F*uncții pentru interfațarea altor aplicații cu* WinMENTOR

DocImpServer este un modul destinat programatorilor care doresc să interfațeze diferite aplicații cu programul **WinMENTOR**.

Ca și componentă soft, DocImpServer este un server **DCOM** ce conține funcții atât pentru consultarea de date din baza de date **WinMENTOR,** cât și pentru actualizarea unor nomenclatoare sau emiterea unor documente.

Deoarece structura bazei de date **WinMENTOR** este destul de complexă, DocImpServer permite programatorilor să dezvolte aplicații care execută constultări și actualizări de date fără a fi necesar ca ei să cunoască structurile de date **WinMENTOR** și relațiile dintre tabele.

Transferul de date dintre aplicația client și **DocImpServer** se face prin blocuri de date de tip *Olevariant* care în esență sunt array-uri de string-uri ce conțin informații concatenate cu ";".

În funcție de specificul dezvoltării pe care doresc să o facă, programatorii primesc ca suport un proiect ce conține exemple de conectare la **DocImpServer** și de lansare a unor funcții de bază.

De asemenea, li se indică funcțiile pe care pot să le folosească pentru a ajunge la rezultatele dorite cât și structura pe care trebuie să trimită sau să primească datele.

Prin cele peste 100 de funcții implementate **DocImpServer** permite :

- Citirea nomenclatorului de Articole;
- Citirea nomenclatorului de Parteneri;
- Citirea nomenclatorului de Gestiuni;
- Citirea nomenclatorului de Personal;
- Consultarea stocurilor;
- Consultarea soldurilor de parteneri;
- Consultarea nomenclatoarelor de categorii de prețuri și prețuri pe articole/clienți;
- Actualizarea nomenclatorului de parteneri;
- Actualizarea nomenclatorului de articole;
- Generearea de comenzi de la clienți;
- Generarea de facturi și avize de expediție către clienți;
- Generearea de facturi de la funizori interni sau externi (DVI-uri);
- Generarea de note de predare în magazie;
- Generarea de bonuri de consum;
- Generarea de transferuri între gestiuni;
- Generarea de monetare cu vânzări în magazin.

I. DESCRIEREA FUNCȚIILOR

Pentru a înțelege explicațiile de mai jos ar trebui să fiți, deja familiarizați cu modul de lucru al aplicației **WinMENTOR**.

Este de peferat ca Server-ul să fie copiat în directorul în care se află *Mentor.exe,* pentru ca sesiunea BDE să poată folosi subdirectorul *Private* existent acolo.

Serverul COM are implementate următoarele funcții :

1. GetListaFirme: OleVariant;

În **WinMENTOR** utilizatorul poate gestiona date pentru un număr nedefint de firme, la intrarea în sesiune el urmând să selecteze firma de lucru.

Această funcție returnează o listă cu denumirile prescurtate ale firmelor ce se găsesc în directorul cu baze de date **WinMENTOR**.

2. GetListaLuni(const NumeFirma: WideString): OleVariant;

Datele în **WinMENTOR** sunt gestionate la nivel de luni de lucru și de aceea utilizatorul trebuie să precizeze la intrarea în sesiune luna de lucru.

Această funcție returnează o listă cu lunile existente pe disc pentru firma care are denumirea prescurtată precizată în parametrul *NumeFirma*. Denumirile de luni de lucru sunt în formatul „aaaa_ll” (așa cum sunt pe disc denumirile subdirectoarelor ce conțin informațiile lunilor de lucru).

3. SetNumeFirma(const NumeFirma: WideString): Integer;

Setează numele firmei de lucru (parametrul **NumeFirma**); Returnează **1**, dacă funcția a reușit sau **0**, dacă nu. Pentru a citi eventualul mesaj de eroare trebuie apelată funcția *GetListaErori* descrisă mai jos.

