects List	DataSets	Fields

Application Objects - Report Objects

Before users call COM interface, few steps must be taken :-

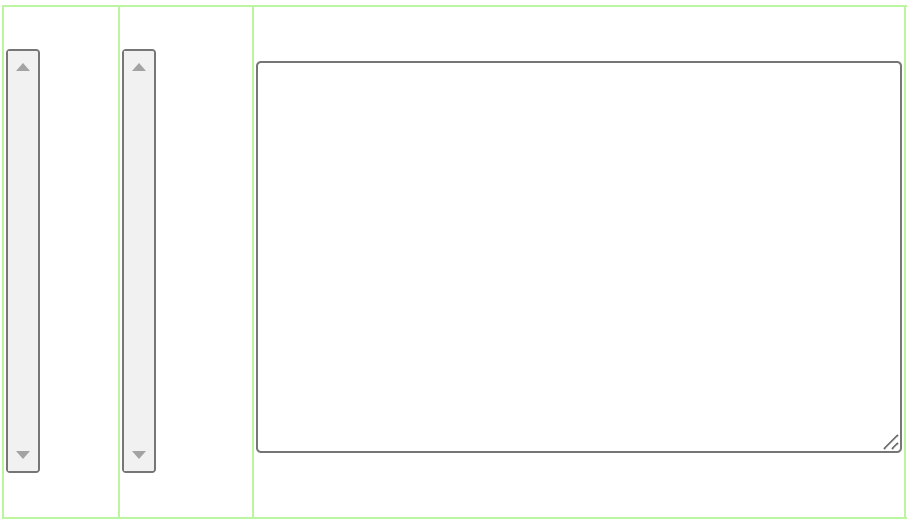
Step 1 : E Stream Application Software (SQL Financial Accounting) must be installed and run.
Step 2 : Users have to write some code to create COM server. [**Show Code**]

1. Report Objects [**Show Code**]
- Collection of report object.
- A report object provides a gateway to access a report in application.
- Examples of report entities are Sales Invoice Report, Customer Statement Report, Agent Listing Report, and etc.
- All report objects possess same methods and properties for uniform access.

Example:

Click here to show a list of report objects.

Report Objects List	DataSets	Parameters



Add Agent Record

The following will show you how to add an agent record into SQL Accounting.

Add Agent

Agent Code:

Description:

Add

Show Code

Edit Agent Record

The following will show you how to edit an agent record in SQL Accounting.

Edit Agent

Agent Code: [ ] Display Agent Info

Description: [ ]

Save

Show Code

Delete Agent Record

The following will show you how to delete an agent record in SQL Accounting.

Delete Agent

Agent Code: [ ] Display Agent Info

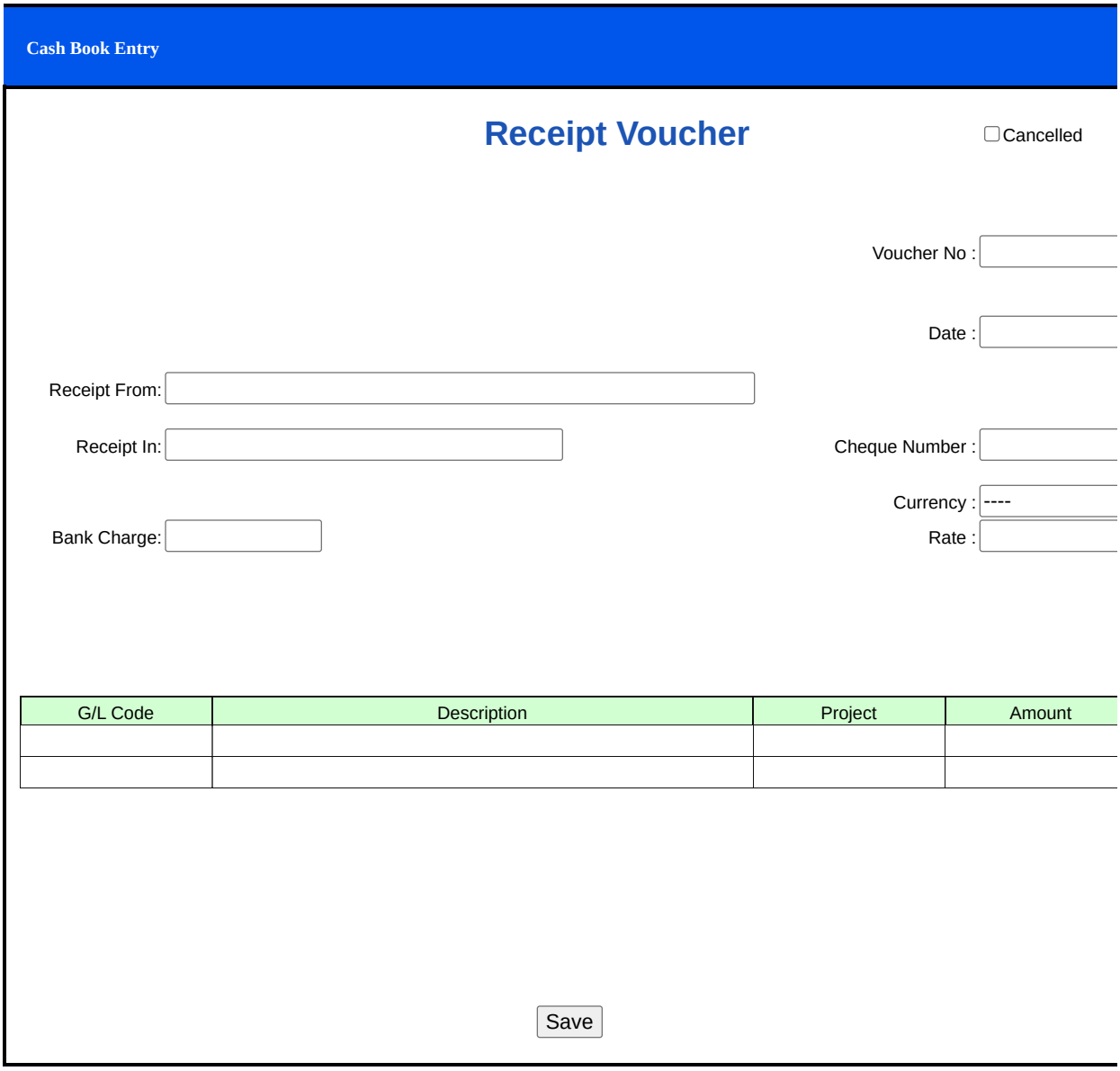
Description: [ ]

Delete

Show Code

Add Receipt Voucher

The following will show you how to add a Receipt Voucher record into SQL Accounting.



