

User Manual

sODBC Plug-in



Somi·t 

Small Office Management Integrated Tools
Solutions de gestion intégrée pour l'entreprise

USER MANUAL OF sODBC PLUG-IN FOR FILEMAKER PRO

By Henri and Mathieu Frappier

©2014 Somi-t inc. All right reserved.



TABLE OF CONTENT

PREAMBLE	8
INTRODUCTION	8
Summary.....	8
Plug-in installation.....	8
Custom functions installation.....	9
Sample files.....	9
EXTERNAL FUNCTIONS	10
List of external functions for sODBC plug-in	10
sODBC_Open("" ; ConnectionString).....	10
sODBC_Close("").....	11
sODBC_Query("" ; Query).....	11
sODBC_NonQuery("" ; Query)	11
sODBC_QueryA("" ; Query ; Get (FileName) ; ScriptName).....	11
sODBC_NonQueryA("" ; Query ; Get (FileName) ; ScriptName)	12
sODBC_FieldSeparator	12
sODBC_RecordSeparator.....	12
sODBC_GetSchema("" ; SchemaName ; AttributeFilter)	13
sODBC_Register(Name ; Key)	13
sODBC_Version({ Format })	14
Text Encoding	14
sODBC error codes and meaning.....	15
CUSTOM FUNCTIONS – ACOMBA.....	16
List of custom functions for the Acomba software	16



Aco.AccountStatement.ASAmountDue (CuNumber ; InInvoiceNumber)	17
Aco.Charter.ChCategory (ChNumber)	17
Aco.Charter.ChDescription (ChNumber)	17
Aco.ControlCustomer.Interest	17
Aco.Customer.Exist (CuNumber)	17
Aco.Customer.NewUpdate (CuNumber ; CuSortKey ; CuName ; CuCareOf ; CuAddress ; CuCity ; CuPostalCode ; CuISOCountryCode ; CuPhoneNumber1 ; CuPhoneExtension1 ; CuPhoneNumber2 ; CuEMail ; CuWebAddress ; CuReceivable ; CuTaxGroupNumber ; CuTermNumber ; CuInterest ; CuMarginLimit ; CuComment)	18
Aco.CustomerPaymentLine.RecCardPos (CPInvoiceARCP)	18
Aco.InvoiceAR.Exist (InCustomerSupplierNumber ; InInvoiceNumber)	18
Aco.InvoiceAP.Exist (InSupplierNumber ; InInvoiceNumber)	19
Aco.Supplier.Exist (SuNumber)	19
Aco.Supplier.NewUpdate (SuNumber ; SuSortKey ; SuName ; SuCareOf ; SuAddress ; SuCity ; SuPostalCode ; SuISOCountryCode ; SuPhoneNumber1 ; SuPhoneExtension1 ; SuPhoneNumber2 ; SuEMail ; SuWebAddress ; SuPurchaseAccountNumber ; SuTaxGroupNumber ; SuPaymentTermNumber ; SuComments)	19
Aco.SupplierTrans.Balance (SuNumber ; InInvoiceNumber)	20
Aco.TransAR.Begin	20
Aco.TransAR.Cancel	20
Aco.TransAR.End	20
Aco.TransARHeader.New (InInvoiceNumber ; InCustomerSupplierNumber ; InTransactionType ; InDate ; InAllPaid ; InTaxGroupNumber ; InReference ; InDescription ; TANumPosts)	20
Aco.TransARDetail.Update (PACharterNumber ; PAAmount ; PaDescription ; PAIsTax ; TANum)	21
Aco.TransAP.Begin	21
Aco.TransAP.Cancel	21
Aco.TransAP.End	21



Aco.TransAPHeader.New (InInvoiceNumber ; InSupplierNumber ; InType ; InDate ; InAmount ; InSubTotal ; InReference ; InDescription ; TAnumPosts)	21
Aco.TransAPDetail.Update (PACharterNumber ; PAAmount ; PaDescription ; PAIsTax ; TaNum)	22
Aco.Utills.CityProvince (City ; Province)	22
Aco.Utills.DebitCreditAR (Amount ; TransactionType ; TaNum)	22
Aco.Utills.DebitCreditAP (Amount ; TransactionType ; TaNum)	22
Aco.Utills.DecimalSeparator (Number)	22
Aco.Utills.ESQ (Text)	23
Aco.Utills.ISOCountryCodeEn (Country)	23
Aco.Utills.ISOCountryCodeFr (Country)	23
Aco.Utills.Phone (Text)	23
CUSTOM FUNCTIONS – SAGE 50 (Simple Comptable)	24
List of the custom functions for the Sage 50 software	24
s50.sASUserld	25
s50sASOrgld.....	25
s50.Customer.Exist (sName).....	25
s50.Customer.New (sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; dCrLimit ; sEmail ; sWebSite ; nLangPref ; dtSince ; sMemo).....	25
s50.Customer.Update (lld ; sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; sEmail ; sWebSite ; sMemo)	26
s50.Invoice.New (sName ; lCustomrld ; dtDate ; sReference ; sTaxCode ; dSubTotal ; dTotal ; dTotalTaxA ; lAccountTaxA ; dTotalTaxB ; lAccountTaxB ; sComment)	27
s50.InvoiceDetail.New (sInvoiceNumber ; sPartCode ; lQuantity ; dPrice ; sTaxCode ; dAmountTaxA ; lAccountTaxA ; dAmountTaxB ; lAccountTaxB)	27
s50.InvoiceDetailAccount.New (sInvoiceNumber ; lAccount ; sDescription ; lQuantity ; dPrice ; sTaxCode ; dAmountTaxA ; lAccountTaxA ; dAmountTaxB ; lAccountTaxB)	28



s50.Invoice.Freight (sInvoiceNumber ; dFreightAmount ; sFreightTaxCode ; dFreightTaxA ; lFreightAccountTaxA ; dFreightTaxB ; lFreightAccountTaxB)	28
s50.Invoice.Commit.....	28
s50.Invoice.Rollback	28
s50.InvoicePayment.New (sInvoiceNumber ; lCustomrId ; dtDate ; dAmount ; lType ; lBankAccount ; sReference ; sComment)	29
s50.Purchase.New (sName ; lVendorId ; dtDate ; sReference ; sDescription ; sTaxCode ; dSubTotal ; lChargeAccount ; dTotalTaxA ; lAccountTaxA ; dTotalTaxB ; lAccountTaxB ; sNonTaxableDescription ; dNonTaxableAmount ; lNonTaxableAccount ; dTotal ; sComment)	29
s50.Vendor.exist (sName).....	30
s50.Vendor.New (sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; sEmail ; sWebSite ; nLangPref ; dtSince ; sMemo) .	30
s50.Vendor.Update (lId ; sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; sEmail ; sWebSite ; sMemo) .	30
s50.Utills.CheckSqlSyntax (Sql)	31
s50.Utills.Date (fmDate)	31
s50.Utills.DateFirstOfLastMonth (fmDate).....	31
s50.Utills.ESQ (Text)	31
s50.Utills.FormatDecimal (Number).....	31
s50.Utills.getNextId (TableId)	32
s50.Utills.getNextInvoiceNumber.	32
s50.Utills.getNextReceiptNumber.....	32
s50.Utills.getInvoiceBalance (sInvoiceNumber ; lCustomrId) .	32
s50.Utills.getPurchaseBalance (sInvoiceNumber ; lVendorId).....	33
s50.Utills.getTaxCodeId (sTaxCode)	33
s50.Utills.getTaxInfo (sTaxCode ; lTaxAccountNumber) .	33
s50.Utills.setNextId (TableId)	33



s50.Utils.setNextInvoiceNumber (Number)	33
s50.Utils.setNextReceiptNumber (Number).....	33
REFERENCES	34
Technical notes.....	34
Troubleshooting	35
About FileMaker, Inc.	35



