

sc\$UI

Users Interface Objects

Version: 1.0

29/11/2021



Περιεχόμενα

Περιεχόμενα	2
Εισαγωγή	7
Ορολογία	7
Enumerator	7
GUID	7
Properties	7
Methods	8
frm	8
Γενικά.....	8
Methods	8
Equals()	8
GetHashCode()	8
GetType().....	8
ToString()	8
Dispose().....	8
Επιστρεφόμενες Τιμές.....	9
poDialogRes	9
Methods	9
scDialogComplexResult	9
sc\$UI.Dialogs	10
Methods	10
Question(caption, text).....	10
InputText(caption, label, value, allowEmpty)	10
ToIgnoreAnyDataChanges(caption).....	11
AskSaveDataChanges(caption)	11



DeleteObject(caption)	12
ShowAsyncStorageItem (asyncHandleID).....	12
ShowError (caption , text)	12
ShowException(ex)	13
ShowInformation(caption, text)	13
ShowActionLogger(logger)	13
ShowActionLogger(logger, ex).....	14
ShowOpenFileDialog(title, initialDir, dialogFilter)	14
ShowSaveFileDialog(title, initialDir, dialogFilter)	15
Παραδείγματα DialogFilter	15
Progress Methods.....	15
Methods	15
ComplexProgressStart(caption).....	15
ComplexProgressNewBar(key, label).....	15
ComplexProgressBarUpdate(progressKey, barkey, pos, max, stepLabel).....	16
ComplexProgressBarClose(progressKey, barkey).....	16
ComplexProgressStop(progressKey).....	16
Παράδειγμα χρήσης ProgressMethods.....	16
scFormArgs (poFormArgs).....	17
Properties	17
AccessRightsInfo.....	17
DesignerIsActive	18
EditValue.....	18
FantomArgs	18
FormCaption.....	18
InArgs.....	18
IsDetailForm	18





OpenMode.....	18
OutArgs.....	18
UnPinned	18
ZoomCollection.....	18
Methods	19
InArg(name)	19
OutArg(name)	19
scForm	19
Properties	19
Args.....	19
IsDirty.....	19
RealType	19
Methods	19
ExecuteFormAction(actionName)	19
GetControlDisplayValue(controlName).....	20
GetControlEditValue(controlName)	20
GetCurrentObject(controlName).....	20
GetGridColumnDisplayValue(gridName, columnName)	20
GetGridColumnValue(gridName, columnName).....	21
GetProperty(name).....	21
GetTreeColumnDisplayValue(treeName, columnName)	22
GetTreeColumnValue(treeName, columnName).....	22
InvokeMethod(name, args)	22
InvokeMethodAsync(name, args).....	22
SetControlEditValue(controlName, value)	22
SetGridColumnValue(gridName, columnName, value).....	23
SetProperty(name)	23





SetTreeColumnValue(treeName, columnName, value)	23
Show().....	23
ShowDialog().....	23
sc\$UI.Forms	24
Create(domain, name, args)	24
GetFormType()	24
sc\$UI.Actions	25
Methods	25
FormShow(domain, name, args)	25
FormShowDialog(domain, name, args)	25
FormGetType()	25
FormCreate(domain, name, args).....	26
Invoke(domain, name, args)	26
IsEnabled(domain, name)	27
sc\$UI.Elements	27
Properties	27
scFormArgsResult	27
Methods	27
FormShowDialog(name, formArgs, inArgs)	27
ResultsShowDialog(name, formArgs, ds)	27
ResultsShowDialogWithResult(name, formArgs, ds).....	28
sc\$UI.Reports	28
Methods	28
AddReportFilter(args, filtername, filterValue)	28
SetReportTextFilterWhereType(args, filterName, whereType)	29
ShowReport(domain, name, viewName, args).....	29
ExecuteReport(domain, name, viewName, args).....	29





ShowCustomReport(reportID, viewName, args)	29
ShowCustomGrid(reportID, viewName, args)	30
ExecuteCustomGrid(reportID, viewName, args)	30
ShowCustomPivot(reportID, viewName, args)	30
ExecuteCustomPivot(reportID, viewName, args)	30
ShowCustomDataExport(reportID, viewName, args)	31
ExecuteCustomDataExport(reportID, viewName, args)	31
ShowERGReport(reportID, viewName, args)	31
ExecuteERGReport(reportID, viewName, args)	31
ShowERGGrid(reportID, viewName, args)	31
ExecuteERGGrid(reportID, viewName, args)	32
ShowERGPivot(reportID, viewName, args)	32
ExecuteERGPivot(reportID, viewName, args)	32
ShowERGDataExport(reportID, viewName, args)	32
ExecuteERGDataExport(reportID, viewName, args)	32
Βοηθήματα	32



Εισαγωγή

Στο κείμενο αυτό περιγράφονται τα βασικά Properties και Methods του **sc\$UI**. Μέσω αυτού μπορείτε να υλοποιήσετε με κώδικα: User Interface , Reporting Actions , Form Actions Η χρήση του σε κώδικα περιορίζεται στα σημεία που επιτρέπεται από την εφαρμογή αντίστοιχη λειτουργικότητα.

Προσοχή

Οι πληροφορίες αυτού του κειμένου απευθύνονται σε χρήστες με τουλάχιστον βασικές γνώσεις προγραμματισμού. Λανθασμένη Χρήση κώδικα μπορεί να προκαλέσει σοβαρές δυσλειτουργίες στην εφαρμογή και να οδηγήσει σε απώλεια δεδομένων. Η συγγραφή και οι δοκιμές θα πρέπει πάντα να γίνονται σε δοκιμαστικό περιβάλλον και στην συνέχεια να μεταφέρεται ο κώδικας στην εγκατάσταση για την οποία προορίζεται. Επίσης τα παραδείγματα που παρατίθενται είναι καθαρά για εκπαιδευτικούς σκοπούς και δεν θα πρέπει να μεταφέρονται αυτούσια.

Πληθώρα κειμένων και εκπαιδευτικών γενικά για κώδικα σε περιβάλλον Microsoft JScript μπορείτε να αναζητήσετε στα επίσημα sites της Microsoft.

Ορολογία

Enumerator

Τύπος = Int16 (Μικρός ακέραιος).

Για μεγαλύτερη ευελιξία αλλά και αποδοτικότητα αντί σε πινακοποιημένα πεδία να αποθηκεύονται αλφαριθμητικές τιμές η πληροφορία αυτή αποθηκεύεται σαν αριθμός. Με αυτό τον τρόπο επιτρέπεται τυποποίηση, ευκολότερη μετάφραση των λεκτικών που αντιστοιχίζονται και ταυτόχρονα οικονομία στο πλήθος των Bytes που αποθηκεύονται στη βάση.

GUID

Globally Unique Identifier ή Unique Identifier. Τύπος δεδομένων 16 byte ο οποίος εγγυάται μοναδικότητα ανάμεσα σε πεδία από διαφορετικούς πίνακες, βάσεις δεδομένων ή Server. Τα πεδία αυτού του τύπου στο PYLON έχουν πάντα κατάληξη ID.

Properties

Ιδιότητες. Τιμές που έχουν αποθηκευτεί σε κάποιο αντικείμενο (object) οι οποίες μπορούν να προστεθούν, να μεταβληθούν ή να διαγραφούν. Εξαιρέση όσες έχουν χαρακτηριστεί Read Only.