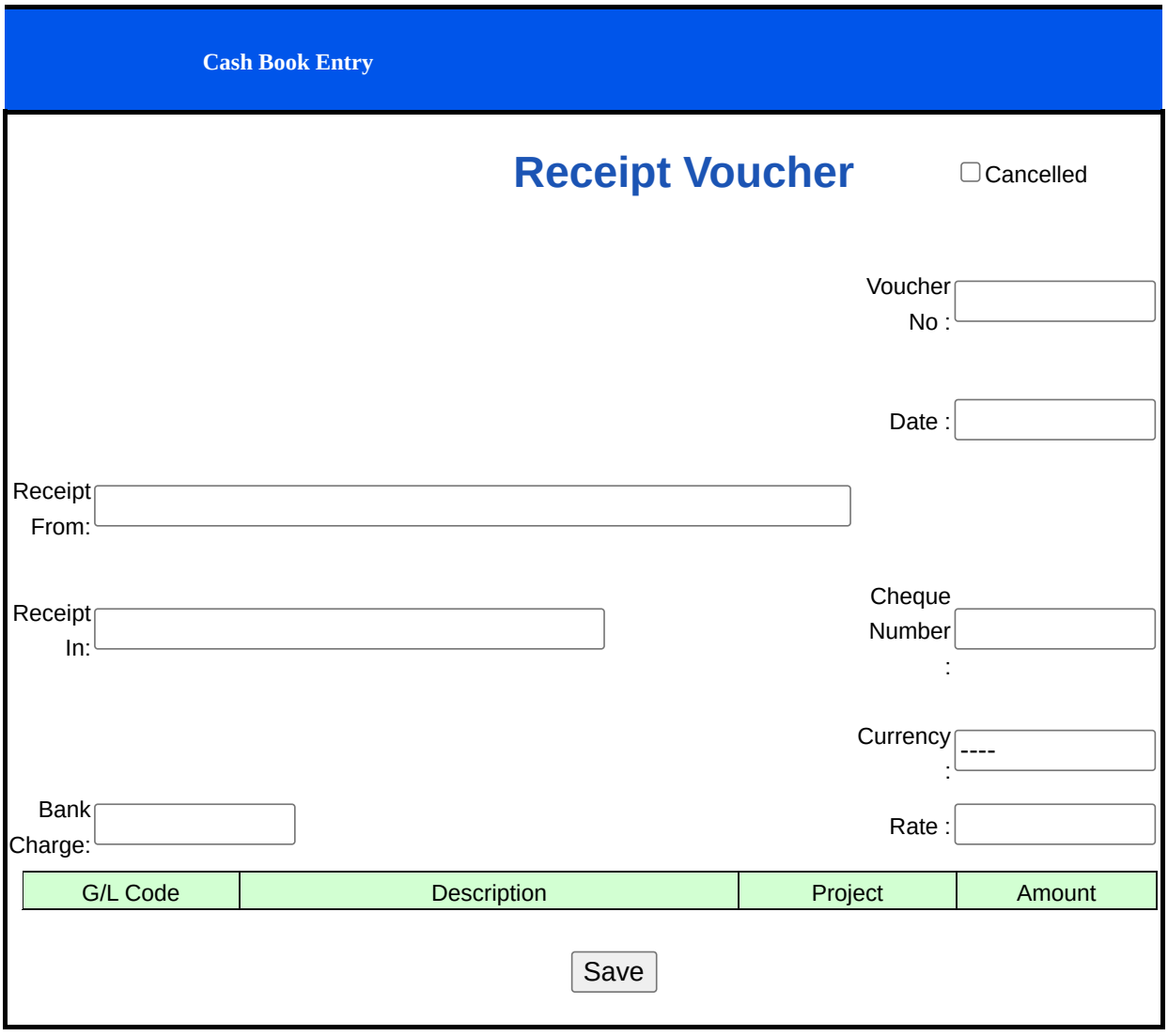
Show Code

Edit Receipt Voucher Record

The following will show you how to edit a Receipt Voucher record in SQL Accounting.

Please provide an Voucher Number to load the Receipt Voucher record.

Voucher No. : [ ] OK



Show Code

Delete Receipt Voucher Record

The following will show you how to delete a Receipt Voucher record in SQL Accounting.

Please provide an Voucher Number to load the Receipt Voucher record.

Voucher No. : [ ] OK

Cash Book Entry

Receipt Voucher

☐ Cancelled

Voucher No : [ ]

Date : [ ]

Receipt From: [ ]

Receipt In: [ ]

Cheque Number : [ ]

Currency : [----]

Bank Charge: [ ]

Rate : [ ]