Observație: În **WinMENTOR,** baza de date este structurată pe directoare, fiecare firmă având alocat un director al cărui nume este dat de numele precurtat al firmei.

Pentru fiecare lună contabilă de lucru se crează câte un subdirector cu denumirea AAAA_LL;

Exemplu: datele firmei EXEMPLU se află în directorul

c:\winment\data\exemplu.

Datele aferente fiecărei luni contabile se află în subdirectore de genul

c:\winment\data\exemplu\2022_04

4. **SetLunaLucru(An, Luna: Integer): Integer;**

Setează luna de lucru (parametri **An**, **luna)**;

Returnează **1** dacă funcția a reușit sau **0** dacă nu. Mesajul de eroare se preia cu funcția *GetListaErori*.

5. SetDocsData(DocData: OleVariant);

Transmite într-o listă de stringuri (parametrul **DocData**) conținutul unui pachet de date care conține documente ce urmează a fi importate în **WinMENTOR**.

Iată un exemplu scris în Delphi privind utilizarea acestei funcții:

VA := VarArrayCreate([0, Lista.Items.Count -1], varOleStr);

for i := 0 to Lista.Items.Count - 1 do VA[i] := Lista.Items[i];

WinMDocImp.SetDocsData(VA);

Lista este o înșiruire de stringuri în care am încărcat linie cu linie un fișier text ce se dorește transmis.

WinMDocImp este o instanță a **DocImpServer**.

6. DateValide: Integer;

Funcția validează conținutul unui pachet de facturi de ieșire trimis prin funcția *SetDocsData.*

Returnează , dacă datele sunt valide sau **0**, dacă nu. Mesajele de eroare se preiau cu funcția *GetListaErori*;

7. GetListaErori: OleVariant;

Transmite o listă de stringuri ce conțin mesaje de eroare returnate de server în urma execuției unor funcții.

8. ImportaFacturi: Integer;

Importă facturile din datele transmise cu *SetDocsData*. Returnează numărul total de facturi importate.

Este utilă apelarea funcției *GetListaErori* deoarece, chiar dacă s-au importat facturi, este posibil ca unele din ele să nu se fi putut modifica sau șterge, deoarece erau editate de alți utilizatori.

Lista returnată de *GetListaErori* poate conține astfel de informații.

Iată un exemplu de utilizare a funcțiilor descrise mai sus (Considerăm *DocImp* o instanță IDocImpObject pe care o veți crea la lansarea programului client):

```
dacă (DocImp.SetNumeFirma('TEST') == 0)
atunci
 {
  ListaErori= DocImp.GetListaErori;
  Afișează listă erori
  return;
 }

dacă (DocImp.SetLunaLucru(2003, 4) == 0)
atunci
 {
  ListaErori= DocImp.GetListaErori;
  Afișează listă erori
  return;
 }

dacă (DocImp.Datevalide == 0) atunci
 {
  ListaErori= DocImp.GetListaErori;
  Afișează listă erori
 }
altfel
  { nr=DocImp.ImportaFacturi; // nr. va conține numărul de facturi preluate
    ListaErori= DocImp.GetListaErori; // citesc eventuale erori sau mesaje de avertizare
  }
```

9. GetListaParteneri(out Error: Integer): OleVariant;

Returnează o listă de stringuri cu informații despre Parteneri. Fiecare linie conține informații despre un partener în formatul:

1. *ID Partener;*
2. *Denumire;*
3. *CodFiscal;*
4. *Localitate;*
5. *Adresa;*
6. *Telefon;*
7. *Persona de contact;*
8. *Simbol Clasa de caracterizare; //* din tabelul NCLASEP.DB
9. *Clasa de caracterizare;//*din tabelul NCLASEP.DB
10. *Simbol categorie pret; //* din tabelul NCATPRET.DB
11. *Denumire categorie pret; //* din tabelul NCATPRET.DB
12. *Marca Agent;* // Coloana MARCA din tabelul NPERS.DB

