



New-Vision-Soft
softwarehouse

Calender 4

SDK V1.1

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Einleitung.....	3
Vorraussetzungen:	3
Das Objekt IApp4.....	3
Eigenschaften IApp4:.....	3
Methoden IApp4:	6
Das Objekt ITask4	7
Eigenschaften ITask4:.....	7
Methoden ITask4:.....	9
Das Objekt INOTE4	9
Eigenschaften INote4:	10
Methoden INote4:.....	10
Das Hauptobjekt NVSCalender4.....	10
Methoden des Hauptobjekts Calender.NVSCALENDER4	10
Beispiele: Delphi	12

Einleitung

Der Calender bietet die Möglichkeit sich von Fremdanwendungen über eine COM Schnittstelle (Common Object Model¹) fernsteuern zu lassen.

Dabei fungiert der Calender Client (calendar.exe), welcher auf den Arbeitsstationen installiert sein muss, als COM-Automatisierungs-Server.

Vorraussetzungen:

1. Der Calender muss als normale Anwendung funktionsfähig arbeiten, d.h. u.a. der Calender Server ist installiert, läuft und normales Arbeiten mit der Calender Software ist möglich.)

2. Der Calender Client muss als COM Server registriert sein.

Dies muss einmalig mit dem Aufruf `calendar.exe /regserver` in dem Verzeichnis in dem der Calender installiert ist geschehen. Dies muss mit der Command-Shell als Administrator durchgeführt werden.

Das COM Objekt des Calender heißt

Calender.NVSCALENDER4

Dieses Objekt besitzt mehrere Methoden und 3 weitere Objekte IApp4, ITask4 und INote4.

Das Objekt IApp4

Das Objekt **IApp4** stellt einen Termin dar und besitzt folgende Eigenschaften und Methoden.

Eigenschaften IApp4:

NR	Variant	Länge 12	<i>nur lesen</i>
Interne Nummer des Datensatzes			
typ	Variant	Länge 3	<i>nur lesen</i>
Typ des Termins, mögliche Werte			
TER = normaler Termin			
TS = Terminserie			
TSA= Ausnahme Terminserie			
TSL= gelöschter Termin in Terminserie			
person	Variant	Länge 3	<i>lesen /schreiben</i>
Verwaltungseinheit (Kürzel)			
Wert muss existent sein, (VWE muss im Calender angelegt sein)			
Start	Datetime		<i>lesen /schreiben</i>
Beginn des Termin			

¹ Common Object Model (COM) ist eine objektorientierte offene Architektur über die Client-Server-Anwendungen von verschiedenen Plattformen transparent miteinander kommunizieren können.

End_ Datetime lesen /schreiben
Ende des Termin

AllDayEvent Boolean lesen /schreiben
ganztägiges Ereignis

Ereinerungseigenschaften, Setzen später mit Methode
SetRemindInfos

inquirystate integer nur lesen
Informationen zum Anfragestatus eines Termins
Dabei sind die mögliche Werte:

0 = keine Anfrage gestellt
10 = Termin angefragt
20 = Termin angefragt und abgelehnt
30 = Termin angefragt und akzeptiert (genehmigt)

Inquiryreplyper Variant Länge 3 nur lesen
VWE, welcher den Anfragestatus gesetzt hat
Nurlesen, zum Setzen verwenden Sie die Funktion SetAnfragestatus

Inquiryreplydatetime TDateTime nur lesen
Zeitpunkt des Setzens des Anfragestatus eines Termins
Nurlesen, zum Setzen verwenden Sie die Funktion SetAnfragestatus

place Variant Länge 50 lesen /schreiben
Ort

Category Integer lesen /schreiben
interne Nummer der Kategorie

Status Integer lesen /schreiben
Status:
0 = nicht begonnen
1 = in Bearbeitung
5 = Aufgeschoben
9 = erledigt