PREAMBLE

- This manual assumes you have basic knowledge about the FileMaker Pro database development platform, the use of plug-in with external functions and custom functions.
- This manual, as well as files that are included, are provided under license of agreement and may not be copied, photocopied, reproduced, translated or converted to electronic format, in whole or in part without written permission in advance from Somi-t Inc.
- The "FileMaker" term used in the descriptions in this manual generally refers to software versions FileMaker Pro and FileMaker Pro Advanced.
- Somi-t Inc. is a proud FileMaker Business Alliance (FBA) member since 2006.
- Does Somi-t Inc. assumes no liability for errors or inaccuracies in this manual.
- Do not hesitate to contact us at **plugin@somi-t.com**
- Web site: **http://www.somi-t.com** or **http://www.FollowTheGuide.ca**
- Last update made to this manual: November 24th, 2014.

INTRODUCTION

Summary

- **The sODBC plug-in allows FileMaker Pro solutions to communicate with any ODBC data source installed on a Windows or Mac OS X system.**
- After installation of the sODBC plug-in, it will be easy to use all the eleven external functions and the basic SQL functions: Select, Update, Delete and Insert.
- The sODBC plug-in allows you to control the actions of opening and closing of the ODBC connection unlike "Execute SQL" script that opens and closes the connection automatically without consulting you.
- It uses the syntax of the SQL language used by the ODBC driver of the target database. Example: MySQL syntax for MySQL software.

Plug-in installation on Windows

- Summary of installation steps:
 - 1) The sODBC.fmx plug-in must be located under C:\Program Files\FileMaker\FileMaker Pro [version]\Extensions\ folder before opening FileMaker Pro.
 - 2) The ODBC driver for the source software must be installed on the local computer according to the manufacturer's recommendations.
 - 3) Verify if the data source is available through the Control Panel\ODBC Data Source\ [Drivers tab]
 - 4) Use User DSN or System DSN to add ODBC source to the list.
 - 5) The ODBC source is now available for sODBC using external functions from FileMaker.



Plug-in installation on Mac OS X

- Summary of installation steps:
 - 1) Install iODBC Framework in Mac OS X available at this address
<http://www.iodbc.org/downloads/iODBC/iodbc-sdk-3.52.7-macosx-10.5.dmg>
 - 2) The sODBC. fmpugin plug-in must be located under /Application/FileMaker Pro [version]/Extensions/ folder before opening FileMaker Pro.
 - 3) The ODBC driver for the application must be installed on the computer according to the manufacturer's recommendations.
 - 4) Although the ODBC connection manager "ODBC Manager" is available in the /Application/Utilities, we recommend that you install "OpenLink ODBC Administrator" available here at this address
<http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/Downloads>
 - 5) Use "User DSN" or preferably the "System DSN" to add (ADD) the source to the list.
 - 6) The ODBC source is now available using external functions provided with the sODBC plug-in for FileMaker Pro.

Custom functions installation

- Only for users of FileMaker Pro Advanced! You can import or copy and paste our custom functions in a different file from our FileMaker included with the set corresponding to the target software.
- It is not possible for users of FileMaker Pro to import the custom functions.
- Steps to run to import the custom functions:
 - 1) Open the destination file you want to import the custom functions;
 - 2) On the File menu, select > Manage> Custom Functions;
 - 3) In the dialog Manage custom functions, click Import;
 - 4) Open the file that contains the custom functions to import;
 - 5) If an account name and password are asked to open the file, enter them in the dialog box and click OK;
 - 6) In the Import dialog box custom functions, select functions to import then click OK;
 - 7) The Import summary dialog box appears;
 - 8) Click Open Log File button to view the log file, or click OK to close the Import summary dialog box;
 - 9) Close the dialog box Manage Custom Functions.

Sample files

- The FileMaker database containing custom functions, are provided only in sets of custom functions sold separately from sODBC plug-in for FileMaker. For example, with all the custom features for Acomba and Sage 50.



EXTERNAL FUNCTIONS

List of external functions for sODBC plug-in

- sODBC_Open("" ; ConnectionString)
- sODBC_Close("")
- sODBC_Query("" ; Query)
- sODBC_NonQuery("" ; Query)
- sODBC_QueryA("" ; Query ; Get (FileName) ; ScriptName)
- sODBC_NonQueryA("" ; Query ; Get (FileName) ; ScriptName)
- sODBC_FieldSeparator
- sODBC_RecordSeparator
- sODBC_GetSchema("" ; SchemaName ; AttributeFilter)
- sODBC_Register(Name ; Key)
- sODBC_Version({ Format })

sODBC_Open("" ; ConnectionString)

- Description: Opens the connection to the ODBC data source.
- "ConnectionString" Parameter: Connection parameter to the ODBC data source.
 - o Example:
 - sODBC_Open("");"DSN=MonAcomba;UID=Supervisor;PWD=DEMO;"
 - sODBC_Open("");"DSN=MonFp7;UID=test;PWD=test;"
 - sODBC_Open("");"DSN=Northwind;"
 - sODBC_Open("");"Driver={Microsoft Access Driver (*.mdb)};DBQ=c:\bin\Northwind.mdb"
- Result: Returns 1 if the connection was successfully established otherwise returns the error code preceded by "!!".



sODBC_Close("")

- Description: Closes the connection to the ODBC data source.
- Parameter "": Not used, stay empty with "".
- Result: Returns 1 if the connection was successfully closed otherwise returns the error code preceded by "!!".

sODBC_Query("" ; Query)

- Description: Execute a SELECT SQL query.
- Parameter "": Not used under Windows, stay empty with "".
For the Mac version, you can change the text encoding of input and output. See "Text Encoding" section of this document.
- "Query" Parameter: SQL query to execute.
 - Example:
 - sODBC_Query("");"SELECT * FROM Contacts WHERE Status='Active'")
- Result: Returns the query result. If more than one field is selected, the field separator will be used. Also, if more than one record is selected, the record separator will be used. See sODBC_FieldSeparator and sODBC_RecordSeparator.

sODBC_NonQuery("" ; Query)

- Description: Execute a INSERT, UPDATE or DELETE SQL query
- Parameter "": Not used under Windows, stay empty with "".
For the Mac version, you can change the text encoding of input and output. See "Text Encoding" section of this document.
- "Query" Parameter: SQL query to execute.
 - Example:
 - sODBC_NonQuery("");"UPDATE Contacts SET Status='Inactive' WHERE ID=101")
 - sODBC_NonQuery("");"INSERT INTO Contacts (FirstName, LastName) VALUES ('Linda','L''Amour')")
- Result: Returns the number of records modified by the query.

sODBC_QueryA("" ; Query ; Get (FileName) ; ScriptName)

WINDOWS ONLY

- Description: Same as sODBC_Query but the query runs in its own execution thread, freeing the user interface.
- Parameter "": Not used, stay empty with "".