Methods

Ενέργειες (εκτελέσιμες συναρτήσεις), οι οποίες αποθηκεύονται σαν ιδιότητες αντικειμένων.

frm

Για τα παραδείγματα που αναφέρονται στο παρόν κείμενο χρησιμοποιείται ως όνομα μεταβλητής για την φόρμα. **<frm>**.

Γενικά

Γενικές μέθοδοι που χρησιμοποιούνται από διάφορα αντικείμενα και αφορούν ισότητα, τύπο, περιγραφή αντικειμένου κλπ.

Methods

Equals()

Ελέγχει την ισότητα ανάμεσα σε δύο Objects και επιστρέφει (Boolean) true ή false.

Παράμετροι: obj (Object).

Παράδειγμα: **<obj>.Equals(<obj>);**

GetHashCode()

Επιστρέφει Int32 hashCode

Παράδειγμα: **var hCode = <obj>.GetHashCode();**

GetType()

Επιστρέφει obj ([Type](#)) με πληροφορίες για τον τύπο ενός αντικειμένου.

Παράδειγμα: **var oType = <obj>.GetType();**

ToString()

Επιστρέφει την περιγραφή του αντικειμένου.

Παράδειγμα: **var lStr = <obj>.ToString();**

Dispose()

Μέθοδος που καλείται προκειμένου να ελευθερωθούν τα resources που έχουν δεσμευτεί από το αντικείμενο.

Παράδειγμα: **<obj>.Dispose();**



Επιστρεφόμενες Τιμές

poDialogRes

Dialog Result Enumerator. Επιστρέφεται από εκτέλεση παραθύρων διαλόγου (προκαθορισμένοι) βάσει του Button που πατήθηκε από τον χρήστη.

Enumerators

- poDialogRes.Yes = 0
- poDialogRes.No = 1
- poDialogRes.Ok = 2
- poDialogRes.Cancel = 3

Ο έλεγχος μπορεί να γίνει χρησιμοποιώντας είτε το αντίστοιχο Enumerator Value είτε την μεταβλητή.

Πχ `if (result == 0)` ή `if (result == poDialogRes.Yes)`

Methods

- poDialogRes.ToString(). Το αποτέλεσμα ως λεκτικό.
- poDialogRes.ToString(format). Το αποτέλεσμα ως λεκτικό με προκαθορισμένο format. Επιτρεπόμενα format "G", "g", "X", "x", "F", "f", "D" ή "d".
- poDialogRes.HasFlag(poDialogRes.Ok)
Επιστρέφει true ή false ανάλογα με το αν στον διάλογο από τον οποίο προέκυψε η τιμή επιστροφής εμφανίστηκε Button με την συγκεκριμένη τιμή.

scDialogComplexResult

Το αποτέλεσμα εκτέλεσης ενός σύνθετου παραθύρου διαλόγου (προκαθορισμένοι). Επιστρέφεται ως Object με δύο τιμές

- Data (Object). Τα δεδομένα που προκύπτουν από το παράθυρο διαλόγου.
- DialogResult ([poDialogRes](#)) : Το button που πατήθηκε ανάλογα με το είδος διαλόγου.

Παράδειγμα: [InputText](#)

sc\$UI.Dialogs

Χρησιμοποιείται για εμφάνιση διαλόγων με διαφορετική μορφή.

Μπορεί να χρησιμοποιηθεί σε Scripts με την προϋπόθεση ότι εκτελούνται **Client Side** άρα επιτρέπεται και UI, καθώς και σε Presenters. Η χρήση τους σε Server Side scripts θα προκαλέσει μήνυμα λάθους.

Methods

Question(caption, text)

Εμφανίζει ερώτηση στον χρήστη.

Παράμετροι:

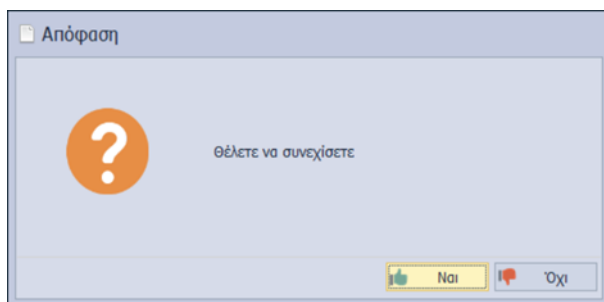
- **caption** (String) . Η επικεφαλίδα του διαλόγου.
- **text** (String) . Το κείμενο της ερώτησης.

Επιστρέφει: true ή false.

Παράδειγμα:

```
function main() {  
    var res = sc$UI.Dialogs.Question("Απόφαση", "Θέλετε να συνεχίσετε");  
    if ( res == true ) {    sc$UI.Dialogs.ShowInformation("Αποτέλεσμα","Ο χρήστης συμφωνεί") }  
    else {    sc$UI.Dialogs.ShowInformation("Αποτέλεσμα","Ο χρήστης δεν συμφωνεί");    }  
}
```

Αποτέλεσμα:



InputText(caption, label, value, allowEmpty)

Εμφάνιση προκαθορισμένου Edit Form όπου ο χρήστης μπορεί να πληκτρολογήσει κείμενο.

Παράμετροι:

- **caption** (String). Η επικεφαλίδα του διαλόγου.
- **label** (String) . Η επικεφαλίδα του κειμένου.
- **value** (String) Προτεινόμενη τιμή ή null.
- **allowEmpty** (Boolean). True ή false. Αν επιλεγεί false τότε το Button «Ναι» δεν ενεργοποιείται αν δεν πληκτρολογήσετε κείμενο ή δεν έχει προτεινόμενη τιμή «**value**».

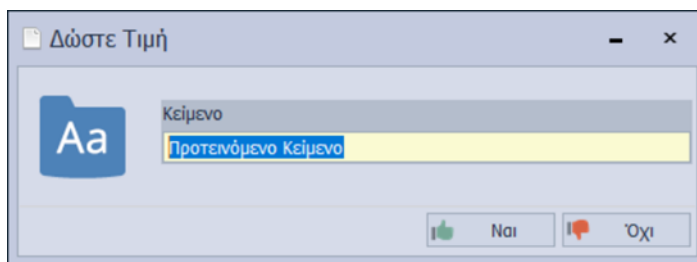


Επιστρέφει: [scDialogComplexResult](#). Τα Buttons αντιστοιχούν σε [poDialogRes.Ok](#) , [poDialogRes.Cancel](#).

Παράδειγμα:

```
function main() {  
    var res = sc$UI.Dialogs.InputText("Δώστε Τιμή", "Κείμενο", "Προτεινόμενο Κείμενο", false);  
    if (res.DialogResult == poDialogRes.Ok ) {  
        sc$UI.Dialogs.ShowInformation("Αποτέλεσμα", res.Data);  
    } else {  
        sc$UI.Dialogs.ShowInformation("Αποτέλεσμα", "Ακύρωση από τον χρήστη");  
    }  
}
```

Αποτέλεσμα:



[ToIgnoreAnyDataChanges\(caption \)](#)

Εμφανίζει προκαθορισμένη ερώτηση για απόρριψη αλλαγών.