13. Nume Agent; // Coloana NUME din tabelul NPERS.DB

14. Prenume Agent; // Coloana PRENUME din tabelul NPERS.DB

15. Scadenta;

16. Discount;

17. Denumire Criteriu de discount asociat;

18. Denumiri sedii partener; // separate prin „~” dacă sunt mai multe;

19. Cod Extern;// CODEXTERN din tabelul NPART.DB

20. Partner blocat ;// Flagul poate fi DA sau NU

21. Credit la vanzare;

22. Cod fiscal;

23. Cont banca;

24. Localitati sedii; // separate prin „~” dacă sunt mai multe, corespunzător ordinii de la câmpul 18

25.Tara;

26. Marcile agentilor asignati la sedii; // separați prin „~” dacă sunt mai mulți, corespunzător ordinii de la câmpul 18

27. Observatii;

28. Flag (D) care indica daca sediile enumerate au statut de sediu social // separate cu „~” , corespunzător ordinii de la câmpul 18

29. Coduri postale Sedii; // separate cu „~”, corespunzător ordinii de la câmpul 18

30. Adrese de email sedii // separate cu „~”, corespunzător ordinii de la câmpul 18

31. Numere de telefon persoane de contact;

32. Flag care indica daca partenerul este persoana fizica sau juridica(PJ sau PF)

33. Moneda implicita partener;

34. Data adaugarii in baza de date;

35.Trasee asignate partenerului;

36. Puncte acumulate;

37. Coduri fiscale sedii // separate cu „~”, corespunzător ordinii de la câmpul 18

38. Informatii privind tipul sediului // separate cu „~”, corespunzător ordinii de la câmpul 18.

Dacă, din diverse motive, lista nu a putut fi transmisă, parametrul *Error* ia valoarea **1**, iar mesajele de eroare se preiau cu funcția *GetListaErori*.

10. GetStocArticole(out Error: Integer): OleVariant;

Returnează o listă de stringuri cu informații despre articolele existente în stoc. Fiecare linie conține informații despre un partener în formatul:
„CodExtern;Denumire;UM;PretVanzare;Stoc;SimbolClasa;DenClasa;IDProducator;Den.Pro

ducator;IDFurnizor;DenFurnizor;SimbolGestiune;DenGestiune;CotaTVA;Flag;PretCuTVA;StocRezervat de comezi; ”

Flag // precizează dacă TVA-ul este inclus în prețul de vânzare (D/N);

Dacă, din diverse motive lista nu a putut fi transmisă, parametrul Error ia valoarea **1**, iar mesajele de eroare se preiau cu funcția *GetListaErori*.

11. GetStocArticol(const ArticolID: WideString;out Error: Integer): OleVariant;

Returnează un string cu informații despre stocul unui articol în formatul *„CodExtern;Denumire;UM;PretVanzare;Stoc*”.

Ca și mai sus, parametrul Error ia valoarea **1**, dacă a intervenit o eroare.

12. GetSoldPart(const PartID: WideString; out Error: Integer): OleVariant;

Returnează un string cu informații despre soldul unui partener în formatul *„CodExtern;Denumire;Sold”*. Parametrul Error are aceeași semnificație ca mai sus.

13. GetListaPersonal

Returnează:

„Nume;Prenume;Marca;CNP;flag EsteACtiv (Da/Nu);flag EsteAgent (Da/Nu);Serie Buletin;Numar buletin;CodPostal;”

14. GetSoldDetaliat(PartID: WideString; out Error: Integer)

Detaliază soldul pe facturi neîncasate și avansuri.

Liniile privind fiecare factură au structura:

„Factura;NrFactura;DataFactura;Rest_de_Incasat”;

Avansurile (dacă sunt) sunt returnate după facturile neîncasate pe structura:

„Avans;DocumentAvans;DataDocument;SumaAvans;”

În cele două tipuri de linii, *„Factura”* și „Avans” sunt texte care definesc tipul liniei; la avans, SumaAvans va fi negativă.