- "Query" Parameter: SQL query to execute.
- "Get (FileName)" Parameter: File name where the script defined by ScriptName.
- "ScriptName" Parameter: FileMaker script name to run after executing the SQL query.
- Result: None
- Note: To access the query result inside the script ScriptName, use the Get (ScriptParameter) function from FileMaker Pro.

sODBC_NonQueryA("" ; Query ; Get (FileName) ; ScriptName)

WINDOWS ONLY

- Description: Same as sODBC_NonQuery but the query runs in its own execution thread, freeing the user interface.
- Parameter "": Not used, stay empty with "".
- "Query" Parameter: SQL query to execute.
- "Get (FileName)" Parameter: File name where the script is defined by ScriptName.
- "ScriptName" Parameter: FileMaker script name to run after executing the SQL query.
- Result: None
- Note: To access the query result inside the script ScriptName, use the Get (ScriptParameter) function of FileMaker.

sODBC_FieldSeparator

- Description: Returns the field delimiter used and/or assigns the field separator to use.
- Parameter: String(s) to be used as a separator.
 - o Example:
 - sODBC_FieldSeparator
 - sODBC_FieldSeparator(Char(31))
 - sODBC_FieldSeparator("<MySeparator1>")
- Result: Returns the field separator used.
- Note: The default field separator is the ASCII code 31, equivalent to the function Char (31) of FileMaker.

sODBC_RecordSeparator

- Description: Returns the record delimiter used and/or assigns the record separator to use.
- Parameter: String(s) to be used as a separator.
 - o Example:
 - sODBC_RecordSeparator



- sODBC_RecordSeparator(Char(30))
- sODBC_RecordSeparator("<MySeparator2>")
- Result: Returns the record separator used.
- Note: The default field separator is the ASCII code 30, equivalent to the function Char (30) of FileMaker.

sODBC_GetSchema("" ; SchemaName ; AttributeFilter)

WINDOWS ONLY

- Description: Provides access to Schema information from the data source.
- "SchemaName" Parameter: Name of Schema.
 - o Possible values:
 - "Restrictions"
 - "Tables"
 - "Columns"
 - "Views"
 - "ViewColumns"
 - "Users"
 - "DataTypes"
 - "ReservedWords"
 - "Procedures"
 - "ProcedureParameters"
 - "Indexes"
 - "IndexColumns"
 - "ForeignKeys"
 - "DataSourceInformation"
 - "MetaDataCollections"
- "AttributeFilter" Parameter: Filter results by attribute name.
 - o Example
 - sODBC_GetSchema("", "Tables", "TABLE_NAME")
- Result: Returns schema information as a value list. If the filter attribute is not used, the attributes are separated by the field separator. See sODBC_FieldSeparator.

sODBC_Register(Name ; Key)

- Description: Register the plug-in. His condition switches from Demonstrator to Registered.
- "Name" Parameter: Organization name you provided to Somi-t when you purchased the plug-in.
- "Key" Parameter: Registration key provided by Somi-t.



sODBC_Version({ Format })

- Description: Returns the version number and/or the name of the plug-in.
- Parameter: "", "short", "long"

Text Encoding

The Mac version of the plug-in sODBC offers the possibility to change the text encoding.

The default encoding used by the plug-in sODBC Mac version is iso_8859_1. To change the encoding of the input text, use the keyword "inputEncoding =" followed by the desired encoding. To change the encoding of the output text, use the keyword "outputEncoding =" followed by the desired encoding.

Examples with sODBC_Query() :

- sODBC_Query("inputEncoding=UTF8";"SELECT * FROM Province WHERE Nom = 'Québec'")
- sODBC_Query("outputEncoding=UTF8";"SELECT * FROM Province WHERE Nom = 'Québec'")
- sODBC_Query("inputEncoding=UTF8;outputEncoding=UTF8";"SELECT * FROM Province WHERE Nom = 'Québec'")

Example with sODBC_NonQuery() :

- sODBC_NonQuery("inputEncoding=UTF8";"UPDATE Province SET nom = 'Québec' WHERE nom = 'Quebec'")



List of available encoding:

Native	UTF8	ASCII_DOS	ASCII_Windows	ASCII_Mac
ISO_8859_1	ISO_8859_2	ISO_8859_3	ISO_8859_4	ISO_8859_5
ISO_8859_6	ISO_8859_7	ISO_8859_8	ISO_8859_9	ISO_8859_15
ShiftJIS_Mac	ShiftJIS_Win	Korean_Mac	Korean_Win	Korean_Johab
ChineseTrad_Mac	ChineseTrad_Win	ChineseSimp_Mac	ChineseSimp_Win	Cyrillic_Mac
Cyrillic_Win	CentralEurope_Mac	EasternEurope_Win	Turkish_Mac	Turkish_Win
Baltic_Win	Arabic_Mac	Arabic_Win	Greek_Mac	Greek_Win
Hebrew_Mac	Hebrew_Win			

sODBC error codes and meaning

- !!-101 : Wrong license, (_Open)
- !!-1 : Not connected (_Open, _Query, _NonQuery, _GetSchema)
- !!-2 : Empty parameter (_Open, _Query, _NonQuery, _GetSchema)
- !!-3 : Connection open specific error (_Open)
- !!-4 : Query command specific error (_Query)
- !!-5 : Query read specific error (_Query)
- !!-6 : NonQuery command specific error (_NonQuery)
- !!-7 : NonQuery read specific error (_NonQuery)
- !!-8 : Connection closing specific error (_Close)
- !!-10 : GetSchema specific error (_GetSchema)