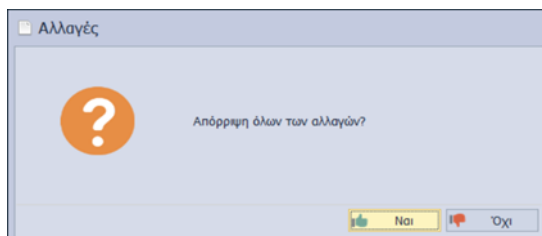
Παράμετροι: **caption** (String) . Η επικεφαλίδα του διαλόγου.

Επιστρέφει: true ή false.

Παράδειγμα:

```
function main() {  
    var res = sc$UI.Dialogs.ToIgnoreAnyDataChanges("Αλλαγές");  
    if ( res == true ) { sc$UI.Dialogs.ShowInformation("Αποτέλεσμα", "Ο χρήστης συμφωνεί") }  
    else {sc$UI.Dialogs.ShowInformation("Αποτέλεσμα ", "Ο χρήστης δεν συμφωνεί»); }  
}
```

Αποτέλεσμα:



[AskSaveDataChanges\(caption \)](#)

Εμφανίζει προκαθορισμένη ερώτηση για αποθήκευση αλλαγών.

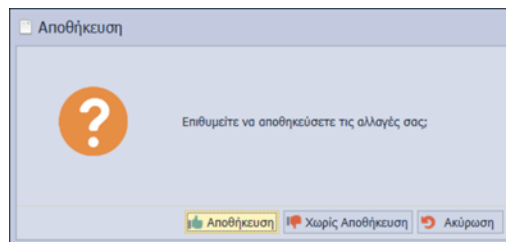
Παράμετροι: **caption** : (String). Η επικεφαλίδα του διαλόγου.

Επιστρέφει: [poDialogRes](#). Τα Buttons αντιστοιχούν σε (poDialogRes.Yes , poDialogRes.No, poDialogRes.Cancel) .

Παράδειγμα:

```
function main() {  
    var res = sc$UI.Dialogs.AskSaveDataChanges("Αποθήκευση");  
    sc$UI.Dialogs.ShowInformation("Επιλέχθηκε", res);  
}
```

Αποτέλεσμα:



DeleteObject(caption)

Εμφανίζει προκαθορισμένη ερώτηση για διαγραφή.

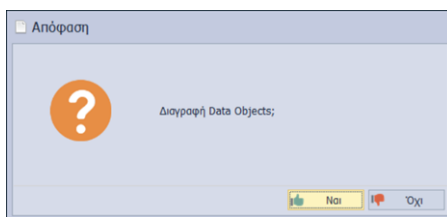
Παράμετροι: caption : (String). Η επικεφαλίδα του διαλόγου.

Επιστρέφει: true ή false.

Παράδειγμα:

```
function main() {  
    var res = sc$UI.Dialogs.DeleteObject("Απόφαση");  
    if ( res == true ) { sc$UI.Dialogs.ShowInformation("Αποτέλεσμα", "Ο χρήστης συμφωνεί" ) }  
    else { sc$UI.Dialogs.ShowInformation("Αποτέλεσμα", "Ο χρήστης δέν συμφωνεί" ); }  
}
```

Αποτέλεσμα:



ShowAsyncStorageItem (asyncHandleID)

Εμφανίζει τα αποτελέσματα **εργασίας** που έχει εκτελεστεί ασύγχρονα. Πχ. Αναφορές εφαρμογής. Οι εργασίες αυτές καταγράφονται στον πίνακα poAsyncStorage.

Παράμετροι: asyncHandleID (String GUID) poAsyncStorage. poAsyncHandle

Παράδειγμα : `sc$UI.Dialogs.ShowAsyncStorageItem(asyncHandleID);`

ShowError (caption , text)

Χρησιμοποιείται για εμφάνιση μηνύματος σφάλματος.

Παράμετροι:

- `caption` (String). Η επικεφαλίδα του μηνύματος
- `text` (String). Το μήνυμα

Παράδειγμα : `sc$UI.Dialogs.ShowError("Λάθος", "<Μήνυμα Λάθους>");`

ShowException(ex)

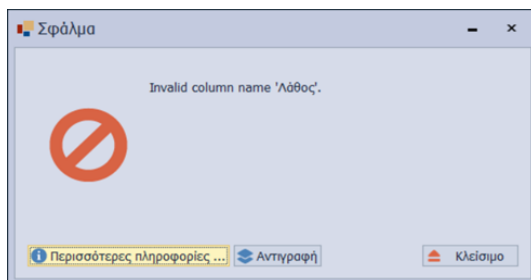
Εμφανίζει αναλυτικά μηνύματα λάθους.

Παράμετροι: `ex` (System. Exception) .

Παράδειγμα:

```
function main() {
  try {
    //Λανθασμένο Query
    var g = sc$Db.ExecuteScalar("select Λάθος from heitems", null);
  }
  catch (err) {
    var ex = err.ToException(err);
    sc$UI.Dialogs.ShowException(ex);
  }
  finally {}
}
```

Αποτέλεσμα :



ShowInformation(caption, text)

Χρησιμοποιείται για εμφάνιση πληροφοριακού μηνύματος.

Παράμετροι:

- `caption` (String) . Η επικεφαλίδα του μηνύματος
- `text` (String) . Το μήνυμα

Παράδειγμα: `sc$UI.Dialogs.ShowInformation("Επικεφαλίδα", "Ενημερωτικό μήνυμα");`

ShowActionLogger(logger)

Χρησιμοποιείται για εμφάνιση μηνυμάτων μέσω [Action Logger](#).

Παράμετροι: **logger** (scActionLogger)

Παράδειγμα:

```
function main() {
    var logger = new scActionLogger();
    // Κώδικας ....
    logger.AddInfo("info", "Η εργασία εκτελέστηκε ");
    sc$UI.Dialogs.ShowActionLogger(logger);
}
```

ShowActionLogger(logger, ex)

Εμφάνιση logger με αναλυτικά μηνύματα από Exceptions.

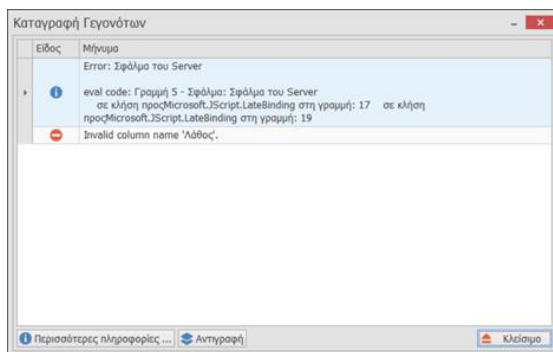
Παράμετροι:

- **logger** (scActionLogger).
- **ex** Exception στην εκτέλεση.

Παράδειγμα:

```
function main() {
    try {
        var logger = null;
        var ex = null;
        var g = sc$db.ExecuteScalar("select λάθος from heitems", null);
    }
    catch (err) {
        var ex = err.ToException(err);
        logger = scActionLogger.CreateFromException(ex);
        logger.AddInfo(err,ex.StackTrace);
    }
    finally { if ( logger != null ) sc$UI.Dialogs.ShowActionLogger(logger,ex); }
}
```

Αποτέλεσμα:



ShowOpenFileDialog(title, initialDir, dialogFilter)

Εμφανίζει διάλογο ανοίγματος αρχείου: (Open File Dialog).

