

Sage Accpac ERP 5.6A

SageCRM 6.2 | Integration Developer Tools Guide



© 2009 Sage Software, Inc. All rights reserved. Sage, the Sage logos, and all Sage Accpac product and service names mentioned herein are registered trademarks or trademarks of Sage Software, Inc., or its affiliated entities. All other trademarks are the property of their respective owners.

Sage Software, Inc.
Publisher

Contents

UtilityFuncs.js.....	1
Grid and Table Creation.....	1
TableStart.....	1
TableHead.....	1
TableContent.....	2
TableEnd.....	2
TD.....	3
TDEx.....	3
TDROW2.....	3
TDClass.....	4
TR.....	4
GridTitle.....	4
GridStartTable.....	5
GridStartTableWidth.....	6
GridEndTable.....	6
GridHeader.....	6
GridHeaderWidth.....	7
GridData.....	7
GridDataWidth.....	8
Field Formatting.....	8
SplitPhone.....	8
SplitName.....	10
formatphoneEx.....	10
formatmulticurrency.....	11
formatCurrencyEx.....	12
Field Modifying.....	12
ltrim.....	12
rtrim.....	12
trim.....	13
IsEmpty.....	13
StringToNumberString.....	14

GetDateObjectFromString	14
ACCPACDate	14
ACCPACDateEX	15
ACCPACDateYear	15
escapeQuotes	16
escapeSingleQuotes	16
enclose	16
PadL	17
trimRight	17
ACCPACNumber	18
Miscellaneous	18
checkifblank	18
HidBucket	19
Bucket	19
eWareImportFromACCPACCustomer	20
eWareImportFromACCPACVendor	20
UrlRoot	21

UtilityFuncs.js

Grid and Table Creation

TableStart

Definition	TableStart (<i>sContent</i> , <i>sWidth</i>); STRING <i>sContent</i> ; STRING <i>sWidth</i> ;
Description	Returns a string that will specify the HTML code to create the beginning of a table.
Parameters	<i>sContent</i> Optional. Table class name. If left blank, will not have a class name. <i>sWidth</i> Optional. Width of the table. If left blank, will set the width to 100%.

TableHead

Definition	TableHead (<i>sHeader</i>); STRING <i>sHeader</i> ;
Description	Returns a string that will specify the HTML code to create the header for a table. This will be formatted in the standard style as a header looks in the core system.
Parameters	<i>sHeader</i> Table header, this should be the title to show on top of the table.

TableContent

Definition	TableContent (<i>sContent</i>); STRING <i>sContent</i> ;
Description	Returns the content, formatted to be correctly display in standard CRM table format.
Parameters	<i>sContent</i> Table content, this should be a full <TABLE> tag.
Example	<pre>var MainTable = eWare.StartTable(); MainTable += TableHead('Standard CRM Table'); var sContent = '<TABLE WIDTH=100% BORDER=0 CELLPADDING=5 CELLSPACING=0 RULES=NONE CLASS=CONTENT>'; sContent += TR(TD('First Name', 'Kylie') + TD('Last Name', 'Ward')); sContent += TR(TD('Company', 'Panopoly Ltd') + TD('Status', 'Active')); sContent += TableEnd(); MainTable += TableContent(sContent) +TableEnd(); Response.Write(MainTable);</pre>

TableEnd

Definition	TableEnd ();
Description	Returns a string that will specify the HTML code to create the end code for a table.
Parameters	None.

TD

Definition	TD (<i>sTitle</i> , <i>sInfo</i>); STRING <i>sTitle</i> ; STRING <i>sInfo</i> ;
Description	Returns a string that will specify the HTML code to create a table cell and populate it with information.
Parameters	<i>sTitle</i> Title of the cell. <i>sInfo</i> Information to be included in the cell.

TDEx

Definition	TDEx (<i>sTitle</i> , <i>sInfo</i> , <i>sWidth</i>); STRING <i>sTitle</i> ; STRING <i>sInfo</i> ; STRING <i>sWidth</i> ;
Description	Returns a string that will specify the HTML code to create a table cell and populate it with information. Same as the TD function, except that TDEx also lets you specify the width of the table cell.
Parameters	<i>sTitle</i> Title of the cell. <i>sInfo</i> Information to be included in the cell. <i>sWidth</i> Width of the cell.

