Axpert Functions (A - Z)

Function : Abs
Syntax : Abx(Number)
Description : Returns the absolute value

Function : AddToDate
Syntax : AddToDate(DateValue,NumberToIncrease)
Description : Adds the given number to Date.
Example :
AddToDate(Ctod({01/04/2007}, 10)
Result : 10/04/2007

Function : AddToMonth
Syntax : AddToDate(DateValue,NumberToIncrease)
Description : Adds the given number to the month of Date.
Example :
AddToMonth(Ctod({01/04/2007}, 5)
Result : 10/08/2007

Function : AmtWord
Syntax : AmtWord(Amount : Numeric)
Description : Converts the given Amount to words.
Example :
AmtWord(1356.50)
Result : One thousand three hundred and fifty six and paise fifty only.

Function : Bulkexecute
Syntax : Bulkexecute(SQLText)
Description : Fires the given SQL. Expects the first column in the SQL result to be an SQL text. Fires this SQL text from every row in the result. This can be used to do complex SQL processing.

Function: BulkPrinting

Syntax: bulkprinting(sqlstring:String)

Description: This function can be called from expressionlist and userdefinedtask.
'sqlstring' should be a select statement.
The 'select' statement should have transid,recordid and formname columns.

Example

Define an action in source transaction. The action event should be after save transaction and task should be a *userdefined task*. Content of the userdefined task should be as

```
 s := {select 1 as slno, 'teste' as transid,
'10111000018588' as recordid, 'eprint.doc' as formname
from dual union select 2 as slno, 'testa' as transid,
'10146000015108'   as recordid, 'testprint.doc' as
formname from dual union select 3 as slno,'teste' as
transid, '10122000019105' as recordid, 'eprint.doc' as
formname from dual} bulkprinting(s)
```

Function : CheckStock
Syntax : CheckStock(ItemId, OldItemId, DocDate, OldDocDate, PlusOrMinus, Qty, OldQty)
Description : Checks if enough stock is available to carry out the transaction for the given ItemId as on given DocDate for the Qty. In case of modification mode, the old values will be used if the modification can be allowed. The function will return T if the transaction can be allowed or else a message will be returned.

Function : CMonthYear
Syntax : CMonthYear(Dt : DateTime)
Description : Returns the character month and four digit year from the given date.
Example :
CMonthYear('01/04/2004')
Result : April 2004

Function : Constructtable
Syntax : Constructtable(SQLText, TableName)
Description : Expects the SQL result to contain fieldname, datatype, width and decimal as columns.
A table with the TableName specified will be created at the backend .

Function : Convertmd5
Syntax : convertmd5(string)
Description : Converts the given stringvalue to MD5

Function : CopyFileTo
Syntax : CopyFileTo(Source, Destination)
Description : CopyFileTo is a function used to copy the file specified in Source to the file specified in Destination. CopyFileTo will return False if the file in

Destination already exists

Function : CreateDbf
Syntax : CreateDbf(SQLText, DirName,TableName)
Description : Creates a dbf file in the directory given by DirName. The result of the SQL text is created as the contents of the table. The name of the table will be TableName.

Function : CreateWordDoc
Syntax : CreateWordDoc(FileName)
Description :Creates a new MS Word file and opens MS Word for editing.

Funtion : CTOD
Syntax : CTOD(Dt : String)
Description : Converts character to DateTime type

Function : CurrAmtWord
Syntax : CurrAmtWord(Amount : Numeric, Currency, SubCurrency, InMillions: Boolean, Decimals :integer)
Description : Converts the given amount to words as per the specification.
Currency - Name of the currency to convert to.
SubCurrency - Name of the sub currency
InMillions - {T} if amount should be in millions.
Decimals - number of decimal digits in the amount.
Example :
`CurrAmtword(1200000, {Dollars}, {Cents}, {T}, 2)`
Result : One million two hundred thousand only.
CurrAmtword(1200000, {Rupees}, {Paise}, {T}, 2)
Result: Twelve lacs only.

Function : Date
Syntax : Date( )
Description : Returns today's date.

Function : Dayofdate
Syntax : Dayofdate(dt : DateTime)
Description : Returns the day of date.