Παράμετροι:

- **title** (String) . Η επικεφαλίδα του διαλόγου.

- `initialDir` (String) . Ο αρχικός κατάλογος από τον οποίο θα ξεκινήσει η αναζήτηση
- `dialogFilter` (String) . Φίλτρο

Επιστρέφει : (String) Το πλήρες όνομα του αρχείου <Διαδρομή>\<Όνομα ή null εφόσον ζητηθεί ακύρωση.

Παράδειγμα: `var importFile = sc$UI.Dialogs.ShowOpenFileDialog("Επεξεργασία", "c:\\", "Αρχεία Excel (*.xlsx)|*.xlsx");`

ShowSaveFileDialog(title, initialDir, dialogFilter)

Εμφανίζει διάλογο αποθήκευσης αρχείου: (SaveFile Dialog)

Παράμετροι:

- `title` (String) . Η επικεφαλίδα του διαλόγου.
- `initialDir` (String) . Ο αρχικός κατάλογος από τον οποίο θα ξεκινήσει η αναζήτηση
- `dialogFilter` (String) . Φίλτρο

Επιστρέφει : (String) Το πλήρες όνομα του αρχείου <Διαδρομή>\<Όνομα ή null εφόσον ζητηθεί ακύρωση.

Παράδειγμα : `var fileDlg = sc$UI.Dialogs.ShowSaveFileDialog("Αποθήκευση Δεδομένων", "c:\\", "Αρχεία Excel (*.xls)|*.xlsx") ;`

Παραδείγματα DialogFilter

```
var txtFilter = "Αρχεία κειμένου (*.txt)|*.txt|Όλα τα αρχεία (*.*)|*.*";
var txtFilter2 = "Αρχεία κειμένου (*.txt)|*.txt|ASCII|*.ASC|Όλα τα αρχεία (*.*)|*.*";
var ExcelFilter = "Αρχεία Excel |*.xls?|Αρχεία CSV|*.csv";
var ExcelFilter2 = "Αρχεία Excel |*.xls?";
var pdfFilter = "Αρχεία PDF |*.PDF";
var XMLfilter = "Αρχεία XML |*.xml";
```

Progress Methods

Μέθοδοι που χρησιμοποιούνται προκειμένου να εμφανιστεί παράθυρο προόδου εργασιών. Δίνει την δυνατότητα εμφάνισης πολλαπλών Progress Bar ανάλογα με τις εργασίες που εκτελούνται.

Methods

ComplexProgressStart(caption)

Εκκινεί διαδικασία προόδου εργασιών.

Παράμετροι: `caption` (String). Το λεκτικό του παραθύρου.

Επιστρέφει: (String GUID). Μοναδικό Αναγνωριστικό.

ComplexProgressNewBar(key, label)

Περιγραφή: Δημιουργεί μια νέα Μπάρα προόδου.

Παράμετροι:

- **key** (String GUID) . Μοναδικό Αναγνωριστικό που επιστρέφεται από [ComplexProgressStart](#)
- **label** (String) . Το λεκτικό της Μπάρας.

Επιστρέφει: (String GUID). Μοναδικό Αναγνωριστικό.

ComplexProgressBarUpdate(progressKey, barkey, pos, max, stepLabel)

Ανανεώνει τον δείκτη και το λεκτικό της μπάρας προόδου.

Παράμετροι:

- **progressKey** Μοναδικό Αναγνωριστικό που επιστρέφεται από [ComplexProgressStart](#) .
- **barkey** Μοναδικό Αναγνωριστικό που επιστρέφεται από [ComplexProgressNewBar](#) .
- **pos** Αριθμός που δηλώνει την τρέχουσα θέση του δείκτη που πρέπει να ανανεωθεί.
- **max** Αριθμός που δηλώνει την μεγαλύτερη τιμή (πλήθος) που μπορεί να δεχθεί.
- **stepLabel** (String) . Κείμενο ως νέο λεκτικό της Μπάρας.

ComplexProgressBarClose(progressKey, barkey)

Σταματά την εκτέλεση ενός Progress Bar.

Παράμετροι:

- **progressKey** Μοναδικό Αναγνωριστικό που επιστρέφεται από [ComplexProgressStart](#)
- **barkey** Μοναδικό Αναγνωριστικό που επιστρέφεται από [ComplexProgressNewBar](#)

ComplexProgressStop(progressKey)

Σταματά την εκτέλεση μιας διαδικασίας προόδου.

Παράμετροι: **progressKey** Μοναδικό Αναγνωριστικό που επιστρέφεται από [ComplexProgressStart](#)

Παράδειγμα χρήσης ProgressMethods

```
function main() {  
    //Δήλωση μεταβλητών  
    var ProgressKey = null;  
    var ComplexProgressKey = null;;  
    var progressstr = null ;  
    var TotalRecords ;  
  
    //Δημιουργία παραθύρου προόδου  
    ComplexProgressKey = sc$UI.Dialogs.ComplexProgressStart("Εργασίες");  
    //Προσθήκη Μπάρας  
    ProgressKey = sc$UI.Dialogs.ComplexProgressNewBar(ComplexProgressKey, "Πρώτη Εργασία");  
    TotalRecords = 300;  
    for ( var i = 1 ; i <= TotalRecords; i++ ) {  
        //Ενημέρωση Λεκτικού Μετρητή προόδου  
        progressstr = "Εγγραφή : " + Convert.ToString(i) + " Από: " + Convert.ToString(TotalRecords);  
        //Ενημέρωση Δείκτη Προόδου  
        sc$UI.Dialogs.ComplexProgressBarUpdate(ComplexProgressKey,ProgressKey,i,TotalRecords,progressstr);  
    }  
}
```



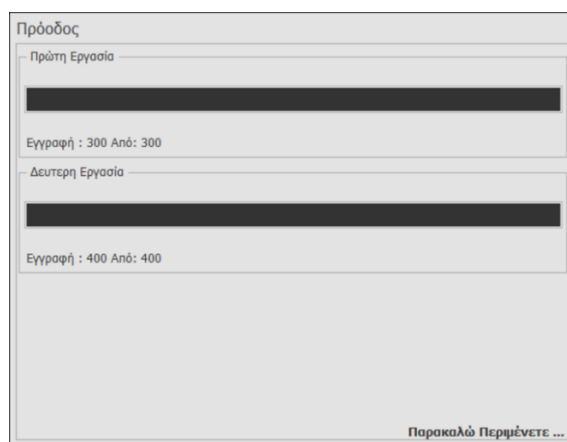
```

ProgressKey = sc$UI.Dialogs.ComplexProgressNewBar(ComplexProgressKey, "Δευτερη Εργασία");
TotalRecords = 400;
for ( var k = 1 ; k <= TotalRecords; k++ ) {
    progressstr = "Εγγραφή : " + Convert.ToString(k) + " Από: " + Convert.ToString(TotalRecords);

    sc$UI.Dialogs.ComplexProgressBarUpdate(ComplexProgressKey,ProgressKey,k,TotalRecords,progress
str);
}
//Τερματισμός παραθύρου προόδου
if ( ComplexProgressKey != undefined && ComplexProgressKey != null ) {
    sc$UI.Dialogs.ComplexProgressStop(ComplexProgressKey);
    sc$UI.Dialogs.ComplexProgressBarClose(ComplexProgressKey, ProgressKey);
}
}

```

Αποτέλεσμα:



scFormArgs (poFormArgs)

Arguments κλήσης μιας φόρμας. (Properties).