CUSTOM FUNCTIONS – ACOMBA

List of custom functions for the Acomba software

- Aco.AccountStatement.ASAmountDue (CuNumber ; InInvoiceNumber)
- Aco.Charter.ChCategory (ChNumber)
- Aco.Charter.ChDescription (ChNumber)
- Aco.ControlCustomer.Interest
- Aco.Customer.Exist (CuNumber)
- Aco.Customer.NewUpdate (CuNumber ; CuSortKey ; CuName ; CuCareOf ; CuAddress ; CuCity ; CuPostalCode ; CuISOCountryCode ; CuPhoneNumber1 ; CuPhoneExtension1 ; CuPhoneNumber2 ; CuEmail ; CuWebAddress ; CuReceivable ; CuTaxGroupNumber ; CuTermNumber ; CuInterest ; CuMarginLimit ; CuComment)
- Aco.CustomerPaymentLine.RecCardPos (CPIInvoiceARCP)
- Aco.InvoiceAP.Exist (InSupplierNumber ; InInvoiceNumber)
- Aco.InvoiceAR.Exist (InCustomerSupplierNumber ; InInvoiceNumber)
- Aco.Supplier.Exist (SuNumber)
- Aco.Supplier.NewUpdate (SuNumber ; SuSortKey ; SuName ; SuCareOf ; SuAddress ; SuCity ; SuPostalCode ; SuISOCountryCode ; SuPhoneNumber1 ; SuPhoneExtension1 ; SuPhoneNumber2 ; SuEmail ; SuWebAddress ; SuPurchaseAccountNumber ; SuTaxGroupNumber ; SuPaymentTermNumber ; SuComments)
- Aco.SupplierTrans.Balance (SuNumber ; InInvoiceNumber)
- Aco.TransAP.Begin
- Aco.TransAP.Cancel
- Aco.TransAP.End
- Aco.TransAPDetail.Update (PACharterNumber ; PAAmount ; PaDescription ; PAIsTax ; TNum)
- Aco.TransAPHeader.New (InInvoiceNumber ; InSupplierNumber ; InType ; InDate ; InAmount ; InSubTotal ; InReference ; InDescription ; TNumPosts)
- Aco.TransAR.Begin
- Aco.TransAR.Cancel
- Aco.TransAR.End
- Aco.TransARDetail.Update (PACharterNumber ; PAAmount ; PaDescription ; PAIsTax ; TNum)
- Aco.TransARHeader.New (InInvoiceNumber ; InCustomerSupplierNumber ; InTransactionType ; InDate ; InAllPaid ; InTaxGroupNumber ; InReference ; InDescription ; TNumPosts)
- Aco.Utils.CityProvince (City ; Province)
- Aco.Utils.DebitCreditAR (Amount ; TransactionType ; TNum)



- Aco.Utills.DebitCreditAP (Amount ; TransactionType ; TaNum)
- Aco.Utills.DecimalSeparator (Number)
- Aco.Utills.ESQ (Text)
- Aco.Utills.ISOCountryCodeEn (Country)
- Aco.Utills.ISOCountryCodeFr (Country)
- Aco.Utills.Phone (Text)

Aco.AccountStatement.ASAmountDue (CuNumber ; InInvoiceNumber)

- Description: Returns the balance of an invoice.
- "CuNumber" Parameter: Customer number
- "InInvoiceNumber" Parameter: Customer invoice number

Aco.Charter.ChCategory (ChNumber)

- Description: Returns the category of the account number.
- "ChNumber" Parameter: Account number

Aco.Charter.ChDescription (ChNumber)

- Description: Returns the description of the account number.
- "ChNumber" Parameter: Account number

Aco.ControlCustomer.Interest

- Description: Returns the default interest rate.

Aco.Customer.Exist (CuNumber)

- Description: Returns the customer internal number of Acomba (RecCardPos) if the customer exist (>0).
- "CuNumber" Parameter: Customer number



[Aco.Customer.NewUpdate \(CuNumber ; CuSortKey ; CuName ; CuCareOf ; CuAddress ; CuCity ; CuPostalCode ; CuISOCountryCode ; CuPhoneNumber1 ; CuPhoneExtension1 ; CuPhoneNumber2 ; CuEMail ; CuWebAddress ; CuReceivable ; CuTaxGroupNumber ; CuTermNumber ; CuInterest ; CuMarginLimit ; CuComment \)](#)

- Description: Add or update information from a customer.
- "CuNumber" Parameter: Customer number
- "CuSortKey" Parameter: Customer sort key
- "CuName" Parameter: Customer name
- "CuCareOf" Parameter: Contact name (A/S)
- "CuAddress" Parameter: Customer address
- "CuCity" Parameter: Customer city (see Aco.Utils.CityProvince())
- "CuPostalCode" Parameter: Customer postal code
- "CuISOCountryCode" Parameter: Customer country (two-letter code, see Aco.Utils.ISOCountryCodeEn())
- "CuPhoneNumber1" Parameter: Phone number of #1 customer
- "CuPhoneExtension1" Parameter: Extension of phone number #1
- "CuPhoneNumber2" Parameter: Phone number of #2 customer
- "CuEMail" Parameter: Customer email address
- "CuWebAddress" Parameter: Customer website
- "CuReceivable" Parameter: Number of the account receivable for that customer
- "CuTaxGroupNumber" Parameter: Acomba Group Number taxes for that customer
- "CuTermNumber" Parameter: Term of payment for that customer
- "CuMarginLimit" Parameter: Line of credit for that customer
- "CuComment" Parameter: Comment about that customer

[Aco.CustomerPaymentLine.RecCardPos \(CPIInvoiceARCP \)](#)

- Description: Returns the number of internal payment of Acomba (RecCardPos).
- "CPIInvoiceARCP" Parameter: Internal invoice number of Acomba (RecCardPos)

[Aco.InvoiceAR.Exist \(InCustomerSupplierNumber ; InInvoiceNumber \)](#)

- Description: Returns the number of customer internal invoice of Acomba (>0).
- "InCustomerSupplierNumber" Parameter: Customer number
- "InInvoiceNumber" Parameter: Customer invoice number



Aco.InvoiceAP.Exist (InSupplierNumber ; InInvoiceNumber)

- Description: Returns the number of supplier internal invoice of Acomba (>0).
- "InSupplierNumber" Parameter: Supplier number
- "InInvoiceNumber" Parameter: Supplier invoice number

Aco.Supplier.Exist (SuNumber)

- Description: Returns the supplier internal number of Acomba (RecCardPos) if the customer exist (>0).
- "SuNumber" Parameter: Supplier number

Aco.Supplier.NewUpdate (SuNumber ; SuSortKey ; SuName ; SuCareOf ; SuAddress ; SuCity ; SuPostalCode ; SuISOCountryCode ; SuPhoneNumber1 ; SuPhoneExtension1 ; SuPhoneNumber2 ; SuEMail ; SuWebAddress ; SuPurchaseAccountNumber ; SuTaxGroupNumber ; SuPaymentTermNumber ; SuComments)

- Description: Add or update information from a supplier.
- "SuNumber" Parameter: Supplier number
- "SuSortKey" Parameter: Supplier sort key
- "SuName" Parameter: Supplier name
- "SuCareOf" Parameter: Contact name (A/S)
- "SuAddress" Parameter: Supplier address
- "SuCity" Parameter: Supplier city (see Aco.Utils.CityProvince())
- "SuPostalCode" Parameter: Supplier postal code
- "SuISOCountryCode" Parameter: Supplier country (two-letter code, see Aco.Utils.ISOCountryCodeEn())
- "SuPhoneNumber1" Parameter: Phone number of #1 supplier
- "SuPhoneExtension1" Parameter: Extension of phone number #1
- "SuPhoneNumber2" Parameter: Phone number of #2 supplier
- "SuEMail" Parameter: Supplier email address
- "SuWebAddress" Parameter: Supplier website
- "SuPurchaseAccountNumber" Parameter: Number of the account receivable for that supplier
- "SuTaxGroupNumber" Parameter: Acomba taxes group number for that supplier
- "SuPaymentTermNumber" Parameter: Term of payment for that supplier
- "SuComments" Parameter: Comment about that supplier



Aco.SupplierTrans.Balance (SuNumber ; InInvoiceNumber)

- Description : Returns the balance to pay for a supplier transaction.
- "SuNumber" Parameter: Supplier number
- "InInvoiceNumber" Parameter: Supplier transaction number

Aco.TransAR.Begin

- Description: Begins a customer transaction type.

