IMPORT LANGUAGE

Πίνακας τροποποιήσεων / Προσθηκών

Ημερομηνία	Έκδοση	Θέμα	Σελίδα
26/6/2013	3.1.2650	GETFIELDVALUE	7

Εκφράσεις

Οι εκφράσεις της γλώσσας χωρίζονται μεταξύ τους με ελληνικό ερωτηματικό **';'**. Μπορούν να καταλαμβάνουν μια ή περισσότερες γραμμές ή και να υπάρχουν περισσότερες από μια σε κάθε γραμμή. Μπορεί επίσης ο χρήστης να ομαδοποιήσει μια η περισσότερες εκφράσεις (block) αν τις γράψει μέσα σε αγκύλες **{ }**.

Τύποι, Σταθερές & Μεταβλητές

Οι τύποι δεδομένων που υποστηρίζονται από τη γλώσσα είναι οι ακέραιοι αριθμοί (4 byte Integers) οι πραγματικοί (doubles) οι συμβολοσειρές (strings) και οι πίνακες (arrays). Δεν υπάρχει κανένας άμεσος τρόπος ορισμού από το χρήστη τύπων για τα δεδομένα τα οποία πρόκειται να διαχειρισθεί. Ο τύπος των δεδομένων ορίζεται έμμεσα από τα ίδια τα δεδομένα.

Σταθερές ορίζονται οι εκφράσεις οι οποίες δεν μπορούν να μεταβάλλουν την τιμή τους. Ανήκουν σε κάποιον από τους προαναφερθέντες τύπους και αναλυτικότερα είναι:
Οι ακέραιες σταθερές που μπορούν να έχουν τιμές στο διάστημα από –2147483648 έως +2147483647. π.χ. 1, -2 ,-3432, 2*2, 1002 1+12+1 κ.λπ.
Οι πραγματικές σταθερές που παίρνουν δεκαδικές τιμές και το ακέραιο τμήμα ξεχωρίζει από το δεκαδικό με το σύμβολο '.' Πχ 1.1 -10.23, 100.09, 30.0 κ.λπ.

Οι συμβολοσειρές οι οποίες αποτελούνται από μια σειρά οποιονδήποτε γραμμάτων, αριθμών ή συμβόλων θέλουμε αρκεί να βρίσκονται κλεισμένα μέσα σε αποστρόφους πχ 'asdfasdf', 'George', '2345$fe_', 'A'+'1' κ.λπ.

Τέλος, οι πίνακες οι οποίοι ορίζονται από μια σειρά στοιχείων οποιονδήποτε τύπων τα οποία βρίσκονται κλεισμένα σε [και], π.χ.
[1, 1, 1.1+1, '1234'], [10+1, 5.01+1.2,'AB'+'01'] κ.λπ.

Οι μεταβλητές σε αντίθεση με τις σταθερές μπορούν να μεταβάλλουν το περιεχόμενό τους κατά τη διάρκεια της εκτέλεσης των εκφράσεων της γλώσσας. Δεν ανήκουν σε κανένα τύπο δεδομένων και έτσι μπορούν να δεχθούν περιεχόμενο οποιουδήποτε τύπου. Χρησιμοποιούνται για την προσωρινή ή μόνιμη αποθήκευση δεδομένων, καθώς και για την επεξεργασία τους. Το όνομα τους πρέπει να ξεκινά οπωσδήποτε από χαρακτήρα του Λατινικού αλφαβήτου ή το σύμβολο '_' ενώ στην συνέχεια μπορούν να χρησιμοποιηθούν και τα ψηφία από '0' έως '9'.
Παραδείγματα σωστών ονομάτων είναι τα I, X, _DATE, X_1, Y12 κ.λπ.
Παραδείγματα ονομάτων που δεν επιτρέπεται να χρησιμοποιούνται σαν ονόματα μεταβλητών είναι τα 1K, $S, %ENA κ.λπ.

Ορισμός Μεταβλητών

Ορίζουμε νέες μεταβλητές με τη δεσμευμένη λέξη "var" ακολουθούμενη από τις μεταβλητές οι οποίες χωρίζονται μεταξύ τους με κόμμα ','.
Η σύνταξη δηλαδή είναι της μορφής:

var μεταβλητή1, μεταβλητή2, ..., μεταβλητήN;

Επίσης, υπάρχει η δυνατότητα να δοθεί και αρχική τιμή στην μεταβλητή, αν με τον ορισμό της ακολουθεί έκφραση ανάθεσης τιμής πχ var Var1=1, Var2 = 0;

Η δήλωση των νέων μεταβλητών μπορεί να γίνει σε οποιοδήποτε σημείο του script, με την προϋπόθεση ότι δεν διακόπτει κάποια άλλη έκφραση.