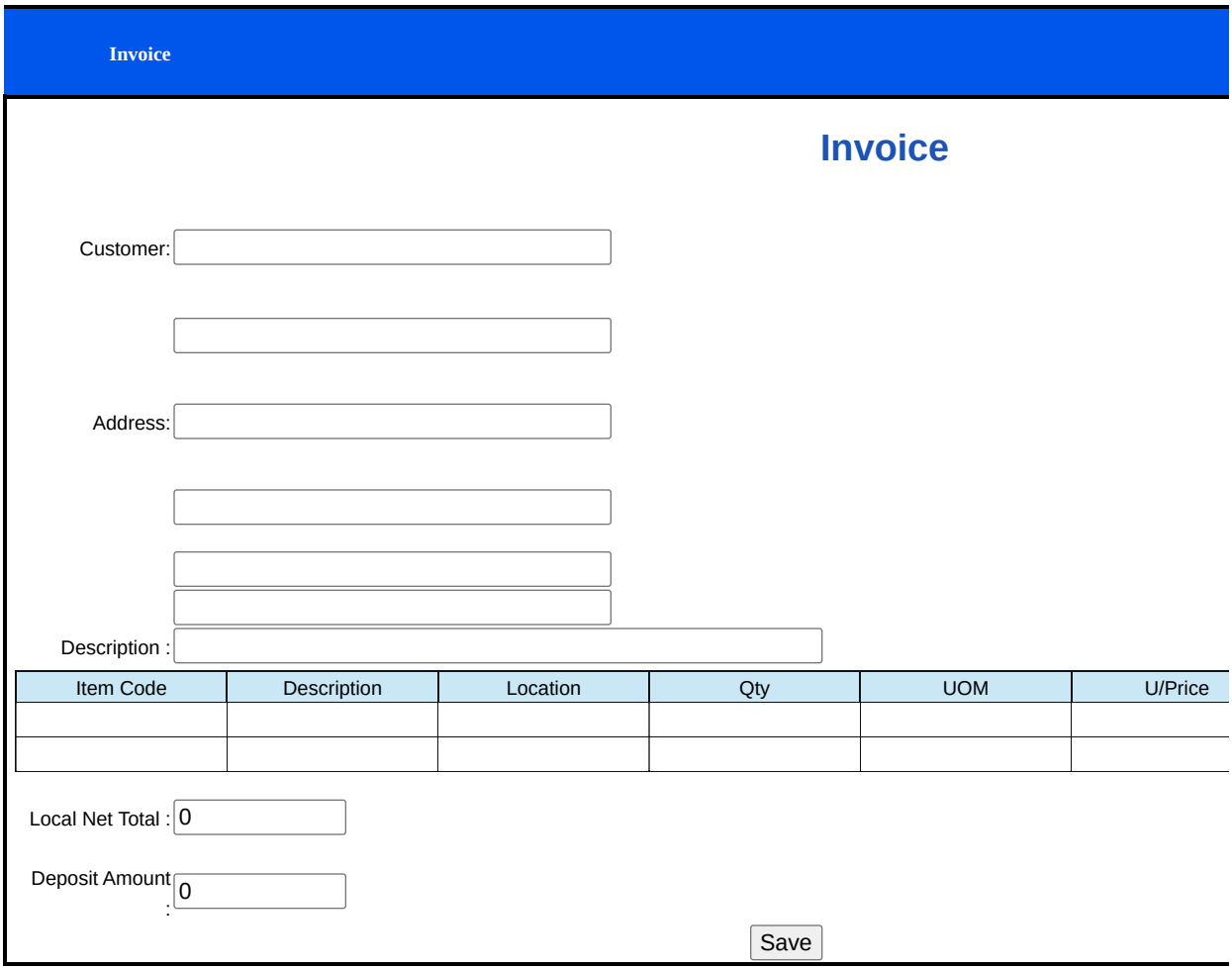
G/L Code	Description	Project	Amount

Delete

Show Code

Add Invoice

The following will show you how to add a Sales Invoice record into SQL Accounting.



Show Code

Edit Invoice Record

The following will show you how to edit a Sales Invoice record in SQL Accounting.

Please provide an Invoice Number to load the Invoice record.

Invoice No. : OK

Invoice

Invoice

Cancelled

Customer:

Invoice No :

Date :

Address:

Agent :

Terms :

Currency : ----

Rate : 1

Description :

Item Code	Description	Location	Qty	UOM	U/Price	Discount	Sub Total

Local Net Total : 0

Net Total. 0

Deposit Amount : 0

Save

Show Code

Delete Invoice Record

The following will show you how to delete a Sales Invoice record in SQL Accounting.

Please provide an Invoice Number to load the Invoice record.
Invoice No. : OK

Invoice

Invoice

Cancelled

Customer:

Invoice No :

Date :

Address:

Agent :

Terms :

Currency : ----

Rate : 1

Description :

Item Code	Description	Location	Qty	UOM	U/Price	Discount	Sub Total

Local Net Total : 0

Net Total. 0

Deposit Amount : 0

Delete

Show Code

Add Customer Payment Record

The following will show you how to add a Customer Payment record into SQL Accounting and how to knockoff document.

Please provide a Customer Code to load record.
Customer Code : OK

Customer Payment Entry

Customer Code : Currency : ---- Rate : 1

Customer Payment

Cancelled

O/R No :

Paid By : Date :

Received In : Project :

Bank Charge : Paid Amount :

Cheque No :

Local Amount :

Description :

Unapplied Amt :

Knock Off Invoices / Debit Notes

Type	Date	Doc No	Amount	Outstanding	Pay	

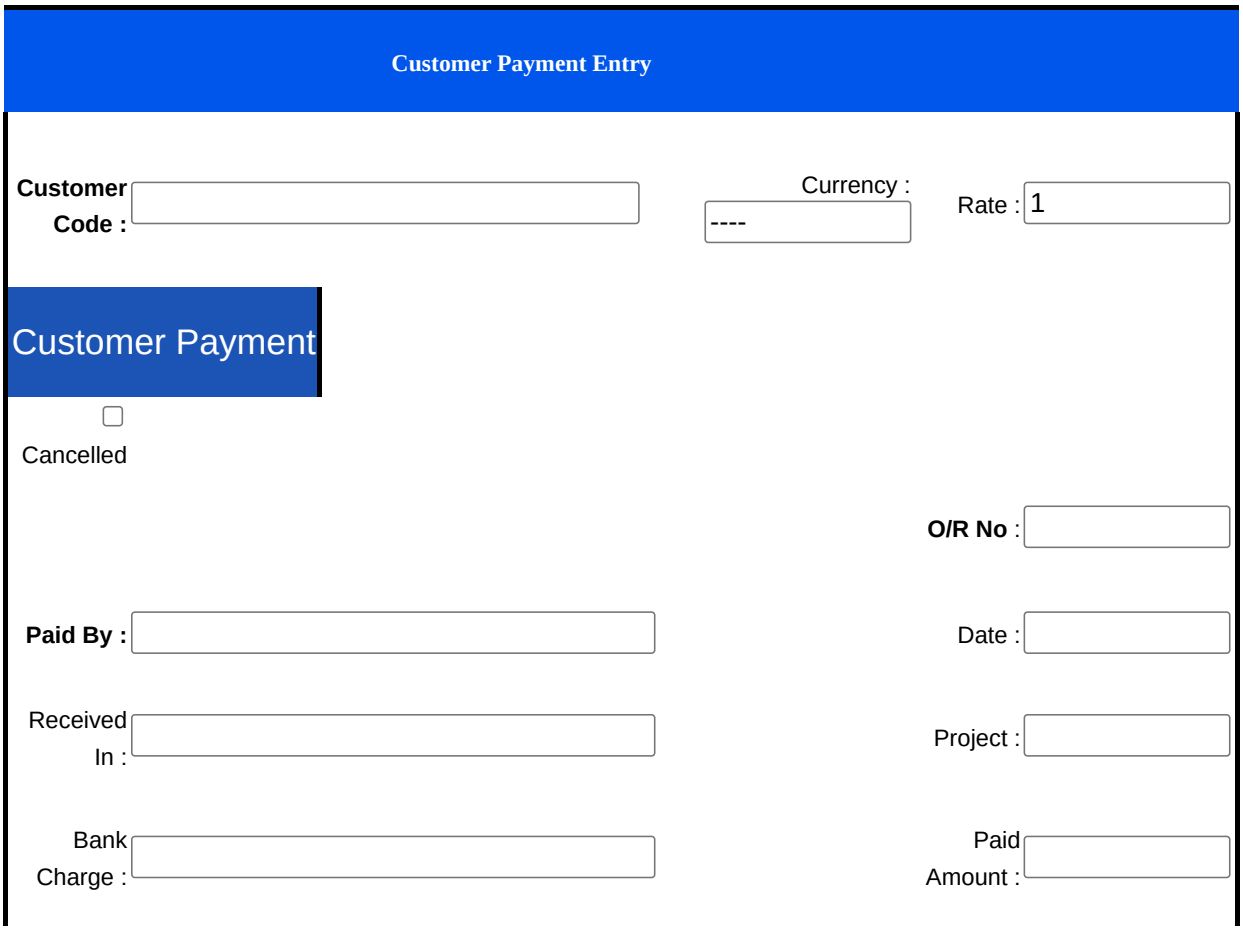
Save

Show Code

Edit Customer Payment Record

The following will show you how to edit a Customer Payment record in SQL Accounting and how to knockoff document.