TDROW2

Definition	TDROW2 (<i>sTitle</i> , <i>sInfo</i>); STRING <i>sTitle</i> ; STRING <i>sInfo</i> ;
------------	--

Grid and Table Creation

Description Returns a string that will specify the HTML code to create a table cell that spans 2 columns.

Parameters *sTitle*
Title of the cell.

sInfo
Information to be included in the cell.

TDClass

Definition **TDClass**(*sInfo*, *sClass*);
STRING *sInfo*;
STRING *sClass*;

Description Returns a string that will specify the HTML code to create a table cell as well as the class information.

Parameters *sInfo*
Information to be included in the cell.

sClass
Optional. Class information. If not specified, will leave the class information out.

TR

Definition **TR**(*sRow*);
STRING *sRow*;

Description Returns a string that will specify the HTML code to create a table row as well as the row information.

Parameters *sRow*
Row information.

GridTitle

Definition **GridTitle**(*sTitle*);
STRING *sTitle*;

Description Returns a string that will specify the HTML code to create a grid and give the grid a title.

Parameters *sTitle*
Title of the grid.

Example

```

Var MainGrid = eWare.StartTable();
MainGrid += GridTitle('Standard CRM Grid');

Var sContent = GridStartTable();
sContent += TR(GridHeader('Sample Names', false) +
GridHeader('Sample Dates', false));

sRowClass = 'ROW1';
sContent += TR(GridData('Ryan', ", sRowClass) + GridData('20',
", sRowClass));

sRowClass = 'ROW2';
sContent += TR(GridData('Rory', ", sRowClass) + GridData('18',
", sRowClass));

sRowClass = 'ROW1';
sContent += TR(GridData('Sophie', ", sRowClass) + GridData('11',
", sRowClass));
sContent + GridEndTable();

MainGrid += sContent+ TableEnd();

Response.Write(MainGrid);

```

GridStartTable

Definition **GridStartTable();**

Description Returns a string that will specify the HTML code to create the start of a grid.

Parameters None.

GridStartTableWidth

Definition	GridStartTableWidth (<i>sWidth</i>); STRING <i>sWidth</i> ;
Description	Returns a string that will specify the HTML code to create the start of a grid. Same as GridStartTable, except the user can specify the width of the grid.
Parameters	<i>sWidth</i> Width of the grid.

GridEndTable

Definition	GridEndTable ();
Description	Returns a string that will specify the HTML code to create the end of the grid.
Parameters	None

GridHeader

Definition	GridHeader (<i>sHeader</i> , <i>bOrderBy</i>); STRING <i>sHeader</i> ; BOOL <i>bOrderBy</i> ;
Description	Returns a string that will specify the HTML code to set up a header for a grid. Will also set up the anchor tags for the header if desired.
Parameters	<i>sHeader</i> Header information. <i>bOrderBy</i> Set to true if you want to set the anchor tags, false if you want a simple header.

GridHeaderWidth

Definition	GridHeaderWidth (<i>sHeader</i> , <i>bOrderBy</i> , <i>sWidth</i>); STRING <i>sHeader</i> ; BOOL <i>bOrderBy</i> ; STRING <i>sWidth</i> ;
Description	Returns a string that will specify the HTML code to set up a header for a grid. Will also set up the anchor tags for the header if desired. Same as GridHeader except the user specifies a width.
Parameters	<i>sHeader</i> Header information. <i>bOrderBy</i> Set to true if you want to set the anchor tags, false if you want a simple header. <i>sWidth</i> Header width.

GridData

Definition	GridData (<i>sValue</i> , <i>sJump</i> , <i>sClass</i>); STRING <i>sValue</i> ; STRING <i>sJump</i> ; STRING <i>sClass</i> ;
Description	Returns a string that will specify the HTML code for the data in a grid cell. The user is able to specify a “jump” location that will open up a new window when the data in the cell is clicked.
Parameters	<i>sValue</i> Value in the cell. <i>sJump</i> Optional. If not specified, there will be no jump location, simply the cell with the specified value. If specified, the jump location will be sJump. <i>sClass</i> Class name.