Λόγω του ότι η γλώσσα διαχειρίζεται αυτόματα τους τύπους των μεταβλητών δεν υπάρχει άμεσος τρόπος ορισμού τους. Μπορεί όμως να ορισθεί έμμεσα μέσω της τιμής τους. Έτσι

στην έκφραση var X=10.2; ορίζεται έμμεσα ο τύπος της μεταβλητής X σαν πραγματικό αριθμό (double).

Από τη στιγμή αυτή και μετά οι μεταβλητές είναι πλέον γνωστές και μπορούν να παίρνουν τιμές και γενικά να συμμετέχουν σε εκφράσεις του script.

Πίνακες

Ένας πίνακας αντιπροσωπεύει μια διευθετημένη συλλογή από στοιχεία, τα οποία μπορεί να προσπελαστούν είτε ενιαία, μέσω της μεταβλητής πίνακα, είτε μεμονωμένα μέσω του πίνακα και ενός ακεραίου που εκφράζει τη θέση του στοιχείου στον πίνακα.

Οι πίνακες περιέχουν στοιχεία που μπορεί να ανήκουν σε διαφορετικούς τύπους ή ακόμα και να είναι πίνακες. Το μέγεθός τους μεταβάλλεται δυναμικά ώστε να καλύπτει τις ανάγκες που επιβάλλουν τη χρήση τους.

Υπάρχουν δυο είδη πινάκων:

- Οι πίνακες των οποίων τα στοιχεία είναι τιμές
- Οι πίνακες των οποίων τα στοιχεία είναι μεταβλητές.

Ανάθεση τιμής σε μεταβλητές

Οι μεταβλητές, τόσο κατά τη δήλωσή τους, με τη χρήση της "var"όσο και σε οποιοδήποτε σημείο του script, μπορούν να αλλάξουν περιεχόμενο, η αλλαγή αυτή γίνεται χρησιμοποιώντας το σύμβολο '=' σε εκφράσεις της μορφής μεταβλητή=τιμή; όπως x=10;, y=5;, St='κάτι' αλλά και πολυπλοκότερες όπως x=x + 10/5 + 2*y;, St=St+' άλλο';

Τελεστές

Οι τελεστές συμπεριφέρονται σαν τις ενσωματωμένες συναρτήσεις της γλώσσας.
Για παράδειγμα, η έκφραση (X + Y) αποτελείται από τις μεταβλητές X και Y που λέγονται τελεσταίοι μαζί με τον τελεστή +. Όταν οι X και Y περιέχουν ακέραιες ή πραγματικές τιμές, η (X + Y) επιστρέφει το άθροισμά τους.
Στους τελεστές της γλώσσας ανήκουν τα παρακάτω σύμβολα και δεσμευμένες λέξεις: not ή !, *, /, div, mod, and, +, –, or, xor, ==, >, <, !=, <= και >=.

Ο τελεστής not ή !, είναι μοναδιαίος (δέχεται έναν τελεσταίο). Όλοι οι άλλοι τελεστές είναι δυαδικοί (δέχονται δυο τελεσταίους), εκτός από τους + και – που μπορούν να λειτουργήσουν και σαν μοναδιαίοι και σαν δυαδικοί. Οι μοναδιαίοι τελεστές προηγούνται του τελεσταίου τους (παράδειγμα, -B, +5). Οι δυαδικοί τελεστές τοποθετούνται ανάμεσα στους δυο τελεσταίους (παράδειγμα, A == 7).

Μερικοί τελεστές συμπεριφέρονται με τρόπο εξαρτώμενο από τον τύπο των δεδομένων. Παράδειγμα ο + για ακεραίους δίνει το άθροισμά τους, ενώ για συμβολοσειρές (strings) την ένωσή τους.
Οι τελεστές ανήκουν στις παρακάτω κατηγορίες:

- **Αριθμητικοί τελεστές:** Λειτουργούν με πραγματικούς ή ακεραίους, στους οποίους συμπεριλαμβάνονται οι +, –, *, /, div και mod.
  Δυαδικοί
  + πρόσθεση
  - αφαίρεση
  ***** πολλαπλασιασμός
  / πραγματική διαίρεση
  div ακέραια διαίρεση
  mod υπόλοιπο
  Μοναδιαίοι
  + , **-** προσδιορίζουν πρόσημο.

