

CurrentTime - Current simulation time during the run.
CurrentSim - Current simulation number which starts at 0.
StartTime - Starting time of the simulation. This is set in the Simulation Setup dialog.
EndTime - Ending time of the simulation. This is set in the Simulation Setup dialog.
AnimationOn - Animation state. Value is 0 if Animation is off and value is 1 if Animation is on.

<u>Operator</u>	<u>Description</u>
i += value;	Equivalent to $i = i + \text{value}$
i -= value;	Equivalent to $i = i - \text{value}$
i ++;	Equivalent to $i = i + 1$
i --;	Equivalent to $i = i - 1$

<u>Operator</u>	<u>Description</u>
AND or &&	Combination
OR or 	Combination
!= or <>	Not equal to
==	Equal to

```

if (a < B)                                If Statement
    Statement1;
else if (B > C and C != A)
{
    Statement2;
    Statement3;
}
else
    UserError("Wow. How did this happen");
    
```

```

X = 10;                                       For Loop
Y = 1;
For( i = 1; i <= 3; i ++ )
{
    X = X + 5;
    Y = Y * 2;
}
    
```

Ceil (real x) - Returns the smallest integral value that is not less than x. **Common Functions**
Floor (real x) - Returns the largest integral value that is not greater than x.
FixDecimal (real x, integer fixFigs) - Sets the number of figures after the decimal place.
Min2 (real x, real y) - Returns the minimum of the two arguments.
Max2 (real x, real y) - Returns the maximum of the two arguments.
RandomReal () - Uniform pseudo-random real number in the range $\{0.0 \leq x < 1\}$
RealMod (real x, real y) - Returns the remainder of x divided by y.
UserError ("string s") - Opens a dialog with an OK button displaying the string s.
MyBlockNum () - Global number of the block in which the function is called.
GetTimeUnits () - Returns the currently selected Time Units from the Simulation Setup dialog.
ConvertTimeUnits (real value, integer FromType , integer ToType) - Converts from one type of time unit to another.

Common Animation Functions

Animation Example

AnimationShow (integer obj) - Shows the hidden object.
AnimationHide (integer obj, integer outsidelcon) - Hides the object.
AnimationText (integer obj, String text) - Animates the text.
AnimationTextTransparent (integer obj, String text) - Animates the text with a transparent background.
AnimationTextSize (integer obj, integer Size) - Sets the text size.
AnimationTextAlign(integer obj, integer justification) - Aligns the text.
AnimationEColor (integer obj, integer eColorValue) - Sets the color.
AnimationLevel (integer obj, real level) - level between 0 and 1.

```

if(AnimationOn == True)
{
    AnimationText (-1, "Utilization:" + ### );
    AnimationShow (-1);
}
else
    AnimationHide (-1, False);

// object number is negative for H-blocks
    
```

Common Database Functions

DATABASE

DBDatabaseGetIndex ("databaseName")

TABLE

DBTableGetIndex (integer databaseIndex, "tableName")

DBTablesGetNum (integer databaseIndex)

FIELD

DBFieldGetIndex (integer databaseIndex, integer tableIndex, "fieldname")

DBFieldsGetNum (integer databaseIndex, integer tableIndex)

RECORD

DBRecordsDelete (integer databaseIndex, integer tableIndex, integer startRecord, integer endRecord)

DBRecordsInsert (integer databaseIndex, integer tableIndex, integer insertAtRecord, integer numberRecords)

DBRecordsGetNum (integer databaseIndex, integer tableIndex)

READ & WRITE FUNCTIONS

DBDataGetAsNumber (integer databaseIndex, integer tableIndex, integer fieldIndex, integer recordIndex)

DBDataGetAsString (integer databaseIndex, integer tableIndex, integer fieldIndex, integer recordIndex)

DBDataSetAsNumber (int databaseIndex, integer tableIndex, integer fieldIndex, integer recordIndex, real valueDouble)

DBDataSetAsString (int databaseIndex, integer tableIndex, integer fieldIndex, integer recordIndex, Str255 valueString)

READ & WRITE FUNCTIONS WITH A CHILD FIELD

Integer parentArray[3]; // you must define this integer array to be passed into the DBDataGetParent function.

RecordIndex = **DBDataGetParent** (int DBIndex, int tableIndex, int fieldIndex, integer recordIndex, integer parentArray)

// [0] contains the parents table index; [1] contains the parents field index; [2] contains the parents record index

DBDataSetAsParentIndex (int databaseIndex, int tableIndex, integer fieldIndex, integer recordIndex, integer liIndex)

// liIndex is an integer variable that indicates the index of the parent value to be written.

Database Functions Example

// Read and Write Data

lrValue = **DBDataGetAsNumber** (DBIdx, myTableTIdx, myReadingFieldFldx, myRecordRIdx);

DBDataSetAsNumber (DBIdx, myTableTIdx, myWritingFieldFldx, myRecordRIdx, lrValue);

// Deleting All Records in a table

liNumOfRecords = **DBRecordsGetNum** (DBIdx, myTableTIdx);

DBRecordsDelete (DBIdx, myTableTIdx, 1, liNumOfRecords);

// Appending 1 Record to a table

liNumOfRecords = **DBRecordsGetNum** (DBIdx, myTableTIdx);

DBRecordsInsert (DBIdx, myTableTIdx, liNumOfRecords + 1, 1);

// Read from a Child field

Integer parentArray[3];

DBDataGetParent (DBIdx, myTableTIdx, myReadingFieldFldx, myRecordRIdx , parentArray);

liValue = parentArray[2];

// Write into a Child field

DBDataSetAsParentIndex (DBIdx, myTableTIdx, myWritingFieldFldx, myRecordRIdx , liValue);