Aco.TransAR.Cancel

- Description: Roll back a transaction previously opened with Aco.TransAR.Begin.

Aco.TransAR.End

- Description: Closes a previously opened transaction with Aco.TransAR.Begin.

Aco.TransARHeader.New (InInvoiceNumber ; InCustomerSupplierNumber ; InTransactionType ; InDate ; InAllPaid ; InTaxGroupNumber ; InReference ; InDescription ; TANumPosts)

- Description: Create the header of a customer transaction/invoice.
- "InInvoiceNumber" Parameter: Customer invoice number
- "InCustomerSupplierNumber" Parameter: Customer number
- "InTransactionType" Parameter: Type of transaction
- "InDate" Parameter: Transaction date
- "InAllPaid" Parameter: 1 If an invoice is paid in full or 0 otherwise
- "InTaxGroupNumber" Parameter: Acomba taxes group number
- "InReference" Parameter: Invoice reference
- "InDescription" Parameter: Description of the invoice
- "TANumPosts" Parameter: Number of details for the bill, generally 1 detail for an account receivable, 1 detail more per item plus 1 detail per tax.

Ex.: TANumPosts = 5, for an invoice of 1.00\$ + taxes

1 – Account receivable 1.14\$

2 – Apple 1 x 0.50\$

3 - Banana 1 x 0.50\$

4 – GST 0.05\$ (5%)

5 – QST 0.09\$ (8.5%)



Aco.TransARDetail.Update (PACHarterNumber ; PAAmount ; PaDescription ; PAIsTax ; TaNum)

- Description: Create a detailed invoice for a transaction/invoice.
- "PACHarterNumber" Parameter: Account number
- "PAAmount" Parameter: Amount
- "PaDescription" Parameter: Description (general description of the account, see Aco.Charter.ChDescription())
- "PAIsTax" Parameter: 1 if this detail is a tax, 0 if not
- "TaNum" Parameter: Sequential number of invoice detail

Aco.TransAP.Begin

- Description: Begins a supplier transaction type.

Aco.TransAP.Cancel

- Description: Roll back a transaction previously opened with Aco.TransAP.Begin.

Aco.TransAP.End

- Description: Closes a previously opened transaction with Aco.TransAP.Begin.

Aco.TransAPHeader.New (InInvoiceNumber ; InSupplierNumber ; InType ; InDate ; InAmount ; InSubTotal ; InReference ; InDescription ; TAnumPosts)

- Description: Create the header of a supplier transaction/invoice.
- "InInvoiceNumber" Parameter: Supplier invoice number
- "InSupplierNumber" Parameter: Supplier number
- "InType" Parameter: Type of transaction
- "InDate" Parameter: Transaction date
- "InAmount" Parameter: Total amount of the transaction
- "InSubTotal" Parameter: Sub-total amount of the transaction
- "InReference" Parameter: Invoice reference
- "InDescription" Parameter: Invoice description



- "TANumPosts" Parameter: Number of details for the bill, generally 1 detail for an account receivable, 1 detail more per item plus 1 detail per tax.
(see example of Aco.TransARHeader.New())

Aco.TransAPDetail.Update (PACHarterNumber ; PAAmount ; PaDescription ; PAIsTax ; TaNum)

- Description: Create a detailed invoice for a supplier transaction/invoice.
- "PACHarterNumber" Parameter: Account number
- "PAAmount" Parameter: Amount
- "PaDescription" Parameter: Description (general description of the account, see Aco.Charter.ChDescription())
- "PAIsTax" Parameter: 1 if this detail is a tax, 0 if not
- "TaNum" Parameter: Sequential number of invoice detail

Aco.Utills.CityProvince (City ; Province)

- Description: Returns a string consisting of the city and province.
- "City" Parameter: City
- "Province" Parameter: Province (two letters, ex.: QC)

Aco.Utills.DebitCreditAR (Amount ; TransactionType ; TaNum)

- Description: Formats the amount depending on the customer invoice.
- "Amount" Parameter: Amount
- "TransactionType" Parameter: Type of transaction
- "TaNum" Parameter: Sequential number of invoice detail

Aco.Utills.DebitCreditAP (Amount ; TransactionType ; TaNum)

- Description: Formats the amount depending on the supplier invoice.
- "Amount" Parameter: Amount
- "TransactionType" Parameter: Type of transaction
- "TaNum" Parameter: Sequential number of invoice detail

Aco.Utills.DecimalSeparator (Number)

- Description: Converts the number format used by FileMaker for a format for SQL
(100,50 => 100.50 , decimal numbers must be separated by a dot ".").



- "Number" Parameter: Amount

[Aco.Utils.ESQ \(Text \)](#)

- Description: Converts single quotes ' for double apostrophe '' for the SQL.
- "Text" Parameter: Text

[Aco.Utils.ISOCountryCodeEn \(Country \)](#)

- Description: Returns the two-character ISO code for the country name in English.
- "Country" Parameter: Country name (ex.: "Germany" => "DE")

[Aco.Utils.ISOCountryCodeFr \(Country \)](#)

- Description: Returns the two-character ISO code for the country name in French.
- "Country" Parameter: Country name (ex.: "Allemagne" => "DE")

[Aco.Utils.Phone \(Text \)](#)

- Description: Returns a phone number of 7, 10 or 11 characters.
- "Text" Parameter: Phone number



CUSTOM FUNCTIONS – SAGE 50 (Simple Comptable)