- **Λογικοί τελεστές:** Δέχονται σαν τελεσταίους ακέραιες εκφράσεις και επιστρέφουν τις ακέραιες τιμές 1 ή 0.

not ή **!** επιστρέφει 0 αν η έκφραση είναι διάφορη του μηδενός και 1 αν είναι 0. Παράδειγμα not (X==5).
and Λογικό 'και' των τελεσταίων, επιστρέφει 1 αν και οι δυο τελεσταίοι είναι δίαφοροι του μηδενός και μηδέν στις άλλες περιπτώσεις. Παράδειγμα (K and 1) ή (X==1) and (Y >= 5).
or Λογικό 'ή' επιστρέφει 1 αν τουλάχιστον ένας από τους τελεσταίους είναι 1 και 0 αν είναι μηδέν και οι δύο τελεστές.
xor Επιστρέφει 1 μόνο αν ο ένας τελεσταίος είναι μηδέν και ο άλλος διάφορος του μηδενός και μηδέν αν και οι δυο είναι μηδέν ή και οι δυο ίσοι με μηδέν.
Στις λογικές εκφράσεις πρέπει να γνωρίζουμε ότι η γλώσσα θεωρεί 'σωστή - True' οποιαδήποτε ακέραια τιμή είναι διαφορετική του μηδενός και False την τιμή 0.

- **Τελεστές συμβολοσειρών:** Είναι μόνο ο τελεστής + οποίος δέχεται σαν τελεσταίους δύο συμβολοσειρές και επιστρέφει την ένωσή τους.

- **Σχεσιακοί τελεστές:** Χρησιμοποιούνται για να συγκρίνουν δυο τελεσταίους.
  == Ελέγχει για ισότητα τους τελεσταίους και φέρνει σαν αποτέλεσμα 1 αν είναι ίσοι και 0 αν είναι διαφορετικοί. Παράδειγμα (X==Y).

  != Ελέγχει αν οι τελεσταίοι είναι διαφορετικοί και φέρνει σαν απότέλεσμα 1 αν είναι διαφορετικοί και 0 αν είναι ίσοι. Παράδειγμα (X != Y).

  < Μικρότερος από, επιστρέφει 1 αν ο πρώτος τελεσταίος είναι μικρότερος από το δεύτερο και 0 αν είναι μεγαλύτερος ή ίσος του. Παράδειγμα (X < Y).

  > Μεγαλύτερος από, επιστρέφει 1 αν ο πρώτος τελεσταίος είναι μεγαλύτερος από το δεύτερο και 0 αν είναι μικρότερος ή ίσος του. Παράδειγμα (Len > 0).

  <= Μικρότερος ή ίσος, επιστρέφει 1 αν ο πρώτος τελεσταίος είναι μικρότερος ή ίσος από το δεύτερο και 0 αν είναι μεγαλύτερός του. Παράδειγμα (Cnt <= I).

  >= Μεγαλύτερος ή ίσος επιστρέφει 1 αν ο πρώτος τελεσταίος είναι μεγαλύτερος ή ίσος από το δεύτερο και 0 αν είναι μικρότερός του. Παράδειγμα (I >= 1).

  Οι τελεσταίοι πρέπει να ανήκουν σε συμβατούς τύπους ή αλλιώς σε τύπους που μπορούν να συγκριθούν.

  Οι συμβολοσειρές ή strings συγκρίνονται με βάση την κατάταξή τους κατά αλφαβητική σειρά. Οι χαρακτήρες στις συγκρίσεις, θεωρούνται συμβολοσειρές με μήκος 1 χαρακτήρα.

Εντολές Ελέγχου και Αλλαγής της Ροής Εκτέλεσης του Script

Συχνά κατά τη συγγραφή ενός τμήματος κώδικα, κάποιο τμήμα του πρέπει να εκτελείται μόνο αν πληρούνται κάποιες προϋποθέσεις και όχι πάντα.

Ο τρόπος με τον οποίο ικανοποιείται η παραπάνω ανάγκη είναι η χρήση της εντολής 'if'. Υπάρχουν δυο μορφές του if η "**if...then**" και η "**if...then...else**".