Function : DaysElapsed
Syntax : DaysElapsed(date1, date2)
Description : Returns the number of days elapsed between date2 and date1.

Function : DoMRP
Syntax : DoMRP(StartDate, EndDate)
Description : Runs the MRP for all records in the demand table between given start and end date. It expects the stock available in the stockvalue table, item master to be present in itemmaster table, bom to be present in Bomtable.

Function : DTOC
Syntax : DTOC(Dt : DateTime)
Description : Converts a Datetime to String.Useful in passing date type of variables to functions that expect string as a parameter.

Function : Eval
Syntax : Eval(ExpressionVar)
Description : ExpressionVar is a variable that should consist of an axpert expression. That expression will be evaluated and the value will be returned.

Function : Extractnum
Syntax : Extractnum(StringValue)
Description : Extracts only the number that is embedded in a string. This can be useful in getting numeric values from a string when reading from XML tags.

Funtion : FindAndReplace
Syntax : FindAndReplace(S, FindWhat, ReplaceWith: String)
Description : Finds the string given by FindWhat and replaces it with ReplaceWith in String S and returns the new string.
Example :
FindAndReplace({xxxyyyzzz}, {xxx}, {aa})
Result : aayyyzz.

Function : Firesql
Syntax : Firesql(SQLName, SQLText)
Description : Fires the given SQL and stores the result set in a dataset identified by the given SQLName. The result set can be accessed in subsequent expressions using the FindRecord and SQLGet functions. To these functions the name of the SQL should be provided.

Function:FormatAmount

Syntax:
FormatAmount(Value,DecLen:numeric,withComma,MillionRep,NegativeRep,Pos
itiveRep : String)

Description: This will format the number given in Value parameter.

DecLen is number of decimals.

WithComma - if {T} will insert commas.

MillionRep - if {T} will insert commas as it is in a millions system.

NegativeRep - if {br} then shows negative amounts within parenthesis else appends the given string to the negative number.

PositiveRep - if {br} then shows negative amounts within parenthesis else appends the given string to the positive number.

Example :

FormatAmount(100000, 2, {F}, {br}, {}) -> 1,00,000.00

FormatAmount(-100000, 2, {F}, {br}, {}) -> (1,00,000.00)

FormatAmount(100000, 2, {F}, {Cr}, {}) -> 1,00,000.00 Cr

FormatAmount(100000, 2, {T}, {}, {Db}) -> 100,000.00 Db

Function : FormatDateTime
Syntax : FormatDateTime(formatstring, datetimevariable)
Description : The given datetimevariable will be converted using the given formatstring.
If the formatstring is empty, the datetimevariable is formatted as if a 'c' format specifier had been given.
The first form of FormatDateTime is not thread-safe, because it uses localization information contained in global variables. The second form of FormatDateTime, which is thread-safe, refers to localization information contained in the FormatSettings parameter. Before calling the thread-safe form of FormatDateTime, you must populate FormatSettings with localization information. To populate FormatSettings with a set of default locale values, call GetLocaleFormatSettings.
Date Time Format Strings are composed from specifiers that represent values to be inserted into the formatted string. Some specifiers (such as "d"), simply format numbers or strings. Other specifiers (such as "/") refer to locale-specific strings

from global variables.
In the following table, specifiers are given in lower case. Case is ignored in formats, except for the "am/pm" and "a/p" specifiers.