15. GetListaGestiuni(out Error: Integer): OleVariant

Generează lista gestiunilor în forma *Simbol;Denumire*

Daca Error <> 0 operatia a esuat (citește listă erori).

16. GetClaseParteneri(out Error: Integer): OleVariant

Generează lista claselor de parteneri în forma *Simbol;Denumire*;

Daca Error <> 0 operatia a esuat (citește listă erori).

17. ExistaFactura(Numar: Integer): Integer;

Testează existența facturii cu numărul *„NUMAR”.*

Funcția poate returna valorile:

- **-1** : eroare execuție funcție(citește listă erori);
- **0** : factura nu există;
- **1** : factura există și este operată;
- **2** : factura există și este neoperată.

18. GenCodArticole: Integer;

Generează coduri externe în nomenclatorul de articole.

Returnează **-1** dacă au fost erori (citește listă erori) sau numărul de articole actualizate;

Pentru citirea articolelor din **WinMENTOR** trebuie să apelați funcția:

19. GetProducts(const LastSyncDate: WideString; out Error: Integer): OleVariant;

Ca parametru de intrare se trimit data și ora ultimei sincronizări pe structura:

'dd.mm.yyyy hh:mm:ss';

Funcția returnează datele pe structura:

IDArticol;

Denumire;

Den_UM;

IDProducator;

Denumire Producator;

TipSerie;

DataAdaugarii;

DataUltimeiModificari;

Tip unitate de masura;

Cod Intern WinMentor;

Simbol Clasa

20. GetStergeriProduse(const LastSyncDate: WideString; out Error: Integer): OleVariant;

Returnează lista articolelor șterse ulterior datei și orei trimise în parametrul *LastSyncDate.*

Structura returnată va fi:

”

Cod Intern WinMentor;Data si ora stergerii;”

Pentru adăugarea de parteneri noi se folosește funcția:

21. AdaugaPartener(const InfoPart: WideString): Integer;

Structura unei linii InfoPart este:

1. *ID Partener;*
2. *Denumire partener;*
3. *Cod Fiscal;*
4. *Sediul in localitatea;*
5. *Adresa sediu;*
6. *Telefon sediu;*
7. *Persoane de contact; //* separate prin „~”
8. *Simbol Clasa;*
9. *Simbol categorie de pret;*
10. *ID Agent implicit;*
11. *Nr. Registrul comertului;*
12. *Observatii;*
13. *Simbol banca; //* separate prin „~” dacă sunt mai multe;
14. *Nume banca;//* separate prin„ ~” dacă sunt mai multe;
15. *Localitate banca; //* separate prin „~”dacă sunt mai multe;
16. *Cont banca; //* separate prin „~” dacă sunt mai multe*;*
17. *Zi implicita plata;*
18. *Nume sediu secundar ;//* separate prin „~”, dacă sunt mai multe*;*
19. *Adresa sediului secundar; //* separate prin „~”,dacă sunt mai multe*;*
20. *Telefonul sediului secundar;//* separate prin „~”, dacă sunt mai multe*;*
21. *Localitatea sediului secundar;//* separate prin „~”, dacă sunt mai multe*;*
22. *ID Agent pentru sediului secundar; //* separate prin „~”,dacă sunt mai multe;
23. *CodExtern*
24. *Simboluri auto judete sedii de livrare*
25. *Simbol auto judet sediu principal*
26. *Flag care indica daca partenerul este persoana fizica (PF)*
27. *Scadenta implicita la vanzare*
28. *Simbol tip contabil implicit*
29. *Flag (P) care indica daca partenerul este producator*
30. *adresa eMail sediu social*

*31. adrese eMail sedii de livrare (*concatenate cu „~”)

32. TVAIncasare (daca este D atunci este partener cu TVA la încasare)

33. SerAI

34. NrAI

35. Simbol Auto tara Sediu principal

Dacă sunt coloane pe care nu le gestionați în aplicația dvs. sau nu intentionați să ajungă în **WinMENTOR**, veți trimite șir vid (adică ;;). Numărul de separatori „;” trebuie însă musai să fie respectat.