Η σύνταξη μιας έκφρασης "**if...then**" είναι:

if παράσταση then statement

Όπου η ακέραια παράσταση επιστρέφει ακέραια τιμή. Αν η παράσταση επιστρέψει τιμή διάφορη του μηδενός το statement εκτελείται, αν επιστρέψει μηδέν το statement δεν εκτελείται.
Παράδειγμα:
```
if J != 0 then Res = I / J;
```

Η σύνταξη μιας έκφρασης "**if...then...else**" είναι:

if παράσταση then statement1 else statement2

Επίσης, η παράσταση πρέπει να επιστρέφει ακέραια τιμή. Αν επιστρέψει τιμή διάφορη του μηδενός εκτελείται το statement1, αλλιώς το εκτελείται το statement2.
Παράδειγμα:
```
if J == 0 then
  return 0;
else
```

```
  return I/J;
```
Τα τμήματα then και else μπορούν να περιέχουν μια ή περισσότερες εντολές Παράδειγμα:

```
if J != 0 then
{
  Res = I/J;
  Count = Count + 1;
}
else
  if Count == Last then
    Done = 1;
  else
    return 0;
```

Μια πολύ ειδική κατάσταση δημιουργείται όταν υπάρχει μια σειρά από συνεχόμενες if εκφράσεις. Το θέμα προκύπτει επειδή μερικές εκφράσεις if έχουν τμήματα else και κάποιες άλλες όχι, κάνοντας δύσκολη τη διάκριση, λόγω του ότι η σύνταξη και στις δυο περιπτώσεις είναι ίδια. Σε μια σειρά από συνεχόμενες συνθήκες στην οποία υπάρχουν λιγότερα τμήματα "else" απ' ότι "if", δεν είναι ξεκάθαρο ποιό else αντιστοιχεί σε ποιό if. Ας θεωρήσουμε την παρακάτω έκφραση:

if expression1 then if expression2 then statement1 else statement2;

Αυτή μπορεί να ερμηνευθεί με δυο διαφορετικούς τρόπους:

```
if παράσταση1 then [ if παράσταση2 then statement1 else statement2 ];
if παράσταση1 then [ if παράσταση2 then statement1 ] else statement2;
```

Η γλώσσα ερμηνεύει πάντα με τον πρώτο τρόπο. Έτσι ο παρακάτω κώδικας
```
if ... { παράσταση1 } then
  if ... { παράσταση2 } then
    ... { statement1 }
  else
    ... { statement2 }
```

είναι ισοδύναμος του

```
if ... { παράσταση1 } then
{
  if ... { παράσταση2 } then
    ... { statement1 }
  else
    ... { statement2 }
}
```

Ο κανόνας στην περίπτωση των συνεχόμενων συνθηκών είναι ο ακόλουθος: Ξεκινώντας από την εσωτερικότερη συνθήκη, αντιστοιχούμε κάθε "else" στο πλησιέστερο του από πάνω μη αντιστοιχισμένο "if". Για να ερμηνεύσει η γλώσσα το παραπάνω παράδειγμα με το δεύτερο τρόπο πρέπει να το διατυπωθεί με ρητό τρόπο όπως:

```
if ... { παράσταση 1 } then
{
  if ... { παράσταση 2 } then
    ... { statement1 }
end
}
  ... { statement2 }
```

Εντολές ανακύκλωσης (while)

Η εντολή "**while**" χρησιμοποιείται για την υλοποίηση τμημάτων κώδικα, τα τμήματα επαναλαμβάνονται υπό κάποια συνθήκη η οποία λέγεται συνθήκη ελέγχου. Ο υπολογισμός

της συνθήκης ελέγχου πραγματοποιείται πριν την πρώτη εκτέλεση των εντολών που ακολουθούν. Εκεί αν η συνθήκη έχει τιμή μηδέν ή αλλιώς false οι εντολές που ακολουθούν δεν θα εκτελούνται.

Η σύνταξη της είναι:

while έκφραση do εντολή [ή εντολές σε {}]

Η έκφραση που ακολουθεί τη "while" πρέπει να έχει ακέραια τιμή και οι εντολές που ακολουθούν μπορεί να και πολλές αν περιέχονται σε {}.