GridDataWidth

Definition	GridDataWidth (<i>sValue</i> , <i>sJump</i> , <i>sClass</i> , <i>sWidth</i>); STRING <i>sValue</i> ; STRING <i>sJump</i> ; STRING <i>sClass</i> ; STRING <i>sWidth</i> ;
Description	Returns a string that will specify the HTML code for the data in a grid cell. The user is able to specify a “jump” location that is a hyperlink. Similar to the function GridData except that the jump location does not open up a new window. Also GridDataWidth allows the user to specify the width of the grid cell.
Parameters	<i>sValue</i> Value in the cell. <i>sJump</i> Optional. If not specified, then there will be no jump location, simply the cell with the specified value. If specified, then the value will be the jump location. <i>sClass</i> Class name. <i>sWidth</i> Width of the cell.

Field Formatting

SplitPhone

Definition	SplitPhone (<i>sAll</i> , <i>P_PersPhoneObj</i> , <i>P_CompPhoneObj</i> , <i>PersonObject</i> , <i>CompanyObject</i> , <i>IsBusiness</i> , <i>UseAreaCode</i> , <i>sEntity</i>); STRING <i>sAll</i> ; *PersonObject <i>P_PersPhoneObj</i> ; *CompanyObject <i>P_CompPhoneObj</i> ; *PersonObject <i>PersonObject</i> ; *CompanyObject <i>CompanyObject</i> ; BOOL <i>IsBusiness</i> ; BOOL <i>UseAreaCode</i> ; STRING <i>sEntity</i> ;
------------	---

Description	Splits the phone/fax number passed into two parts—area code, and phone number—and assigns them to the person or company object passed into the function.
Parameters	<p><i>sAll</i> String to be evaluated.</p> <p><i>P_PersPhoneObj</i> Person phone object to be filled with the area code and phone number of <i>sAll</i>, if <i>sEntity</i> is 'Person'.</p> <p><i>P_CompPhoneObj</i> Company phone object to be filled with the area code and phone number of <i>sAll</i>, if <i>sEntity</i> is not 'Person'.</p> <p><i>PersonObject</i> Person object passed to function (required when <i>sEntity</i> is 'Person').</p> <p><i>CompanyObject</i> Company object passed to function (required when <i>sEntity</i> is not 'Person').</p> <p><i>IsBusiness</i> If true, then the value of <i>PersonObject</i> (when <i>sEntity</i> is 'Person') or <i>CompanyObject</i> (when <i>sEntity</i> is not 'Person') will have the phone number set; otherwise, it will have the fax number set.</p> <p><i>UseAreaCode</i> Specifies if an area code is to be used. If true, then the first 3 numbers in the phone number entered will be designated as the area code. If false, the phone number entered will be used as is.</p> <p><i>sEntity</i> If 'Person' (case sensitive), the phone/fax number belongs to a person (so fields in <i>P_PersPhoneObj</i> and <i>PersonObject</i> will be set). Otherwise, that number belongs to a company (so fields in <i>P_CompPhoneObj</i> and <i>CompanyObject</i> will be set).</p>
Example	<i>SplitPhone('1234567890', pph, cph, p, c, false, true, 'Person');</i>

Field Formatting

Result
pph('Phon_AreaCode') = '123'
pph('Phon_Number') = '4567890'
p('Pers_FaxAreaCode') = '123'
p('Pers_FaxNumber') = '4567890'

SplitName

Definition **SplitName**(*sAll*, *PersonObject*, *OldSalute*);
 STRING *sAll*;
 **PersonObject* *PersonObject*;
 STRING *OldSalute*;

Description Splits the full name into salutation, first name, and last name.

Parameters *sAll*
 Full name to be split

PersonObject

Person object to be initialized with the salutation, first, and last name.

OldSalute

Alternate salutation to try to match the first part of the *sAll* string against (in addition to matching against the standard salutations). Set it to "" if you will only match against the standard salutations (those include "M", "Mr.", "Mrs", "Mrs.", "Ms", "Ms.", "Dr", and "Dr.).