21.1 GetNextPartID(out Error: Integer): WideString

Folosită pentru a obține următorul ID partener disponibil pentru parteneri. (Acest mecanism presupune ca ID partener sunt numerice).

22. ModificaPartener(const InfoPart: WideString): Integer;

InfoPart are aceeași structură cu cea de la *AdaugaPartener*.

23. GetListaBanci(out Error: Integer): OleVariant;

Returnează lista băncilor la nivel național din **WinMENTOR** pe structura:
„Simbol;Denumire”

24. ComenziValide: Integer;

Validează pachetul de comenzi trimis prin SetDocsData. Returnează **1**, dacă pachetul este valid și **0**, dacă nu (caz în care trebuie citită lista de erori).

26. ImportaComenzi: Integer;

Funcția importă pachetul de comenzi și returnează numărul de comenzi importate.

25. MonetareValide : Integer;

Validează pachetul de monetare trimis prin funcția *SetDocsData*. Returnează **0**, dacă pachetul conține erori și **1**,dacă pachetul este valid.

26. ImportaMonetare: Integer;

Importă pachetul de monetare și returnează numărul de monetare importate.

27. GetOferte(out Error : integer) : OleVariant.

Structura datelor de iesire :
„PartID;ArtID;DataInceputOferta;DataSfarsitOferta;Pret;Cantitate”

28. GetClaseArticole(out Error: Integer): OleVariant;

Returnează:
„SimbolClasa;NumeClasa;”

Pentru citirea nomenclatorului de articole din **WinMENTOR** se folosește:

29. GetNomenclatorArticole(out Error: Integer): OleVariant;

Returnează nomenclatorul de articole pe structura:

CodExtern(Intern);
Denumire;
DenUM;
PretVanzare;
SimbolClasa;
DenClasa;
CodExtern(Intern)Producator;
Den.Producator;
GestImplicita;
CodExtern (util de știut când identificatotul unic este *CodIntern)*;
CotaTVA;
DenUMSecundaraImplicita;
ParitateUMSecundaraImplicita;
Masa;
Serviciu;
CodVamal;
PretMinim;
CantImplicita;
PretValuta;
DataAdaug;
Masa;
PretVCuTVA;
Locatie;
PretReferinta;

30. FactIntrareValida : Integer;

Validează pachetul cu facturi de intrare transmis prin *SetDocsData*. Returnează **0**, dacă pachetul conține erori și **1**, dacă pachetul este valid.
În cazul când pachetul este invalid, erorile apărute se citesc cu funcția *GetListaErori*.

Importul propriu-zis de documente se face cu funcția:

31. ImportaFactIntrare : Integer;

Returnează numărul de facturi importate.
Iată un exemplu de utilizare a acestor 2 funcții de mai sus:

```
Erori.Clear;
  if WinMDocImp.FactIntrareValida = 0 then begin // am erori de validare
    ListaErori := WinMDocImp.GetListaErori;
    if VarIsArray(ListaErori) then
for i := VarArrayLowBound(ListaErori, 1) to VarArrayHighBound(ListaErori, 1) do
Erori.Items.Add(ListaErori[i]);
  end
  else ShowMessage('Au fost importate ' + Inttostr(WinMDocImp.ImportaFactIntrare));
```

32. GetVanzariExt(out Error: Integer): OleVariant;

Returnează lista vânzărilor din luna de lucru pe structura:
„PartID; Zi; PrefixDoc; NrDoc; ArtID; Cant; DenUM; Pret; DenGest; CodInternArt; Locatie client;
Adresa; Comision; CodFisca; MarcaAgent; ValAchizitie; CodPostal; ClasaArticol;”

33. GetIncasariClienti(An1, Luna1, An2, Luna2: Integer; const PartID: WideString; out Error: Integer): OleVariant

Returnează doar valoarea totală a încasărilor într-un anumit interval.