Specifier	Displays
c	Displays the date using the format given by the ShortDateFormat global variable, followed by format given by the LongTimeFormat global variable. The time is not displayed if the date-tim midnight precisely.
d	Displays the day as a number without a leading zero (1-31).
dd	Displays the day as a number with a leading zero (01-31).
ddd	Displays the day as an abbreviation (Sun-Sat) using the strings given by the ShortDayNames g
dddd	Displays the day as a full name (Sunday-Saturday) using the strings given by the LongDayNar variable.
ddddd	Displays the date using the format given by the ShortDateFormat global variable.
dddddd	Displays the date using the format given by the LongDateFormat global variable.
e	(Windows only) Displays the year in the current period/era as a number without a leading zero and Taiwanese locales only).
ee	(Windows only) Displays the year in the current period/era as a number with a leading zero (Ja and Taiwanese locales only).
g	(Windows only) Displays the period/era as an abbreviation (Japanese and Taiwanese locales o
gg	(Windows only) Displays the period/era as a full name. (Japanese and Taiwanese locales only)
m	Displays the month as a number without a leading zero (1-12). If the m specifier immediately f specifier, the minute rather than the month is displayed.
mm	Displays the month as a number with a leading zero (01-12). If the mm specifier immediately f specifier, the minute rather than the month is displayed.
mmm	Displays the month as an abbreviation (Jan-Dec) using the strings given by the ShortMonthNa variable.
mmmm	Displays the month as a full name (January-December) using the strings given by the LongMo variable.
yy	Displays the year as a two-digit number (00-99).
yyyy	Displays the year as a four-digit number (0000-9999).
h	Displays the hour without a leading zero (0-23).
hh	Displays the hour with a leading zero (00-23).
n	Displays the minute without a leading zero (0-59).
nn	Displays the minute with a leading zero (00-59).
s	Displays the second without a leading zero (0-59).
ss	Displays the second with a leading zero (00-59).
z	Displays the millisecond without a leading zero (0-999).
zzz	Displays the millisecond with a leading zero (000-999).
t	Displays the time using the format given by the ShortTimeFormat global variable.
tt	Displays the time using the format given by the LongTimeFormat global variable.
am/pm	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'am' for any hour befor for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the resu

	accordingly.
a/p	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'a' for any hour before any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is dis accordingly.
ampm	Uses the 12-hour clock for the preceding h or hh specifier, and displays the contents of the Tin variable for any hour before noon, and the contents of the TimePMString global variable for ar
/	Displays the date separator character given by the DateSeparator global variable.
:	Displays the time separator character given by the TimeSeparator global variable.
'xx'/"xx"	Characters enclosed in single or double quotes are displayed as-is, and do not affect formatting

Function : Filewrite
Syntax : Filewrite((Formatfile, TargetFile)
Description : Used in a TStruct to convert the data related to a transaction into any ASCII file format.
The format of the ASCII should be created and stored prior to using this function. Fields are represented in the file within curly braces. In case one of the lines in the format file consists of grid fields, then the line will be repeated for the rows in the grid.
This can be used to create any kind of file formats for interfacing with other devices like barcode printer or to enable electronic data interchange (EDI)

Function : Findrecord
Syntax : Findrecord(SQLName, ColumnName, ColumnValue)
Description : Finds the record that has the given ColumnValue in the given columnname in the SQL result identified by SQLName.

Function : Getdelimitedstr
Syntax :Getdelimitedstr(SQLName, ColumnName, Delimiter)
Description : All the values in the column will be concatenated to a single string separated by the delimiter.
Example : A comma separated string can be created by setting the delimiter as ‘,’.

Function : GetId
Syntax : GetId({FieldName}, RowNo: Integer)
Description : Returns the record id of a normalized selection field at the given rowno.

Function : GetInteger
Syntax : GetInteger(Value :numeric)
Description : Returns the integer part in a decimal number.

Example :
GetInteger(12.33)
Result : 12.

Function : GetLength
Syntax :GetLength({value})
Description : Returns the length of the string.
Example :
GetLength({xyz})
Result : 3.

Function : GetMax
Syntax : GetMax({ColumnName })
Description : Applicable only to grid fields.Returns the row that contains the maximum value in a column in a grid DC.
Use in : TStructs
Example:
Consider a grid frame as follows

Sno	Category	Amount
1	A	100.00
2	A	200.00
3	B	300.00
4	C	400.00
5	A	500.00

GetMax({Amount})
Result : 5 (5th row contains the maximum value in the amount column)

Function : GetMin
Syntax : GetMin(ColumnName)
Description : Applicable only to grid fields.Returns the row in the grid that contains the minimum value in the column given by fieldname.
Use in : TStructs
Example:
Consider a grid frame as follows

no	Category	Amount
1	A	100.00
2	A	200.00
3	B	300.00
4	C	400.00

5	A	500.0

GetMin({Amount})
Result : 1 (the 1st row contains the minimum value in the amount column)

Function : GetMod
Syntax : GetMod(numerator, denominator)
Description : Returns the reminder after division.