Παράδειγμα:

```

var fargs = new scFormArgs();
fargs.OpenMode = 1;
fargs.EditValue = <Τιμή>

```

Στην συνέχεια αυτά τα argument θα χρησιμοποιηθούν σε Forms (Create, FormShow)

Properties

AccessRightsInfo

Τα δικαιώματα (poAccessRightsInfo) για την συγκεκριμένη φόρμα, σε επίπεδο χρήστη.

- View
- Modify
- Delete



- Insert

Κάθε ένα από αυτά μπορεί να έχει τιμές 0 ή 1.

DesignerIsActive

Δηλώνει αν είναι active (διαθέσιμο) το Form Designer.

EditValue

Το ID της εγγραφής εφόσον πρόκειται για Master Form και το [OpenMode](#) είναι ίσο με 1 .

FantomArgs

Arguments κλήσης προκειμένου να γίνουν override **Access Rights**

FormCaption

Το Caption της φόρμας. Μπορεί να μεταβληθεί εφόσον η φόρμα έχει κληθεί με Create.

InArgs

Περιγραφή: Arguments κλήσης.

IsDetailForm

Ορίζει αν πρόκειται για κύρια ή φόρμα λεπτομερειών.

OpenMode

Enumerator που περιγράφει τον τρόπο που θα

Τιμές :

- Normal (browser)
- LocateRecord Επεξεργασία στοιχείων εγγραφής. Θα πρέπει υποχρεωτικά να έχει δοθεί τιμή στο Property [EditValue](#) . Σε περίπτωση που το EditValue είναι null
- NewRecord Νέα Εγγραφή (default για Quick Forms)
- Zoom
- NewRecordFromZoom
- Preview

OutArgs

Arguments Επιστροφής.

UnPinned

Καθορίζει αν η φόρμα εμφανίζεται σε Float μορφή και όχι με την κλασική Pinned (εμφάνιση μέσα σε Tab)

ZoomCollection

Αναφέρεται σε συνδεδεμένο collection. Παράδειγμα χρήσης μπορείτε να βρείτε σε Script εισαγωγής δεδομένων . Εισπρακτέα Αξιόγραφα, πληρωτέα Αξιόγραφα.



Methods

InArg(name)

Επιστρέφει τιμή από συγκεκριμένο Argument κλήσης.

Παράμετροι: `name` (String). Το όνομα του Argument.

Επιστρέφει: `<T>`

OutArg(name)

Επιστρέφει τιμή από συγκεκριμένο Argument κλήσης.

Παράμετροι: `name` (String). Το όνομα του Argument.

Επιστρέφει: `<T>`

scForm

Η φόρμα σαν αντικείμενο.

Properties

Args

Ορίσματα κλήσης της φόρμας [scFormArgs](#)

IsDirty

Δηλώνει το αν η φόρμα είναι σε κατάσταση Edit. Τιμές true ή false.

Παράδειγμα : `<frm>.IsDirty = true;` ή έλεγχος με `if (frm.IsDirty == true) { Κώδικας .. }`

RealType

Επιστρέφει το Actual type της φόρμας.

Παράδειγμα : `var Type = <frm>.RealType;`

Methods

ExecuteFormAction(actionName)

Εκτελεί FormAction. (Εργασία από το menu **Ενέργειες**) εφόσον το συγκεκριμένο action είναι ενεργό για εκτέλεση από τον χρήστη.

Παράμετροι: `actionName` (String) . Το όνομα του Action που θέλετε να εκτελεστεί:

Παράδειγμα: [GetProperty\(\)](#)

Σημείωση: Τα διαθέσιμα Actions μπορείτε να τα δείτε μέσω του Action Center.



Controls & Data Security			
↑ Επέκταση Όλων ↓ Σύμπτυξη Όλων			
Type Hercules.Win.FIN.Traders.heTradersF			
Structure			
	Name	Caption	Type
☐	JoinContacts	Ενέργειες;Σύνδεση με Επαφή	poFormMenuItem

GetControlDisplayValue(controlName)

Επιστρέφει την εμφανιζόμενη περιγραφή ενός LookupEdit, LookupTreeEdit, LookupMaster control. Παράδειγμα ο Κωδικός ή Επωνυμία Πελάτη σε ένα παραστατικό πώλησης. Η τιμή του Control είναι πάντα το **heID** του πελάτη ενώ η τιμή εμφάνισης είναι ο κωδικός ή η επωνυμία.

Παράμετροι: controlName (String). Το όνομα του Control.

Παράδειγμα: `var ctrlDVal = <frm>.GetControlDisplayValue("<controlName>");`

Το Control Name για το εκάστοτε control μπορείτε να το δείτε ενεργοποιώντας το και πατώντας ALT_CTRL_SHIFT_I

GetControlEditValue(controlName)

Επιστρέφει την τιμή (Edit Value) ενός control.

Παράμετροι: controlName (String)

Παράδειγμα: `var ctrlEVal = <frm>.GetControlEditValue("<controlName>");`

Το Control Name για το εκάστοτε control μπορείτε να το δείτε πατώντας ALT_CTRL_SHIFT_I

GetCurrentObject(controlName)

Επιστρέφει το αντικείμενο που είναι συνδεδεμένο με το control.

Παράμετροι: controlName (String)

Παράδειγμα: `var cob = <frm>.GetCurrentObject("<controlName>");`

Το Control Name για το εκάστοτε control μπορείτε να το δείτε πατώντας ALT_CTRL_SHIFT_I

GetGridColumnDisplayValue(gridName, columnName)

Επιστρέφει την τιμή εμφάνισης (DisplayValue) ενός κελιού grid. Παράδειγμα είναι η στήλη κωδικός είδους σε παραστατικά όπου η τιμή που δέχεται το πεδίο είναι το **heID** του είδους ενώ στον χρήστη στην στήλη του grid εμφανίζεται ο κωδικός.

Παράμετροι:

- gridName (String)
- columnName (String)

Παράδειγμα: `var ColDispV = <frm>.GetGridColumnDisplayValue("<GridName>","<columnName>");`

Τα Grid Name και columnName μπορείτε να τα δείτε πατώντας ALT_CTRL_SHIFT_I επάνω σε κάποιο grid.

GetGridColumnValue(gridName, columnName)

Επιστρέφει την τιμή (Value) ενός κελιού grid. Αφορά την τιμή για αυτή την στήλη. Παράδειγμα είναι η στήλη κωδικός είδους όπου η τιμή που δέχεται το πεδίο είναι το **heID** του είδους ενώ στον χρήστη στην στήλη του grid εμφανίζεται ο κωδικός. Πάντα για την επιλεγμένη γραμμή του Grid.

Παράμετροι:

- **gridName** (String). Το όνομα του grid.
- **columnName** (String). Το όνομα της στήλης.

Παράδειγμα: `var colV = <frm>.GetGridColumnValue("<GridName>","<controlName>");`

Τα Grid Name και columnName μπορείτε να τα δείτε πατώντας ALT_CTRL_SHIFT_I επάνω σε κάποιο grid.

GetProperty(name)

Επιστρέφει τιμή για συγκεκριμένο property (ιδιότητα) της φόρμας.

Παράμετροι: **name** (string). Το όνομα της παραμέτρου.