List of the custom functions for the Sage 50 software

- s50.sASUserId
- s50.sASOrgId
- s50.Customer.Exist (sName)
- s50.Customer.New (sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; dCrLimit ; sEmail ; sWebSite ; nLangPref ; dtSince ; sMemo)
- s50.Customer.Update (lId ; sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; sEmail ; sWebSite ; sMemo)
- s50.Invoice.New (sName ; lCustomrId ; dtDate ; sReference ; sTaxCode ; dSubTotal ; dTotal ; dTotalTaxA ; lAccountTaxA ; dTotalTaxB ; lAccountTaxB ; sComment)
- s50.InvoiceDetail.New (sInvoiceNumber ; sPartCode ; lQuantity ; dPrice ; sTaxCode ; dAmountTaxA ; lAccountTaxA ; dAmountTaxB ; lAccountTaxB)
- s50.InvoiceDetailAccount.New (sInvoiceNumber ; lAccount ; sDescription ; lQuantity ; dPrice ; sTaxCode ; dAmountTaxA ; lAccountTaxA ; dAmountTaxB ; lAccountTaxB)
- s50.Invoice.Freight (sInvoiceNumber ; dFreightAmount ; sFreightTaxCode ; dFreightTaxA ; lFreightAccountTaxA ; dFreightTaxB ; lFreightAccountTaxB)
- s50.Invoice.Commit
- s50.Invoice.Rollback
- s50.InvoicePayment.New (sInvoiceNumber ; lCustomrId ; dtDate ; dAmount ; lType ; lBankAccount ; sReference ; sComment)Aco.InvoiceAR.Exist (InCustomerSupplierNumber ; sInvoiceNumber)
- s50.Purchase.New (sName ; lVendorId ; dtDate ; sReference ; sDescription ; sTaxCode ; dSubTotal ; lChargeAccount ; dTotalTaxA ; lAccountTaxA ; dTotalTaxB ; lAccountTaxB ; sNonTaxableDescription ; dNonTaxableAmount ; lNonTaxableAccount ; dTotal ; sComment)
- s50.Vendor.Exist (sName)
- s50.Vendor.New (sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; sEmail ; sWebSite ; sTaxCode ; nLangPref ; dtSince ; sMemo)
- s50.Vendor.Update (lId ; sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; sEmail ; sWebSite ; sMemo)
- s50.Utils.CheckSqlSyntax (Sql)
- s50.Utils.Date (fmDate)
- s50.Utils.DateFirstOfLastMonth (fmDate)
- s50.Utils.ESQ (Text)
- s50.Utils.FormatDecimal (Number)



- s50.Utls.getNextId (TableId)
- s50.Utls.getNextInvoiceNumber
- s50.Utls.getNextReceiptNumber
- s50.Utls.getInvoiceBalance (sInvoiceNumber ; ICustomrId)
- s50.Utls.getPurchaseBalance (sInvoiceNumber ; IVendorId)
- s50.Utls.getTaxCodeId (sTaxCode)
- s50.Utls.getTaxInfo (sTaxCode ; ITaxAccountNumber)
- s50.Utls.setNextId (TableId)
- s50.Utls.setNextInvoiceNumber (Number)
- s50.Utls.setNextReceiptNumber (Number)

s50.sASUserId

- Description: Function to change the name of the user you want to use.

s50.sASOrgId

- Description: Function to change the name of the organization that you want to use.

s50.Customer.Exist (sName)

- Description: Returns the unique number of the customer in Sage 50 if there is otherwise returns "".
- "sName" Parameter: Customer company name in Sage50.

s50.Customer.New (sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; dCrLimit ; sEmail ; sWebSite ; nLangPref ; dtSince ; sMemo)

- Description: Adding a client. Returns the unique number of the customer in Sage 50.
- "sName" Parameter: Customer company name.
- "sCntcName" Parameter: Name of main contact at customer company.
- "sStreet1" Parameter: Company address (line #1).
- "sStreet2" Parameter: Company address (line #2).
- "sCity" Parameter: City name.
- "sProvState"Parameter: Name of province.
- "sCountry" Parameter: Name of country.



- "sPostalZip" Parameter: Postal code.
- "sPhone1" Parameter: Phone number #1.
- "sPhone2" Parameter: Phone number #2.
- "sFax" Parameter: Fax number.
- "dCrLimit" Parameter: Credit Limit.
- "sEmail" Parameter: Email address.
- "sWebSite" Parameter: Website address.
- "nLangPref" Parameter: Preferred language. (0=English, 1=Spanish, 2=French)
- "dtSince" Parameter: Date customer was added to Sage 50.
- "sMemo" Parameter: Notes.

s50.Customer.Update (lId ; sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; sEmail ; sWebSite ; sMemo)

- Description: Updating general customer information.
- "lId" Parameter: Unique number of the customer in Sage 50.
- "sName" Parameter: Customer company name.
- "sCntcName" Parameter: Name of main contact at customer company.
- "sStreet1" Parameter: Company address (line #1).
- "sStreet2" Parameter: Company address (line #2).
- "sCity" Parameter: City name.
- "sProvState" Parameter: Name of province.
- "sCountry" Parameter: Name of country.
- "sPostalZip" Parameter: Postal code.
- "sPhone1" Parameter: Phone number #1.
- "sPhone2" Parameter: Phone number #2.
- "sFax" Parameter: Fax number.
- "dCrLimit" Parameter: Credit Limit.
- "sEmail" Parameter: Email address.
- "sWebSite" Parameter: Website address.
- "sMemo" Parameter: Notes.



s50.Invoice.New (sName ; lCustomrId ; dtDate ; sReference ; sTaxCode ; dSubTotal ; dTotal ; dTotalTaxA ; lAccountTaxA ; dTotalTaxB ; lAccountTaxB ; sComment)

- Description: Creates the header of a transaction/customer invoice and returns the invoice number used by Sage 50.
- "sName" Parameter: Customer company name.
- "lCustomrId" Parameter: Unique number of the customer in Sage 50.
- "dtDate" Parameter: Transaction date.
- "sReference" Parameter: External reference.
- "sTaxCode" Parameter: Tax code to use.
- "dSubTotal" Parameter: Transaction sub-total.
- "dTotal" Parameter: Transaction total.
- "dTotalTaxA" Parameter: A tax amount.
- "lAccountTaxA" Parameter: Tax A account number.
- "dTotalTaxB" Parameter: B tax amount.
- "lAccountTaxB" Parameter: Tax B account number.
- "sComment" Parameter: Notes.
- Note: the 'Commit' is not automatic. You must use the functions s50.Invoice.Commit() ou s50.invoice.Rollback().

s50.InvoiceDetail.New (sInvoiceNumber ; sPartCode ; lQuantity ; dPrice ; sTaxCode ; dAmountTaxA ; lAccountTaxA ; dAmountTaxB ; lAccountTaxB)

- Description: Adds a line to a transaction/customer invoice.
- "sInvoiceNumber" Parameter: Invoice number used by Sage 50.
- "sPartCode" Parameter: The product or service code displayed in the section 'Stocks & Services' by Sage 50.
- "lQuantity" Parameter: Quantity of the product or service.
- "dPrice" Parameter: Price of the product or service.
- "sTaxCode" Parameter: Tax code to use.
- "dAmountTaxA" Parameter: A tax amount.
- "lAccountTaxA" Parameter: Tax A account number.
- "AmountTaxB" Parameter: B tax amount.
- "lAccountTaxB" Parameter: Tax B account number.
- Note: the 'Commit' is not automatic. You must use the functions s50.Invoice.Commit() ou s50.invoice.Rollback().