Please provide a O/R No to load payment record.
O/R No : OK



Cheque No :

Local Amount :

Description :

Unapplied Amt :

Knock Off Invoices / Debit Notes

Type	Date	Doc No	Amount	Outstanding	Pay	

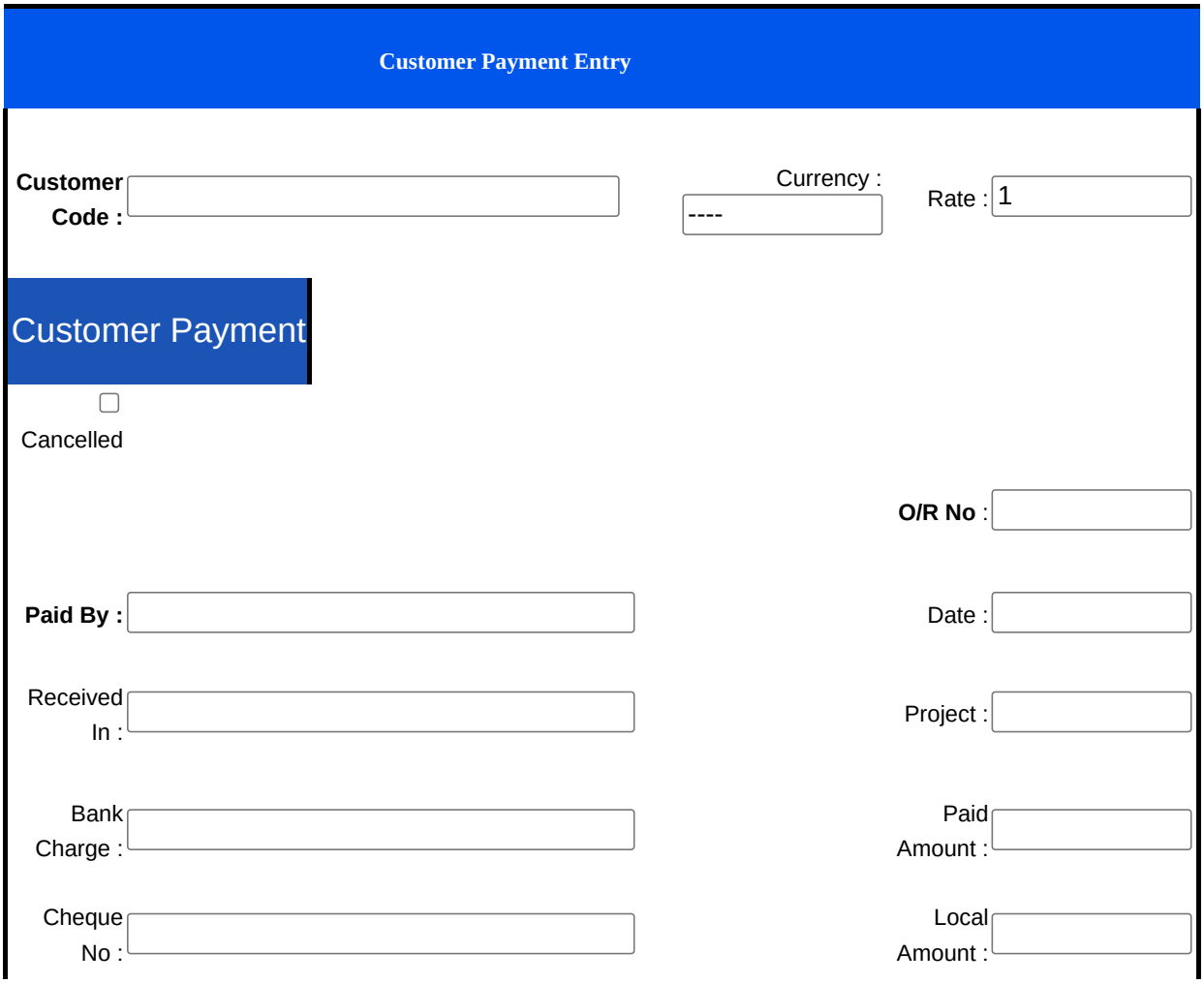
Save

Show Code

Delete Customer Payment Record

The following will show you how to delete a Customer Payment record in SQL Accounting.

Please provide a O/R No to load payment record.
O/R No : OK



Description :

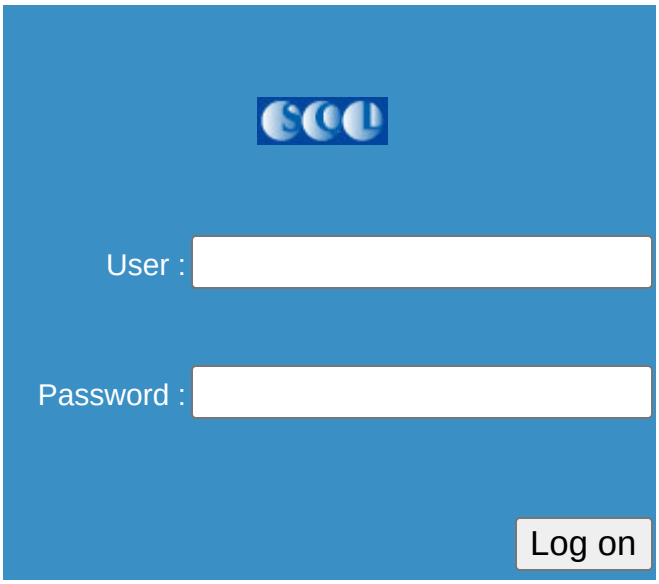
Unapplied Amt :

Knock Off Invoices / Debit Notes

Type	Date	Doc No	Amount	Outstanding	Pay	

Delete

Show Code


User :
Password :
Log on

How to Set COM Object Class Instancing property ?

- Class Instancing property

  determines how COM objects are instantiated.

- Instancing determines how many

  instances of your object clients can create in a single executable.