Η "while" εκτελεί τις εντολές που ακολουθούν συνεχώς, ελέγχοντας την έκφραση πριν από κάθε εκτέλεση. Για όσο χρόνο η τιμή της έκφρασης υπολογίζεται ίση με 1, η εκτέλεση των εντολών συνεχίζεται.

Παραδείγματα:

```
while Data[I] <> X do I := I + 1;
while I > 0 do
{
  if I mod 2 == 1 then Z := Z * X;
  I = I div 2;
}
```

Ενσωματωμένες Συναρτήσεις και Διαδικασίες

- **ExecuteDBProc**(ProcName: string; InParams: string;,
  InputData: array; OutParams: string;
  Var OutData: VarArray);

  Εκτελεί την store procedure με όνομα ProcName στη βάση.
  ProcName: Είναι το όνομα της store procedure που θα εκτελεστεί.

  InParams string: Δίνουμε τα ονόματα των μεταβλητών χωρισμένα με ελληνικό ερωτηματικό π.χ. 'x;y;i'.

  InputData array: Οι τιμές των παραπάνω μεταβλητών με τη μορφή πίνακα π.χ. [12.54, 'λαλα', 10].

  OutParams string: Τα ονόματα των μεταβλητών που θα μας επιστρέψει, χωρισμένα με ελληνικό ερωτηματικό π.χ. 'x;y;i'.

  Var OutData VarArray: Πίνακας του οποίου τα στοιχεία είναι μεταβλητές και στα οποία θα τοποθετηθούν οι τιμές τις οποίες θα επιστρέψει η procedure και τις οποίες έχουμε αναφέρει στην προηγούμενη παράμετρο.

- **ShowMessage**(st: string [DialogType: Integer]);
  Εκτυπώνει ένα μήνυμα ή κάνει μια απλή ερώτηση με μορφή διαλόγου.
  DialogType:

  0: Information
  1: Error
  2: Warning
  3: Ερώτηση Ναι/Όχι
  4: Ερώτηση Ναι/Όχι/Άκυρο.

  Αν το DialogType είναι 3 ή 4 επιστρέφει τις παρακάτω τιμές:

  0: Τίποτε
  1: Πατήθηκε το πλήκτρο [Ok]
  2: Πατήθηκε το πλήκτρο [Άκυρο]
  6: Πατήθηκε το πλήκτρο [Ναι]
  7: Πατήθηκε το πλήκτρο [Όχι]

- **VarIsNull**(Variabe): Integer;
  Επιστρέφει 1 αν η δοσμένη μεταβλητή περιέχει την τιμή Null. Αν η μεταβλητή περιέχει οποιαδήποτε άλλη τιμή η συνάρτηση επιστρέφει 0.

- **Copy**(st: string, Position, Length: Integer): string;
  "St" είναι έκφραση τύπου string. Position και Length ακεραίου τύπου εκφράσεις.

  Η "Copy" επιστρέφει τμήμα string που αποτελείται από "Length" χαρακτήρες ξεκινώντας από το "Position".

  Αν το "Position" είναι μεγαλύτερο από το μήκος του "St", η "Copy" επιστρέφει ένα άδειο string.

  Αν η "Count" αναφέρεται σε περισσότερους χαρακτήρες από όσους είναι διαθέσιμοι, μόνο οι χαρακτήρες από τη θέση St[Position] έως το τέλος του "St" επιστρέφονται.

- **Pos**(SubStr, Str: string): Integer;
  Η "Pos" αναζητά ένα string μέσα σε ένα άλλο. Οι παράμετροι "Substr" and "Str" είναι εκφράσεις τύπου string.

  Αναζητά το "SubStr" μέσα στο "Str" και επιστρέφει μια ακέραια τιμή η οποία εκφράζει τη θέση του πρώτου χαρακτήρα του "SubStr" στο "Str". Κάνει διάκριση μικρών/κεφαλαίων χαρακτήρων. Αν το "SubStr" δεν βρεθεί επιστρέφει μηδέν.

- **Len**(Str: string): Integer;
  Η "Length" επιστρέφει τον αριθμό των χαρακτήρων που περιέχει το string.

- **Char**(Byte): Character;
  Επιστρέφει το χαρακτήρα που αντιστοιχεί στον ascii κωδικό της παραμέτρου.

- **Ascii**(Character): Integer;
  Επιστρέφει τον ascii κωδικό του χαρακτήρα που ορίζουμε στην παράμετρο.