Example *SplitName("Mr. Bob Brown", a, "");*

Result returns true
 a('Pers_Salutation') = "Mr."
 a('Pers_LastName') = "Bob"
 a('Pers_FirstName') = "Brown"

formatphoneEx

Definition **formatphoneEx**(*uphone*);
 STRING *uphone*;

Description	Strips out characters -, (,), and spaces, and then formats phone numbers in the form (123) 456-7890.
Parameters	<i>uphone</i> Phone number to be evaluated.
Example	<i>formatphoneEx("((123) 45- 678- 90)");</i>
Result	returns "(123) 456-7890"

formatmulticurrency

Definition	formatmulticurrency (<i>num</i> , <i>persontype</i> , <i>customerid</i>); NUMBER <i>num</i> ; STRING <i>persontype</i> ; STRING <i>customerid</i> ;
Description	Converts a number to the currency format, with the currency specified by <i>persontype</i> and <i>customerid</i> . The number is always rounded to two decimal places and negatives always use the sign '-'. <i>num</i> Number to be evaluated <i>persontype</i> (Values are case insensitive.) If 'customer', use the currency from the AR0024 customer view for <i>customerid</i> . If 'vendor', use the currency from the AP0015 vendor view for <i>customerid</i> . If 'functional', use the Sage Accpac company's home currency. <i>customerid</i> If <i>persontype</i> is 'customer', this is the Sage Accpac customer number. If <i>persontype</i> is 'vendor', this is the Sage Accpac vendor number. If <i>persontype</i> is 'functional', this parameter is ignored (so set it to "").
Example	<i>// Vendor MYSTORE uses currency USD.</i> <i>formatmulticurrency(-3333.456, 'vendor', 'MYSTORE');</i>
Result	returns "\$ 3,333.46"

formatCurrencyEx

Definition	formatCurrencyEx (<i>num</i>); NUMBER <i>num</i> ;
Description	Converts a number to currency format, rounded to two decimal places.
Parameters	<i>num</i> Number to be evaluated.
Example	<i>formatCurrencyEx</i> (-3333.456);
Result	returns “-\$3,333.46”

Field Modifying

ltrim

Definition	ltrim (<i>s</i>); STRING <i>s</i> ;
Description	Function that will replace any whitespace characters at the beginning of the string passed, with “”. Effectively removes any leading spaces.
Parameters	<i>s</i> String to be trimmed.
Example	<i>ltrim</i> (" hello world");
Result	returns “hello world”

rtrim

Definition	rtrim (<i>s</i>); STRING <i>s</i> ;
Description	Function that will replace any whitespace characters at the end of the string passed, with “”. Effectively removes any trailing spaces.

Parameters *s*
String to be trimmed.

Example *rtrim("hello world ");*

Result returns "hello world"

trim

Definition **trim**(*s*);
STRING *s*;

Description Performs both an ltrim and an rtrim on the variable passed into the function, and returns the result. Effectively removes any leading and trailing spaces.

Parameters *s*
String to be trimmed.

Example *trim(" hello world ");*

Result returns "hello world"

IsEmpty

Definition **IsEmpty**(*sTestEmpty*);
STRING *sTestEmpty*;

Description Checks to see if the string passed contains a value, or is an empty string.

Parameters *sTestEmpty*
String to be evaluated.

Example *IsEmpty("hello world");*

Result returns false

StringToNumberString

Definition	StringToNumberString (<i>sNumberString</i>); STRING <i>sNumberString</i> ;
Description	Converts a string to a number string, and inserts commas for every three digits. Also truncates extraneous zeroes.
Parameters	<i>sNumberString</i> String to be evaluated.
Example	<i>StringToNumberString(-123456789.8880)</i> ;
Result	returns -123,456,789.888

GetDateObjectFromString

Definition	GetDateObjectFromString (<i>cd</i>); STRING <i>cd</i> ;
Description	Formats a date object based on the string passed. This function checks to see if user has set a date format already, and returns a date object in one of the following format: mm/dd/yyyy, dd/mm/yyyy, or yyyy/mm/dd.
Parameters	<i>cd</i> String containing the date.
Example	<i>a = new Date();</i> <i>a = GetDateObjectFromString(12252001)</i> ;
Result	a.month = 12 a.date= 25 a.year= 2001