Priority Integer lesen /schreiben

Priorität
 9 = sehr niedrig
 7 = niedrig
 5 = normal
 3 = hoch
 1 = sehr hoch

Private_	WordBool		<i>lesen /schreiben</i>
Kennzeichen, ob Termin privat ist			
Subject	Variant	80	<i>lesen /schreiben</i>
Betreff des Termins			
Text	Variant	Länge 2500	<i>lesen /schreiben</i>
Text des Termins			
CrPerson	Variant	Länge 3	<i>nur lesen</i>
VWE, welchen Termin angelegt			
CrDatetime	Tdatetime		<i>nur lesen</i>
Datum und Zeit, wann der Termin erstellt wurde			
ModPers	Variant	Länge 3	<i>nur lesen</i>
VWE, welchen Termin das letzte mal geändert hat			
ModDatetime	Tdatetime		<i>nur lesen</i>
Datum und Zeit, wann der Termin das letzte Mal geändert wurde			
additiv1	Variant	Länge 40	<i>lesen /schreiben</i>
1. frei definierbare Zusatzfeld			
Additiv2	Variant	Länge 40	<i>lesen /schreiben</i>
2. frei definierbare Zusatzfeld			
Additiv3	Variant	Länge 40	<i>lesen /schreiben</i>
3. frei definierbare Zusatzfeld			
Additiv4	Variant	Länge 40	<i>lesen /schreiben</i>
4. frei definierbare Zusatzfeld			
RefPerNr	Variant	Länge 12	<i>nur lesen</i>

Nummer des Mastertermins

Der Mastertermin ist der Termin, von dem allen anderen abhängen

persinv	Variant	Länge 255	<i>nur lesen</i>
Alle beteiligten VWE eines Termins, getrennt durch Kommas			
AppGroup	Variant	Länge 12	<i>Lesen/schreiben</i>
Interne Gruppennummer. Alle Termine, welche zu einer Gruppe gehören, haben dieselbe Nummer			
split	WordBool		<i>Lesen/schreiben</i>
Ist true, wenn es sich um einen Splittermin handelt			
totaltime	Integer		<i>Lesen/schreiben</i>
Länge der Gesamttermine einer Splitgruppe. Wird nur Kundenspezifisch verwendet			

Methoden IApp4:

Procedure GetRemindInfos(VAR count: integer; VAR unit_: PChar; VAR email: WordBool)
Holt Informationen zur Erinnerung, dabei steht in Count die Anzahl der Einheiten, in unit_ die Einheiten (erlaubte Werte sind M = Minuten, S = Stunden, T = Tage)
Email ist ein Schalter, ob die Erinnerung auch über Email erfolgen soll

Procedure SetRemindInfos(count: integer; unit_: PChar; email: WordBool)
setzt Erinnerungseigenschaften
weitere Infos siehe GetRemindInfos

Procedur SetInquiryState(status: Integer; mkz: OleVariant);
Setzen des Anfragestatus
Parameter Status siehe Feld inquiryState
mkz ist die VWE, welche den Status setzt

Funktion Getcategoryname: :OLEVariant
Liefert den Namen der Kategorie

Prozedur Setcategoryname(kategorienname :OLEVariant)
Setzt die Kategorie des Termins über den Namen

Prozedur SetAppRecurrency(rectype, recparam1, recparam2, recendtype, recendparam: OleVariant);
Setzt die Parameter für einen Serientermin

Rectype	Recparam1	Recparam2	Beschreibung der Serie
T	A (Arbeitstage)	----	Jeden Arbeitstag
T	---	x	Jeden x. ten Tag
W	1010000 Folge von 7 Nullen und Einsen, Start mit Montag, letzter tag ist der Sonntag 1 steht für an diesem tag findet Serie statt	x	Alle x. Wochen am Montag., Mittwoch,g ..
M	X (Monat)	Y (Tag)	Jeden x. Monat am y.ten
O	X (Monat)	1.Stelle Y, 2. Stelle Z Y = 1 =>Montag Y = 2 =>Dienstag Y = 3 =>Mittwoch Y = 4 =>Donnerstag Y = 5 =>Freitag Y = 6 =>Samstag Y = 7 => Sonntag Z= 1 => ersten Z= 2 => zweiten Z= 3 => dritten Z= 4 => vierten Z= 5 => letzten	Jeden x. Monat am z.ten Y
J	X (Monat) 1= Januar 2 = Februar ...	Y (Tag)	Jedes Jahr am Y.Tag des X.Monats