s50.InvoiceDetailAccount.New (sInvoiceNumber ; lAccount ; sDescription ; lQuantity ; dPrice ; sTaxCode ; dAmountTaxA ; lAccountTaxA ; dAmountTaxB ; lAccountTaxB)

- Description: Uses a G/L account instead of a catalog item.
- "sInvoiceNumber" Parameter: Invoice number used by Sage 50.
- "lAccount" Parameter: G/L account number used.
- "sDescription" Parameter: Line invoice details description.
- "lQuantity" Parameter: Quantity of the product or service.
- "dPrice" Parameter: Price of the product or service.
- "sTaxCode" Parameter: Tax code to use.
- "dAmountTaxA" Parameter: A tax amount.
- "lAccountTaxA" Parameter: Tax A account number.
- "AmountTaxB" Parameter: B tax amount.
- "lAccountTaxB" Parameter: Tax B account number.
- Note: the 'Commit' is not automatic. You must use the functions s50.Invoice.Commit() ou s50.invoice.Rollback().

s50.Invoice.Freight (sInvoiceNumber ; dFreightAmount ; sFreightTaxCode ; dFreightTaxA ; lFreightAccountTaxA ; dFreightTaxB ; lFreightAccountTaxB)

- Description: Use to include the costs of transport in a typical customer invoice.
- "sInvoiceNumber" Parameter: Customer invoice number.
- "dFreightAmount" Parameter: Costs of transportation.
- "sFreightTaxCode" Parameter: Tax code to use.
- "dFreightTaxA" Parameter: A tax amount.
- "lFreightAccountTaxA" Parameter: A tax account number to use.
- "dFreightTaxB" Parameter: B tax amount.
- "lFreightAccountTaxB" Parameter: A tax account number to use.

s50.Invoice.Commit

- Description: Recorded permanently Sage 50 database the records generated by the functions s50.Invoice.New() and s50.InvoiceDetail.New().

s50.Invoice.Rollback

- Description: Cancel transaction to the Sage 50 database 50 generated by functions s50.Invoice.New() and s50.InvoiceDetail.New().



s50.InvoicePayment.New (sInvoiceNumber ; lCustomrId ; dtDate ; dAmount ; lType ; lBankAccount ; sReference ; sComment)

- Description: Adds a payment to a transaction/customer invoice.
- "sInvoiceNumber" Parameter: Invoice number used by Sage 50.
- "lCustomrId" Parameter: Unique number of the customer in Sage 50.
- "dtDate" Parameter: Payment date.
- "dAmount" Parameter: Payment amount.
- "lType" Parameter: Payment type. (0=Cash, 1=Cheque, >1=Credit card)
- "lBankAccount" Parameter: Account number where the payment will be recorded.
- "sReference" Parameter: Cheque number or name of the credit card.
- "sComment" Parameter: Notes.

s50.Purchase.New (sName ; lVendorId ; dtDate ; sReference ; sDescription ; sTaxCode ; dSubTotal ; lChargeAccount ; dTotalTaxA ; lAccountTaxA ; dTotalTaxB ; lAccountTaxB ; sNonTaxableDescription ; dNonTaxableAmount ; lNonTaxableAccount ; dTotal ; sComment)

- Description: Creates the header and the details of a transaction/purchase supplier.
- "sName" Parameter: Supplier company name.
- "lVendorId" Parameter: Unique number of the supplier in Sage 50.
- "dtDate " Parameter: Transaction date.
- "sReference" Parameter: External reference.
- "sDescription" Parameter: Purchase description.
- "sTaxCode" Parameter: Tax code to use.
- "dSubTotal" Parameter: Transaction sub-total.
- "lChargeAccount" Parameter: Account number where the charge will be recorded.
- "dTotalTaxA" Parameter: A tax amount.
- "lAccountTaxA" Parameter: Tax A account number.
- "dTotalTaxB" Parameter: B tax amount.
- "lAccountTaxB" Parameter: Tax B account number.
- "sNonTaxableDescription" Parameter: Description of the part without tax.
- "dNonTaxableAmount" Parameter: Amount of the part without tax.
- "lNonTaxableAccount" Parameter: Account number where the part without tax will be recorded.
- "dTotal" Parameter: Transaction total.
- "sComment" Parameter: Notes.



s50.Vendor.Exist (sName)

- Description: Returns the unique number of the supplier in Sage 50 if there is otherwise returns "" (empty string).
- "sName" Parameter: Supplier company name in Sage 50.

s50.Vendor.New (sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; sEmail ; sWebSite ; nLangPref ; dtSince ; sMemo)

- Description: Adding a supplier. Returns the unique number of the supplier in Sage 50.
- "sName" Parameter: Supplier company name.
- "sCntcName" Parameter: Name of main contact at supplier company.
- "sStreet1" Parameter: Company address (line #1).
- "sStreet2" Parameter: Company address (line #2).
- "sCity" Parameter: City name.
- "sProvState" Parameter: Name of province.
- "sCountry" Parameter: Name of country.
- "sPostalZip" Parameter: Postal code.
- "sPhone1" Parameter: Phone number #1.
- "sPhone2" Parameter: Phone number #2.
- "sFax" Parameter: Fax number.
- "sEmail" Parameter: Email address.
- "sWebSite" Parameter: Website address.
- "nLangPref" Parameter: Preferred language. (0=English, 1=Spanish, 2=French)
- "dtSince" Parameter: Date supplier was added to Sage 50.
- "sMemo" Parameter: Notes.

s50.Vendor.Update (IId ; sName ; sCntcName ; sStreet1 ; sStreet2 ; sCity ; sProvState ; sCountry ; sPostalZip ; sPhone1 ; sPhone2 ; sFax ; sEmail ; sWebSite ; sMemo)

- Description: Update general supplier information.
- "IId" Parameter: Unique number of the supplier in Sage 50.
- "sName" Parameter: Supplier company name.
- "sCntcName" Parameter: Name of main contact at supplier company.
- "sStreet1" Parameter: Company address (line #1).
- "sStreet2" Parameter: Company address (line #2).