ACCPACDate

Definition	ACCPACDate (<i>sDate</i>); STRING <i>sDate</i> ;
------------	--

Description	Takes a string and returns the date in Sage Accpac format (i.e. 'ddd mmm nn 00:00:00 UTC+0100 yyyy'). Because, often, there is no number after the UTC, just the year, the function currently takes the first 11 characters, plus the last 4.
Parameters	<i>sDate</i> String object to be evaluated.
Example	<code>ACCPACDate("Wed May 10 12:30:25 UTC+0200 2002");</code>
Result	returns "Wed May 10 2002"

ACCPACDateEX

Definition	ACCPACDateEX (<i>ViewFieldDateValue</i>); STRING <i>ViewFieldDateValue</i> ;
Description	Takes a string in date format and returns the date in Sage Accpac recognized format.
Parameters	<i>ViewFieldDateValue</i> String representation of the date.
Example	<code>ACCPACDateEX('dec 23 1900');</code>
Result	returns "Dec 23, 1900"

ACCPACDateYear

Definition	ACCPACDateYear (<i>sDate</i>); STRING <i>sDate</i> ;
Description	Takes a string and returns the date in the appropriate format for the credit info screen. The string returned is in yyymmdd format.
Parameters	<i>sDate</i> String representing a date.
Example	<code>ACCPACDateYear("Apr 28 2004");</code>
Result	returns "20040428"

escapeQuotes

Definition	escapeQuotes (<i>s</i>); STRING <i>s</i> ;
Description	Escapes each double quote in a string by putting a backslash in front of it. This function is used when constructing a database query or browse filter, where double quotes enclose field values, but those field values can have embedded double quotes within them. If the quotes were not escaped, the query or browse could fail.
Parameters	<i>s</i> The string whose double quotes are to be escaped.
Example	<i>escapeQuotes('The "Hello World" Program');</i>
Result	returns ‘The \<"Hello World\<" Program’

escapeSingleQuotes

Definition	escapeSingleQuotes (<i>s</i>); STRING <i>s</i> ;
Description	Escapes each single quote in a string by putting another single quote in front of it. This function is used when constructing a database query or browse filter, where single quotes enclose field values, but those field values can have embedded single quotes within them. If the quotes were not escaped, the query or browse could fail.
Parameters	<i>s</i> The string whose single quotes are to be escaped.
Example	<i>escapeSingleQuotes("The 'Hello World' Program");</i>
Result	returns “The "Hello World" Program”

enclose

Definition	enclose (<i>s</i>); STRING <i>s</i> ;
------------	---

Description	Encloses a string in double quotes (by adding one before the string and one after the string).
Parameters	<i>s</i> The string to be enclosed in double quotes.
Example	<i>enclose('abc');</i>
Result	returns "abc"

PadL

Definition	PadL (<i>sString</i> , <i>nLength</i> , <i>sChar</i>); STRING <i>sString</i> ; NUMBER <i>nLength</i> ; STRING <i>sChar</i> ;
Description	Pads the left side of a string with characters. The total length of the string is also given by the user.
Parameters	<i>sString</i> The string to be padded. <i>nLength</i> The total length of the string after padding. <i>sChar</i> The character that will be used to pad the string.
Example	<i>PadL("hello", 10, "x");</i>
Result	returns "xxxxxhello"

trimRight

Definition	trimRight (<i>sCustomer</i>); STRING <i>sCustomer</i> ;
Description	Trims/removes the empty spaces on the right side of the string to be evaluated.

Miscellaneous

Parameters *sCustomer*
String to be evaluated.

Example *trimRight("hello world ");*

Result returns “hello world”

ACCPACNumber

Definition **ACCPACNumber**(*sNumber*);
STRING *sNumber*;

Description Rounds a number entered to 2 decimal places. If the number does not need to be rounded, then it is returned.

Parameters *sNumber*
Number to be evaluated

Example *ACCPACNumber(-100.2345);*

Result returns -100.23

Miscellaneous

checkifblank

Definition **checkifblank**(*s*);
STRING *s*;

Description Checks to see if a string to be evaluated is empty or not. If it is empty, then the function returns a period (“.”); otherwise, it returns the string entered.