Recendtype	Recendparam	Beschreibung
K	----	Endet nie
D	X (Datum)	Endet am X
W	X (Anzahl)	Endet nach X Vorkommen

Prozedur GetAppRecurrency(VAR rectype, recparam1, recparam2, recendtype, recendparam: OleVariant);

Holt die Parameter für einen Serientermin

siehe SetAppRecurrency

Prozedur SetAppNoRecurrency

Setzt alle Parameter so, dass Termin ein normaler (kein Serientermin) ist.

Das Objekt ITask4

Das Objekt ITask4 stellt eine Aufgabe dar und besitzt folgende Eigenschaften und Methoden.

Eigenschaften ITask4:

NR Interne Nummer des Datensatzes	Variant	Länge 12	<i>nur lesen</i>
typ Typ der Aufgabe, Wert immer TOD	Variant	Länge 3	<i>nur lesen</i>
Start Datum Beginn der Aufgabe	Datetime		<i>lesen /schreiben</i>
Due Datum der Fälligkeit der Aufgabe	Datetime		<i>lesen /schreiben</i>
TaskType Art der Aufgabe 0 = private Aufgabe 1 = Allgemeine Aufgabe 2 = delegierte Aufgabe	Integer		<i>lesen /schreiben</i>
<p>Achtung: damit eine Aufgabe z.B. delegiert sein kann, müssen auch die Felder Taskfor und Taskfrom richtig gesetzt werden. Dann erst das Kennzeichen Tasktype setzen.</p>			
Taskfor Verwaltungseinheit für den die Aufgabe ist Leer bei allgemeiner Aufgabe	Variant	Länge 3	<i>lesen /schreiben</i>
Taskfrom Verwaltungseinheit von dem die Aufgabe kommt, angelegt wurde Muss bei z.B. delegierter Aufgabe unterschiedlich zur Taskfor sein	Variant	Länge 3	<i>lesen /schreiben</i>
Subject Betreff der Aufgabe	Variant	80	<i>lesen /schreiben</i>
Text Text des Aufgabe	Variant	Länge 2500	<i>lesen /schreiben</i>
Category interne Nummer der Kategorie	Integer		<i>lesen /schreiben</i>
Status	Integer		<i>lesen /schreiben</i>

Status:

0 = nicht begonnen

1 = in Bearbeitung

5 = Aufgeschoben

9 = erledigt

Priority	Integer		<i>lesen /schreiben</i>
Priorität			
9 = sehr niedrig			
7 = niedrig			
5 = normal			
3 = hoch			
1 = sehr hoch			

Completed	Datetime		<i>lesen /schreiben</i>
Datum wann die Aufgabe erledigt wurde			

place	Variant	Länge 50	<i>lesen /schreiben</i>
Ort			

additiv1	Variant	Länge 40	<i>lesen /schreiben</i>
1. frei definierbare Zusatzfeld			

Additiv2	Variant	Länge 40	<i>lesen /schreiben</i>
2. frei definierbare Zusatzfeld			

Additiv3	Variant	Länge 40	<i>lesen /schreiben</i>
3. frei definierbare Zusatzfeld			

Additiv4	Variant	Länge 40	<i>lesen /schreiben</i>
4. frei definierbare Zusatzfeld			

Methoden ITask4:

Procedure ClearProperties
setzt alle Eigenschaften zurück

Das Objekt INOTE4

Das Objekt INote4 stellt eine Notiz dar und besitzt folgende Eigenschaften und Methoden.