- **LTrim**(S: string): string;
  Η "TrimLeft" επιστρέφει ένα αντίγραφο του S από το οποίο έχουν αφαιρεθεί τα κενά και οι χαρακτήρες ελέγχου που υπάρχουν στην αρχή του.

- **RTrim**(S: string): string;
  Η "TrimRight" επιστρέφει ένα αντίγραφο του S από το οποίο έχουν αφαιρεθεί τα κενά και οι χαρακτήρες ελέγχου που υπάρχουν στο τέλος του.

- **Trim**(S: string): string;
  Αφαιρεί τα κενά και τους χαρακτήρες ελέγχου που υπάρχουν στην αρχή και το τέλος του string S.

- **StrOfChar**(Count: Integer, Ch: Char): string;
  Επιστρέφει ένα string το οποίο περιέχει Count φορές το χαρακτήρα που ορίζει η παράμετρος Ch.

  Παράδειγμα:
  η έκφραση S = StrOfChar(10, 'A');
  Θέτει στην μεταβλητή S το string 'AAAAAAAAAA'.

- **Abs**(X): Double;
  Η "Abs" γυρίζει την απόλυτη τιμή της παραμέτρου X.
  Η παράμετρος X είναι έκφραση τύπου ακεραίου ή πραγματικού αριθμού.
- **Sign**(X): integer;
  Επιστρέφει το πρόσημο της παραμέτρου X.
  Η παράμετρος X είναι έκφραση τύπου ακεραίου ή πραγματικού αριθμού.
  Συγκεκριμένα, επιστρέφει –1 αν η παράμετρος έχει αρνητική τιμή ή 1 αν έχει θετική.

- **Round**(X: Double; Decimals: integer): Double;
  Κάνει στρογγύλευση του πραγματικού αριθμού X σε τόσα δεκαδικά ψηφία όσα ορίζει η ακέραια παράμετρος Decimals.

  Παράδειγμα:
  η έκφραση R = Round(10.324, 2);
  Θέτει στην μεταβλητή R την τιμή 10.32.

- **Upper**(S: string): string
  Η "UpperCase" μετατρέπει όλους τους χαρακτήρες ανάμεσα στους 'a' και 'z' στα κεφαλαία. Προσοχή μετατρέπει μόνο τους Λατινικούς χαρακτήρες.

- **RaiseException**(Message: string);
  Τερματίζει την εκτέλεση του script με μήνυμα λάθους.

- **StrToDate**(S: string): TDate;
  Μετατρέπει το string S σε ημερομηνία. Αν η μορφή του S δεν ακολουθεί τη μορφή που ορίζεται στις παραμέτρους του συστήματος για τις ημερομηνίες, η συνάρτηση προκαλεί exception, με αποτέλεσμα τον τερματισμό της εκτέλεσης του script.

  SQL statements, τα οποία θα εκτελεστούν
  Τα statements που επιτρέπονται είναι τα INSERT, UPDATE, DELETE, τα υπόλοιπα πρέπει να ξεκινάνε με '$'.

  Π.χ.
  [CusCode, CusName] = SELECT CODE, NAME FROM CUSTOMER WHERE ID=:CUSID;
  ή
  X = SELECT CODE, NAME FROM CUSTOMER WHERE ID=:CUSID;
  [CusCode, CusName] = X

  UPDATE CUSTOMER SET ACTIVE = 1
  WHERE ACTIVE = 0;
  UPDATE C01IMPORT SET ATLUPD=1
  WHERE ATLUPD IS NULL;
  $DROP TABLE CUSTOMER;

- **GETFIELDVALUE**(S: string): Variant;
  Δέχεται όρισμα ένα string S: αποτελεί το όνομα του πεδίου του οποίου ζητείται η τρέχουσα τιμή. Προαιρετικά μπορεί να οριστεί και το όνομα του πίνακα στην εξής μορφή:

  [TableName.]FieldName

  Σε περίπτωση που δεν οριστεί το πρόθεμα TableName, τότε επιλέγεται ο κυρίως πίνακας του MODULE που έχει οριστεί στο Import script. Η επιστρέφει την τρέχουσα τιμή του πεδίου εφόσον αυτό υπάρχει. Σε αντίθετη περίπτωση εμφανίζει μήνυμα σφάλματος.

  Πχ:

  GetFieldValue('NAME);
  GetFieldValue('CUSTOMER.NAME');