Function : GetRow
Syntax : GetRow({ColumnName }, Fieldvalue)
Description : Gets the first row that matches the given fieldvalue in the column.
Use in : TStructs
Example:
Consider a grid frame as follows:

Sno	Category	Amount
1	A	100.00
2	A	200.00
3	B	300.00
4	C	400.00
5	A	500.00

GetRow({Category}, {A}) returns 1. Only the first matching rowno is returned.

Function : GetRowCount
Syntax : GetRowCount({Columnname})
Description : Returns the number of rows in the grid under the given column name.
Use in : TStructs
Example:
Consider a grid frame as follows

Sno	Category	Amount
1	A	100.00
2	A	200.00
3	B	300.00
4	C	400.00
5	A	500.00

GetRowCount({Category})
Result : 5

Function : GetStockValue
Syntax :GetStockValue(ItemMasterID, TransDate, Qty, ValuationMethod, LocationId)
Description : Returns the issue value for Qty of an item as on the date given by TransDate. The ItemMasterID is the record id of the required item. The ValuationMethod can be 'f' for FIFO or 'W' for weighted average. The LocationId parameter is optional. If this is provided, the issue value will be calculated considering the stock in the given location.

Function : GetValue
Syntax : GetValue({Fieldname} , RowNo: numeric)
Description : Returns the value of the field at the given row.FieldName is the name of the field. For fields in a non grid DC, give the rowno as 1.
Use in : TStructs
Example:
Consider a grid frame as follows

Sno	Category	Amount
1	A	100.00
2	A	200.00
3	B	300.00
4	C	400.00
5	A	500.00

GetValue({Category}, 3)
Result : B

Function : IIF
Syntax : IIF(Expression, True_Result, False_Result : String)
Description : Evaluates the expression.
If expression returns True, then the True_result is returned else the false_result is returned.
Example :
iif(10 > 5, {Greater}, {Lesser})
Result : Greater
iif((10-6) > 5, {Greater}, {Lesser})
Result : Lesser
iif(10 < 5, {Lesser 1}, iif(10 < 11, {Greater 2}, {Lesser 2}))

Result : Greater 2
iif(10 < 5, {Lesser 1}, iif(10 > 11, {Greater 2}, {Lesser 2}))
Result : Lesser 2
iif((10-6) < 5, {Lesser 1}, iif(10 > 11, {Greater 2}, {Lesser 2}))
Result : Lesser 1

Function : IsEmpty
Syntax : IsEmpty(Value)
Description : Returns true if the given value is null.

Function : IsEmptyValue
Syntax : IsEmptyValue(Value, Datatype)
Description : Returns true if the given value is empty. However, zero in case of numeric is not considered as empty.

Function : IsVarEmpty
Syntax : IsVarEmpty(Variablename)
Description : Returns true if the variable is empty.

Function : LastDayOfMonth
Syntax : LastDayOfMonth(DateVar)
Description :Returns the last day of the month in datevar. It can be 28, 29, 30 or 31.

Function : LeftPad
Syntax : LeftPad(S: String; MaxLength: integer; c: Char)
Description : If the length of the given string S is less than MaxLength, then pads it to the left with the character given by c to make its length equal to MaxLength. If the length of the given word is greater or equal to the MaxLength then the given word is returned as it is.
Example :
LeftPad('abc', 5, 'x')
Result : 'xxabc'

Function : Lower
Syntax : Lower(Str : String)
Description : Converts the given string to lower case.

Function : MakeDate
Syntax :MakeDate(day,month,year)
Description :Returns the date

Function : Monthofdate
Syntax :Monthofdate(Date)
Description :Returns the month of date

Function : MandY
Syntax : MandY(PDate : Date)
Description : Returns the month and year in the given date in the format YYYYMM.
Example :
MandY({01/04/2005})
Result : 200504
MandY({01/10/2006})
Result : 200610

Function :OpenDialog
Syntax : OpenDialog
Description : Displays the file open dialog to the user. The selected file name is returned to the user.