Παράδειγμα: Στο συγκεκριμένο παράδειγμα καλείται η φόρμα με OpenMode =1 και χωρίς Edit Value προκειμένου να δημιουργήσουμε νέα Εγγραφή. Στην συνέχεια καλείται η GetProperty για να μας δώσει τον presenter που συνδέεται με την φόρμα. Μέσω του presenter αποκτούμε πρόσβαση στο DataCollection της φόρμας.

```
function main() {  
    var args = new scFormArgs;  
    args.OpenMode=1;  
    var frm = sc$UI.Forms.Create("Hercules","Traders", args)  
    var pres = frm.GetProperty("Presenter");  
    var col = pres.Collection;  
    var trader = col[0];  
    frm.Show();  
    frm.ExecuteFormAction("JoinContacts");  
    frm.Dispose();  
}
```

Properties για κλήση μέσω της GetProperty().

- AllowCustomization
- Presenter
- BindingSources
- Caption
- Layout
- Height

- Width
- FormMenu
- UniqueName

GetTreeColumnDisplayValue(treeName, columnName)

Επιστρέφει την τιμή (Display Value) που επιστρέφεται από το αντίστοιχο Control.

Παράμετροι:

- **treeName** (String). Το όνομα του δέντρου.
- **columnName** (String). Το όνομα της στήλης.

Παράδειγμα: *<frm>*. GetTreeColumnDisplayValue ("<GridName> ", "<controlName> ");

Τα tree Name και Column Name μπορείτε να τα δείτε πατώντας ALT_CTRL_SHIFT_I

GetTreeColumnValue(treeName, columnName)

Επιστρέφει την πραγματική τιμή που επιστρέφεται από το αντίστοιχο Control.

Παράμετροι:

- **treeName** (String). Όνομα Δέντρου
- **columnName** (String). Όνομα στήλης

Παράδειγμα: *<frm>*. GetTreeColumnDisplayValue ("<GridName> ", "<controlName> ");

Τα tree Name και Column Name μπορείτε να τα δείτε πατώντας ALT_CTRL_SHIFT_I

InvokeMethod(name, args)

Χρησιμοποιείται για την κλήση μεθόδων της φόρμας.

Παράμετροι:

- **name** (String)
- **args** (Object[])

Επιστρέφει: Obj

InvokeMethodAsync(name, args)

Λειτουργεί όπως και η InvokeMethod αλλά η εκτέλεση γίνεται ασύγχρονα.

SetControlEditValue(controlName, value)

Αναθέτει τιμή σε ένα Control.

Παράμετροι:

- **controlName** (String). Το όνομα του Control
- **value** Η τιμή που θα ανατεθεί στο Control.

Σύνταξη: *frm*. SetControlEditValue ("<controlName> ", "<value> ");

Το Control Name για το εκάστοτε control μπορείτε να το δείτε πατώντας ALT_CTRL_SHIFT_I





SetGridColumnValue(gridName, columnName, value)

Αναθέτει τιμή σε ένα κελί GRID. (**Προσοχή**. Για την γραμμή που είναι επιλεγμένη).

Παράμετροι:

- **gridName** (String) Το όνομα του Grid
- **columnName** (String) Το όνομα της στήλης
- **value** Η τιμή που θα δοθεί στο κελί.

Παράδειγμα: `<frm>.SetGridColumnValue("<GridName>","<controlName>","<value>");`

Τα Control Name και Grid Name μπορείτε να τα δείτε πατώντας ALT_CTRL_SHIFT_I

SetProperty(name)

Αναθέτει τιμή για συγκεκριμένο property (ιδιότητα) της φόρμας.

Παράμετροι: **name** (String)

SetTreeColumnValue(treeName, columnName, value)

Περιγραφή: Αναθέτει τιμή σε στήλη μιας Δενδροειδούς λίστας.

Παράμετροι:

- **treeName** (String)
- **columnName** (String)

value

Show()

Εμφανίζει την φόρμα. Χρησιμοποιείται μετά την Create για να δείξει την φόρμα στον χρήστη.

Παράδειγμα: `<frm>.Show();`

ShowDialog()

Εμφανίζει την φόρμα ως παράθυρο διαλόγου (Modal). Χρησιμοποιείται μετά την Create για να δείξει την φόρμα στον χρήστη. Ο χρήστης δεν μπορεί να μετακινηθεί σε άλλο παράθυρο της εφαρμογής έως ότου κλείσει η φόρμα.

Παράδειγμα: `<frm>.ShowDialog();`



sc\$UI.Forms

Create(domain, name, args)

Δημιουργεί μια νέα φόρμα ([scForm](#) object) . Η φόρμα δεν θα είναι ορατή έως κληθεί η αντίστοιχη [Show](#) ή [ShowDialog](#)

Παράμετροι:

- **domain** (String) Το Domain της φόρμας. Κατά κανόνα Hercules
- **name** (String) Το όνομα της φόρμας.
- **args** ([scFormArgs](#))

Επιστρέφει: [scForm](#).

Παράδειγμα:

//Νέα Εγγραφή

```
function main() {  
    var args = new scFormArgs;  
    args.OpenMode=1;  
    var frm = sc$UI.Forms.Create("Hercules","Traders", args)  
    frm.Show();  
    frm.Dispose();  
}
```

//Ανοιγμα φόρμας για επεξεργασία συγκεκριμένης εγγραφής

```
function main() {  
    var args = new scFormArgs;  
    args.OpenMode=1;  
    args.EditValue = "da7058bf-d07c-eb11-bcea-588a5a424a82"; //heID Συναλλασόμενου  
    var frm = sc$UI.Forms.Create("Hercules","Traders", args)  
    frm.Show();  
    frm.Dispose();  
}
```

GetFormType()

Επιστρέφει τον τύπο μιας φόρμας.

Παράδειγμα: `<frm>.GetFormType();`

sc\$UI.Actions

Methods

FormShow(domain, name, args)

Δίνει την δυνατότητα εμφάνισης Action Form της εφαρμογής. Η φόρμα γίνεται Create με βάση τα arguments (args) και στην συνέχεια εμφανίζεται. Χρησιμοποιείται για Action Forms που δείχνουν δεδομένα όπως οι σελίδες οικονομικών στοιχείων , και καλούνται μέσα από κάποια οντότητα πχ παραστατικά, είδη, συναλλασσόμενοι κλπ.

Παράμετροι:

- **domain** (String) Το domain της φόρμας. Hercules ή Poseidon.
- **name** (String) . Το όνομα της φόρμας.
- **Args** ([scFormArgs](#))

Παράδειγμα:

```
function main() {  
    var args = new scFormArgs();  
    args.EditValue = "2625b9b7-df6c-e911-83fe-408d5ca6c094"; //heID είδους  
    args.OpenMode = 1;  
    sc$UI.Actions.FormShow("Hercules", "Items.Item.FinancialData", args);  
}
```

FormShowDialog(domain, name, args)

Εμφανίζει ένα Action Form ως Modal. Αυτό σημαίνει ότι έως ότου κλείσει η φόρμα ο χρήστης δεν θα έχει δυνατότητα να μετακινηθεί σε άλλη ενότητα της εφαρμογής.