- If you specify a Single Instance model, for example, then once a client has instantiated your object, COM removes the application from view so that other clients must launch their own instances of the application.

- By Default, SQL application's com object instancing property is set to **Multi Instance**.

- There are 3 possible values to

  set Com Object Instancing property :

Value	Instancing	Meaning
0	Internal	The COM object is created by the same process as the COM server. That is, an external application cannot create an instance of this object directly. Instead, external processes must call a method of the application that creates the document object.
1	Single Instance	Allows only a single instance of the COM object for each executable (application). If this single instance is not shared across multiple clients, then each client must launch its own instance of the executable. Each clients launch new executable (application).
2	Multi Instance	The COM object is

		created as one of multiple instances within the same executable. Any time a client requests service, a separate instance of the object gets invoked. Multi clients share the same executable (application).

Example

```
<script language="JScript">
var ComServer;

function CreateSQLAccServer () {
ComServer = new ActiveXObject('SQLAcc.BizApp');
}

function SetClassInstancing () {
CreateSQLAccServer (); /* Create Com Server


*/
ComServer.ClassInstancing = 1; /* Single Instance */
}
</script>
```

How to create SQL Accounting COM Server ?

Javascript / JScript

```
<script

language="JScript">
var ComServer;

function CreateSQLAccServer () {
ComServer = new ActiveXObject('SQLAcc.BizApp');
}
</script>
```

VBScript

```
<script

language="VBScript">
<!--
Function CreateSQLAccServer
Set CreateSQLAccServer = CreateObject("SQLAcc.BizApp")
End Function
-->
</script>
```

ASP

```
<script
language="JScript" runat="server">

var ComServer = new ActiveXObject('SQLAcc.BizApp');
<script>
```

Delphi

```
uses

ComObj;

var ComServer: variant;

function CreateSQLAccServer: variant;
begin
ComServer := CreateOleObject('SQLAcc.BizApp');
end;
```

How to open an Invoice through Invoice business object by Document Number ?

Javascript / JScript

```
<script

language="JScript">
var ComServer;

function CreateSQLAccServer() {
ComServer = new ActiveXObject('SQLAcc.BizApp');
}

function Login() {
var Result = true;

/* Firstly initialize to true */
CreateSQLAccServer();

/* Create Com Server */

if (!ComServer.IsLogin) { /* if user hasn't

logon to SQL application */
try {
/* Call login method */
ComServer.Login('ADMIN', 'ADMIN', 'C:\\Program
Files\\eStream\\SQLAccounting\\DEFAULT.DCF',

'ACC-0001.FDB');

} catch (e) {

/* Catch exception */
Result = false;
alert(e.message);

/* Display error message catch from SQL Application */
}
}
return (Result);
}

function OpenInvoice() {
CreateSQLAccServer();
```

```
/* Create Com Server */
if (Login()) {
BizObject = ComServer.BizObjects.Find('SL_IV');

/* Create Biz Object */
lDocNo = 'IV-00001';

/* Assigned Document Number here */
lDocKey = BizObject.FindKeyByRef('DocNo', lDocNo); /* FindKeyByRef
will find Document key based on lDocNo

*/
BizObject.Params.Find('DocKey').Value = lDocKey;

/* Set lDocKey to BizObject DocKey field */

if (lDocKey != null) {

/* if lDocKey is not null mean key is found */
BizObject.Open();

/* Open BizObject */
alert ("Document found.");
}
else {
alert ("Document not found.");


/* not found */
}
}
}
</script>
```

VBScript

```
<script

language="VBScript">
<!--
Function CreateSQLAccServer
Set CreateSQLAccServer = CreateObject("SQLAcc.BizApp")
End Function

Function Login
Dim ComServer
Login = True
```

```
'Firstly, initialize to True
Set ComServer = CreateSQLAccServer 'Create Com Server
If not ComServer.IsLogin Then 'if

user hasn't logon to SQL application
On Error Resume Next
'Call login method
ComServer.Login "ADMIN", "ADMIN", "C:\\Program
Files\\eStream\\SQLAccounting\\DEFAULT.DCF",

"ACC-0001.FDB"

If not ComServer.IsLogin then
Err.Raise vbObjectError 'Assigned your error number

here
MsgBox Err.Description 'Prompt error catch from SQL

application
Login = False
End If
End If
Err.Clear

'Clear the error.
End Function

Function OpenInvoice
Dim ComServer, BizObject, lDocKey, lDocNo
Set ComServer = CreateSQLAccServer

'Create Com Server
If Login Then
Set BizObject = ComServer.BizObjects.Find("SL_IV")

'Create Biz Object
lDocNo = "IV-00001"

'Assigned Document Number here
lDocKey = BizObject.FindKeyByRef("DocNo", lDocNo)

'FindKeyByRef will find Document key based on lDocNo
BizObject.Params.Find("DocKey").Value = lDocKey

'Set lDocKey to BizObject DocKey field

If not IsNull(lDocKey) Then

'if lDocKey is not null mean key is found
BizObject.Open
```

```
'Open BizObject
MsgBox "Document found."
Else

'not found
MsgBox "Document not found."
End If
End If
End Function

-->
</script>
```

ASP

```
<%@

LANGUAGE="JAVASCRIPT" %>
<%
var ComServer;

function CreateSQLAccServer() {
ComServer = new ActiveXObject('SQLAcc.BizApp');
}

function Login() {
var Result = true;

/* Firstly initialize to true */
CreateSQLAccServer();

/* Create Com Server */

if (!ComServer.IsLogin) {

/* if user hasn't logon to SQL application */
try {
/* Call login method */
ComServer.Login('ADMIN', 'ADMIN', 'C:\\Program
Files\\eStream\\SQLAccounting\\DEFAULT.DCF',

'ACC-0001.FDB');
} catch (e) {

/* Catch exception */
Result = false;
Response.Write (e.message);

/* Display error message catch from SQL Application */
}
}
return (Result);
}

function OpenInvoice() {
CreateSQLAccServer();

/* Create Com Server */
if (Login()) {
BizObject = ComServer.BizObjects.Find('SL_IV');
```

```
/* Create Biz Object */
lDocNo = 'IV-00001';

/* Assigned Document Number here */
lDocKey = BizObject.FindKeyByRef('DocNo', lDocNo); /*

FindKeyByRef will find Document key based on lDocNo */
BizObject.Params.Find('DocKey').Value = lDocKey;

/* Set lDocKey to BizObject DocKey field */

if (lDocKey != null) {

/* if lDocKey is not null mean key is found */
BizObject.Open();

/* Open BizObject */
Response.Write ("Document found.");
}
else {
Response.Write ("Document not found.");


/* not found */
}
}
}
%>
```