Function : Pad
Syntax : Pad(S: String; MaxLength: integer; c: Char)
Description :Pads the string S to the right with the character given in C. If the length of the given word is greater or equal to the MaxLength then the given word is returned as it is.
Example :
Pad('abc', 5, 'x')
Result : 'abcxx'

Function : Power
Syntax :Power(ConstVal, Exponent)
Description : The result will be Constval raised to the given Exponent.

Function : RegVar
Syntax : RegVar(varname, datatype, value)
Description : Registers the varname as a variable. This can be used in axpert

expressions within the current structure

Function : Rnd
Syntax : Rnd(Amount, RoundTo : Numeric)
Description : Rounds off the given amount to the nearest given RoundTo figure. The RoundTo could be 100 to round off the amount to nearest 100. If RoundTo is 50, the amount will be rounded to the nearest 50. This could be useful to roundoff amounts to the nearest rupee or nearest 50 paise
Example :
Rnd(100.20, 50) Result : 100.00
Rnd(100.45, 50) Result : 100.50
Rnd(100.65, 50) Result : 100.50
Rnd(100.20, 100) Result : 100.00
Rnd(100.65, 100) Result : 101.00

Function : Round
Syntax : Round(Num , Decimals:Numeric)
Description : Rounds the given number to given decimals.
Example :
Round(100.334,2) Result : 100.33
Round(100.337,2) Result : 100.34

Function : SaveDialog
Syntax : SaveDialog
Description : Displays the save file dialog to the user. The selected file name is returned to the user

Function : SetExtSequence
Syntax : SetExtSequence(TransId, FieldName, Prefix)
Description :The sequences defined in another TStruct could be used to generate numbers for a field in the current TStruct. The TransId parameter is the name of the other TStruct. This will enable sharing a single sequence across different TStructs

Function : SetSequence
Syntax : SetSequence(FieldName, Prefix:String)
Description : Auto generated fields can have multiple sequences. Each sequence is identified by a unique four letter prefix. There may be a need to set the sequence for number generation based on the value set in another field. In the on exit even of the field a user defined task could be written with a call to this function.. The name of the auto generated field and the prefix that should be used for number

generation are passed as parameters.
Note : The functions related to numbering sequence viz SetSequence and SetExtSequence cannot be used in OnFormLoad. Moreover, the case of the parameters should exactly match the ones defined in the tables. If the field name is defined in lower case, then parametric field name should also be in lower case .
Example:
Consider a TStruct that has a field name Type. This can have either A or B as its value. If it is A, then the DocNo field following this field should have the 'AAA-' sequence selected. If type is not A, then the sequence for DocNo should be 'BBB-'.
seq := iif(Type = {A}, {AAA-}, {BBB-})
SetSequence({DocNo}, Seq)

Function : SetValue
Syntax : SetValue({FieldName },Rowno: integer;Value)
Description : Sets the value for the field at the given row to the given value. For non grid fields, the rowno should be 1.
Example: Consider a TStruct that has a field named ItemCode in a Grid DC. If the value of the itemcode in row 2 is to be changed to 'x1' use SetValue({ItemCode},2,{x1}).

Function : Sqlget
Syntax : Sqlget(sqlname, columnname)
Description : Returns the value of the column in the SQL result from the current record.

Function : SQLRegVar
Syntax : SQLRegVar(SQLText)
Description : This function will register the result of the SQL for using them in expressions.The SQL text will be fired and a result set will be obtained. The SQL result should have the following columns : VarName, VarType, VarValue The function will register the given varname with the VarValue.
Example : If the SQL result is as follows:

VarName	VarType	VarValue
a1	n	100.00
a2	c	abc
a3	d	01/04/2004

Three variables will be registered as follows : a1 = 100.00 a2 = abc a3 = 01/04/2004

These variables could be used in subsequent expressions.