Eigenschaften INote4:

Im Aufbau

Methoden INote4:

Im Aufbau

Das Hauptobjekt NVSCalender4

Methoden des Hauptobjekts Calendar.NVSCALENDER4

Funktion GetAPP4:IAPP4

Liefert die Schnittstelle zu dem APP4 Objekt zurück, wichtig bei Visual Basic zum Zuweisen des APP Objekts (siehe Visual Basic Beispiele)

Funktion GetTASK4:ITASK4

Liefert die Schnittstelle zum ITASK4 Objekt, siehe auch GetAPP4

Funktion GetNOTE4:INOTE4

Liefert die Schnittstelle zum INOTE4 Objekt, siehe auch GetAPP4

Prozedur SetActUser(Param1: OleVariant)

Meldet sich beim Calendar als Benutzer mit dem internen Kürzel Param1 an

Prozedur SetActUserbyLogin(Param1: OleVariant)

Meldet sich beim Calendar als Benutzer mit dem Login Param1 an

Function GetUserbyLogin(Param1: OleVariant):OleVariant

Gibt die interne MitarbeiterId zurück, die zu dem Login Param1 des Benutzers gehört

Function GetUserbyName(Param1: OleVariant):OleVariant

Gibt die interne MitarbeiterId zurück, die zu dem Namen (Vollständiger Name) Param1 des Benutzers gehört

Funktion App_new: OleVariant

Legt einen neuen Termin an und verwendet die Eigenschaften in dem aktuellen IAPP4 Objekt
Rückgabewert: Nummer des angelegten Datensatzes

Wenn der Wert -1 zurückgegeben wird, dann ist ein Fehler aufgetreten

Funktion App_New_sc: OleVariant

Wie App_new, nur wir hier ein modaler Bildschirm für Benutzereingaben geöffnet

Prozedur App_Edit_sc

Zeigt einen modalen Bildschirm zum Bearbeiten des Termins an.

Funktion App_delete(Param1: OleVariant): WordBool

Lädt den Termin mit der Nummer Param1 in das IAPP4 Objekt und löscht in aus der Datenbank

Funktion App_load(Param1: OleVariant): WordBool

Lädt den Termin mit der Nummer Param1 in das IAPP4 Objekt

Funktion App_save: WordBool

Speichert den Termin, welcher in der Variablen IAPP4 liegt ab. Termin muss vorher existieren.
(ansonsten bitte Funktion App_new verwenden)

Funktion Task_new: OleVariant

Legt eine neue Aufgabe an und verwendet die Eigenschaften in dem aktuellen IAPP4 Objekt

Rückgabewert: Nummer des angelegten Datensatzes

Wenn der Wert -1 zurückgegeben wird, dann ist ein Fehler aufgetreten

Funktion Task_new_sc: OleVariant

Wie Task_new, nur wir hier ein modaler Bildschirm für Benutzereingaben geöffnet

procedure Task_Edit_sc

Wie Task_Edit, nur wir hier ein modaler Bildschirm für Benutzereingaben geöffnet

Prozedur Task_load(Param1: OleVariant): WordBool

Lädt eine Aufgabe in das ITASK4 Objekt

Prozedur Task_save

Speichert das aktuelle ITASK4 Objekt

Prozedur Task_delete

Löscht das aktuelle ITASK4 Objekt

Prozedur ComServerdisableactive

intern verwendet

Prozedur changeview(Param1: Integer)

Wechsel die Ansicht im Calendar.

Mögliche Werte für Param1 sind:

0 = Tagesansicht

1 = Wochenansicht

2 = Monatsansicht

3 = Jahresansicht

Prozedur Jumpto(day: datetime)

Springt an den Tag, welcher in day angegeben ist

Prozedur Minimize

Minimiert den Calendar

Prozedur Show

Zeigt den Calendar an (macht in sichtbar)

Funktion GetCategorybyName(Param1: OleVariant): OleVariant

Liefert die interne Nummer der Kategorie mit Namen Param1 zurück

Funktion GetUserbyLogin (Name: OleVariant): OleVariant
Liefert das interne Kürzel des Benutzers mit dem Login Name.