Delphi

```
uses

ComObj;

var ComServer, BizObject: variant;

function CreateSQLAccServer: variant;
begin
Result := CreateOleObject('SQLAcc.BizApp');
end;

function Login: boolean;
begin
Result := True;

//Firstly, initialize to True
ComServer := CreateSQLAccServer;
```

```
if not ComServer.IsLogin then begin //if user hasn't logon to

SQL application
try
// Call login method
ComServer.Login('ADMIN', 'ADMIN', 'C:\Program

Files\eStream\SQLAccounting\DEFAULT.DCF',

'ACC-0001.FDB');
except
Result := False;
raise; //raise exception
end;
end;
end;

procedure OpenInvoice;
var lDocKey, lDocNo: variant;
begin
ComServer := CreateSQLAccServer;

//Create Com Server
if Login then begin
BizObject := ComServer.BizObjects.Find('SL_IV');

//Create Biz Object
lDocNo := 'IV-00001'; //Assigned Document Number here
lDocKey := BizObject.FindKeyByRef('DocNo', lDocNo);

//FindKeyByRef will find Document key based on lDocNo
BizObject.Params.Find('DocKey').Value := VarToStr(lDocKey); //Set

lDocKey to BizObject DocKey field

if not VarIsNull(lDocKey) then begin

//if lDocKey is not null mean key is found
BizObject.Open; //Open BizObject
ShowMessage('Document found.')
end else
ShowMessage('Document not found');


//not found
end;
end;
```

```
uses ComObj;

var ComServer, BizObject: variant;

function CreateSQLAccServer: variant;
```

```
begin
   Result := CreateOleObject('SQLAcc.BizApp'); end;


function Login: boolean;
```

```
begin

   Result := True; //Firstly, initialize to True ComServer :=
CreateSQLAccServer; if not ComServer.IsLogin then begin //if user hasn't
logon to SQL application try

   // Call login method ComServer.Login('ADMIN', 'ADMIN', '<font
face="Courier New" size="2">C:\Program
Files\eStream\SQLAccounting\DEFAULT.DCF</font>', 'ACC-0001.FDB');
except

   Result := False;

   raise; //raise exception end;

   end;

end;



procedure DeleteDetailRecord;

var lMainDataSet, lDetailDataSet, lDocNo, lDocKey: variant; begin

   ComServer := CreateSQLAccServer; //Create Com Server if Login then
begin BizObject := ComServer.BizObjects.Find('SL_IV'); //Create Biz
Object lDocNo := 'IV-00001'; //Assigned Document Number here lDocKey
:= BizObject.FindKeyByRef('DocNo',lDocNo); //FindKeyByRef will find
Document key based on lDocNo BizObject.Params.Find('DocKey').Value
:= VarToStr(lDocKey); //Set lDocKey to BizObject DocKey field

   lMainDataSet := BizObject.DataSets.Find('MainDataSet');
//lMainDataSet contains master data lDetailDataSet :=
BizObject.DataSets.Find('cdsDocDetail'); //lDetailDataSet contains detail
data
```

```
    if not VarIsNull(lDocKey) then begin //if lDocKey is not null mean key is found try

    BizObject.Open; //Open BizObject before edit BizObject.Edit; //Edit BizObject

    while (lDetailDataSet.RecordCount > 0) do begin //while record count > 0

    lDetailDataSet.First; //move to first record lDetailDataSet.Delete; //delete record end;

    BizObject.Save; //Save changes ShowMessage('Detail records deleted.'); except

    raise; //raise exception end;

    end else begin

    ShowMessage('Document not found.'); //not found end;

    end;

end;
```

How to check if the SQL Accounting is logon ?

Javascript / JScript

```
<script

language="JScript">
var ComServer;

function CreateSQLAccServer() {
ComServer = new ActiveXObject('SQLAcc.BizApp');
}

function IsLogin() {
CreateSQLAccServer();

/* Create Com Server */
var Result = ComServer.IsLogin; /* Return the logon status - true


or false */
alert (Result);
return (Result);
}
</script>
```

VBScript

```
<script

language="VBScript">
Function CreateSQLAccServer
Set CreateSQLAccServer = CreateObject("SQLAcc.BizApp")
End Function

Function IsLogin
Dim ComServer
Set ComServer = CreateSQLAccServer 'Create Com Server
IsLogin = ComServer.IsLogin


'Return the logon status - true or false
End Function
</script>
```

ASP

```
<script

language="JavaScript" runat="server">
var ComServer = new ActiveXObject("SQLAcc.BizApp"); /* Create Com
Server

*/
Result = ComServer.IsLogin;


/* Return the logon status - true or false */
Response.Write(Result);
</script>
```

Delphi

```
uses

ComObj;

var ComServer: variant;

procedure CheckLogin;
begin
ComServer = CreateOleObject('SQLAcc.BizApp'); // Create Com Server
ShowMessage(ComServer.IsLogin);


// Return the logon status - true or false
end;
```

How to retrieve a list of records from business object ?

Select Data has six parameters, and return a

result string :

```
Select(Fields,

Where, OrderBy, OutputFormat, Delimiter, OutputFile):
String;
```

There are 4 types of output format :

1.
AD

Output the data to string

separated by the Delimiter specified by users in **Delimiter** parameter.

If users set delimiter to comma, it will forms the data in comma separator

strings. Users may provide any characters to **Delimiter** parameter. In

order to use this type, users must remember to provide a file name in **OutputFile** parameter. The output will be saved in the file specified in

OutputFile. If no file name specified, the system will create a new

file, save into the file, and return the file name to users.

Examples:

Select all agent records :
`FileName =`

```
BizObject.Select("Code,Description", "", "Code", "AD", ",",
"C:\Data.txt");
```

Select agent records with condition :

`FileName =`

```
BizObject.Select("Code,Description", "Code='----'", "Code",
"AD", ",", "C:\Data.txt");
```

2.

AX

Output the data in XML

strings format, and save the result in a file specified by the users in **OutputFile** parameter. If no file name specified, the system will create

a new file, save into the file, and return the file name to users. Users do

no need to specify **Delimiter**, users may set it to empty string "".

Examples :

Select all agent records :

`FileName =`

```
BizObject.Select("Code,Description", "", "Code", "AX", "",
"C:\Data.xml");
```

Select agent records with condition :

`FileName =`

```
BizObject.Select("Code,Description", "Code='----'", "Code",
"AX", "", "C:\Data.xml");
```

3.

SD

Output the data to string

separated by the Delimiter specified by users in **Delimiter** parameter.

If users set delimiter to comma, it will forms the data in comma separator

strings. Users may provide any characters to **Delimiter** parameter. The

output will be returned to the users as a result strings. In this case,

users do no need to specify **OutputFile**, users may set it to empty

string "".