Παράμετροι:

- **Domain** (String) . Το Domain της φόρμας. Hercules ή Poseidon.
- **Name** (String). Το όνομα της φόρμας
- **args** ([scFormArgs](#))

FormGetType()

Επιστρέφει τον τύπο της φόρμας.

Παράδειγμα: `<frm>.GetFormType();`

FormCreate(domain, name, args)

Εκκινεί την διαδικασία δημιουργίας μιας φόρμας με βάση τα ορίσματα (παράμετροι) που δέχεται. Θα πρέπει στην συνέχεια για να εμφανιστεί η φόρμα στον χρήστη, να γίνει κλήση στην [Show](#)

Παράμετροι:

- **domain** (String). Το Domain της φόρμας. Hercules ή Poseidon.
- **Name** (String). Το όνομα της φόρμας
- **Args** ([scFormArgs](#))

Παράδειγμα:

```
function main() {  
    var args = new scFormArgs();  
    args.EditValue = "4aa7bdbe-ef35-e911-bc53-5cea1d9a70e3"; //helD είδους  
    args.OpenMode = 1;  
    var frm = sc$UI.Actions.FormCreate ("Hercules", "Items.Item.FinancialData", args);  
    frm.Show();  
}
```

Επιστρέφει : [scForm](#)

Invoke(domain, name, args)

Χρησιμοποιείται για την εμφάνιση φόρμας που καλείται με κλήση μέσω delegate. Αυτός ο τρόπος κλήσης χρησιμοποιείται για την εμφάνιση Action Forms που ενεργοποιούνται αφού ο χρήστης κάνει κάποιες επιλογές. Πχ. Φόρμες οικονομικών στοιχείων όταν καλούνται από το menu και όχι από ενέργεια σε συγκεκριμένη οντότητα, όπου εμφανίζεται η φόρμα στοιχείων αλλά ταυτόχρονα και μια αναδυόμενη φόρμα για την επιλογή οντότητας.

Παράμετροι:

- **Domain** (String) . Το Domain της φόρμας. Hercules ή Poseidon.
- **Name** (String) . Το όνομα της φόρμας.
- **Args** ([scFormArgs](#))

Επιστρέφει: null ή Object.

Παράδειγμα:

```
function main() {  
    var args = new scFormArgs();  
    var invObj = sc$UI.Actions.Invoke("Hercules", "Items.Item.FinancialData", args);  
}
```

IsEnabled(domain, name)

Ελέγχει αν η φόρμα είναι ενεργοποιημένη. (Διαθέσιμη στον χρήστη).

Παράμετροι:

- **domain** (String) . Το Domain της φόρμας. Hercules ή Poseidon.
- **name** (String) . Το όνομα της φόρμας.

sc\$UI.Elements

Χρησιμοποιείται για την διαχείριση Parameters Form ή Results Grid Form. Πρόκειται για Custom Φόρμες που δημιουργούνται μέσα στο περιβάλλον εργασίας του Script. Πληροφορίες και παραδείγματα μπορείτε να βρείτε μέσω του συνδέσμου [Περιβάλλον Εργασίας Συγγραφής Κώδικα](#)

Properties

scFormArgsResult

Το αποτέλεσμα εκτέλεσης ενός Element Dialog Form.

Παράμετροι Επιστροφής:

- **args** Οι τιμές
- **OK** true ή false ανάλογα με το button που πατήθηκε από τον χρήστη.

Methods

FormShowDialog(name, formArgs, inArgs)

Χρησιμοποιείται για την εμφάνιση ενός Parameter Form προκειμένου ο χρήστης να εισάγει ή να επεξεργαστεί στοιχεία, που θα χρησιμοποιηθούν στην συνέχεια για τις ανάγκες του Script.

Παράμετροι:

- **name** Το όνομα που δόθηκε κατά την παραμετροποίηση της φόρμας.
- **formArgs** Τα ορίσματα της φόρμας
- **inArgs** Προκαθορισμένες τιμές

Επιστρέφει : ([scFormArgsResult](#))

Αναλυτικό παράδειγμα μέσω του συνδέσμου [Περιβάλλον Εργασίας Συγγραφής Κώδικα](#)

ResultsShowDialog(name, formArgs, ds)

Χρησιμοποιείται για την εμφάνιση ενός Result Grid Form προκειμένου να εμφανιστούν δεδομένα στον χρήστη.

Παράμετροι:

- **name** (String) . Το όνομα που δόθηκε κατά την παραμετροποίηση της φόρμας.



- **formArgs** Τα ορίσματα της φόρμας.
- **ds** To Dataset

Αναλυτικό παράδειγμα μέσω του συνδέσμου [Περιβάλλον Εργασίας Συγγραφής Κώδικα](#)

ResultsShowDialogWithResult(name, formArgs, ds)

Χρησιμοποιείται για την εμφάνιση ενός Result Grid Form προκειμένου να εμφανιστούν δεδομένα στον χρήστη με δυνατότητα επεξεργασίας και στην συνέχεια αυτά τα δεδομένα να χρησιμοποιηθούν για τις ανάγκες του Script. Η λειτουργία και ο τρόπος

Παράμετροι:

- **name** (String) . Το όνομα που δόθηκε κατά την παραμετροποίηση της φόρμας.
- **formArgs** Τα ορίσματα της φόρμας.
- **ds** To Dataset

Επιστρέφει: [scFormArgsResult](#)

sc\$UI.Reports

Methods

AddReportFilter(args, filtername, filterValue)

Δίνει την δυνατότητα να επεξεργαστείτε τις τιμές των φίλτρων που εμφανίζονται σε αναφορές της εφαρμογής, ERG, Σχεδιαζόμενες, προκειμένου στην συνέχεια να εκτελεστούν οι αναφορές αυτές με προ συμπληρωμένα τα φίλτρα αυτά.

Παράμετροι:

- **args** ([scFormArgs](#))
- **filterName** (String). Το όνομα του πεδίου φίλτρου της αναφοράς. Για να βρείτε το όνομα του εκάστοτε πεδίου φίλτρου ανοίξτε την αναφορά και στην φόρμα των φίλτρων επιλέξτε το πεδίο που σας ενδιαφέρει. Με τον συνδυασμό ALT_CTRL_SHIFT_I θα εμφανιστεί οθόνη που σας δίνει πληροφορίες σχετικά με το πεδίο και την φόρμα που το περιέχει. Αντιγράψτε την τιμή από το ControlName
- **filterValue** . Η τιμή του φίλτρου

Παράδειγμα:

```
var args = new scFormArgs();
sc$UI.Reports.AddReportFilter(args, "Fsyrr_1", sc$AppSessionInfo.CurrentFiscalYearID);
sc$UI.Reports.AddReportFilter(args, "tblItemTransCmbrheID_1", sc$AppSessionInfo.CurrentCompBranchID);
```

SetReportTextFilterWhereType(args, filterName, whereType)

Αναθέτει σε ένα λεκτικό φίλτρο τον τρόπο που θα εφαρμοστεί στο query.

Παράμετροι:

- **args** ([scFormArgs](#))
- **filterName** (String). Το όνομα του πεδίου φίλτρου της αναφοράς
- **whereType** (int16)
 0. Περιέχει
 1. Ξεκινάει Με
 2. Τελειώνει Με
 3. Ισούται

ShowReport(domain, name, viewName, args)

Εμφανίζει προεπιλεγμένη αναφορά έτοιμη για εκτέλεση.