- "sCity" Parameter: City name.
- "sProvState" Parameter: Name of province.
- "sCountry" Parameter: Name of country.
- "sPostalZip" Parameter: Postal code.
- "sPhone1" Parameter: Phone number #1.
- "sPhone2" Parameter: Phone number #2.
- "sFax" Parameter: Fax number.
- "sEmail" Parameter: Email address.
- "sWebSite" Parameter: Website address.
- "sMemo" Parameter: Notes.

s50.Utills.CheckSqlSyntax (Sql)

- Description: Check the SQL query for missing values.
- "Sql" Parameter: SQL query.

s50.Utills.Date (fmDate)

- Description: Convert a FileMaker date into format required by Sage 50.
- "fmDate" Parameter: Date.

s50.Utills.DateFirstOfLastMonth (fmDate)

- Description: Returns the start date of the previous month fmDate. Ex.: 28/09/2013 returns 01/08/2013.
- "fmDate" Parameter: Date.

s50.Utills.ESQ (Text)

- Description: Convert single quote ' in double quote " for the SQL query.
- "Text" Parameter: Text.

s50.Utills.FormatDecimal (Number)

- Description: Change the number format with a dot (.) As decimal separator.
- "Number" Parameter: Number to format.



s50.Utills.getNextId (TableId)

- Description: Returns the next unique number to use for the table TableId.
- "TableId" Parameter: Unique number from Sage 50 table.
- Table ID :

20 tAccount 24 tBRSum 25 tBTR 26 tBRSumJE 30 tProject
40 tVendor 43 tVenTr 44 tVenTrDt 50 tCustomr 53 tCusTr
54 tCusTrDt 60 tPurOrdr 62 tSalOrdr 70 tEmpUS 72 tPyDetU
80 tEmp 82 tPyDet 90 tInvent 97 tPrclList 100 tJourEnt
103 tJEntLY 106 tJEntCsh 110 tUser 113 tCurrncy 115 tCCard
116 tShipper 119 tTask 121 tStmntDwn 122 tSDLine 123 tUserWnd
126 tTaxDtl 150 tITRec 160 tRCGen 163 tRCPur 166 tRCSal
169 tRCPrI 171 tRCPrIU 173 tRCTrf 176 tRCAdj 179 tRCPay
191 tTaxAuth 192 tTaxCode 200 tActHRel 201 tActHDat 210 tRCBOM
212 tRCTSlp 220 tTSRec 222 tPmtHdr 223 tRcpHdr 224 tPrIHdr
225 tActDpt 255 tActRang 256 tJobCat 261 tCnActRI 268 tDeposit
269 tDSCheq 270 tDSCash 272 tDSDenom 273 tDSMult 304 tRptOpts
307 tRptGrp 350 tInvLoc 352 tSDSHdr 353 tSDSCHqs 354 tSDSCash
369 tRQRet 381 tGrpExp 386 tCustShp 388 tInvLot 390 tInvCate
391 tUsrRole 400 tInvSNum 404 tSBom 407 tJEntPH 408 tJELYPH
410 tCCardTr 414 tSDNode 415 tSDUUID 438 tGovForm 440 tSDDgst
445 tSDTax

s50.Utills.getNextInvoiceNumber

- Description: Returns the next customer invoice number to use.

s50.Utills.getNextReceiptNumber

- Description: Returns the next receipt number to use.

s50.Utills.getInvoiceBalance (sInvoiceNumber ; lCustomrId)

- Description: Returns the balance of a customer invoice.
- "sInvoiceNumber" Parameter: Invoice number used by Sage 50.
- "lCustomrId" Parameter: Unique number of the customer in Sage 50.



s50.Utills.getPurchaseBalance (sInvoiceNumber ; IVendorId)

- Description: Returns the balance on a supplier invoice.
- "sInvoiceNumber" Parameter: Invoice number used by Sage 50.
- "IVendorId" Parameter: Unique number of the supplier in Sage 50.

s50.Utills.getTaxCodeId (sTaxCode)

- Description: Returns the unique number of a tax group.
- "sTaxCode" Parameter: Tax code used by Sage 50.

s50.Utills.getTaxInfo (sTaxCode ; ITaxAccountNumber)

- Description: Returns information about the tax code. (ref.: source code).
- "sTaxCode" Parameter: Tax code used by Sage 50.
- "ITaxAccountNumber" Parameter: Account number of the sale or purchase.

s50.Utills.setNextId (TableId)

- Description: Assigns the next unique number to be used by the table TableId. (see s50.Utills.getNextId())
- "TableId" Parameter: Unique ID number of the table in Sage 50.

s50.Utills.setNextInvoiceNumber (Number)

- Description: Assigns the next number of customer invoice to use.
- "Number" Parameter: Number

s50.Utills.setNextReceiptNumber (Number)

- Description: Assigns the next receipt number to use.
- "Number" Parameter: Number



REFERENCES

Technical notes

- The sODBC plug-in requires the operating system Windows XP version or later.
- The sODBC plug-in requires the operating system Mac OS X version 10.8 (Mountain Lion) or later.
- Under Windows OS, sODBC standard requires that Microsoft. NET 3.5 is installed with Windows Vista and 7. In the case of Windows XP, install the resources .NET 3.5 available from Microsoft. Consult the Web page <http://www.microsoft.com/net/>
- If your OS is Windows 7 in 64 bits mode (or XP 64 bits), use the following ODBC Manager for all 32 bits ODBC drivers: C:\Windows\SysWOW64\odbcad32.exe
- The sODBC plug-in is compatible with versions 9, 10, 11, 12 and 13 of FileMaker Pro, FileMaker Pro Advanced and FileMaker Server.
- The sODBC plug-in can be used by the Acomba ODBC driver with versions 9, 10, 11 and 13 of FileMaker Pro and FileMaker Pro Advanced. However, it cannot be used with version 12 of FileMaker Pro and FileMaker Pro Advanced.
- The sODBC plug-in can be used by Sage 50 software with versions 9, 10, 11, 12 and 13 of FileMaker Pro and FileMaker Pro Advanced.
- The sODBC plug-in is compatible with FileMaker Server-side scripting (SSS) technology.
- Codes of items must match between Sage 50 and your FileMaker solution.
- The tax code for 'No Tax' in Sage 50 must be empty.
- There must be at least one client and one provider created in Sage 50 to allow export from FileMaker.
- There are no management rebate (discount) for client in Sage 50. We recommend that you configure the default values to zero (0) in the menu Settings/Customers & Sales.
- The 'Commit' is not automatic for functions s50.Invoice.New() and s50.InvoiceDetail.New(). Do not forget to use the functions s50.Invoice.Commit() or s50.invoice.Rollback().
- Refer to the demo file delivered with the custom functions kit for more details on the settings for Sage 50 software.
- A problem for the receipt number is related to the behavior of Sage 50 Software.
Example: If you open the billing module, the next invoice number is loaded in memory and is updated when closing Sage 50. So if you do the normal work in addition to export invoices into Sage 50, the number that had been stored is no longer correct. There are system in place to try to manage this behavior with the invoice number.
We recommend that you do not use the same Sage 50 session to work and to export. The best would be to close Sage 50 and reopen before export and not to go elsewhere using different section of the software. So we open Sage 50, we do not touch anything and than you do the exportation. If you had a duplicate payment, you could simply delete it and recreate by hand.



Troubleshooting

- As this is the first user manual for sODBC plug-in for FileMaker Pro, we invite users to communicate with us to help us build this section of the manual.

About FileMaker, Inc.

- FileMaker develops database program awards since 1990. Its products include the legendary FileMaker Pro product line for Windows, Mac and FileMaker Go for iPhone, iPod and iPad. FileMaker has won over 52 awards, more than his eight competitors combined from 2003 to 2009 in the United States, and a total of 134 awards in the world during those years. Millions of users - from individuals to large companies - rely on FileMaker to manage, analyze and share information. FileMaker, Inc. is a subsidiary of Apple Inc.