Examples :

Select all agent records :

```
Data =

BizObject.Select("Code,Description", "", "Code", "SD", ",",
"");
```

Select agent records with condition :

```
Data =

BizObject.Select("Code,Description", "Code='----'", "Code",
"SD", ",", "");
```

4.

SX

Output the data in XML

strings format and return the XML strings as result strings. In this case,

users do no need to specify **Delimiter and OutputFile**, users may set

it to empty string "".

Examples :

Select all agent records :

```
Data =

BizObject.Select("Code,Description", "", "Code", "SX", "",
"");
```

Select agent records with condition :

```
Data =

BizObject.Select("Code,Description", "Code='----'", "Code",
"SX", "", "");
```

Note

:-

- Double backslashes (\\)

is required for **OutputFile** in Javascript.
- In order to select all records, users may set **Where** to empty

string "".
- For SD and SX, data string will be returned by function **Select**.
- For AD and AX, file name will be returned by function **Select**. If

users did not specify **OutputFile** then function will return a file

name to users. If users specified
OutputFile then the file name returned by the function is

same as the file name specified by users.

JavaScript / JScript

```
<script

language="JScript">
var ComServer;

function CreateSQLAccServer() {
ComServer = new ActiveXObject("SQLAcc.BizApp");
if (!ComServer.IsLogin) { /* check whether user has logon */
```

```
ComServer.Login('ADMIN', 'ADMIN'); /* UserName, Password */
}
}

function SelectAgentData() {
var BizObject, FileName;
CreateSQLAccServer();
BizObject = ComServer.BizObjects.Find('Agent'); /* Create Agent Biz

Object */
/* Select Agent Data */
FileName = BizObject.Select("Code,Description", "Code='----'",
"Code",


"AD", ",", "C:\\AgentData.txt");
}
</script>
```

VBScript

```
<script

language="VBScript">
<!--
Function CreateSQLAccServer
Set CreateSQLAccServer = CreateObject("SQLAcc.BizApp")
End Function

Function SelectAgentData
Dim ComServer, BizObject, FileName
Set ComServer = CreateSQLAccServer 'Create Com Server
If not ComServer.IsLogin Then 'if

user hasn't logon to SQL application
ComServer.Login "ADMIN", "ADMIN"
END IF
Set BizObject = ComServer.BizObjects.Find("Agent") 'Create

Biz Object
'Select Agent Data
FileName = BizObject.Select("Code,Description", "Code='----'",
"Code",


"AD", ",", "C:\\AgentData.txt")
End Function
```

```
-->
</script>
```

ASP

```
<%@

LANGUAGE="JAVASCRIPT" %>
<%
var ComServer;

function CreateSQLAccServer() {
ComServer = new ActiveXObject('SQLAcc.BizApp');
}

function SelectAgentData() {
var BizObject, FileName; /* Firstly initialize to true */
CreateSQLAccServer();

/* Create Com Server */

if (!ComServer.IsLogin) {

/* if user hasn't logon to SQL application */
ComServer.Login('ADMIN', 'ADMIN', 'C:\\Program
Files\\eStream\\SQLAccounting\\DEFAULT.DCF',

'ACC-0001.FDB');
}

BizObject = ComServer.BizObjects.Find('Agent'); /* Create Agent Biz

Object */
/* Select Agent Data */
FileName = BizObject.Select("Code,Description", "Code='----'",
"Code",


"AD", ",", "C:\\windows\\temp\\AgentData.txt");
}
%>
```

Delphi

```
uses

ComObj;

var ComServer: variant;
```

```
function CreateSQLAccServer: variant;
begin
Result := CreateOleObject('SQLAcc.BizApp');
end;

procedure SelectAgentData;
var BizObject, FileName: variant;
begin
ComServer := CreateSQLAccServer;
if not ComServer.IsLogin then begin //if user hasn't logon to SQL

application
ComServer.Login('ADMIN', 'ADMIN');
end;
BizObject := ComServer.BizObjects.Find('Agent'); // Create Agent
Biz

Object
// Select Agent Data
FileName := BizObject.Select('Code,Description', 'Code=''----''',
'Code',


'AD', ',', 'C:\AgentData.txt');
end;
```

How to retrieve data from Customer Invoice Listing Report?

Step 1 : E Stream Application Software (SQL Financial Accounting) must be installed and run.

Step 2 : Users have to write some code to create COM server.

Step 3 : Logon to SQL Financial Accounting.

Step 4 : Find and create report object. e.g, Customer.IV.RO . (For more report objects, please refer to Report Objects under Application Objects).

Step 5 : Spool report parameters.

Step 6 : Perform report calculations. e.g RptObject.CalculateReport.

Step 7 : Retrieve the output. There are few ways to retrieve output,

a) User may save the output to XML file by provide a file name.

Example :

```
RptObject.SaveAsXML('C:\Output.xml');
```

b) Loop a particular dataset in report object to get the output.

Example :

```
R =
ComServer.RptObjects.Find('Common.Agent.RO');

R.CalculateReport;
D = R.DataSets.Find('cdsMain');

lDataSet.First();

while (!lDataSet.eof) {

    edOutput.value =
edOutput.value + lDataSet.FindField('Code').AsString + '\n';
    lDataSet.Next();

}
```

c) Pump the data into a local ClientDataSet. (Delphi only)

Example: lDataSet.Data :=

```
RptObjects.DataSets.Find('cdsMain').Data;
```

Example :

Click here to Calculate Report

Javascript / JScript

```
<script

language="JScript">
var ComServer;

function CreateSQLAccServer() {
ComServer = new ActiveXObject("SQLAcc.BizApp");
if (!ComServer.IsLogin) { /* check whether user has logon */
ComServer.Login('ADMIN', 'ADMIN'); /* UserName, Password */
}
}

function GetData() {
var RptObject, lDataSet, lDateFrom, lDateTo;
edOutput.value = '';
/* Step 2: Create Com Server object */
CreateSQLAccServer();

/* Step 4: Find and Create the Report Objects */
RptObject = ComServer.RptObjects.Find('Customer.IV.RO');

/* Step 5: Spool parameters */
RptObject.Params.Find('AllAgent').Value = false;
RptObject.Params.Find('AllArea').Value = true;
RptObject.Params.Find('AllCompany').Value = true;
RptObject.Params.Find('AllCurrency').Value = true;
RptObject.Params.Find('AllDocument').Value = true;
RptObject.Params.Find('AllPaymentMethod').Value = true;
RptObject.Params.Find('AllDocProject').Value = true;
RptObject.Params.Find('AllItemProject').Value = true;
RptObject.Params.Find('SelectDate').Value = true;
RptObject.Params.Find('DateFrom').Value = 0;
RptObject.Params.Find('DateTo').Value = 0;
RptObject.Params.Find('AgentData').Value = '----\nHALIM'; /* filter

agents, separated by \n when more than one agents */

/* Step 6: Perform Report calculation */
RptObject.CalculateReport();
lDataSet = RptObject.DataSets.Items(0);

/* Step 7 Retrieve the output */
lDataSet.First();
while (!lDataSet.eof) {
edOutput.value = edOutput.value +

lDataSet.FindField('DocNo').AsString + ' ' +

lDataSet.FindField('DocDate').AsString + ' '+
lDataSet.FindField('CompanyName').AsString + ' ' +


lDataSet.FindField('DocAmt').AsString + '\n';
lDataSet.Next();
}
}
</script>
```