Παράμετροι:

- **domain**
- **name** (String). Το όνομα της αναφοράς.
- **viewName** (String). null ή "default" για την προεπιλογή ή όνομα προβολής
- **args** ([scFormArgs](#))

Για να βρείτε τα ονόματα των παραμέτρων που χρειάζεστε, ανοίξτε την αναφορά και στην φόρμα των φίλτρων, με τον συνδυασμό ALT_CTRL_SHIFT_I θα εμφανιστούν, το domain και το name στο τμήμα Report Action.

Παράδειγμα: Εμφάνιση αναφοράς (Αποθήκη→Αναφορές→Εκτυπώσεις→Είδη/Υπηρεσίες/Πάγια→Λίστα→Απογραφή→Απογραφή)

```
function main() {  
    var args = new scFormArgs();  
    sc$UI.Reports.AddReportFilter(args, "Fsyrr_1", sc$AppSessionInfo.CurrentFiscalYearID);  
    sc$UI.Reports.AddReportFilter(args, "tblItemTransCmbrheID_1", sc$AppSessionInfo.CurrentCompBranchID);  
    sc$UI.Reports.ShowReport("Hercules", "Viewer.Items.Inventory.Report", "Συνοπτικό", args);  
}
```

ExecuteReport(domain, name, viewName, args)

Εκτελεί προεπιλεγμένη αναφορά και παρουσιάζει το αποτέλεσμα. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowReport](#)

ShowCustomReport(reportID, viewName, args)

Περιγραφή: Δίνει την δυνατότητα εμφάνισης σχεδιαζόμενης αναφοράς.

Παράμετροι:

- **reportID** (String GUID) Το ID της αναφοράς
- **viewName** (String) . null ή "default" για την προεπιλογή ή όνομα προβολής εφόσον υπάρχει κάποια σχεδιασμένη.
- **args** ([scFormArgs](#))

Παράδειγμα:

```
function main() {  
    var args = new scFormArgs();  
    sc$UI.Reports.AddReportFilter(args, "ItemREportItemREportheCode_1", "ΕΙΔΗ-00000043");  
    sc$UI.Reports.AddReportFilter(args, "flt@$UserParameter1$_1",  
    sc$AppSessionInfo.FiscalYear.GetFiscalYearStartDate(sc$AppSessionInfo.CurrentFiscalYearID));  
    sc$UI.Reports.AddReportFilter(args, "flt@$UserParameter2$_1",sc$AppSessionInfo.WorkDate);  
    sc$UI.Reports.ShowCustomReport("c961c3bb-075f-eb11-bce2-588a5a424a82","default",args);  
}
```

Θα πρέπει στο δικό σας παράδειγμα να αντικαταστήσετε την τιμή «c961c3bb-075f-eb11-bce2-588a5a424a82» με helD από υπαρκτό Custom Report.

[ShowCustomGrid](#)(reportID, viewName, args)

Εμφανίζει σχεδιαζόμενη αναφορά με προεπιλεγμένη την εμφάνιση σε GRID προκειμένου να προστεθεί ότι άλλο χρειάζεται για την εκτέλεση από τον χρήστη. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowCustomReport](#)

[ExecuteCustomGrid](#)(reportID, viewName, args)

Εκτελεί σχεδιαζόμενη αναφορά και παρουσιάζει το αποτέλεσμα σε μορφή Grid . Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowCustomReport](#)

[ShowCustomPivot](#)(reportID, viewName, args)

Εμφανίζει σχεδιαζόμενη αναφορά με προεπιλεγμένη την εμφάνιση σε PIVOT προκειμένου να προστεθεί ότι άλλο χρειάζεται για την εκτέλεση από τον χρήστη. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowCustomReport](#)

[ExecuteCustomPivot](#)(reportID, viewName, args)

Εκτελεί σχεδιαζόμενη αναφορά και παρουσιάζει το αποτέλεσμα σε μορφή Pivot . Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowCustomReport](#)



ShowCustomDataExport(reportID, viewName, args)

Εμφανίζει σχεδιαζόμενη αναφορά για εξαγωγή δεδομένων προκειμένου να προστεθεί ότι άλλο χρειάζεται για την εκτέλεση από τον χρήστη. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowCustomReport](#)

ExecuteCustomDataExport(reportID, viewName, args)

Εκτελεί σχεδιαζόμενη αναφορά για εξαγωγή δεδομένων. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowCustomReport](#).

ShowERGReport(reportID, viewName, args)

Εμφανίζει ERG αναφορά προκειμένου να προστεθεί ότι άλλο χρειάζεται για την εκτέλεση από τον χρήστη.

Παράμετροι:

- reportID Το ID της αναφοράς
viewName null ή "default" για την προεπιλογή ή όνομα προβολής εφόσον υπάρχει κάποια σχεδιασμένη.
- args [scFormArgs](#)

Παράδειγμα:

```
function main() {  
  
    var args = new scFormArgs();  
  
    sc$UI.Reports.AddReportFilter(args, "ItemsItemsheCode_1", "ΕΙΔΗ-00000043");  
  
    sc$UI.Reports.AddReportFilter(args, "flt@$UserParameter1$_1",  
    sc$AppSessionInfo.FiscalYear.GetFiscalYearStartDate(sc$AppSessionInfo.CurrentFiscalYearID));  
  
    sc$UI.Reports.AddReportFilter(args, "flt@$UserParameter2$_1", sc$AppSessionInfo.WorkDate);  
  
    sc$UI.Reports.ShowERGReport("3384155c-295e-eb11-bce1-588a5a424a82", "default", args);  
  
}
```

ExecuteERGReport(reportID, viewName, args)

Εκτελεί ERG αναφορά. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowERGReport](#)

ShowERGGrid(reportID, viewName, args)

Εμφανίζει ERG αναφορά με προεπιλεγμένη εξαγωγή σε μορφή GRID προκειμένου να προστεθεί ότι άλλο χρειάζεται για την εκτέλεση από τον χρήστη.. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowERGReport](#)



ExecuteERGGrid(reportID, viewName, args)

Εκτελεί ERG αναφορά σε μορφή Grid. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowERGReport](#)

ShowERGPivot(reportID, viewName, args)

Εμφανίζει ERG αναφορά με προεπιλεγμένη εξαγωγή σε μορφή PIVOT προκειμένου να προστεθεί ότι άλλο χρειάζεται για την εκτέλεση από τον χρήστη. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowERGReport](#)

ExecuteERGPivot(reportID, viewName, args)

Εκτελεί ERG αναφορά σε μορφή Grid. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowERGReport](#)

ShowERGDDataExport(reportID, viewName, args)

Εμφανίζει ERG αναφορά για εξαγωγή δεδομένων προκειμένου να προστεθεί ότι άλλο χρειάζεται για την εκτέλεση από τον χρήστη. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowERGReport](#)

ExecuteERGDDataExport(reportID, viewName, args)

Εκτελεί ERG αναφορά για εξαγωγή δεδομένων. Ο τρόπος κλήσης και οι παράμετροι είναι ίδιοι με την [ShowERGReport](#)

Βοηθήματα

[Περιβάλλον Εργασίας Συγγραφής Κώδικα](#)

[Scripting Χρήση Sc\\$AppSessionInfo](#)

[Microsoft.JScript Namespace](#)

[Action Logger](#)

[MS-Office-JScript](#)