Parameters *s*
String to be evaluated.

Example *checkifblank("hello");*

Result returns “hello”

HidBucket

Definition	HidBucket (<i>sName</i> , <i>sClass</i> , <i>sValue</i>); STRING <i>sName</i> ; STRING <i>sClass</i> ; STRING <i>sValue</i> ;
Description	Returns a string to create a read-only table cell with name “HIDBUCKET + <i>sName</i> ”, with class “ <i>sClass</i> ”, and with value “ <i>sValue</i> ”. This function is used in Company_CreditInfo.asp to assign hidden text boxes for the credit terms (30, 60, 90 days).
Parameters	<i>sName</i> The cell name as a string. <i>sClass</i> The class name as a string. <i>sValue</i> The value as a string/ number.

Bucket

Definition	Bucket (<i>sName</i> , <i>sClass</i> , <i>sValue</i>); STRING <i>sName</i> ; STRING <i>sClass</i> ; STRING <i>sValue</i> ;
Description	Returns a string to create a table cell. Contains an onChange method that assigns one more than the value of the current bucket, to the next larger bucket, except for the fourth bucket. When populating the fourth bucket, the value remains the same. This function is used in Company_CreditInfo.asp to show the credit terms (30, 60, 90 days).
Parameters	<i>sName</i> The cell name as a string. <i>sClass</i> The class name as a string. <i>sValue</i> The value as a string/ number.

eWareImportFromACCPACCustomer

Definition	eWareImportFromACCPACCustomer (<i>Customer_Fields</i> , <i>sRowClass</i> , <i>UseAreaCode</i>); *xapiFields <i>Customer_Fields</i> ; STRING <i>sRowClass</i> ; BOOL <i>UseAreaCode</i> ;
Description	Imports a Sage Accpac customer into CRM. The function also creates Company, Person, Address, Email, and Phone records, plus links.
Parameters	<i>Customer_fields</i> The collection of fields from the Sage Accpac customer view (AR0024). <i>sRowClass</i> Class information for the HTML tags that will be generated. (This parameter is currently ignored. Set it to “”). <i>UseAreaCode</i> If true, the area code will be used in function; otherwise, it will not.
Example	<pre>// NOTE: ACCPAC and GetPhoneFormat come from AccPac.js. Cust_View = ACCPAC.OpenView("AR0024", "AR"); Cust_View.Browse("", true); Cust_View.Fetch(); // gets the first customer bUseAreaCode = (GetPhoneFormat() > 0); eWareImportFromACCPACCustomer(Cust_View.Fields, "", bUseAreaCode);</pre>
Result	(The first Sage Accpac customer is imported into CRM.)

eWareImportFromACCPACVendor

Definition	eWareImportFromACCPACVendor (<i>Vendor_Fields</i> , <i>sRowClass</i> , <i>UseAreaCode</i>); *xapiFields <i>Vendor_Fields</i> ; STRING <i>sRowClass</i> ; BOOL <i>UseAreaCode</i> ;
------------	---

Description	Imports a Sage Accpac vendor into CRM. The function also creates Company, Person, Address, Email, and Phone records, plus links.
Parameters	<i>Vendor_fields</i> The collection of fields from the Sage Accpac vendor view (AP0015). <i>sRowClass</i> Class information for the HTML tags that will be generated. <i>UseAreaCode</i> If true, the area code will be used in function; otherwise, it will not.
Example	<pre>// NOTE: ACCPAC and GetPhoneFormat come from AccPac.js. Vend_View = ACCPAC.OpenView("AP0015", "AP"); Vend_View.Browse("", true); Vend_View.Fetch(); // gets the first vendor bUseAreaCode = (GetPhoneFormat() > 0); eWareImportFromACCPACVendor(Vend_View.Fields, "ROW1", bUseAreaCode);</pre>
Result	(The first Sage Accpac vendor is imported into CRM.)

UrlRoot

Definition	UrlRoot();
Description	Returns the UrlRoot of a URL address in lowercase, everything on the left of the “custompages” portion of the URL.
Parameters	None
Example	If the URL is http://www.example.com/test/custompages
Result	returns http://www.example.com/test/