VBScript

```
<script

language="VBScript">
<!--
Function CreateSQLAccServer
Set CreateSQLAccServer = CreateObject("SQLAcc.BizApp")
End Function

Function GetVBData
Dim ComServer, RptObject, lDataSet, lDateFrom, lDateTo
edOutput.value = ""
'Step 2: Create Com Server object
Set ComServer = CreateSQLAccServer 'Create Com Server
If not ComServer.IsLogin Then 'if user hasn't logon to SQL application
ComServer.Login "ADMIN", "ADMIN"
END IF
'Step 4: Find and Create the Report Objects
Set RptObject = ComServer.RptObjects.Find("Customer.IV.RO")

'Step 5: Spool parameters
RptObject.Params.Find("AllAgent").Value = false
RptObject.Params.Find("AllArea").Value = true
RptObject.Params.Find("AllCompany").Value = true
RptObject.Params.Find("AllCurrency").Value = true
RptObject.Params.Find("AllDocument").Value = true
RptObject.Params.Find("AllPaymentMethod").Value = true
RptObject.Params.Find("AllDocProject").Value = true
RptObject.Params.Find("AllItemProject").Value = true
RptObject.Params.Find("SelectDate").Value = true

lDateFrom = CDate("July 1, 2004")
lDateTo = CDate("December 31, 2004")

RptObject.Params.Find("DateFrom").Value = lDateFrom
RptObject.Params.Find("DateTo").Value = lDateTo
RptObject.Params.Find("AgentData").Value = "----" & vbCRLF & "HALIM"

'Separated by vbCRLF when more than one agents

'Step 6: Perform Report calculation
RptObject.CalculateReport()
Set lDataSet = RptObject.DataSets.Find("cdsMain")

'Step 7 Retrieve the output
lDataSet.First
While (not lDataSet.eof)
edOutput.value = edOutput.value &

lDataSet.FindField("DocNo").AsString & " " &

lDataSet.FindField("DocDate").AsString &_

" " & lDataSet.FindField("CompanyName").AsString & " " &

lDataSet.FindField("DocAmt").AsString & vbCRLF
lDataSet.Next
Wend
End Function
-->
</script>
```

ASP

```
<%@

LANGUAGE="JAVASCRIPT" %>
<%
var ComServer;

function CreateSQLAccServer() {
ComServer = new ActiveXObject("SQLAcc.BizApp");
if (!ComServer.IsLogin) { /* check whether user has logon */
ComServer.Login('ADMIN', 'ADMIN', 'C:\\Program Files\\eStream\\SQLAccounting\\DEFAULT.DCF',

'ACC-0001.FDB');
}
}

function GetData() {
var RptObject, lDataSet, lStr;
/* Step 2: Create Com Server object */
CreateSQLAccServer();

/* Step 4: Find and Create the Report Objects */
RptObject = ComServer.RptObjects.Find('Customer.IV.RO');

/* Step 5: Spool parameters */
RptObject.Params.Find('AllAgent').Value = false;
RptObject.Params.Find('AllArea').Value = true;
RptObject.Params.Find('AllCompany').Value = true;
RptObject.Params.Find('AllCurrency').Value = true;
RptObject.Params.Find('AllDocument').Value = true;
RptObject.Params.Find('AllPaymentMethod').Value = true;
RptObject.Params.Find('AllDocProject').Value = true;
RptObject.Params.Find('AllItemProject').Value = true;
RptObject.Params.Find('SelectDate').Value = false;
RptObject.Params.Find('DateFrom').Value = 0;
RptObject.Params.Find('DateTo').Value = 0;
RptObject.Params.Find('AgentData').Value = '----\nHALIM'; /* filter

agents, separate by \n when more than one agents */

/* Step 6: Perform Report calculation */
RptObject.CalculateReport();
lDataSet=RptObject.DataSets.Items(0);

/* Step 7 Retrieve the output */
lStr = '';
lDataSet.First();
while (!lDataSet.eof) {
lStr = lStr + lDataSet.FindField('DocNo').AsString + ' ' +

lDataSet.FindField('DocDate').AsString + ' ' +
lDataSet.FindField('CompanyName').AsString + ' ' +


lDataSet.FindField('DocAmt').AsString + '<br>';
lDataSet.Next();
}


Response.Write (lStr);
}
%>
```

Delphi

```
uses

ComObj;

var ComServer: variant;
```

```
function CreateSQLAccServer: variant;
begin
Result := CreateOleObject('SQLAcc.BizApp');
end;

procedure GetData;
var lRptObject: variant;
begin
ListBox1.Clear;

// Step 2: Create Com Server object
ComServer := CreateSQLAccServer;

// Step 3: Login as usual
ComServer.Login('ADMIN', 'ADMIN', '', '');

// Step 4: Find and Create the Report Objects
lRptObject := ComServer.RptObjects.Find('Customer.IV.RO');

// Step 5: Spool the parameters
lRptObject.Params.Find('AllAgent').Value := False;
lRptObject.Params.Find('AllArea').Value := True;
lRptObject.Params.Find('AllCompany').Value := True;
lRptObject.Params.Find('AllCurrency').Value := True;
lRptObject.Params.Find('AllDocument').Value := True;
lRptObject.Params.Find('AllPaymentMethod').Value := True;
lRptObject.Params.Find('AllDocProject').Value := True;
lRptObject.Params.Find('AllItemProject').Value := True;
lRptObject.Params.Find('SelectDate').Value := True;
lRptObject.Params.Find('DateFrom').Value := StrToDate('01/07/2004');
lRptObject.Params.Find('DateTo').Value := StrToDate('31/12/2004');
lRptObject.Params.Find('AgentData').Value := '----'#13#10'HALIM';


//Separated by #13#10 when more than one agents


// Step 6: Perform Report calculation
lRptObject.CalculateReport;


// Step 7 Retrieve the output
ClientDataSet1.Data := lRptObject.DataSets.Items[0].Data;
end;
```