34. GetVanzariLuna

Returnează lista facturilor emise într-o lună de lucru.
Structura returnată este :

IDPartener;
Zi;
NrFactura;
ID Articol;
NumarComanda;
Cant;
DenUM;
Pret;
MarcaAgent;
Valoare Factura.

35. GetSolduriExt - returnează detalierea soldurilor clienților, soldurile fiind constituite din facturi și avansuri.

Pentru o linie care descrie un sold provenit dintr-o factură structura este:

ID Partener;
Factura; (un text care-mi spune proveniența soldului)
NrFactura;
DataFactura;
Rest de plata;
Termen de plata;
Locatia partener pentru care s-a emis factura;
Marca Agent
Valoare factura;
Observatii factura;

Pentru o linie de avans structura este:

ID Partener;
Avans; (un text care-mi spune provenienta soldului);
Tip Document plata avans (exemplu Chit, CEC, OP etc);
Nr Document plata avans;
Data Document plata avans;
Rest avans;
Marca Agent incasator;
Valoare initiala avans;

36. GetComenziNefacturate
Returnează lista articolelor comandate, dar încă nefacturate pe structura:
IDArticol;
Numar Comanda;
Cant;
Den Articol;

37. IncasariValideExt - validează pachetul cu încasări trimis prin SetDocsData
38. ImportaIncasariExt – importă pachetul cu încasări.

Pentru setarea câmpului de identificare a partenerilor în Baza de date se folosește:

39. Procedure SetIDPartField(const FieldName: WideString);
FieldName poate lua valorile: CodExtern, CodFiscal sau CodIntern;

Pentru setarea câmpului de identificare a articolelor în baza de date se folosește:

40. Procedure SetIDArtField(const FieldName: WideString);
FieldName poate lua valorile : CodExtern sau CodIntern.

41. GenCodParteneri: Integer

Funcția returnează următorul cod intern, cod extern sau cod fiscal, în funcție de valoarea constantei *„Cod pentru identificare partener”* din *Constante generale > Import date din alte aplicații*.

42. GetPretVanzare(const ArtID, PartID: WideString; out Error: Integer): OleVariant;

Funcția returnează prețul de vânzare sau o listă cu prețurile de vânzare în cazul în care sunt prețuri multiple ale unui articol pentru un partener dat.

43. GetListaCatPret(out Error: Integer): OleVariant;

Returnează o listă cu categoriile de preț pentru firma curentă.

44. GetListaArtCatPret(out Error: Integer): OleVariant;

Returnează o listă cu categoriile de preț specificate pentru fiecare articol în parte.

45. GetListaLocalitati(out Error: Integer): OleVariant;

Returnează o listă cu localitățile fiecărui partener al firmei curente.

46. GetSolduri(out Error: Integer): OleVariant;

Returnează o listă de stringuri cu informații despre încasările pe agent. Fiecare linie conține informații în formatul:

„NrDoc;DataDoc;Rest;Termen;Agent;Valoare”

Avansurile pe agenți sunt returnate în formatul:

„Doc;NrDoc;DataDoc;Suma;Agent;Valoare”

47. GetListaCarnete(out Error: Integer): OleVariant;

Funcția returnează o listă cu carnetele de documente folosite la ieșirile către clienții interni și la transferurile între gestiuni.

48. GetNumarFactura(const SimbolCarnet: WideString; out Error: Integer): Integer;

Funcția verifică dacă un număr de factură este folosit într- un carnet de document dat.

49. GetIncasariClienti(An1, Luna1, An2, Luna2: Integer; const PartID: WideString; out Error: Integer): OleVariant;

Funcția returnează o listă cu încasările unui client într-un interval An1.Luna1 - An2.Luna2 dat.