Funktion GetUserbyName (Name: OleVariant): OleVariant
Liefert das interne Kürzel des Benutzers mit dem Namen Name.

function FetchApps(User: OleVariant; StartDate: TDateTime; EndDate: TDateTime): OleVariant;
Liefert einen String, welcher durch Komma getrennte Terminnummer enthält, welche zu dem internen Benutzerkennzeichen User gehören und sich in dem Datumszeitraum Startdate bis Enddate befinden.

function FetchRelatedApps(AppGroupNumber: OleVariant): OleVariant; safecall;
Liefert zu der Termingruppe mit der Nummer AppGroupNumber alle zugehörigen Terminnummer durch Komma gtrennt.

Der Calender 4 besitzt ein Dual Interface, somit können die Anwendungen wählen ob Sie das IUnkown oder das IDispatch Interface nutzen wollen, bzw. ob frühe oder späte Bindung erwünscht ist.

Beispiele: Delphi

```
procedure TForm1.SchalterneueAufgabemitBildschrimClick(Sender: TObject);
VAR Calender : INVSCALENDER4;
    htask : ITask4;
    hnr : string;
begin
Calender := CoNVSCALENDER4.Create;
htask := Calender as ITASK4;
htask.Taskfor := '';
htask.Taskfrom := 'ADM';
htask.Tasktype := 1;
htask.text := 'Dies ist eine neue Aufgabe, erstellt vom COM Objekt.';
hnr := Calender.Task_new_sc;
MessageDlg('Die Nummer der neuen Aufgabe lautet: '+hnr, mtInformation, [mbOK], 0);
end;
```

```
procedure TForm1.SchalterneueAufgabeohneBildschrimClick(Sender: TObject);
VAR Calender : INVSCALENDER4;
    htask : ITask4;
    hnr : string;
begin
Calender := CoNVSCALENDER4.Create;
htask := Calender as ITASK4;
htask.Taskfor := '';
htask.Taskfrom := 'ADM';
htask.Tasktype := 1;
htask.text := 'Dies ist eine neue Aufgabe, automatisch erstellt vom COM Objekt.';
hnr := Calender.Task_new;
MessageDlg('Die Nummer der neuen Aufgabe lautet: '+hnr, mtInformation, [mbOK], 0);
```

end;

```
procedure TForm1.SchalterAufgabeaendernClick(Sender: TObject);
```

```
VAR Calender : INVSCALENDER4;
```

```
    htask : ITask4;
```

```
    hnr : string;
```

```
begin
```

```
    Calender := CoNVSCALENDER4.Create;
```

```
    htask := Calender as ITASK4;
```

```
    if Calender.Task_load('10000000024') then
```

```
        begin
```

```
            htask.Set_status(9);
```

```
            Calender.Task_save;
```

```
        end;
```

```
    end;
```

```
procedure TForm1.SchalterAufgabeaendernmitBildschirmClick(Sender: TObject);
```

```
VAR Calender : INVSCALENDER4;
```

```
    htask : ITask4;
```

```
    hnr : string;
```

```
begin
```

```
    Calender := CoNVSCALENDER4.Create;
```

```
    htask := Calender as ITASK4;
```

```
    if Calender.Task_load('10000000024') then
```

```
        begin
```

```
            htask.Set_status(5);
```

```
            Calender.Task_edit_sc;
```

```
        end;
```

```
    end;
```

Visual Basic Beispiel:

```
Sub Calendermakro()
```

```
Dim CALENDER As Calender.NVSCALENDER4
```

```
Dim APP As Calender.APP4
```

```
Set CALENDER = CreateObject("Calender.NVSCALENDER4")
```

```
Set APP = CALENDER.GetAPP4
```

```
APP.Text = "123"
```

```
CALENDER.Show
```

```
CALENDER.App_New_sc
```

```
Set CALENDER = Nothing
```

```
End Sub
```