Function: SQLPost

Syntax:SQlPost({sql},{TargetTransid},{PrimaryField},{GroupField}
Description: This function is similar to GenMap. This function will generate an SQL statement, using the variables in the TStruct to post to another process structure. It offers more flexibility than GenMaps. SQLPost can handle processes in the background with no fornt-end user interaction, unlike with GenMaps. A real-life scenario will be in the case of batch-controlled stock, where stock needs to be issued, without the user choosing the batch from which issue has to be made. SQLPost can handle this easily.
Example:
SQlPost({SELECT POType, ProductDivision, VehicleSalesDealerCode, DealerPODate, DeliveryPlant, DealerPONumber,DeliveryPlant+DealerPONumber PlantPONumber, PartNo, Qty FROM VW_POOUTBOUNDDTL where POId=:recordid order by plantponumber
},{OBPOs},{PlantPONumber},{PlantPONumber})

Syntax:doStoreProcedure({procname},{inparam},{otparam})

Description: Calls a stored procedure using userdefined task

Example: doStoreProcedure({DoCost},{'01/01/2011'},{})

Function : Str
Syntax : Str(Num :Numeric)
Description : Converts given string to numeric.
Example :
Str({123}) Result :123.

Function : Stuff
Syntax :Stuff({Str1 }, {Str2}, P : Numeric)
Description : Inserts the str2 into str1 at position p.
Example:
Stuff({abcd}, {x}, 2) Result : axbcd
Stuff({abcd}, {}, 2) Result : acd
Stuff({abcd}, {xyz}, 2) Result : axyzbcd

Function : SubStr
Syntax : SubStr(S:string; Posn,Num:numeric)
Description : Returns num characters from Posn in the string S.
Example :
Substr({123Abc}, 4, 3) Result : Abc

Function : Sum
Syntax : Sum({ColumnName}, ColValue, {SumField })
Description : Returns the total value of the SumField from all rows in the grid that has the Colvalue in the Columnname. Applicable only to Grid DC.
Use in : TStructs
Example:
Consider a grid frame as follows

Sno	Category	Amount
1	A	100.00
2	A	200.00
3	B	300.00
4	C	400.00
5	A	500.00

Sum({Category}, {A}, {Amount}) will return 800.

Function : SumTill
Syntax : SumTill({ColumnName }, ColValue, {SumField}, RowNo)
Description : Returns the total value of the SumField from all rows in the grid that has the ColValue in the ColumnName and rowno less than or equal to the given RowNo. Applicable only to Grid DC.
Use in : TStructs
Example:
Consider a grid frame as follows

Sno	Category	Amount
1	A	100.00
2	A	200.00
3	B	300.00
4	C	400.00
5	A	500.00

SumTill({ItemCategory}, {A}, {Amount}, 4) will return 300.

Function : Time
Syntax : Time()
Description : Returns current time.

Function : TimeElapsed
Syntax : TimeElapsed(d1, d2)
Description : Returns time elapsed between two given dates along with time.

Function : Total
Syntax : Total({FieldName})
Description : Returns the total value of the given field in a TStruct.FieldName is the name of the grid column.Applicable only to grid DC
Use in : TStructs
Example:
Consider a grid frame as follows

Sno	Category	Amount
1	A	100.00
2	A	200.00
3	B	300.00
4	C	400.00
5	A	500.00

Total({Amount}) gives the result as 1500

Function : Trim
Syntax : Trim({Value})
Description : Trims spaces from the given string.

Function : Trimspace
Syntax : Trimspace({Value})
Description : Trims spaces from the given string.
Example :
Trimspace({ abc })
Result : abc

Function : Upper
Syntax : Upper(Str : String)
Description : Converts the given string to upper case.

Function : Val
Syntax :Val(Num:String)
Description : Converts given string to numeric.
Example :
Val({123})
Result : 123

Function : ValidEncodeDate
Syntax : ValidEncodeDate(year, month, day)
Description : Returns a date. If day is greater than highest day available in the month then the day will be considered as the last day of the month.

Function : Yearofdate
Syntax : Yearofdate(date)
Description : Returns year of date

Function : XRun
Syntax : XRun(ApplicationName)
Description : Runs the given application. This function does a shell execute of the given application name.
The application name can also be document name along with extension. If the given extension is recognized by windows then the document will be opened using the relevant application. For example if an MS word document file name is given as a parameter then XRun will open MS Word and open the document file.
Parameters can be passed to the application. The application name and the parameter name should be separated with a % character. If more than one parameter needs to be passed, then each parameter should be separated with % character.