50. ImportaComenzi: Integer;

Importă comenzile interne din datele transmise cu *SetDocsData*. Returnează numărul total de comenzi importate.

51. ComenziValide: Integer;

Funcția validează dacă structura comenzii interne ce urmează a fi importată este validă.

52. SetIndexNart(const aIndexName: WideString);

O metodă utilizată anterior implementării funcției *SetIDArtField*.
În prezent nu-și mai are rostul, însă am păstrat-o în ideea că cineva o mai folosește încă.

53. ComenziValideExt: Integer;

Funcția validează dacă structura comenzii ce urmează a fi importată este validă.

54. ImportaComenziExt: Integer;

Importă comenzile din datele transmise cu *SetDocsData*. Returnează numărul total de comenzi importate.

55. GetVersiuni(out VerMentor, VerServer: Double): Integer;

56. GetInfoPers(IDPers: Integer; out Error: Integer): OleVariant;

Returnează o listă cu numele și prenumele fiecărui angajat.

57. GetInfoPart(const PartID: WideString; out Error: Integer): OleVariant;

Returnează denumirea partenerului pentru *PartID* trimis.

58. GetListaClienti(AnInceput, LunaInceput: Integer;out Error: Integer): OleVariant;

Returnează o listă de stringuri cu informații despre clienți, începând cu o anumită lună a unui an, în formatul:
„CodIntern;CodExtern;Denumire;CodFiscal;Localitate;Judet;Adr;Tel;MarcaAgent;DataFact; SediiPart;Simbol_Clasa;Denumire_Clasa;LocalitSedii”

59. GetStringConstanta(Id: Integer; const Simbol: WideString): WideString;

60. GetArticoleVandute(const PartID: WideString; MarcaAgent, AnInceput,

LunaInceput: Integer;out Error: Integer): OleVariant;

Funcția returnează un șir cu articolele vândute unui anumit client de un agent începând cu luna.anul dat.

61. GetUltimeleVanzari(const ArtID, PartID: WideString; MarcaAgent, Cate: Integer; out Error: Integer): OleVariant;

GetDocFromFile;
GetVanzariExt
GetIntrari
GetListaSubunit
SetClasaArt
GetInfoBon
GetInfoBonExt
GetInfoIesiri
GetInfoIesiriExt
GetStocArtDetaliat
GetListaArtCatPretExt
BonuriConsumValide
ImportaBonuriConsum
GetListaDelegati
SetCmdImplicitAcceptat
GetOferteClienti
GetCritDiscPeArticole
GetCritDiscPeClase
GetCritDiscPart
GetStocuriPeGestiuni
GetReceptii
GetTransferuri
CheckDocument
GetTranzactiiInCurs
TransferuriValide
ImportaTransferuri
GetSuppliersOrders
AddProduct
SetReceivingList
GetReceivingStatus
GetInventoryOrders
SetInventoryOrders
GetDispozitiiDeLivrare
SetPickedList
ExistaFacturaExt
GetStocArticoleExt
ExistaFacturaIntrare
PotIntroduceDoc
GetDeliveryOrders
SetDeliveryList
GetListacarneteExt
GetInfoCmdNefacturate
GetNomenclatorLocalitati

GetMonede
NCValide
ImportaNoteContabile
GetStergeriProduse
GetInfoCmdSubNefacturate
AdaugaGestiune
ActualizeazaAcceptat
GetReceptiiIntrSubunit
GetInfoCmdBon
GetInfoCmdBonuri
GetIncasariFactura
PlatiValideExt
ImportaPlatiExt
GetNirAtribute
SetDescarcareAutomata
ModiProduct
AddClasaArt
ModifPretValide
ImportaModifPret
GetStocInitialAmbalaj
GetMiscariAmbalaje
GetStocArticoleExt2
ComenziFurnValide
ImportaComenziFurn
GetSoldFactNeop
GetIncasariLuna